

Kim Korkar LINUX'tan?

Can Uğur Ayfer, Ömer Ayfer

**Bu kitap, PUSULA Yayıncılık tarafından LINUX
kullanıcıları için ücretsiz olarak dağıtılır.**



YAYINCILIK ve İLETİŞİM LIMITED ŞİRKETİ

Pusula **103**

Kim Korkar Linux'tan?

ISBN 975-6477-18-0

Can Uğur Ayfer, Ömer Ayfer

1. Baskı: Eylül 2003

2. Baskı: Ekim 2004

3. Baskı: Temmuz 2006

Yayın yönetmeni: Mustafa Arslantunalı

Düzeltili: Suat Özdemirci

Sayfa düzeni: Mehmet Öztürk

Kapak illüstrasyonu/tasarım: Mehmet Öztürk

Dağıtım sorumlusu: Hüseyin Üstünel

Baskı:

Ayhan Matbaası Tel: (0212) 629 01 65

Bu kitabın bütün yayın hakları Pusula Yayıncılık ve İletişim San. ve Tic. Ltd. Şti'ne aittir. Yayınevimizin yazılı izni olmaksızın kısmen veya tamamen alıntı yapılamaz, kopya edilemez, çoğaltılamaz ve yayınlanamaz.

Pusula Yayıncılık ve İletişim San. ve Tic. Ltd. Şti.

İnönü Cad. Emektar Sok. No: 24/B Gümüşsuyu-Beyoğlu/İSTANBUL

Tel. : (0212) 252 42 80 pbx

Faks : (0212) 293 15 44

e-posta: pusula@pusula.com

web : www.pusula.com

Can Uğur Ayfer

1955 yılında doğdu. Bilgisayarların sadece birkaç üniversitede bulunabildiği, 64 KB belleğe sahip olanların parmakla gösterildiği zamanlarda bu işe merak sarmaya başladı. 1973 yılında ilk kullandığı bilgisayarın 4 KB çekirdek belleği vardı; disk ya da teyp gibi manyetik ortamları ise yalnızca kitaplarda okumuştur. Elini sürdüğü ilk büyük bilgisayar, yaz stajlarını yaptığı Ege Üniversitesi'ndeydi. Daha sonra kendi okulu olan ODTÜ'nün Bilgi İşlem Merkezi'nde öğrenci asistanlık yaptı, ardından Türkiye Elektrik Kurumu'ndaki "efsanevi" IBM 370/138 ile tanıştı. Dile kolay! Tam 512 KB belleği ve toplam 420 MB disk kapasitesi vardı.

Can Uğur Ayfer, zamanla çeşitli işlerde, "çok büyük" diye nitelendirilen bilgisayarları kullandı, çalışır durumda tutulmalarından sorumlu oldu. Büyük boy bilgisayar, orta boy bilgisayar, mini sistemler... derken ortalığı PC'ler sardı. Ama Can Uğur Ayfer'in 25 yıllık meslek hayatında bir konu hiçbir zaman değişmedi. Mesleği bilgisayarlarla doğrudan ilgili olmayan, ama bu "gâvur işi aleti" kullanmak zorunda olan insanların sıkıntıları...

Pusula Yayıncılığın kıdemli yazarlarından Can Uğur Ayfer'in **Kim Korkar Hain Bilgisayarlardan?** isimli kitabı 7 baskı yaparak Türkiye'de bilgisayar kitapları alanında bir rekor kırmıştı. Ayrıca yazarın **Kim Korkar Unix'ten?** (2 baskı) ve **Perl ve MySQL ile CGI Programlama** kitapları da yayınevimizden çıktı. Can Uğur Ayfer, halen Bilkent Üniversitesi'nde çalışmakta ve yeni bilgisayar kitapları hazırlamaktadır.

cayfer@bilkent.edu.tr

Ömer Ayfer

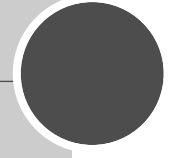
1981 yılında Ankara'da doğdu. Orta öğrenimini Bilkent Üniversitesi Hazırlık Okulu'nda (BUPS) tamamlayıp Purdue Üniversitesi'nde Bilgisayar Mühendisliği eğitime başladı. Üç yılda tamamladığı mühendislik eğitimi sırasında, lise yıllarında tanıştığı UNIX işletim sistemindeki deneyimine LINUX'u da ekleyerek kendini hızla geliştirdi. Halen San Fransisco'da bir biyoteknoloji şirketinde "Bilimsel Yazılım Uzmanı" ve "Sistem Yöneticisi" olarak görev yapmaktadır.

omer@ayfer.net

Reyyan Ayfer'e...
Onsuz hiçbir şey yapamazdık.
Ömer, Uğur

Çok değerli önerileriyle ve bulduğu hatalarla
kitabın daha yararlı olmasını sağlayan
Sayın S. Satuk B. Ercilasun'a sonsuz şükranlarımızla...
Ömer, Uğur

İçindekiler



00. Başlarken	1
• Neden LINUX?	1
01. UNIX Dünyasının Veliht Prensi... LINUX...	3
• UNIX Dünyasının Veliht Prensi LINUX	4
• UNIX İşletim Sistemi	4
• Çok İş, Çok Kazanç	4
• UNIX Her Yerde Aynı UNIX...	5
• Çok İyi Tasarımlanmış Bir İşletim Sistemi	5
• Biraz da Tarih...	6
• LINUX Doğuyor	9
• LINUX Çeşitleri	11
• Bir de “Free BSD” UNIX Varmış...	13
• UNIX Geleneği	13
• UNIX’i UNIX Yapan Özellikler	14
• LINUX’u LINUX Yapan Özellikler	14
• Neden Mandriva LINUX?	15
• Neden İngilizce Kurulum?	15
• LINUX’u “Gerçekten” Öğrenmek İsteyenlere Öğütler	17
• Mandriva LINUX Kurulum Öncesi Hazırlıklar	17
• Kurulum Öncesi Bilinmesi Gerekenler	17
• Nasıl Bir Bilgisayar Gerekli?	17
• Disk Bölümleri (Partition)	18
• Bilgisayarların Açılış Süreci (Boot Süreci)	20
• Disk Dosya Sistemleri (File Systems)	21
• X Window	22

• X Window Pencere ve Masaüstü Yöneticileri (Window Managers, Desktop Managers)	25
• LINUX Çekirdeği (LINUX Kernel)	27
• root	27
• LINUX'ta Disk İsimlendirmesi	28
02. Mandriva LINUX Kurulumu	31
• Varsayımlar	31
• Kurulum	32
03. LINUX'la İlk Tanışma: KDE Masaüstü Yöneticisi	69
• İlk Tanışma	70
• LINUX Bilgisayarların Açılması	70
• LINUX Bilgisayarların Kapatılması	72
• İlk Login	75
• Mandriva LINUX Uygulamaları	82
• KDE Ayarları	82
• Görsel Ayarlar (Look and Feel)	82
• Menü Ayarları	83
• İleri Düzey KDE Ayarları	85
• Kullanıcı Yönetimi: UserDrake	88
• Dosya Yöneticisi: Konqueror	90
• Genel Sistem Yönetimi Araçları: Mandriva Control Center	91
• Ofis Uygulamaları	104
• OpenOffice.org	104
• Grafik Uygulamaları	108
• The Gimp	108
• Diğerleri	109
• Kişisel Uygulamalar	109
• Ximian-Evolution	109
• Gnome-Card	111
• İnternetle İlgili Uygulamalar	112
• Kmail	112
• Konqueror	113
• FTP İstemcisi (gFTP)	114

• Web Sayfası Hazırlama Araçları	115
• Çoklu Ortam (Multimedya) Uygulamaları	116
• XMMS	116
• CD Player	117
• Ve Yüzlerce Uygulama Yazılımı...	117
• Kopete Anında Mesajlaşma Yazılımı	117
• K3B	118
• Gambas	119
• Eclipse	120
• Digikam	121
• Wine	121
• Bundan Sonrası	123
• Konsolda LINUX Kullanımı	124
• “Konsol” Kavramı	124
• “logout”: Sistemden Çıkış	129
• Isınma Hareketleri	131
• Ne Var Ne Yok?	131
• Ben Kimim?	133
• Şifrenizi Değiştirmek İsteddiğinizde...	133
• İmdaaaat!..	135
• Güç... Daha Fazla Güç...	136
• LINUX Dosya-Dizin Yapısı	138
• Dosya-Dizin İsimleri	138
• Her Horoz Kendi Çöplüğünde...	140
• Neredeyim?	141
• Yuvaya Dönüş	142
• Ne Var Ne Yok? (Ama Bu Kez Daha Bir Merakla...)	142
• Dosyalar ve Dizinler	148
• Dosya Yaratmak	149
• cat Komutu	149
• cp Komutu (copy)	152
• Dizin Kopyalamak	153
• Dizin Yaratmak	154
• Çalışma Dizinini Değiştirmek	155
• Dosya Silmek	156

• Dizin Silmek	156
• Dosya / Dizin Adı Değiştirmek	157
• Dosya / Dizin Yeri Değiştirmek	157
• LINUX'ta Erişim Yetkileri	158
• Yazma, Okuma, Program Çalıştırma Yetkileri	158
• Dizinlere Erişim Hakları	163
• umask : Dosya ve Dizinlerin Varsayılan Erişim Yetkileri	164
• Suid Biti ve suid Programlar	166
• Sticky Bit	167
• Dosyaların ve Dizinlerin Sahibini Değiştirmek	168
• Dosyaların ve Dizinlerin Gruplarını Değiştirmek	168
• Dosyaların ve Dizinlerin Sahibini ve Grubunu Birlikte Değiştirmek	169
• bash Kabuk Programı	170
• Dosya İsim Kalıpları	172
• Ortam Değişkenleri	174
• PATH Ortam Değişkeni	175
• Diğer Önemli Ortam Değişkenleri	175
• Ortam Değişkenlerinin Kalıcı Olarak Değiştirilmesi	176
• bash Başlangıç Dosyaları	177
• bash History	178
• ! Aracı	179
• Kendi Komutlarınız: Aliases	180
04. Editörler: Kedit vs. ve En Önemlisi: vi	183
• vi : Visual Editor	185
• vi Bir Kelime İşlemci Değildir!	185
• Beğenseniz de Beğenmeseniz de, vi Öğrenmelisiniz!	185
• Dosya Açmak / Yaratmak	186
• vi Editörünün “Durum”ları	187
• Ekleme Durumu	188
• Komut Durumu	188
• Dosya Açarken Karşılaşılabileceğiniz Sorunlar	189
• İmleç Dolaştırma Komutları	190
• Ekleme Komutları	192

• Yazı Silmek	192
• Yanlışlıkla Silerseniz...	193
• Metin Bloklarının Yerini Değiştirmek	193
• Metin Bloklarını Kopyalamak	196
• Son Komutu Tekrarlamak	197
• Metin Eklemenin / Değiştirmenin Birkaç Değişik Yolu	197
• Metin İçinde Dolaşmak	198
• Metin İçinde Arayarak Dolaşmak	200
• Bulup Değiştirmek	200
• Metnin İçinde Satır Numaralarını Kullanarak Dolaşmak	202
• Tuş Kısaltmaları	203
• vi Başlatırken Verebileceğiniz Komutlar	204
• Dosya İşlemleriyle İlgili Komutlar	205
• vi İçinden UNIX Komutu Vermek	206
• Birden Fazla Dosyayı Peşpeşe İşletmek	206
• .vimrc Dosyası	207
• Daha Ayrıntılı Bilgi İçin...	207
• gvim	209

05. Önemli LINUX Kavram ve Komutları **211**

• Önemli LINUX Kavramları	211
• Standart Giriş ve Standart Çıkış	211
• Giriş ve Çıkışı Yönlendirmek	213
• STDERR	214
• Çekirdek: Kernel	216
• Dosya Sistemleri	217
• fsck	225
• Süreçler	226
• Şeytanlar (Daemons)	230
• Süreç Öldürmek	231
• Link Kavramı ve ln Komutu	232
• “Pipe” Kavramı	235
• Biraz Nefes Alalım	237
• Kullanışlı LINUX Komutları	238
• cal	238

• sleep	239
• watch	239
• wc	239
• tail	240
• more	241
• less	241
• top	242
• which	244
• zip	244
• unzip	245
• gzip	245
• gunzip	246
• bzip2	246
06. Çevreyi Tanıyalım	249
• /dev Dizini	252
• /etc Dizini	254
• lost+found Dizinleri	256
07. bash Kabuğu ve Kabuk Programları	257
• Komut Satırının Yorumlanması ve Parametreler	258
• Kabuk Değişkenleri (ya da Ortam Değişkenleri)	261
• Programları Arka Planda Çalıştırmak	264
• Ön Planda Çalışan Programları Arka Plana Atmak	265
• Kabuk Programlama	267
• İlk Kabuk Programı Örneği	268
• İkinci Kabuk Programı Örneği	270
08. Diskler, Disketler ve mount Komutu	273
• mount Edilmiş Diskleri Görmek	274
• Yeni Disk Takma	275
• Dosya Sistemi Tipleri	277
• fsck: Dosya Sistemi Kontrol Programı	278
• supermount	280
• CD Kullanımı	280
• Disket Kullanımı	281

• <code>umount</code> Komutu	281
• Başka UNIX Bilgisayarlardaki Diskleri Kullanmak	282
• <code>mount</code> İşlemlerini Otomatik Olarak Yapmak (<code>fstab</code>)	282
• Başka Bir Windows Bilgisayarının Diskini Kullanmak	283
• Dosya Sistemi Yaratmanız Gerekirse...	284
09. Önemli LINUX Komutları	287
• <code>find</code>	287
• <code>grep</code>	295
• <code>tar</code> Komutu ve <code>tar</code> Dosyaları	298
• <code>tar</code> Komutu ile Yedekleme	301
• <code>tar</code> Komutunu Kullanırken	
Dikkat Edilmesi Gereken Noktalar	304
• <code>rsh</code>	305
• <code>xargs</code>	308
• <code>at</code>	310
• <code>date</code> , <code>hwclock</code>	313
• <code>lynx</code>	314
• <code>cut</code>	315
• <code>tee</code>	316
• <code>script</code>	317
• <code>split</code>	317
10. Sistem Yönetimi	321
• Sistem Hakkında	323
• Sistemin Açılması	323
• Sistemin Kapatılması	328
• Disklerin Yönetimi	329
• <code>fdisk</code>	331
• <code>mkfs</code>	332
• <code>mount</code>	332
• <code>showmount</code>	333
• <code>lsdf</code>	335
• Bellek Yönetimi	336
• Kullanıcı Yönetimi	339
• Kullanıcı Hesapları	339

• Kullanıcı Grupları	345
• Kullanıcı Disk Kotaları	348
• Log Dosyalarının Yönetimi	351
• Ağ Yönetimi	354
• TCP/IP Ayarları	354
• Yazıcı Yönetimi	361
• Yedekleme	363
• Periyodik İşlerin Yönetimi	366
• LILO Yönetimi	370
• Webmin	375
• “root” Şifresini Unuttuğunuzda...	381
11. Yazılım/Paket Yönetimi	385
• Paket Kavramı	385
• Paket Bağımlılığı (Dependency)	386
• Paket Yöneticileri (Package Managers)	386
• Yazılım Depoları (Media, Repositories)	387
• Yazılım Güncelleme	388
• Yazılım Yükleme	388
• Mandriva rpm Paket Yöneticisi - rpmdrake ile Yükleme	388
• urpmi - Gerçek Paket Yöneticisi	392
• rpm Paketlerinden Yazılım Yükleme	394
• Kaynak rpm Paketlerinden (src-rpm) Yazılım Yükleme	397
• Kaynak Kodundan Derleyerek Kurulum	398
• İşletim Sisteminin ve Uygulama Programlarının Güncellenmesi	400
• Çekirdek Güncellemeleri (Kernel Updates)	403
12. Güvenlik	407
• Sistem Güvenliği	407
• Kolay Tahmin Edilebilecek Şifre	
Kullanmayın ve Kullandırmayın	408
• Sistem Güvenliği ile İlgili Haberleri İzleyin	408
• Olabildiğince Güvenli İletişim Yapan Yazılımlar Kullanın	408
• Güvenlikle İlgili Yamaları İzleyin ve Uygulayın	409
• Gereksiz Yazılım Yükleme	409
• Gereksiz Servisler Çalışmasın	409
• Gereksiz Portlar Açık Olmasın	412

• Bilgisayarınızdaki Tüm Servisler Dünyaya Açık Olmasın	413
• Log Dosyalarınıza Bakın	416
• Sisteminizi Yedekleyin	416
13. Yararlı LINUX Sunucu ve Servis Yazılımları	419
• Yararlı LINUX Sunucu Yazılımları	420
• ftp Sunucusu	420
• apache Web Wunucusu	422
• postfix e-Posta Sunucusu	425
• procmail	428
• samba Sunucusu	429
• named (DNS) Sunucusu	429
• ssh Sunucusu (Secure Shell)	432
• NIS Sunucusu (Network Information Services)	433
• iptables Ateş Duvarı	433
• DHCP Sunucusu	434
• MySQL ve PostgreSQL Veritabanı Sunucuları	436
• squid Proxy Sunucusu	437
• ppp Çevirmeli Ağ Sunucusu	439
EK: GNU Genel Kamu Lisansı GNU Public License (GPL)	441
Sonsöz	450

• Neden LINUX?

Neden LINUX?

“*Bedava da ondan*” diyeceğimizi sanıyorsanız yanılıyorsunuz!

Yanıtımız: “*LINUX iyi bir işletim sistemidir de ondan...*”

Evindeki bilgisayarı eğlence için kullananlardansanız, bilgisayarı İnternet’te dolaşmak, oyun oynamak, internet sohbetlerine katılmak için kullanıyorsanız bu kitabı satın alıp paranızı ve zamanınızı boşa harcamayın!

“*Benim işim bilgisayarlarla; 2000’li yılların yükselen yıldız işletim sistemini öğrenmek istiyorum*” diyorsanız bu kitabı alın; LINUX öğrendiğiniz için pişman olmazsınız.

Yok eğer bilgisayarı internet ve intranet servisi vermek için kullananlardansanız, güvenilir ve güvenli sunucu hizmetleri vermek istiyorsanız bu kitabı almalısınız! LINUX veya UNIX işletim sistemlerini öğrenerek mesleğinizde çok değerli bir ilerleme kaydedebilirsiniz.

LINUX’un bedava olması genel kanısının ve birçok yerde ön plana çıkarılmasının tersine, bu LINUX’un çok da önemli bir özelliği değildir. LINUX’un bizce en önemli özelliği güvenilir ve güvenli olmasıdır. Zaten Google, Yahoo gibi dünyanın en büyük internet sunucularını işleten kuruluşların yatırımlarını LINUX üzerine kurmuş olmaları da bundandır. Hatta Microsoft bile uzun süre Hotmail servisini FreeBSD (aslında FreeBSD, LINUX değildir, ancak gerek felsefesi gerekse teknolojisi eşdeğerdir) ile vermiştir. LINUX bir işletim sistemidir. İnsanlar kullansın diye insanlar tarafından geliştirilmiş bir programdır. Kitabı da vardır, dökümantasyonu da, desteği de...

Bir de üstüne üstlük bedavadır... İşte bu kitabı LINUX'un korkulacak bir yazılım olmadığını vurgulamak için yazdık. LINUX; daha doğrusu UNIX; felsefesi ve gelenekleri olan bir işletim sistemidir. Öğrendikleriniz yıllarca geçerliliğini korur. İşletim sistemi küçük, basit ama işini çok iyi yapan programlardan oluşur. Çok hızlıdır. Az kaynakla çok iş yapabilir. Bu sayede eski bilgisayarlar, geri planda web sunucusu, e-posta sunucusu, güvenlik barajı (firewall) gibi önemli işlerde değerlendirilebilmektedir.

“Ama Windows işletim sistemini kullanması kolay!” sözlerini çok duyuyoruz. Evet, masaüstü uygulamalar ve ev kullanımı için yerinde bir söz; ama sunucu hizmetleri ve güvenilirlik gerektiren ciddi işler söz konusu olduğunda Windows tabanlı sistemlerin yöneticileri için hayat gerçekten çok zor oluyor. Bunu deneyerek öğrendik. Örneğin Bilkent Üniversitesi'nde 10 kadar Windows tabanlı sunucudan oluşan büyükçe bir ağı ayakta tutmak için sistem yöneticilerine nöbet yazmak, vardiyalı iş programları düzenlemek gibi işlerle uğraşırken, bu sunucuları UNIX ortamına geçirerek normal mesai saatlerinde çalışan tek yöneticiyle daha iyi hizmet verdik.

LINUX, “açık sistem” bir işletim sistemidir; yani, kaynak koduna her isteyen erişebilir. Bu sayede, yazılım hataları çok kısa sürede bulunabilmekte ve düzeltme yamaları birkaç saat içinde dünyada yayınlanmaktadır.

Sözün kısası; LINUX güvenli, güvenilir, tutarlı, hızla gelişen, her tür yeni teknolojiye çok kısa sürede uyum sağlayan, ciddi, profesyonel bir işletim sistemidir.

Bu kitap LINUX'u öğretmek için yazılmamıştır. LINUX öğretmek için “kütüphane” yazmak lazım... Bu kitap, LINUX denizine atlamak isteyenlerin ayaklarını ıslatıp, varsa korkularından arındırmak için yazılmıştır.

LINUX dünyasına hoş geldiniz...

UNIX Dünyasının Veliaht Prensi... LINUX...

01

- **UNIX Dünyasının Veliaht Prensi LINUX**
 - **UNIX İşletim Sistemi**
 - **Çok İş, Çok Kazanç**
 - **UNIX Her Yerde Aynı UNIX...**
 - **Çok İyi Tasarımlanmış Bir İşletim Sistemi**
 - **Biraz da Tarih...**
 - **LINUX Doğuyor**
 - **LINUX Çeşitleri**
 - **Bir de "Free BSD" UNIX Varmış...**
 - **UNIX Geleneği**
 - **UNIX'i UNIX Yapan Özellikler**
 - **LINUX'u LINUX Yapan Özellikler**
 - **Neden Mandriva LINUX?**
 - **Neden İngilizce Kurulum?**
 - **LINUX'u "Gerçekten" Öğrenmek İsteyenlere Öğütler**
- **Mandriva LINUX Kurulum Öncesi Hazırlıklar**
 - **Kurulum Öncesi Bilinmesi Gerekenler**
 - **X Window Pencere ve Masaüstü Yöneticileri (Window Managers, Desktop Managers)**
 - **LINUX Çekirdeği (LINUX Kernel)**
 - **root**
 - **LINUX'ta Disk İsimlendirmesi**

UNIX Dünyasının Veliht Prensi LINUX

UNIX İşletim Sistemi



Dünyada hiçbir işletim sistemi, UNIX kadar uzun ve sürekli gelişerek gündemde kalmayı başaramamıştır. IBM PC'ler için geliştirilmiş olan Windows serisi işletim sistemleri kadar yaygın olmamakla birlikte, dolaylı yoldan da olsa, UNIX işletim sisteminin hizmet vermekte olduğu kullanıcı sayısının, Windows ve MS-DOS kullanıcılarının sayısından daha fazla olduğu sanılmaktadır. Örneğin, Nisan 2002'ye kadar Microsoft MSN, abonelerinin kişisel web sayfalarını yayınlayan bilgisayar UNIX altında çalışan bir **Apache** sunucusuydu. (Hala öyle olduğunu iddia edenler var ya, neyse...)

5-6 yıl öncesine kadar UNIX genellikle güçlü bilgisayarlar dünyasının işletim sistemiydi. UNIX felsefesinin temelinde, bir bilgisayarın birden fazla kullanıcı arasında paylaşılması ya da bir kullanıcının aynı anda birden fazla iş yapmasına olanak sağlaması yatmaktadır. Bu nedenle, UNIX ile çalışan bir bilgisayarın, kaynaklarının birden fazla iş arasında paylaşılması durumunda performansını kabul edilebilir düzeyde tutabilecek güçte olması gerekmektedir.

Bilgisayar teknolojisindeki gelişmeler, donanımları hızla güçlendirip ucuzlatınca bunun doğal sonucu olarak, UNIX işletim sistemi kendine kişisel bilgisayar dünyasında da yer aramaya başladı ve sonunda UNIX işletim sisteminin x86 tabanlı kişisel bilgisayarlara bir uyarlaması olan LINUX, bilgisayar tarihindeki yerini aldı.

LINUX ve UNIX temelde eşdeğer işletim sistemleridir. Zaten UNIX'in de çok değişik türevleri bulunmaktadır; ancak bu türevler arasındaki farklar genellikle ayrıntılardadır. Herhangi bir UNIX'i iyi bilen; özellikle de UNIX'in felsefesini iyi kavramış birisi için hepsi aynı sayılır.

Çok İş, Çok Kazanç

UNIX işletim sistemi, bilgisayar bilimcilerinin “çok kullanıcı” (*multi-user*) ve “çok işli” (*multi-tasking*) adını verdikleri çalışma koşullarını sağlar. Bir başka deyişle; UNIX altında çalışan bir bilgisayar, birden fazla kullanıcı birbirlerinden bağımsız olarak ve aynı anda kullanabilir. Bu birlikte kullanım sırasında, bilgisayarın kaynaklarını (merkezi işlem birimini, ana belleğini

(RAM), disk-teyp gibi yan bellek birimlerini, yazıcılarını) paylaşırlar. UNIX, bu kaynakların kullanımını, paylaşımından kaynaklanan performans düşmelerini en aza indireyecek şekilde düzenlemeye çalışır.

Bir kullanıcının aynı anda birden fazla iş yapabilmesi de bir başka kolaylıktır. Bilgisayarınızda uzun bir iş başlattığınızı varsayalım; ancak bu iş, her birkaç dakikada bir sizin klavyeden müdahale etmenizi gerektirsin. Bu durumda, bu uzun işi başlatıp yemeğe gidemezsiniz. Tek iş düzeninde kullanım için tasarlanmış bir işletim sistemi kullanıyorsanız (MS-DOS gibi), söz konusu programın yaptığı iş tamamlanıncaya kadar bilgisayarın karşısında oturmak zorundasınız. Eğer bu işi UNIX altında çalışan bir bilgisayarda yapıyorsanız, uzun programınız bir yandan çalışırken, siz öte yandan (gene aynı ekran ve klavyeyi kullanarak) bir başka iş yapabilirsiniz. Örneğin, bu tür beklemeelerde internet üzerinde bir gezintiye çıkabilirsiniz. Şimdi *“Eeee... Bunda ne var? Windows işletim sisteminde de bunu yapılabiliyorum.”* diyorunuz herhalde. Evet, doğru, kısmen yapabilirsiniz. Bir de binlerce dosyadan oluşan, toplam bir-iki Gigabyte’lık bir dizini silerken ikinci bir iş yapmayı deneyin bakalım neler oluyor! Windows XP-Professional Edition altında dizin silerken diğer tüm servisler duruyor! LINUX altında ise durmuyor!

UNIX Her Yerde Aynı UNIX...

İlk kez 1970 yılında ortaya çıkan UNIX işletim sistemi, ticari bir amaçla tasarlanmamıştı; bu yüzden, bu yeni işletim sistemine ilgi duyan tüm bilgisayar üreticilerine ve bilgisayarını UNIX desteği ile kullanmak isteyen herkese çok küçük bir ücret karşılığında dağıtıldı. Bu sayede, UNIX kısa sürede gelişti ve yayıldı. Bu gelişmelere katkıda bulunan bilgisayarlar, UNIX’in ilk günlerinde ortaya atılan standartları gelenekleştirerek korudular. Böylece sağlam bir UNIX geleneği ve kültürü oluştu. Kullanıcılar açısından bunun anlamı oldukça basittir: UNIX işletim sistemini bir kez öğrendiniz mi, UNIX’le çalışan herhangi bir bilgisayarı kolaylıkla kullanabileceğiniz gibi; alışık olduğunuz komut ve kavramların yüzde 99’u farklı bilgisayarlarda bile aynen geçerli kalacaktır.

Çok İyi Tasarımlanmış Bir İşletim Sistemi

UNIX işletim sistemi’nin 36 yıllık bir geçmişi var. Bu süre bilgisayar endüstrisi için çok ama çok uzundur. Son 36 yıl içinde bilgisayarlar çok değişti, gelişti, hızlandı, küçüldü. 36 yıl önceki donanım tasarımları çoktan unutuldu ama UNIX işletim sistemi, ilk yıllarında sahip olduğu özellikleri ve yetenek-

leriyle dimdik ayakta duruyor. 36 yaşındaki yaşlı UNIX, (belki de sadece “olgun” demek daha doğru) günümüz bilgisayarlarına çok kolay uyum sağladı. Bundan sonraki gelişmelere de rahatça ayak uydurabilecek gibi görünüyor. Bilgisayar dünyası, yaklaşık 50 yıllık tarihinin son 30-35 yılında, işletim sistemlerine ait birçok örnek gördü geçirdi. Bunlardan bazıları çok başarılı oldu, bazıları özel uygulamalara hizmet etti ve ömrünü tamamladı, bazıları piyasaya çıkamadan yok oldu, unutuldu gitti. İşletim sistemleri genellikle donanım üreticileri tarafından, ürettikleri bilgisayar modelleri için özel olarak geliştirildiler. Söz konusu donanım modelleri ortadan kalktıkça, bu bilgisayarların işletim sistemleri de sahneden ayrıldılar.

UNIX için böyle olmadı; çünkü belirli bir marka veya model donanım için tasarılanmamıştı. Donanım modelleri geliştikçe, UNIX bu yeni platformlara uyarlandı ve eski deneyim, yazılım birikimleri zarar görmeden yeni bilgisayar nesillerine taşındı. Bu uyumluluğun yararını gören bilgisayar üreticilerinin neredeyse tamamı, işletim sistemi repertuarlarına UNIX'i eklemek zorunluluğunu hissettiler. Hatta birçok bilgisayar üreticisi dev firma, kendi UNIX türevlerini geliştirdiler. AIX (IBM), ULTRIX (DEC), HPUX (HP), SINIX (SIEMENS) gibi...

Biraz da Tarih...

1960'lı yıllarda kullanılan bilgisayarlar, ancak “Sıralı İş Düzeni”nde (*Batch Processing*) çalışabilmekteydi. Bir diğer deyişle, kullanıcılar ve programcılar, bilgisayarda yapmak istedikleri işle ilgili komut ve/veya programları bilgisayarın operatörüne teslim ederler ve sıranın kendi işlerinin yapılmasına gelmesini beklerlerdi. Bu sıra artık onbeş dakikada mı yoksa üç günde mi gelir, bilinmezdi.

Bu yıllarda, üç önemli kuruluş (AT&T, MIT Üniversitesi ve General Electric) bir arada yürüttükleri bir projeye ilk “Zaman Paylaşımı İşletim Sistemi” (*time sharing operating system*) üzerinde çalışmaya başladılar. Proje, bir bilgisayarın bir anda birden fazla kullanıcıya hizmet etmesini sağlayan, kullanıcıların bilgisayar programında olup bitenleri izleyebileceği ve programlarla etkileşimli (interactive) olarak çalışabilecekleri bir ortam yaratmaya yönelikti. Çalışmalar sonunda MULTICS işletim sistemi ortaya çıktı. (*MULTiplexed Information and Computing System*) Her şey akademik olarak çok iyiydi; fakat, MULTICS yazılımı, o zamanki bilgisayarlar için biraz büyük ve hantal kalıyordu.

MULTICS ekibiyle birlikte çalışan ve uzay araştırmalarında kullanılan benzetim (*simulasyon*) yazılımları üretmekte olan Ken Thompson hayatından pek memnun değildi. Proje arkadaşları, onun üzerinde çalıştığı programların sistem kaynaklarını çok zorladığından sürekli şikayet ediyorlardı. Bu yüzden, Thompson, sadece başkalarının bilgisayarı kullanmadığı zamanlarda çalışabiliyordu. Bu böyle devam edemezdi. Thompson, çalışmalarını kendisine ait olan eski ve küçük bir DEC PDP-7 bilgisayarıda tamamlamaya karar verdi. Ama bu bilgisayarın işletim sistemi de gereksinimlerini karşılamıyordu; bu yüzden kendi istekleri ve gereksinimleri doğrultusunda bir işletim sistemi geliştirmeye koyuldu. MULTICS'in yararlı bulduğu ve beğendiği özelliklerinin tümünü kullandı. Hatta, o kadar ki, yeni işletim sisteminin adını MULTICS'den esinlenerek UNICS koydu. Bu ismin sonradan UNIX'e dönüştürüldüğü Brian Kernighan (C Programlama dilini ve UNIX'i yaratan ekibin önemli isimlerinden) tarafından anlatılmaktadır.

1970 yılında UNIX işletim sisteminin ilk sürümü DEC PDP-7 modeli bir bilgisayarda tamamlanmıştı. İşletim sistemi, programcılar için yararlı olacak şekilde tasarlanmış ve özellikle metin işleme yetenekleri (*text processing*) oldukça gelişmişti. 1971 yılında Bell Labs şirketi UNIX işletim sistemini, yeni metin işleme sistemlerinde kullanılacak standart olarak kabul etti. 1972 Haziran ayında gelindiğinde, artık dünyada 10 kadar bilgisayar UNIX işletim sistemi ile çalışmaktaydı. Bu arada, Dennis Ritchie ve Brian Kernighan, C programlama dili üzerindeki çalışmalarını büyük ölçüde tamamlamışlardı. 1973 yılında, UNIX işletim sistemi, C programlama diliyle baştan yazıldı.

Böylece bilgisayar tarihinin “yüksek seviyeli bir dil ile yazılmış olan ve donanımdan bağımsız” ilk işletim sistemi ortaya çıkmış oldu.

1974 yılından başlayarak, AT&T şirketi, bu yeni işletim sisteminin kaynak programlarını, başta Columbia Üniversitesi olmak üzere birçok üniversite ve yüksek okula ÜCRETSİZ olarak dağıttı. UNIX işletim sisteminin önlenemez yükselişi başlamıştı. (Aslında o tarihlerde bu yükselişi şimdiki gibi önlemek isteyen kimse olduğunu da sanmıyoruz.) 1975 yılına gelindiğinde, AT&T, UNIX Sürüm 6'yı kullanmaktaydı ve artık UNIX kullanmak isteyenler, küçük de olsa bir lisans ücreti ödemek zorundaydı. UNIX, standart bir C kütüphanesi ile birlikte dağıtılmaya başlandı. Böylece; C dili, UNIX işletim sistemi için yazılım geliştirmek isteyenlerin öğrenmesi gereken bir dil olarak yaygınlaştı.

1977 yılında, Berkeley Üniversitesi, ilgilenenlere UNIX üzerindeki birikimlerini, “1BSD: 1st Berkeley Software Distribution” adlı bir ürün olarak dağıtmaya başladı.

1978 yılında Bill Joy ve Özalp Babaoğlu (University of California-Berkeley’de yüksek lisans öğrencisi) UNIX işletim sistemine sanal bellek (*virtual memory*) özelliğini eklediler. Artık UNIX tam bir işletim sistemi olmuştu. (Ref: Unix Administration Guide for System V, Rebecca Thomas, ISBN 0-13-942889-5)

1979 yılında, AT&T yedinci sürümü piyasaya çıkardı. UNIX’in yaratıcılarından Ken Thompson’un Berkeley Üniversitesi’nde ders vermeye başlamasıyla AT&T ve Berkeley ekipleri UNIX’i hızla geliştirmeye başladılar. Sonunda, ABD Savunma Bakanlığı’na bağlı DARPA (Defence Advanced Research Projects Agency: İleri Savunma Araştırma Projeleri) bölümü, UNIX için bir bütçe ayırmaya karar verdi.

1979’da UNIX artık iyice yaygınlaşmıştı. Üniversite yıllarında UNIX öğrenen, kullanan ve beğenen öğrenciler UNIX’i sanayiye taşımaya ve donanım üreticileri, tasarım aşamalarında UNIX işletim sistemini de göz önünde bulundurma zorunluluğunu hissetmeye başladılar.

1980 yılı sonunda, büyük bilgisayar üreticilerinin hepsi, hiç değilse bazı modellerinde, UNIX kullanmaya başladılar.

1983 yılında MIT Üniversitesi’nden Richard Stallman “özgür yazılım” (*free software*) kavramı etrafında GNU organizasyonunu kurdu. Bu organizasyon; bilimin hızlı gelişmesini, bilim adamlarının yaptıkları araştırmaların sonuçlarını makaleler şeklinde yayınlamalarına, başka bilim adamlarının da bunlardan yararlanarak yeni araştırmalara yönelmesi şeklinde açıklanabilir. GNU organizasyonu yazılım dünyasında da benzeri şekilde paylaşım olması durumunda, bilişim dünyasındaki zaten hızlı olan gelişmelerin insanlık yararına daha da hızlanacağını savunuyor.

Richard Stallman, “özgür yazılım”ı tanımlarken “özgür” sözcüğünü

- Bir yazılımı istediğiniz gibi kullanma, kopyalama ve dağıtma özgürlüğü,
- Bir yazılımı istediğiniz gibi değiştirme, geliştirme özgürlüğü,

- Geliştirilmiş yeni yazılımı da istediğiniz gibi çoğaltma ve dağıtma özgürlüğü olarak tanımlıyor.

Stallman'a "GNU ne demek; neyin baş harfleri?" diye sorulduğunda "GNU is not UNIX!" yanıtını vermiş. Ancak bir bilişimciye yaraşacak, özdevinimli (*recursive*) bir yanıt!

LINUX Doğuyor

1991 yılında Hollandalı üstad Prof. Andrew S. Tanenbaum, öğrencilerinin işletim sistemlerinin ayrıntılarını ve işlevlerini daha iyi görebilmeleri hatta geliştirebilmeleri için 8086 serisi mikroişlemciler için MINIX işletim sistemini geliştirdi. Kodu tamamen açık ve dolayısıyla herkesin kullanabileceği ve geliştirebileceği bu minik işletim sistemi tamamen UNIX'den esinlenilmişti. O yıllarda internet üzerinde MINIX kullanıcı ve programcılarının çeşitli teknik tartışmalarını yürüttükleri haber grupları vardı. Bir gün bu gruba şöyle bir mesaj geldi:

From: torvalds@klaava.Helsinki.FI (Linus Benedict Torvalds)
Newsgroups: comp.os.minix
Subject: What would you like to see most in minix?
Summary: small poll for my new operating system
Message-ID: <1991Aug25.205708.9541@klaava.Helsinki.FI>
Date: 25 Aug 91 20:57:08 GMT
Organization: University of Helsinki



Hello everybody out there using minix -
I'm doing a (free) operating system (just a hobby, won't be big and professional like gnu) for 386(486) AT clones. This has been brewing; since april, and is starting to get ready. I'd like any feedback on things people like/dislike in minix, as my OS resembles it somewhat (same physical layout of the file-system (due to practical reasons) among other things). I've currently ported bash(1.08) and gcc(1.40), and things seem to work. This implies that I'll get something practical within a few months, and I'd like to know what features most people would want. Any suggestions are welcome, but I won't promise I'll implement them :-)

Linus (torvalds@kruuna.helsinki.fi)
PS. Yes - it's free of any minix code, and it has a multi-threaded fs. It is NOT portable (uses 386 task switching etc), and it probably never will support anything other than AT-harddisks, as that's all I have:-).

Helsinki Üniversitesi öğrencilerinden Linus Torvalds bu mesajında MINIX kullanıcılarına seslenerek, özetle hobi olarak 386 ve 486 serisi kişisel bilgisayarlara yönelik yeni ve ücretsiz bir işletim sistemi üzerinde çalıştığını ve artık bir yerlere gelmeye başladığını söylüyordu. GNU C derleyicisini ve **bash** kabuk kodunu yeni ortama aktarmayı başardığını ve artık birkaç ay içinde pratik bazı sonuçlar almayı umduğunu da ekliyordu. MINIX kullanıcılarından bu yeni işletim sistemi hakkındaki görüşlerini bildirmelerini isteyen Torvalds, gelebilecek önerilerin hepsini dikkate alamayacağını da belirtirken o zamanki disk arabirim standardı olan AT disk arabirimi dışında diskleri destekleyecek modülleri büyük olasılıkla hiçbir zaman geliştirmeyeceğini de eklemişti.

Linus Torvalds bu yeni işletim sistemine kendi adından esinlenerek LINUX adını verdi. Eylül 1991'de ilk sürümü olan LINUX 0.01 Net'te yerini aldı. 20 gün gibi kısa bir süre içinde o kadar çok değişiklik ve düzeltme geldi ki, 5 Ekim 1991'de Linus Torvalds 0.02 sürümünün dağıtımına başladı. Haftalarla ölçülen sürelerle diğer sürümler takip etti. LINUX'un kodu kolektif olarak geliştiriliyordu. Birileri disket sürücülerini destekleyen kodu geliştirirken başkaları da o zamanların en gelişmiş grafik standartları olan VGA (640 x 480, 256 renk) desteği üzerinde çalışıyordu. Herkes bir ucundan tutmuş, her iki-üç haftada bir yeni sürüm çıkartılıyordu.

Aynı tarihlerde, Richard Stallman ve kurduğu FSF (Free Software Foundation, Açık Yazılım Derneği), "GNU işletim sistemi" adı altında bir araya getirdiği birçok özgür yazılım ile yeni bir projeyi hayata geçirmektedir. Bu işletim sisteminin en önemli eksiklerinden biri yeterince olgun bir çekirdeğe (kernel) sahip olmamasıydı. İşte Torvalds'ın yarattığı ve hızla gelişmekte olan Linux, bu önemli eksiği kapatarak ve Genel Kamu Lisansı (GPL: GNU General Public License) altında korunarak bu projeye katıldı. Bu iki proje, o gün bugündür, birlikte "GNU/Linux işletim sistemi" olarak anılmaktadır.



Genel Kamu Lisansı'na sahip yazılımlar İngilizce "free" sözcüğüyle tanımlanırlar. Bu "free" sözcüğü "bedava" anlamında kullanılmamıştır; "özgür" anlamında kullanılmıştır. GNU lisansı altındaki bir ürünün ücreti olabilir ama mutlaka "özgür" olmalıdır. Özgürlükten kastedilen; ürünün özgürce kullanılması, özellikle kaynak kodunun serbestçe dağıtılması ve bu kaynak kod üzerinde serbestçe geliştirmelerde bulunulabilmesidir. GKL lisansına sahip bir program geliştirildiğinde, yeni ve gelişmiş sürüm de otomatik olarak GKL altında olacaktır; yani kodu geliştirenler yeni kaynak kodlarını da isteyen herkese gene "özgür" olarak vermelidir.

GKL'nın Türkçe tam metnini bu kitabın sonundaki eklerde bulacaksınız; ancak sabırsız okuyucular için ana hatlarıyla GKL mantığı şöyledir:

GKL ile dağıtılan programların kaynak kodu isteyen herkese açık olmalıdır. İsteyen herkes bu kaynak kod üzerinde istediği değişikliği yaparak daha gelişmiş veya farklı bir ürün elde edebilir; ancak bu yeni ürün de GKL altında olmalıdır. Bir yazılım ürününü (program, dokümantasyon, kavram, tasarım) GKL altında serbestçe dağıtabilirsiniz ancak bu ürün üzerinde emeği geçenlerin isimlerini değiştiremez veya yok edemezsiniz. Daha ayrıntılı bilgi isteyenler bu kitabın sonundaki GKL metnini okuyabilir veya www.gnu.org adresine başvurabilir.



LINUX Çeşitleri

Aslında tek çeşit LINUX vardır; o da GNU-LINUX'tur. LINUX dünyasına bir kez ayağınızı attıktan sonra sürekli olarak "Mandriva LINUX", "Mandrake LINUX", "Debian LINUX", "Suse LINUX", "Gelecek LINUX", "Ubuntu LINUX", "RedHat LINUX"... duyacaksınız.

Bu isimler, LINUX çeşitlerinin isimleri değil, çeşitli GNU-LINUX dağıtımlarının isimleridir. "*Dağıtım*" sözcüğü üzerinde durulması gereken bir sözcük, çünkü artık bu sözcüğü çok duyacak ve çok kullanacaksınız.



Aslında, FSF'in kurucusu Richard Stallman, "GNU/LINUX" işletim sisteminin yalnızca "LINUX" olarak anılmasından haklı olarak çok şikayetçidir. "LINUX" sadece çekirdeğin adıdır. Ancak, GNU/LINUX kullanıcıları, geliştiricileri ve konuyla ilgili yazarlar uzun yıllardır kısaca "LINUX" demeyi tercih ettiler. Artık bu alışkanlığı değiştirmek olanaksız hale geldi. Biz de, Stallman'ın affına sığınarak, doğru adı "GNU/LINUX" olmasına rağmen, bu kitabın konusu olan işletim sistemini "LINUX" adıyla anacağız. "LINUX" dediğimizde sadece çekirdekten değil, tüm işletim sisteminden söz ediyor olacağız.



Bildiğiniz gibi üzerinde hiçbir uygulama programı olmayan bir işletim sistemi fazla işe yaramaz. Olası tüm uygulama programlarını da işletim sistemi üreticisinden bekleyemezsiniz. Bu durumda birilerinin işletim sistemini yaygın kullanılan uygulama programlarıyla birlikte paketleyip bir "dağıtım"

oluşturması gerekir. Örneğin Microsoft, XP işletim sisteminin dağıtım paketini oluştururken içine XP işletim sisteminin yanısıra Internet Explorer, Outlook, MediaPlayer, Wordpad gibi bir sürü uygulama paketini de yerleştirir. Eğer standart paketin içinde yer almayan uygulama programlarına gereksiniminiz varsa, onları ayrıca satın alırsınız.

İşte Mandriva, RedHat, SuSE, Debian, Ubuntu gibi firma ya da kuruluşların kendi LINUX dağıtım paketlerini hazırlarken yaptıkları da aynen budur.

Her dağıtım kuruluşu kendi kurulum yazılımını geliştirir, dağıtım paketinde yer almasını uygun gördüğü uygulama programlarını pakete ekler (örneğin web tarayıcı olarak Mozilla Firefox, Netscape, Galeon, Konqueror gibi yazılımları, masa üstü yönetimi için KDE, GNOME gibi yazılımları seçip dağıtımlarına yerleştirir) ve dağıtımı piyasaya sürer. Gerek LINUX, gerekse birçok uygulama yazılımı GKL ile dağıtılan “özgür” yazılımlar olduğu için bunlarla oluşturulan dağıtım paketleri de GKL ile dağıtılır. Neredeyse hiçbirine para ödemek zorunda olmazsınız. Mandriva dağıtımını mı beğendiniz; gider www.mandriva.com'dan en son sürüm dağıtımı indirirsiniz. Eğer internet bağlantınızın hızı uygun değilse ya da yanında kağıda basılmış kurulum kılavuzu istiyorsanız o zaman az bir bedel ödeyip dağıtımın CD ve kitaplarından oluşan bir paket ısmarlayabilirsiniz.

Çeşitli LINUX dağıtımları arasında küçük ve önemsiz farklar olmasına rağmen bazıları diğerlerine göre daha başarılıdır. Örneğin, kurulum programının tanıyabildiği donanım unsurları (ses kartı, video kartı gibi) zenginliği açısından en beğenilen dağıtım **SuSE**'dir. Kurulum açısından en kolay olarak nitelendirilen dağıtım Mandriva'dır. Kurulum sırasında en fazla seçenek sunan dağıtım ise RedHat olarak kabul edilir. Biz bu kitap için kurulumu en kolay olan Mandriva'yı seçtik.

Bir kez kurulduktan sonra, tüm dağıtımlar neredeyse birbirinin aynı LINUX bilgisayarını oluşturur. Örneğin, Mandriva dağıtımında Apache web sunucusu ile ilgili dosyalar **/var/www** dizini altına yerleştirilirken aynı dosyalar bir başka dağıtımda **/home/httpd** dizini altına yerleştirilebilir. Sonuçta her ikisi de web sunumunu Apache ile yapar, Apache ayar dosyalarının isimleri ve içerikleri aynıdır ama yerleri değişiktir.

Dağıtımlar arasında gözleyeceğiniz farklara bir başka örnek de, LINUX ayarlarını grafik arabirimlerle yapmak için dağıtım firmalarının geliştirdikleri yazılımlardır. Örneğin Mandriva'nın "Mandriva Control Center" isimli grafik arayüzlü destek programı yeni başlayanlar için büyük kolaylıktır.

Sözün kısası, LINUX öğrenmek ve kullanmak açısından dağıtımın Mandriva mı olduğu, yoksa Debian mı olduğu pek fazla farketmez. Ancak size önerimiz sık sık dağıtımınızı değiştirmemenizdir. LINUX yüzlerce dizin içine yerleştirilmiş binlerce dosyadan oluşan büyük bir işletim sistemidir ve çeşitli denetim dosyalarını elinizle koymuş gibi bulabilmek önemlidir. Sık sık dağıtım değiştiren sistem yöneticilerinin ortak sıkıntısı, aradıklarını kolay bulamamalarıdır.

Bir de "Free BSD" UNIX Varmış...

Evet var! LINUX'la hiçbir ilgisi olmayan ama LINUX gibi özgür olan bir UNIX dağıtımı daha vardır. İşin "aslına" bakarsanız UNIX profesyonelleri sunucu olarak kullanılacak Intel tabanlı bilgisayarlarına genellikle FreeBSD kurarlar. Hem performans hem güvenlik açısından FreeBSD'nin, GNU-LINUX'a göre üstün olduğu söylenir. Ancak FreeBSD'yi kurmak zordur; hele UNIX dünyasına yeni adım atmış olanlar için... FreeBSD'nin bir de masa üstü işlevlerine ilişkin desteği zayıftır. Örneğin birçok ayar ve kurulum yazılımının grafik arabirimi yoktur. FreeBSD, uzaktan ve/veya konsoldan denetlenecek, veritabanı sunucusu, web sunucusu, e-posta sunucusu gibi hizmetlerde kullanılacak bilgisayarlar için idealdir.



UNIX Geleneği

Çok geniş bir araştırmacı kitlesi tarafından geliştirilmesine rağmen UNIX, ilk tasarımı olduğu günlerdeki özelliklerinden pek uzaklaşmamıştır. Bunun en önemli nedeni, bu araştırmacıların yazılı olmayan geleneklere bağlı kalmış olmalarıdır. Belki de UNIX, başarısını bu gelenekselleşmeye borçludur. (Japon'ların ekonomik mucizesinin de geleneklere bağlılık olduğu söylenmez mi?) Örneğin, dizinlerdeki dosyaların detaylı listesini veren **ls** komutunun 100 Megabyte'dan büyük dosyalarda ortaya çıkan hatası hala düzeltilmemektedir. Her ne kadar çok tutucu bir tablo çizmiş olsak da, 1990'lı yıllarda iki ayrı UNIX ekolü olduğundan söz etmek gerekmektedir: Berkeley Üniversitesi'nin yürüttüğü BSD ekolü ve AT&T şirketinin yürüttüğü AT&T UNIX (SVR4: System 5 Release 4) ekolü. Bu iki tip UNIX, kullanıcıları açısından

pek önemli farklılıklar göstermese de, sistem yöneticileri açısından çok farklıdır. 1992 yılından başlayarak AT&T UNIX'i geliştiren ekipler, BSD UNIX'in üstün özelliklerini AT&T UNIX ile birleştirerek SVR4 UNIX'i ortaya çıkardılar ve BSD ekolüne göre önemli bir üstünlük kazandılar.

UNIX'i UNIX Yapan Özellikler

Belki bazı noktalar tekrar edilmiş olacak ama, UNIX'i UNIX yapan özellikleri bir kez daha sıralamak istiyoruz. Kitabın okunması sırasında ve daha önemlisi LINUX işletim sistemi'ni kullanırken yararlı olacağı inancındayız.

- UNIX “çok kullanıcı” bir işletim sistemidir. Kullanıldığı bilgisayarın bir anda birden fazla kişi tarafından kullanılmasını; daha doğrusu paylaşılmasını sağlayabilmektedir.
- UNIX “çok iş düzeni”ni sağlayan bir işletim sistemidir. Kullanıcıların her birinin, aynı anda birden fazla iş yapmalarına olanak sağlar.
- UNIX, donanımdan bağımsızdır. Hangi bilgisayar üzerinde kullanılırsa kullanılsın, kullanıcılarına görüldüğü şekli aynıdır. Öğrendikleriniz kalıcıdır.
- UNIX iyi tasarlanmıştır. Teknolojideki gelişmelere kolaylıkla uyum sağladığı ve sağlayacağı kanıtlanmıştır.
- UNIX, bir işletim sistemi standardı olarak kabul edilmiştir. Bu sayede farklı marka ve model bilgisayarlar birbirleriyle uyumlu kılınabilmektedir. İnternet'in bu denli hızlı gelişmesi ve yaygınlaşması UNIX'in getirdiği standartlar sayesinde olmuştur.

LINUX'u LINUX Yapan Özellikler

Aslında LINUX, UNIX üzerine geliştirilmiş bir işletim sistemi değildir. LINUX işletim sisteminde bir satır bile UNIX kodu kullanılmamıştır. UNIX kavramları kullanılarak tasarlanmış olmasından gelen özelliklerin yanısıra LINUX'un belki de en önemli özelliği arkasındaki destektir. Sorununuz ne olursa olsun, dünyanın bir yerlerinde size yardımcı olabilecek birileri mutlaka bilgisayarlarının başında çalışmaktadır. Sorunuzu doğru tartışma listelerine ilettiğinizde en fazla 10-15 dakika içinde yanıt alırsınız. İşte biraz da bu yüzden kitabı Mandriva dağıtımının İngilizce kurulumu üzerine yazdık. Japonya'daki bir LINUX uzmanının sorunuzu anlayabilmesi için soruyu İngi-

lizce olarak kaleme (pardon, klavyeye) almak zorundasınız. Bunun için de İngilizce LINUX terminolojisini bilmelisiniz.

Bir başka önemli özelliği hızlı ve sürekli gelişmesidir. Her hafta LINUX için yeni ve çok kullanışlı bir yazılım ortaya atılmaktadır. Bunlar kimi zaman güvenlikle ilgili, kimi zaman yeni bir internet servisiyle ilgili olmaktadır. Bu yazılımlar çoğunlukla da “özgür” olmaktadır. Rahatlıkla, lisans endişesi olmadan bilgisayarınıza kurup deneyebilirsiniz.

LINUX güvenlidir. Herşeyden önce virüs derdi yok denecek kadar azdır. UNIX işletim sisteminin çok kullanıcı ortamlarda işletim sisteminin kendini kullanıcılara karşı korumak için kullandığı standart güvenlik yöntemleri, sistemleri virüslere karşı da rahatlıkla koruyabilmektedir.

LINUX bilgisayarları “kırmak” kolay değildir. Kaynak kodunun açık olması sayesinde LINUX işletim sisteminin yeni eklenen modülleri yüzbinlerce meraklı tarafından didik didik edildiği için güvenlik gedikleri çok çabuk bulunmakta ve düzeltilmektedir.

Daha saymaya devam edersek kitaba bir türlü başlayamayacağız.

Neden Mandriva LINUX?

Açıkçası en kısa yol bildiğiniz yoldur da ondan! Baba-oğul uzun süredir Mandrake ve Mandriva dağıtımlarını kullanıyoruz. Artık hangi ayar dosyası hangi dizinde bulunur, ayarların varsayılan değerleri nedir; iyice alıştık. Bu alışkanlıklar da bizi sadık birer Mandriva kullanıcısı yaptı. Bir başka LINUX dağıtımına geçmek için şimdilik hiçbir neden göremiyoruz. Üstelik Mandriva’nın kurulumu kolay; çok sayıda değişik görüntü kartını, ses kartını sorunsuz destekliyor. İşletim sistemini güncelleştirmek; yani gerek işletim sisteminin modüllerinin, gerekse uygulama yazılımlarının yeni sürümlerini yüklemek çok kolay. Eh, daha ne isteyeceğiz ki...

Neden İngilizce Kurulum?

“Tarih biçim resmi tüm izin girişi çevrilmeden önce son bulur.”

Bu cümle ORACLE veritabanı yönetim sistemi yazılımının Türkçe’leştirilmiş bir sürümünün hata mesajlarından birisidir. İşte bu yüzden “İngilizce dağıtım”...

Çeviri yapmak zor iştir. Hele sözcük sayısı, karakter sayısı gibi sınırlamalar varsa...

Aslında Mandriva dağıtımı, gerek kurulum gerekse kullanım aşamalarında Türkçe'yi gayet güzel destekleyen bir pakettir. Türkçe desteği derken yalnızca Türkçe'ye özgü karakterlerden söz etmiyoruz; mesajlarıyla, açıklamalarıyla, başlıklarıyla Türkçe desteğinden söz ediyoruz.

Bu güçlü desteğe rağmen kitabımızdaki tüm örnekleri, tüm ekran görüntülerini İngilizce kurulum ve kullanım için verdik. Bunun en önemli nedeni İngilizce'nin bilgisayar dünyasının standart dili olması ve bu işle ciddi olarak uğraşanların İngiliz dilini olmasa bile bu dildeki bilgisayar terimlerini bilmesinin kaçınılmaz bir zorunluluk olmasıdır.

İşletim sisteminizi kurduğunuzda başka hiçbir yazılım, kitap veya İnternet kaynağına gereksinim duymayacağınıza emin olabilseniz sorun olmayacaktır ama ne yazık ki gerçekler böyle değil. Hadi LINUX'unuzu Türkçe LINUX olarak kurdunuz diyelim... GIMP uygulamasını ne yapacaksınız, Java VM paketini ne yapacaksınız?

Günün birinde çözmeye uğraştığınız bir sorunla ilgili olarak LINUX tartışma listelerinde bir Japon ya da Yeni Zelandalı tarafından yazılmış notlarda

*"Check your file system;
make sure that the partition table is correct;
make sure that your users are all configured to use the bash shell"*

gibi cümleler göreceksiniz. Gerek genel bilgisayar terimlerinin, gerekse LINUX terimlerinin İngilizcesini biliyorsanız bu notlardan yararlanarak sorununuzu çözmeye şansınız çok yüksektir. Eğer bu gibi İngilizce belgelerde yazılanlar size hiçbir şey ifade etmiyorsa çözümün Türkçe destek sitelerine ve yardım belgelerine yansımaları beklemek zorundasınız demektir.

Aslında şöyle bir düşünürseniz ana dili İngilizce olanların da bilgisayar başına oturduklarında işleri zor: *http, apache, ftp, telnet, file system, mouse, root* gibi sözcükler (ya da bizim dünyamızda sözcükleşmiş kısaltmalar) onlara da pek bir şey ifade etmemektedir. Örneğin UNIX dünyasında pek yaygın kullanılan "root" sözcüğünün günlük İngilizce'deki "root" sözcüğüyle yakından uzaktan bir ilgisi yoktur.

Sözün kısası: Bilişim dünyasının kendine özgü bir dili var ve bu dil tarihsel nedenlerle İngilizce'den türemiş. Bu dili konuşamayanların ve yazamayanların da bilişim dünyasında başarı şansı yok denecek kadar az; onun için ya “*bilingilizce*” öğreneceksiniz ya da bu işten vazgeçeceksiniz.

Pusula Yayıncılığın “İngilizce-Türkçe Ansiklopedik Bilişim Sözlüğü” (ISBN 975-6477-03-2) elinizin altında olursa kendinizi daha rahat hissedebilirsiniz.

LINUX'u “Gerçekten” Öğrenmek İsteyenlere Öğütler

Bilgisayarınızdaki diğer işletim sistemlerini silin ve bir daha yüklemeyin. Bilgisayarınızda LINUX öncesi, yapmaya alıştığınız işlerin hepsini LINUX altında da yapabilirsiniz. Başlangıçta zorluklarla karşılaştığınızda hemen havlu atmayın. LINUX konusunda başkalarıyla yazışabileceğiniz, soru sorabileceğiniz, sohbet edebileceğiniz sitelerle ilgilenin; listelere üye olun. Kitap, dergi okuyun.

İşte size birkaç başlangıç noktası:

- linux.org.tr
- lkd.org.tr
- gelecek.com.tr

Mandriva LINUX Kurulum Öncesi Hazırlıklar

Kurulum Öncesi Bilinmesi Gerekenler

Nasıl Bir Bilgisayar Gerekli?

Bütün yemek tarifleri malzeme listesiyle başlar. Biz de öyle yapalım bari...

- En az 128 Mbyte bellek, 256 MB olsa iyi olur.
- Pentium, Celeron, Athlon, Duran, Sempron, K6 ve Via C3 işlemci.
- 3-4 GigaByte disk, 40 GB veya daha fazlası daha da keyifli olacaktır.
- 1024x768 çözünürlükte 256 renk destekleyecek bir görüntü kartı. (800x600 de olur)

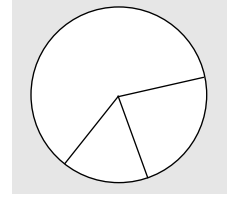


- Fare.
- Ağ bağlantısı için bir Modem veya Ethernet arabirimi, ADSL.
- CD sürücüsü olan bir bilgisayar işinizi görecektir.

Daha güçlü bir bilgisayar elbette daha keyifli olacaktır. LINUX'u diski boş bir bilgisayara; daha doğrusu bilgisayarın boş bir diskine kurmak daha güvenli olacaktır; ancak şart da değildir. Eğer bilgisayarınızda tek bir disk varsa; bu diskte de 3-4 GByte boş bölüm (partition) varsa LINUX'u bu boş alana kurabilirsiniz. Büyük olasılıkla bilgisayarınızda şu anda Windows ailesinden bir işletim programı kurulmuş durumdadır ve doğal olarak diskinizde boş bir bölüm yoktur. Dikkat edin; boş “yer” demiyoruz, boş “bölüm” diyoruz.

Disk Bölümleri (Partition)

Fiziksel bir diski, örneğin 4 bölüme ayırarak, sanki 4 ayrı diskmiş gibi kullanmak olasıdır. Her diskin üstünde bir “bölümlendirme tablosu” (partition table) kayıtlıdır. Bu tablo bölümlenmemiş disklerde bile vardır çünkü “bölümlenmemiş disk” aslında “tek bölümlü” bir disktr.



Disk bölümlendirmenin önemli birkaç yararı vardır. İlk başta, kapasitesi kullanılan işletim sisteminin destekleyebildiği disk kapasitesini aşan çok büyük diskleri parçalamakta kullanılır. Örneğin, 30 Gbyte'ın üstündeki diskler Windows 98 ikinci sürüme kadar sorun yaratmaktaydı.

Disk bölümlendirme disk yedeklemeyi de kolaylaştırır. Şimdilik bunun ayrıntılarına girmeyelim; böylece kabul edin lütfen.

Disk bölümlendirmenin bu aşamada bizim için en önemli yararı bir diske birden fazla işletim sistemi yükleyebilme olanağını yaratmasıdır. Örneğin iki bölüme ayrılmış bir diskin ilk bölümünde Windows ailesinden bir işletim sistemi; ikincisinde de LINUX yer alabilir.

Windows yüklü diskleri veri kaybetmeksizin sonradan bölümlendirmek kolay değildir; daha doğrusu standart Windows araçlarıyla bunu yapamazsınız. Bu iş için **Partition Magic** diye ayrı bir Windows uygulama yazılımı vardır, onu kullanabilirsiniz.

Windows 9x ve Windows ME sürümleri ile NT tabanlı Windows 2000 ve Windows XP sürümlerinin disk formatlama yöntemleri farklı olabilmektedir. **NTFS (NT File System)** adı verilen disk formatı ile Windows 9x serisinin kullandığı FAT16 ve FAT32 formatlarının veri bloklarını disk üzerinde yerleştirilme yöntemleri çok farklıdır. Piyasada sonradan disk bölümleyebilen birçok yazılım olmasına rağmen NTFS serisi formatları da sorunsuz destekleyen Partition Magic yazılımını öneririz. Aslında daha da iyisi LINUX için bilgisayarınıza ek bir disk takmanızdır.

Eğer mevcut diskinizi eski işletim sistemini ve veri/program dosyalarını bozmadan bölümlendirecekseniz diskinizde LINUX'a ayırabileceğiniz en az 3-4 Gbyte'lık boş alan olmalıdır. Bölümlendirme işini yaparken diskte iki yeni bölüm (eskiyle birlikte toplam 3 bölüm) oluşturmalısınız. Eski işletim sistemini içeren ilk bölümün ardından 128-256MB arası bir takas alanı bölümü (*swap partition*) oluşturup kalan bölümü LINUX'a ayırabilirsiniz.

Takas alanı, LINUX tarafından ana belleğin yetmediği durumlarda yardımcı ana bellek gibi kullanılacaktır. Ortalama bellek erişim sürelerinin mikrosaniyelerden milisaniyelere fırlaması yüzünden takas alanını kullanmak zorunda kalan yazılımların çalışması çok ama çok yavaşlayacaktır; ama öte yandan bu yavaşlık programların hiç çalışmamasından daha iyidir.

Eğer Windows işletim sisteminizi korumak istiyorsanız ve tek bir diskle çalışıyorsanız, bölümlmeden sonra diskinizin haritası şöyle olabilir.

Windows için ayrılan bölüm (Ne kadar gerekiyorsa)	Takas alanı*	LINUX dosya sistemleri için ayrılan bölüm (3-4 Gbyte)
--	--------------	--

Eğer LINUX için ayrı bir disk kullanacaksanız ya da diskinizdeki eski işletim sistemini korumak istemiyorsanız bu diskin bölüm haritası şöyle olabilir:

Takas alanı*	LINUX dosya sistemleri için ayrılan bölüm (3-4 Gbyte)
* Takas alanının ana belleğin iki katı olması önerilir. Ancak disk kapasitesini uygun değilse 256 MB'a kadar inilebilir.	

LINUX için ayrı bir disk kullanma olanağınız varsa, bu diskin bölümlenmesini ve takas alanı yaratılması işini LINUX'un kurulumu sırasında yapabilirsiniz.

Takas alanı için yaklaşık olarak bilgisayarın hafızasının iki katı civarında bir yer ayrılır. 256 Mbyte günümüz bilgisayarları için son derece uygun bir takas alanı kapasitesidir.

Her disk üzerinde yapılmış olan bölümlendirmeye ilişkin bilgiler o diskin belli bir alanında “Bölümlendirme Tablosu” (Partition Table) adı altında kaydedilir. Bölümlenmemiş diskler aslında tek bir bölüm olarak düzenlenmiş gibi değerlendirilirler; yani onların da birer bölümlendirme tablosu vardır. Bir diskin bölümlendirmesi değiştirilirse, yeni harita bölümlendirme tablosuna kaydedilmelidir. Bölümlendirmeye ilgili düzenlemeleri ve yeni tablonun kaydedilmesi işini bölümlendirme işlemlerini yaparken kullanacağınız programlar (**fdisk** gibi) yapacaktır.

LINUX'un kurulması sürecinde bilmenizde yarar olan önemli kavramlardan biri de “boot süreci”dir.

Bilgisayarların Açılış Süreci (Boot Süreci)

“Boot” sözcüğü İngilizcedeki “boot strapping” deyiminin kısaltılmışıdır. “Boot strapping” deyimini ise bir insanın botlarının bağcıklarından çekerek kendini havaya kaldırması anlamında bir benzetmedir. Aslında bir bilgisayarın açılış sürecini gerçekten andıran çok hoş bir benzetmedir bu...

Kapalıyken belleğinde hiçbir program yüklü olmayan bir bilgisayarın, koca bir işletim sistemini belleğe yükleyip işe yarar hale gelmesi süreci gerçekten ilginçtir.

Bilgisayarı ilk açtığınızda BIOS (Basic Input-Output System) entegre devresinde (**ROM: Read Only Memory**) kayıtlı olan küçük bir program, işletim sistemi yüklemekte kullanılabilecek disket, disk gibi aday yan bellekleri belirler. Genellikle öncelik disket sürücüsündedir. Eğer disket sürücüde bir disket takılı değilse CD sürücü kontrol edilir; CD de takılı değilse birincil kontrol kanalındaki (Primary IDE Channel) ilk disk kontrol edilir. Bu sırası sisteminizin “Setup” menüsündeki araçlarla değiştirebilirsiniz.

BIOS, işletim sisteminin yükleneceği yan bellek birimini belirledikten sonra, yan belleğin tipine göre önceden bir standart olarak belirlenmiş bir adresteki kaydı belleğe okur. “Master Boot Record” (MBR) adı verilen bu kayıt 446 byte uzunluğunda bir program içerir. Diskte yüklü olan işletim siste-

minin bir parçası olan bu küçük program işletim sisteminin disk üzerinde nasıl yayıldığını bilen bir program olduğu için artık işletim sisteminin çekirdek modüllerinin (kernel) belleğe yüklenmesi kolaydır. LINUX işletim sisteminin MBR içine kaydettiği işletim sistemi çekirdeği yükleme programı olan **LILLO**'ya (**L**inux **L**Oader) ileride daha ayrıntılı bir şekilde değineceğiz.

LILLO, sistemin değişik disklerinde ve bölümlerinde yüklü olan değişik işletim sistemlerinin her birini ayrı ayrı yükleyebilir. Bu sayede bilgisayarınızda LINUX yüklenmeden önce yüklü olan işletim sistemlerini kaybetmezsiniz.

Disk Dosya Sistemleri (File Systems)

Her işletim sistemi bilgisayarın diski üzerinde kaydedilecek dosya ve dizinlerin disk üzerindeki fiziksel yerleşimlerini düzenleyebilmek için kendine göre bir dosya yapısı sistemi kullanır. Genel olarak “dosya sistemi” olarak adlandırılan bu yapılar disklerde dosya ve dizin yaratılması, bunlara hızlı bir şekilde erişilmesinin sağlanması, silinen dosyaların kullandığı alanların geri kazanılması gibi işlerin halledilmesine yönelik oldukça karmaşık veri yapılarıdır. Windows ailesinde işletim sisteminin sürümüne göre FAT16, FAT32 ve NTFS gibi dosya sistemleri kullanılır.

LINUX altında biçimlendirilecek (formatlanacak) diskler üzerinde yaratılacak dosya sistemleri için birden fazla seçeneğiniz vardır. İçinde bulunduğumuz yıllarda en sağlıklı ve gelişmiş dosya sistemleri “reiser-fs” ve “ext3” dosya sistemleridir. Yüksek performans gereken uygulamalar için kurulan ve kesintisiz güç kaynağıyla desteklenen sistemler için en uygun ve en popüler disk dosya sistemi ise “ext2” dosya sistemidir.

Reiser-fs ve ext3, “Journalized file systems” adı verilen bir kategoriye ait dosya sistemleridir. Bu dosya sistemlerinin en önemli özelliği, bilgisayarın güç kesintisi gibi nedenlerle kapanması durumunda disk dosya sisteminde bozukluk olmaması; dolayısıyla bilgi kaybı olmamasıdır. Dosya sistemi üzerinde birçok kaydın yedekli tutulması nedeniyle doğal olarak biraz yavaştır. Bilgisayarınız bir kesintisiz güç kaynağıyla desteklenmiyorsa disklerinizi **Reiser-fs** veya **ext3** kullanarak düzenlemenizi öneririz.

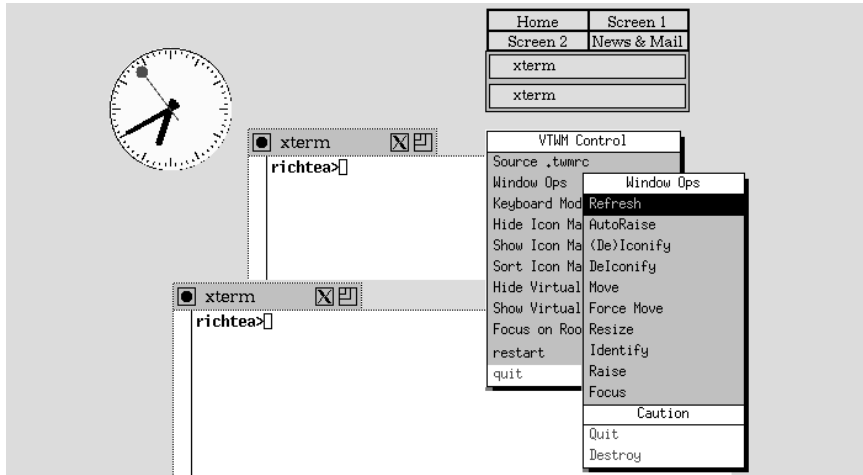
Şimdi de LINUX kullanırken oldukça sık karşılaşacağınız bazı önemli kavramlardan söz etmek istiyoruz. Bu kavramları baştan biraz olsun tanırsanız, gerek kurulum, gerekse kullanım sırasında çok rahat edersiniz.

X Window

Bildiğiniz gibi grafik kullanıcı arabirimleri kişisel bilgisayar pazarına uzun yıllar önce Apple Macintosh'larla girdi. Ekranda bir fareyle dolaşmak, ekranda yazıların ve resimlerin birlikte görülebilmesi, hatta yazıların da resim olarak değişik yazı tipleriyle yazılabilmesi doğal olarak bilgisayar kullanımını çok büyük ölçüde kolaylaştırdı. Kolaylaştırmak bir yana, yepyeni uygulama alanlarına yol açtı.



Genel kanının aksine ilk grafik kullanıcı arabirimi Macintosh'la ortaya çıkmadı; ama Apple bu kavramı “kişisel bilgisayarlara” taşıyan ilk firma oldu. Massachusetts Institute of Technology (MIT) üniversitesinde 1984 yılında başlatılan Athena projesi 1986'da ilk ürünlerini vermeye başladığında **X Window** (kısaca X) adıyla anılmaya başlamıştı. Belki inanmayacaksınız ama UNIX dünyasının grafik kullanıcı arayüzü hala bu X üzerine kuruludur. Elbette o günden bugüne çok şey değişti ama temel ilkeler hala aynı. UNIX'in geleneksellikçi felsefesi sayesinde o ilk yıllarda yazılmış olan programların neredeyse tamamı hala yeni X sürümleri ile uyumludur.



Alçakgönüllü bir X ekranı.

X Window, veya LINUX dünyasında daha yaygın olarak kullanılan adıyla **Xorg** standardı bilgisayarın ekranında grafik öğeler oluşturabilmek için kullanılan bir yazılım alt yapısı ve bu alt yapıya hükmetmek için geliştirilmiş bir

yazılım kütüphanesidir. Bu nedenle X (veya Xorg) tek başına kullanıcının pek bir işine yaramaz.

X'in mekanizmasını anlamak için üç kavramı çok iyi sindirmelisiniz: “X Sunucusu”, “X İstemcisi” ve “X Protokolü”.

X Sunucusu (X Server): “X Sunucusu” genellikle çok yanlış anlaşılan bir terimdir. “X Sunucu” herşeyden önce bir yazılımdır. X grafik kullanıcı arabirimi hangi bilgisayarın ekranında grafik görüntü oluşturacaksa o bilgisayarda bir X Sunucu yazılımı çalışmalıdır. X Sunucu yazılımı ekranın çeşitli grafik uygulamalar arasında paylaşılmasını sağlayan; bir başka deyişle grafik ekranda bir anda birden fazla pencere açılmasını ve bu pencerelerin içinde yazı-resim görüntülenmesini sağlayan, klavye ve farenin denetimini üstlenen yazılımdır.

X İstemcisi (X Client): Yazılma amacına hizmet edebilmek için bir ekrana grafik çizme gereksinimi duyan, fare ve klavyeden komut kabul edecek uygulama yazılımlarına “X istemcisi” denir. Örneğin zamanı akrepli yelkovanlı bir saat resmiyle göstermek isteyen bir uygulama programı ve Netscape web tarayıcısı birer X istemcisidir. X istemcileri, grafik ekranda bir değişiklik yapmak istediklerinde bu isteklerini grafik ekranı denetleyen X sunucusuna iletirler. X sunucusu da, denetlemekte olduğu grafik kartı ile görüşüp görüntü değişikliğinin monitöre yansımısını sağlar. Çarpıcı bir şey söyleyelim mi? X sunucusu ile X istemcisi farklı bilgisayarlarda çalışıyor bile olabilirler.

X Protokolü: X istemcilerinin grafik birimlere çizim yapma isteklerini X sunucularına aktarmak için kullanılan protokoldür. X istemcileri, X sunucularından çizim işlerini yapmalarını isteyebilecekleri gibi; X sunucuların, denetledikleri grafik arabirimler hakkında bilgi göndermelerini de isteyebilirler (renk zenginliği, çözünürlük gibi). X Protokolü tek bir bilgisayar üzerinde çalışan X sunucusu ile X istemcileri arasında yürütülebileceği gibi TCP/IP üzerinden farklı bilgisayarlar arasında da yürütülebilir. “Nasıl yani?” diyorsanız kısacası şöyle: Ankara’da Bilkent Üniversitesi’nde yer alan bir bilgisayar üzerinde çalışan CAD yazılımının grafikleri Boğaziçi Üniversitesi’ndeki bir bilgisayarın ekranına alınabilir. Böyle bir durumda X sunucusu Boğaziçi Üniversitesi’ndeki bilgisayarda; X istemcisi olan CAD yazılımı da Bilkent Üniversitesi’ndeki bilgisayarda çalışıyor olacaktır.

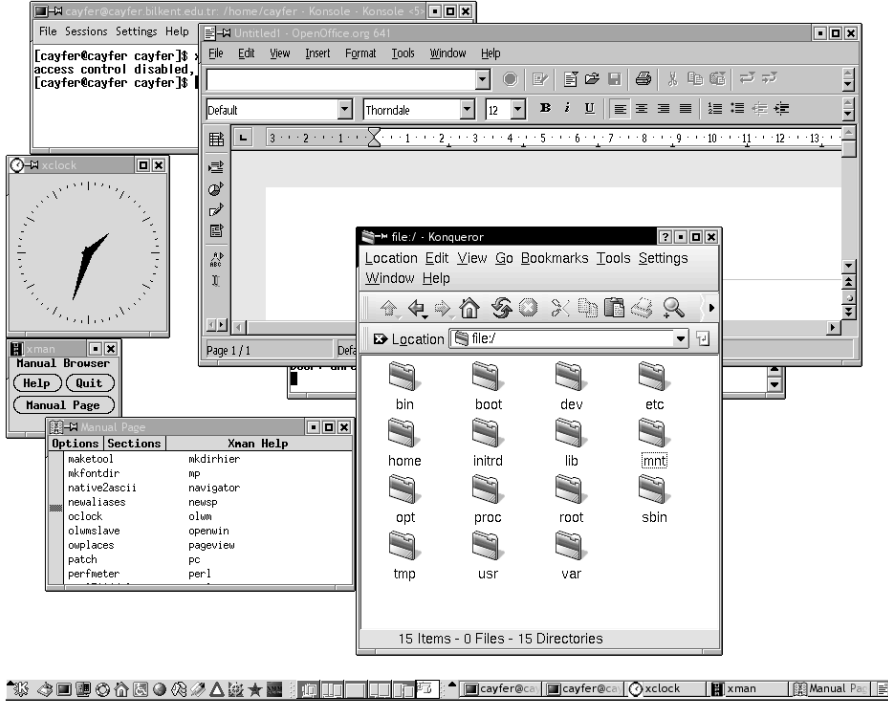
Evet! Birkaç yıldır size “yeni teknoloji” diye sunulan işletim sistemlerinde buna birazcık benzer bir iş yapabilmek için fazladan birkaç bin dolar ödemeniz gerekiyor! Oysa X, 1985’den beri bu işi mükemmel bir şekilde **ücretsiz** ve **özgür** olarak yapıyor.

X istemcileri, bir grafik çizmek istediklerinde “çember çiz, mavi kare çiz, kesikli çizgi çiz, Times Roman yazı tipiyle şu karakterleri göster” gibi temel X komutlarını X protokolü ile bir X sunucusuna gönderirler. Eğer yazılımın çalışma mantığı açısından gerekliyse, önce bu grafiklerin nasıl bir ekranda çizileceklerini de öğrenebilirler.

Bu istekleri alan X sunucusu, istemcinin tanımlanmış yetkileri uygunsa bu grafik unsurları kendi denetiminde olan ekrana çizer. Farenin hareketleri ve klavyede basılan tuşlar hakkında gereken bilgiler de X sunucusu tarafından istemciye gönderilerek uygulama yazılımının davranışı denetlenebilir.

Bir X sunucusu birden fazla bilgisayarda çalışan X istemcilerinin pencerelerini barındırabilir. Yani kendi ekranınızda beş değişik pencere içinde, beş değişik coğrafi bölgede yer alan beş değişik bilgisayarda çalışan beş değişik grafik uygulama programının ekranlarını barındırabilirsiniz.

Yönetmekle sorumlu oldukları sunucuda önemsiz bir parametre değiştirdikten sonra veya yeni bir program yükledikten sonra sistemi “Restart” etmek için ya da yedekleme başlatmak için gece yarısı şirkete giden NT sistem yöneticilerinin kulakları çınlasın...



Bu ekranda gördüğünüz her bir pencere, farklı bir bilgisayarda çalışmakta olan uygulama programlarına aittir. Bir başka deyişle, dört ayrı bilgisayarın ekranı tek bir ekrana toplanabilmektedir.



X Window Pencere ve Masaüstü Yöneticileri (Window Managers, Desktop Managers)

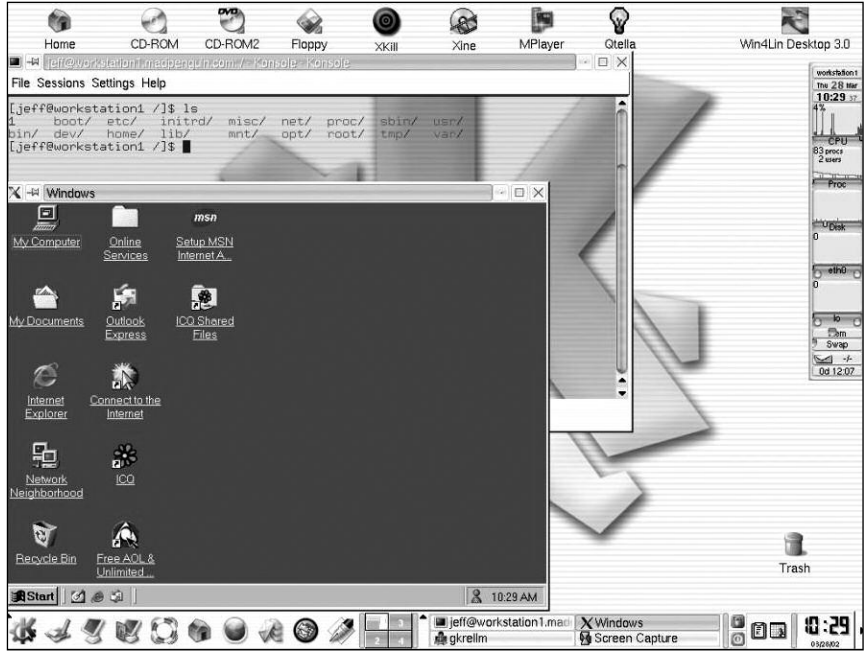
X altında çok sayıda pencere açmak, bunları yönetmek, küçültüp büyütme, ekranda gezdirmek için geliştirilmiş yazılımlara **pencere yöneticisi** denir. Pencere yöneticilerinin bir görevi de pencerelerin ve ekran fonunun estetik özelliklerini düzenlemektir. Standart X altında pencerelerin görsel özellikleri son derece basit ve çirkindir ancak kullanıcılar ekranlarının görsel özelliklerini pencere yöneticilerinin yardımıyla kendi zevklerine göre düzenleyebilirler.

X dünyasının önemli pencere yöneticileri arasında “sawfish”, “enlightenment”, “ICEwm” sayılabilir. Normal koşullarda LINUX kullanıcıları bu

pencere yöneticilerini doğrudan kullanmazlar. Kullanıcının doğrudan iletişimde bulunduğu yazılımlar **masa üstü yöneticileri**'dir.

X altında çalışan pencere yöneticileri altında çeşitli ikonlar, menü sistemleri ile kullanıcılara daha kullanışlı ve anlamlı ekran görüntüleri sunan X uygulamalarına **masa üstü** yöneticisi denir. LINUX dünyasının en önemli masa üstü yöneticileri KDE ve GNOME yazılımlarıdır. Masa üstü yöneticileri, pencere yöneticileri altında çalışır. Pencere yöneticileri de X altında...

Bu kitaptaki tüm ekran görüntüleri KDE masa üstü yöneticisi kullanılarak hazırlanmıştır. LINUX'un kurulumunun bir aşamasında sizin de bir masa üstü yöneticisi seçmeniz gerekecektir. Bu kitapla uyumlu olması açısından KDE seçmenizi öneririz. Zaten istediğiniz zaman değiştirme şansınız olacaktır.



Yukardaki örnek KDE ekranında dikkatinizi çekmek istediğimiz iki öge var: Biri terminal penceresinin yarı saydam olması diğeri de ön plandaki pencerenin bir Windows-98 bilgisayarına ait olması. Yarı saydam uygulama pencereleri, uygun fon resimleri ve renklendirmeyeyle çok hoş masa üstü görüntüleri verebilmektedir. www.kde-look.com adresinde yüzlerce değişik masa üstü teması bulabilirsiniz.

Ön plandaki Windows ekranına gelince... Evet, LINUX altında Windows işletim sistemini çalıştırabileceğiniz gibi birçok Windows uygulamasını da kullanabilirsiniz. Bu iş için VMWare, Win4Lin ve Wine gibi LINUX yazılımları vardır. Ancak, Wine hariç hepsi ücretlidir.

LINUX Çekirdeği (LINUX Kernel)

Her işletim sistemi bir çekirdek ve onun etrafına toplanmış onlarca (UNIX söz konusu olduğu zaman binlerce) komuta ilişkin programlardan oluşur. İşletim sisteminin çekirdeği, sistemin açılmasıyla belleğe yüklenen ve işletim sistemi çalıştığı sürece bellekte kalan yazılım modülleridir. Çekirdeğin temel görevleri arasında bellek yönetimi, merkezi işlem biriminin paralel çalışan süreçler arasında paylaşılmasının koordine edilmesi (zaman paylaşımı), çeşitli arabirimlerin (IDE ve SCSI disk arabirimleri gibi) yönetim ve denetimi sayılabilir.

Normal koşullarda bir LINUX kullanıcısının işletim sistemi çekirdeğini (*kernel*) kurcalaması gerekmez. Ancak, çok ender de olsa bazı durumlarda çekirdeği değiştirmeniz veya değişik bazı yetenekleri ekleyebilmek için yeni baştan derlemeniz gerekebilir. Örneğin, LINUX altında MS Windows 9x işletim sisteminin çalıştırılmasını sağlayan Win4Lin isimli bir yazılım vardır. (www.netraverse.com) Bu yazılım standart LINUX çekirdeklerinde çalışmıyor, daha doğrusu çalışabilmek için çekirdekte bazı eklemeler ve değişiklikler gerektiriyor. Bu programı satın alırsanız, ya LINUX'unuza uygun bir çekirdeği firmanın ftp sunucusundan indirmeniz ya da bu firmanın sayfalarında açıklanan yöntemlerle çekirdeğinizin kaynak kodunu değiştirip yeniden derlemeniz gerekecektir.

LINUX; kaynağı açık bir işletim sistemi olduğu için kaynak kodlarını indirip, birtakım değişiklikler yapıp yeniden derlemek ve yeni işlevlerle donatılmış bir çekirdek elde etmek mümkündür. Bu bilgi bir kenarda dursun; belki lazım olur.

root

UNIX dünyasına yeni girenlerin kısa bir süre için de olsa alışmakta zorluk çektikleri iki anlamlı bir UNIX terimidir.

Eğer insanlardan (kullanıcılardan) bahsedilirken kullanırsa, **root** sözcüğü (“ruut” okunur) “süper kullanıcı” yani “sistem yöneticisi” anlamına gelir.

Tüm UNIX ve LINUX kurulumlarında adı “root” olan bir kullanıcı vardır. Bu kullanıcı sistemdeki tüm kaynakları kullanmaya, tüm dosya ve dizinlerde her türlü değişikliği yapmaya yetkilidir.

“root” sözcüğü disklerden, dizinlerden söz ederken kullanılırsa bir UNIX bilgisayardaki disk dosya sisteminin en üst noktası anlamına gelir. Aslında bu noktanın adının okunuşu “root” dur ama yazılışı “/” dır. Windows işletim sisteminde her diskin C:\, D:\ ile gösterilen kendi “root dizin”leri olmasına karşın UNIX’de durum biraz farklıdır. En önemli fark, üzerinde kaç disk takılı olursa olsun her UNIX bilgisayarda tek bir “/ dizini” olmasıdır. İkinci önemli fark da, root dizinin sembolünün “\” değil, normal bölüm işareti olan “/” olmasıdır.

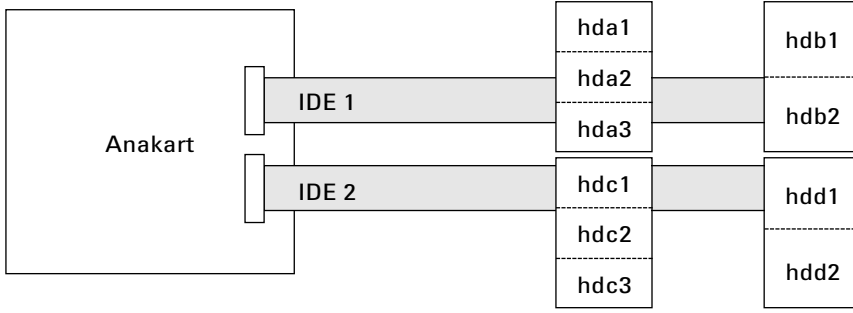
LINUX’ta Disk İsimlendirmesi

Mikrobilgisayar dünyasının ilk profesyonel işletim sistemlerinden biri olan CP/M işletim sisteminin oluşturduğu alışkanlıktan olsa gerek, Microsoft işletim sistemlerinde disklere A:, C:, D: gibi harf ve “iki nokta üstüste”den oluşan isimler verilmiştir. Oysa UNIX dünyasında durum çok farklıdır.

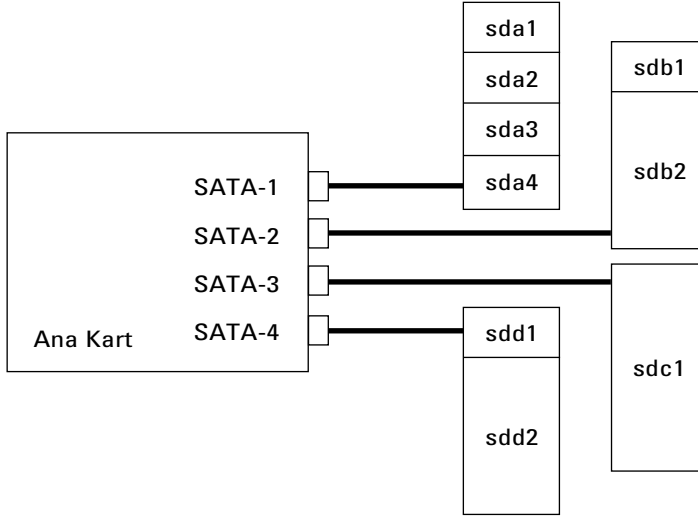
Fiziksel IDE arabirimli disk ve CD sürücüler **hda**, **hdb**, **hdc**, **hdd** gibi isimlerle anılır. SCSI diskler ise **sda**, **sdb** gibi isimlerle... Disket sürücüler ise **fd0**, **fd1** gibi isimlerle... Ancak bu son cümleler sizi yanıltmasın. Bu isimler yalnızca fiziksel birimlerden söz ederken kullanılır. Bu ortamlardaki dizin ve dosyalara erişirken disk, CD ve disket sürücülere kendi uygun göreceğiniz isimlerle ulaşacaksınız.

Şimdilik bu isimlendirme işinin ayrıntılarına girmeyeceğiz. Bu aşamada bilmeniz gereken birincil (*primary*) IDE kanalı üzerindeki ilk (*master*) disk ya da CD sürücünün adının **hda**, gene birincil kanaldaki ikinci birimin (*slave*) adının **hdb**, ikincil kanalın ilk diskinin **hdc** ve sonuncu diskin de **hdd** olarak isimlendirileceğidir.

SCSI diskleriniz varsa onlar da SCSI adreslerine göre **sda**, **sdb** gibi isimlendirilecektir. Disk üzerindeki bölüm isimleri ise **sda1**, **sda2**, ... olacaktır.



SCSI çok esnek bir arabirim standardıdır. Sabit disklerden teyplere, tarayıcılardan yazıcılara kadar birçok donanımda kullanılabilmektedir. SCSI arabirim denetim modülleri de (driver) bir o kadar esnektir. Bu nedenle yeni disk teknolojileri ile ilgili çekirdek modülleri genellikle SCSI arabirim sürücülerine eklenmektedir. Örneğin son yıllarda hızla yaygınlaşan SATA disk arabirim standardı da LINUX işletim sisteminde bir SCSI arabirimi gibi denetlenmekte; bunun için de SATA disk sürücüler sisteminde **sda**, **sdb** gibi isimler altında görünmektedir.



Kurulum öncesi bu kadar teori yeter. Şimdi Mandriva LINUX dağıtımının üç CD'sini ve bir tane boş disketi hazır edip kurulumu başlayabilirsiniz.

BUNLARI BİLİYOR MUYDUNUZ?

UNIX Tasarım İlkeleri (Başarının sırrı mı yoksa...)

LINUX'un çok işlemcili bilgisayarlarda da rahatlıkla kullanılabildiğini biliyor muydunuz? Üstelik 3 işlemciden sonra performansın artmaması gibi bir sorun olmaksızın!

Peki... Çok sayıda (sınırsız sayıda) tek işlemcili LINUX bilgisayarı paralel çalıştırarak bir "süper bilgisayar" yapabileceğinizi biliyor muydunuz? Beowulf veya Openmosix yazılımıyla binlerce LINUX bilgisayarı paralel işlemcili tek bir bilgisayar gibi kullanarak (elbette uygun programlama araçları ve teknikleri gerekecektir) çılgın bir işlem gücü elde edebilirsiniz.

Mandriva LINUX

Kurulumu

02

- Varsayımlar
- Kurulum

Varsayımlar

Bilgisayarınızda donanım özellikleri olarak

- Birincil IDE kanalına “master” olarak takılmış 20 Gbyte kapasiteli bir disk sürücü,
- İkincil IDE kanalına “master” olarak takılmış bir CD sürücü,
- 1024 x 768 çözünürlükte en az 256 renk destekleyen bir grafik kartı,
- Bir ses kartı (ya da ana kart üzerinde bütünleşik bir ses arabirimi) ve hoparlör,
- 128 MByte bellek,
- Tekerli bir fare,
- Bir ADSL bağlantısı,
- Bir disket sürücü,

olduğunu varsayıyoruz.

Ayrıca, sisteminizde Windows XP yüklü olduğunu, buna rağmen 10 GByte boş yer olduğunu, bu diskteki değerli dosyaları yedeklemiş olduğunuzu ve Partition Magic veya eşdeğeri bir programla diskinizi üç bölüme ayırmış olduğunuzu varsayıyoruz.

10 GByte (WinXP bölümü)	512 MByte (Takas alanı)*	9.5 GB (LINUX için ayrılan alan)
* Takas alanı olarak ana belleğin (RAM) iki katı önerilir. Daha fazla olmasında bir sakınca yoktur.		

Diskinizin 10 GByte'lık ilk bölümünde XP işletim sistemi, ikinci 512

MByte'lık bölümünde takas alanı olarak ayrılmış bir alan ve son bölümünde de LINUX için ayrılan alan.



Gelişmiş işletim sistemlerinde bilgisayarın belleği, çalışan programlar arasında dinamik olarak paylaştırılır. İşletim sistemi, belleğe gereksinim duyan programlara bellek verir; işi biten bellek alanlarını da geri alır. Ancak, öyle anlar olur ki, çalışan programların istedikleri toplam bellek, sistemin sahip olduğu bellekten fazladır. Modern işletim sistemleri bu durumlarda çalışmakta olan programlar arasından uygun gördükleri birini ya da birkaçını bellekten diske taşıyarak kalan programlara yer açar. Biraz sonra bellekte kalmayı başaran programları diske indirip, az önce diske taşınmış olan programları belleğe geri alıp çalıştırmaya kaldığı yerden devam ettirir. Bütün bu işler saniyenin küçük kesirlerinde yapılabilse de bilgisayarın çalışmasını büyük ölçüde yavaşlatır. Bu "sanal bellek" (virtual memory) uygulaması aslında yukarda anlattığımızdan elbette daha karışıktır ama ilke olarak bu şekilde; bellekteki programların zaman zaman diske atılarak bellek kazanmaya çalışılması, "bellek yetersizliğinden programları tamamen durdurmak zorunda kalmaktansa sistemi yavaşlatmak daha iyidir" düşüncesidir. Bellekten diske ve diskten belleğe program parçalarının taşınıp durması için diskte ayrılan alana "takas alanı" adı verilir.

Kurulum

Evet, hazırsanız başlayalım...

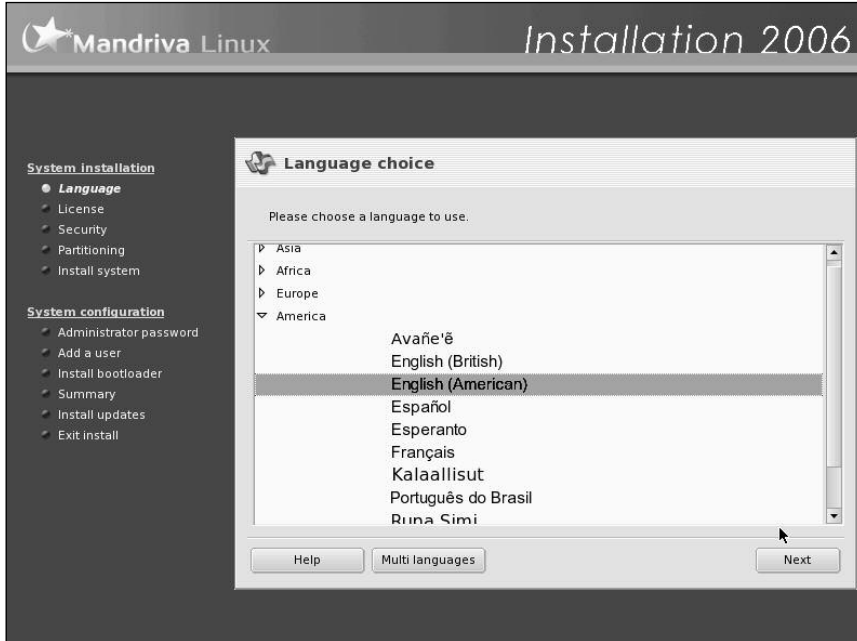
Sisteminizin BIOS ayarlarından önce-likli boot sürücüsünü CD-ROM sürücü olarak seçmeyi unutmadan Mandriva dağıtım CD'lerinden **Install** etiketli olan ilkini sürücüye takıp bilgisayarınızı açınız. Birkaç saniye içinde ekranınızda...



Kurulumu devam etmek için “Enter” tuşuna basınız. Mandriva LINUX dağıtımını kurmanın tek yolu CD’den kurmak değildir. LINUX’unuzu bir diskten veya ağ üzerinden erişebildiğiniz bir bilgisayardan da kurabilirsiniz. Hangi kurulum yöntemi seçilmiş olursa olsun başladıktan sonra gerisi aynı olduğu için biz CD’den kurulumu anlatacağız.

Önce Mandriva LINUX’u hangi dille kurmak ve kullanmak istediğinize ilişkin seçiminizi yapmalısınız. Karşınıza gelecek listeden “English – American” seçebileceğiniz gibi “Turkish” de seçebilirsiniz. LINUX’unuzun dilini “Turkish” olarak belirlerseniz tüm kurulum ekranları ve daha sonra da işletim sisteminin pek çok metni karşınıza Türkçe olarak gelecektir. Ancak Mandriva dağıtımının Türkçeleştirmesi henüz tamamlanmadığı için bazı ekranlar/mesajlar İngilizce olacaktır.

Biz LINUX’unuzu İngilizce olarak kurmanızı ve kullanmanızı öneririz. Eğer bilgisayar ve LINUX terimlerinin İngilizcesini biliyorsanız veya öğrenirseniz zaten sorun yok. Yok eğer bilmiyorsanız, öğrenmeye başlamanın tam zamanıdır. UNIX ve LINUX’la ilgili bütün kitap ve dokümanlar Türkçeleştirilemediği için ve buna da pek olanak olmadığı için LINUX’a özgü terimlerin ve kavramların İngilizce’sini öğrenmek zorundasınız.



Kim Korkar LINUX'tan?

Bu dil seçme ekranının sol tarafındaki liste kurulum boyunca orada kalacak ve kurulumun hangi aşamasında olduğunuzu gösterecektir. Halen içinde bulunduğunuz kurulum aşaması beyaz renkli bir butonla gösterilir.

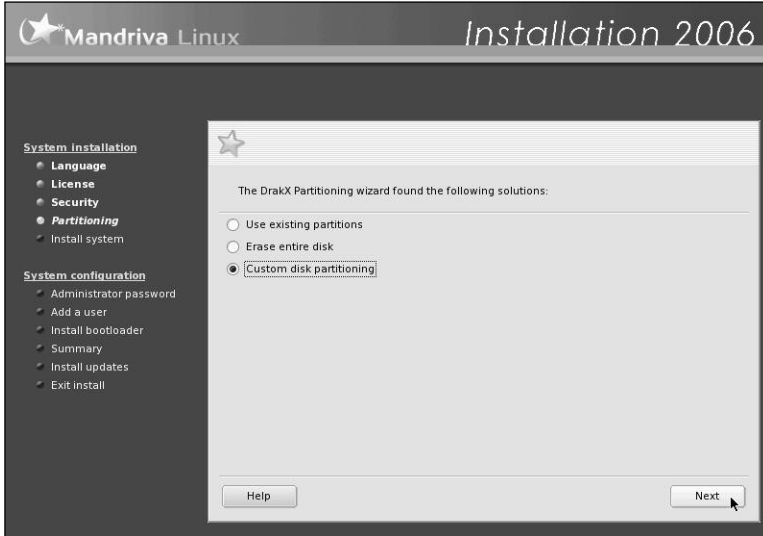


Bu da lisans anlaşmanız... Özet olarak Mandriva LINUX dağıtımının ve logolarının telif hakkının Mandriva S.A şirketine ait olduğunu, bu ekranda “Accept” seçimini yapıp “Next” butonunu tıklamakla bu lisans anlaşmasını kabul etmiş olacağınızı belirtiyor. Bu yazılımları kullanmanızdan ya da kullanamamanızdan dolayı uğrayabileceğiniz hiçbir zarardan kendilerinin sorumlu tutulamayacağını vurguluyor. Ayrıca, dağıtım paketindeki yazılımların çok büyük bir çoğunluğunun ve paketin kendisinin Genel Kamu Lisansı (GNU General Public License) ile dağıtıldığını belirtiyor.

GKL size biraz garip gelebilir. Bu lisans ile dağıtılan programları hiç kimseye para ödemek zorunda kalmadan kullanabilir ve istediğiniz kadar kopyalayıp istediğiniz gibi dağıtabilirsiniz. Microsoft yöneticilerinin “GKL” lafını duyunca neden bir hoş olduklarını şimdi anlıyor musunuz?



LINUX bilgisayarınızın kurulumu sırasında yapılacak güvenlik ayarlarının düzeyi soruluyor. “Standard” seçmenizi öneririz. Daha yüksek güvenlik düzeyleri yeni öğrenenler için biraz fazla güvenli oluyor. İkinci kutuda, sisteminizde çalışacak birtakım güvenlik denetim yazılımlarının raporlarını kime göndereceği soruluyor (*Security Administrator*). Bu kutuya kurulumun ileri aşamalarında kendinize açacağınız hesabın adını girebileceğiniz gibi başkasına ait bir elektronik posta adresi de girebilirsiniz. Önerimiz buraya kendi hesap adınızı (“**cayfer**” gibi) girmenizdir.



Kurulumun önemli bir aşaması... Diskinizin bölümlenmesiyle ilgili seçimlerinizi yapmak üzeresiniz.

- Eğer diskinizin tamamını LINUX'a ayırabilecekseniz “Erase entire disk” seçimini yapınız.
- Eğer diskinizin bölümlenmesi LINUX kurulumu için hazırsa; yani, bir tanesi takas alanı olmak, diğeri LINUX kurulmak üzere en az iki bölüm tanımlıysa “Use existing partitions” seçimini yapabilirsiniz.
- Eğer disk bölümlmeyi kendi kontrolünüz altında yapacaksınız (-ki bu seçimi öneririz) “Custom disk partitioning” seçimini yapınız.

Bundan sonraki ekran görüntüleri 20 GB diski olan, diskin ilk 10 GByte'ında Windows XP kurulu olan bir disk için verilecektir.

Bu noktada amacınız, ekrandaki bölümlenme haritasını aşağıda görünen şekle benzer bir şekle getirmek olacaktır. Yeni bölümlenme haritasında en az iki yeni bölüm yer almalıdır:

1. Takas alanı,
2. LINUX bölümü.

Disk bölümlenme haritası

Windows XP Bölümü	Takas Alanı	LINUX Bölümü
----------------------	-------------	--------------

Öncelikle “Takas Alanı”nı yaratmanızı öneririz. “Takas Alanı” tüm UNIX işletim sistemlerinde olduğu gibi LINUX'ta da çok önemlidir. Özel bir neden yoksa, takas alanı olarak ayıracağınız disk alanının boyu, bilgisayarınızın ana belleğinin iki katı kadar olsa yeter.



Disk bölümlemesini LINUX'a uygun bir şekilde getirmek için:

- Varsa Windows bölümü için bir “mount point” tanımlamanız,
- Bir takas alanı bölümü yaratmanız ve bölüm tipini “Swap” olarak atamanız,
- Kalan alan için de “mount point” olarak “/”; “File system type” (dosya sistemi tipi) olarak da “Journalised FS” tanımlamanız gerekmektedir.



Bu işleri yapmak için

şemasında ilgili disk bölümünü tıkladıktan sonra soldaki menüden

- “Resize” (boyunu değiştir)
- “Mount point” (bağlantı noktası)
- ve üstteki listeden “File system type” (dosya sistemi tipi)

seimlerini ve ayarlarını yapmanız gerekmektedir. Bu işlemleri yaparken önce bir disk bölümünü tıklayıp seçmek, sonra da ayarlarını yapmak gerekmektedir.

Bu işlemler sırasında disk bölümlerini silmek için “Delete”, boyutlarını deęiştirmek için “Resize”, boş disk alanına yeni bölüm yaratmak için de “Create” butonlarını kullanabilirsiniz.

Eğer birden fazla fiziksel diskiniz (disk birimi) varsa, bölüm haritasının üst tarafında “hda”, “hdb” (ya da “sda”, “sdb”) gibi başlıklar göreceksiniz. Hangi diskin bölümlenmesini yapmakta olduğunuza dikkat etmelisiniz.

“Done” butonunu tıklayıncaya kadar yaptığınız hiç bir deęişiklik diske ya da disklere kaydedilmeyecektir, rahat rahat çalışabilirsiniz.

Örneğin takas alanını yaratmak için ekrandaki bölümlenme haritasında boş disk alanını tıklayıp ardından “Create” butonunu tıklayınız. Sonra “File System Type” kutusunda “Linux Swap” seçip, bu bölüm için ayıracağınız alan için “Size in MB” ayarını yaklaşık 512'ye getiriniz ve “OK” butonunu tıklayınız.

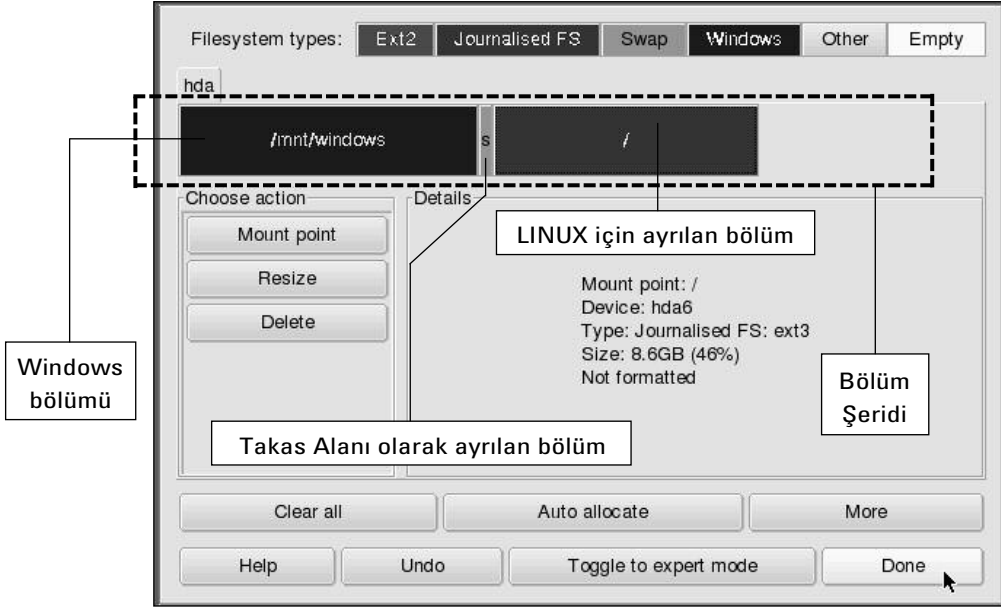
Takas alanını ayırdıktan sonra LINUX işletim sisteminin yerleşeceği bölümü tanımlamalısınız. Bunun için, haritada kalan boş alan üzerine tıklayıp “Create” butonunu tıklayınız.

Diskin kalan alanının tamamını LINUX'a ayırmak için “Size in MB” sürgüsünü en sağa çekiniz.

“Filesystem types” kutusunda “Journalised FS: ext3” seçili olsun, “Mount point” kutusunda da “/” işareti olsun.

Amacınız aşağıdakine benzer bir bölümlenme haritası elde etmek olmalı:

1. Hiç ellenmeyecek bir Windows bölümü (tabii varsa),
2. Ana belleğin iki katı kadar bir takas alanı ve
3. Diskin kalanı kadar bir LINUX bölümü.



- Bilgisayarınızda birden fazla disk sürücü varsa, yukarıdaki harita ekranında bu disklerin bölümlendirme tablolarını “hda”, “hdb”, “hdc”, “hdd” gibi başlıklar altında göreceksiniz.
- LINUX için kullanmayı düşündüğünüz diskte bir Windows bölümüne gereksiniminiz yoksa, o diskte yalnızca bir takas alanı ve LINUX bölümü yer alabilir.



Disklerinizin bölümlenmesini tamamladığınızda disk ya da disklerinizin sayısına ve eski işletim sistemini korumak isteyip istememenize göre aşağıdaki bölümlendirme haritalarından birini elde etmiş olmalısınız:

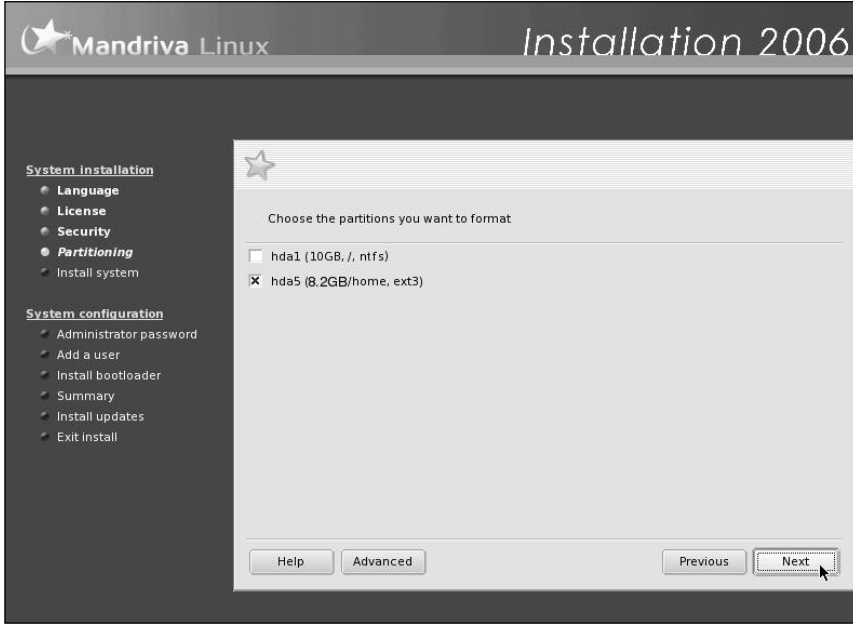
Tek Disk, Tamamı LINUX'a Ayrılmış			
Disk ve Bölümü	Kapasite	“mount” Noktası (mount point)	Dosya Sistemi Tipi (fs Type)
hda, hda5	200-400 MByte	swap	swap
hda, hda6	Kalan kısım	/	“Journalised FS:ext 3”

Tek Disk, Bir Kısım Windows, Kalan Kısım LINUX'a Ayrılmış			
Disk ve Bölümü	Kapasite	“mount” Noktası (mount point)	Dosya Sistemi Tipi (fs Type)
hda, hda1	Her ne ise	–	Win98 FAT32
hda, hda5	200-400 MByte	swap	swap
hda, hda6	Kalan kısım	/	“Journalised FS:ext 3”

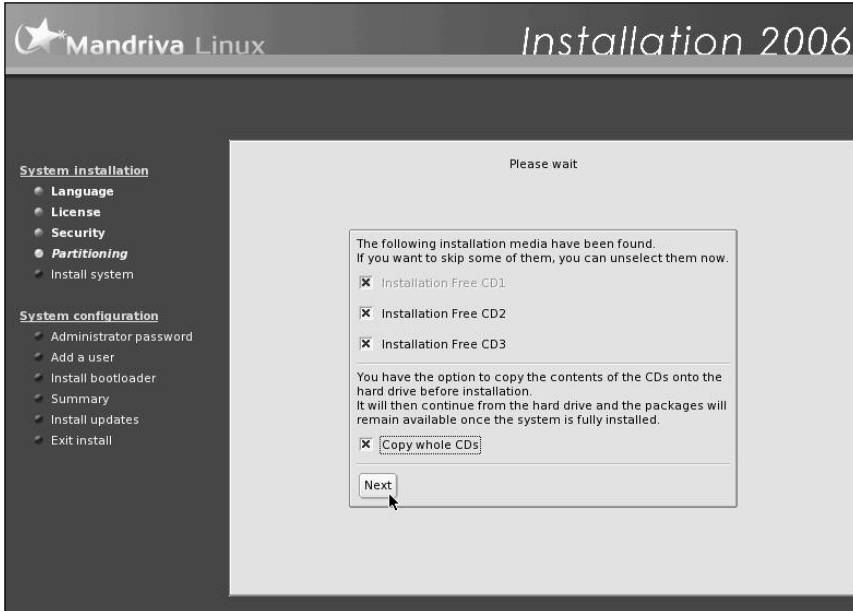
Tek Disk, Bölümleme Büyücüsü Tarafından Düzenlenmiş Bölümleme			
Disk ve Bölümü	Kapasite	“mount” Noktası (mount point)	Dosya Sistemi Tipi (fs Type)
hda, had1	1 GByte	/	“Journalised FS:ext 3”
hda, hda5	256 MByte	swap	swap
hda, hda6	1.5 GByte	/usr	“Journalised FS:ext 3”
hda, hda7	4.2 GByte	/home	“Journalised FS:ext 3”

İki Diskli PC, İkinci Diskin Tamamı LINUX'a Ayrılmış			
Disk ve Bölümü	Kapasite	“mount” Noktası (mount point)	Dosya Sistemi Tipi (fs Type)
hda, hda1	Her ne ise		Win98 FAT32
hdb, hdb5	200-400 MByte	swap	swap
hdb, hdb6	Kalan kısım	/	“Journalised FS:ext 3”

Bölümleme tablonuz hazır olunca “Done” butonunu tıklayınız. Yeni bölümleme tablonuzun diske kaydedileceğine ilişkin “Partition table of drive hda is going to be written to disk!” diye bir mesaj alacaksınız. “Ok” tıklayıp devam ediniz.

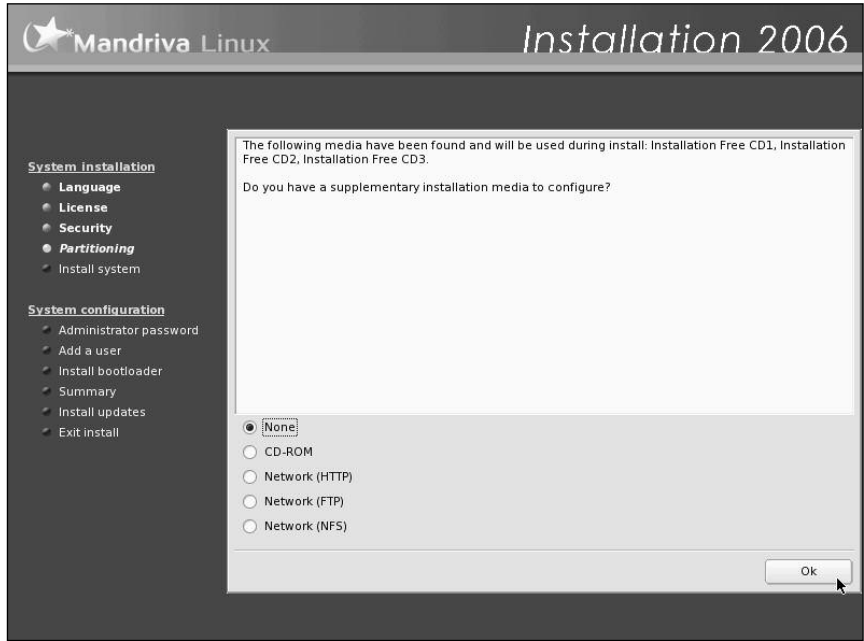


Şimdi yeni tanımlanan LINUX bölümü formatlanacaktır. Bu formatlama, bilgisayarınızın hızına bağlı olarak 10-20 saniye içinde tamamlanacaktır. Eğer diskinizde ayırabileceğiniz 2 GigaByte kadar fazladan yer varsa, Mandriva dağıtım CD'lerini bu aşamada diske oldukları gibi yüklemek iyi bir fikir olacaktır.



Kurulum CD'lerinin kopyası diskinizde yer aldığında çeşitli eksik programları yüklemek ya da bazı paketleri silip yeniden yüklemek istediğinizde kolaylık olacaktır. Eğer CD kopyaları diskinizde bulunmazsa, dağıtım içinden bir program yüklemek istediğinizde, programın içinde bulunduğu CD'yi takmanız istenecektir. İleride CD kopyalarının diskte harcadığı yer sizi rahatsız ederse /var/ftp/pub/Mandrivalinux/media/ altındaki media1, media ve media3 dizinlerini silebilirsiniz.

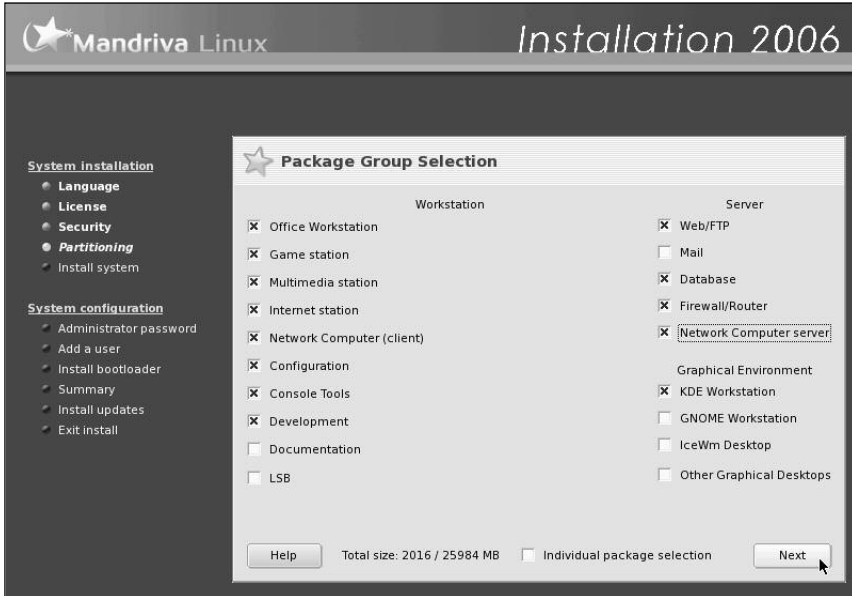
Eğer CD'lerin diske kopyalanmasını tercih ederseniz, önce üç dağıtım CD'sinden başka CD'niz olup olmadığı sorulacak (ki buna "yok - none" demelisiniz) sonra CD'ler diske birer birer kopyalanacak, kurulum ondan sonra devam edecektir.



Copying in progress
(Installation Free CD1)



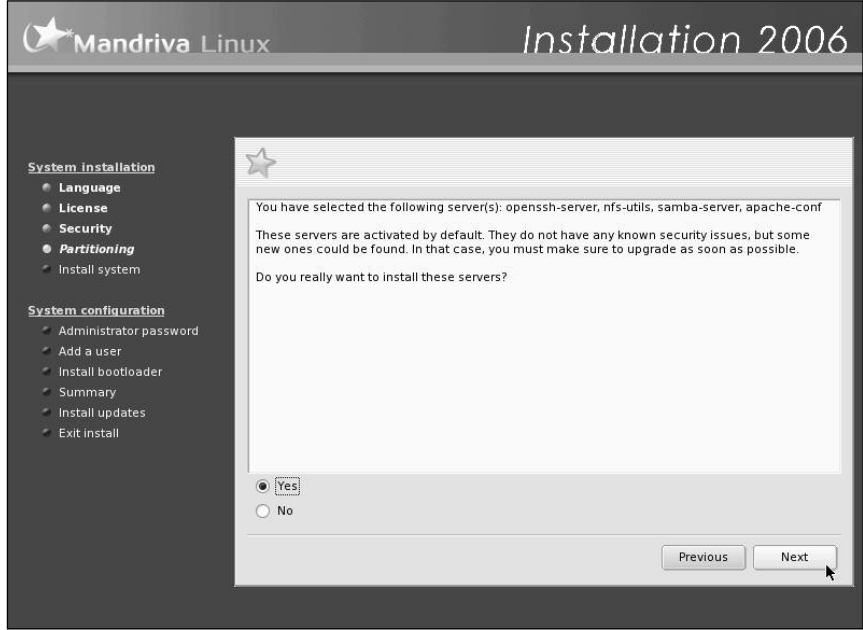
Kurulum işlemi başladığında kullanmayı düşündüğünüz uygulama gruplarını seçmeniz istenecektir. Seçimlerinizi yaparken disk kapasitenizi göz önünde bulundurmalısınız. Siz grup seçtikçe ekranın en altında “Total size” başlığı yanında seçimlerinizin diskte ne kadar yer tutacağı görüntülenecektir. Tipik ve bol keseden yapılmış bir seçim listesi örneğini aşağıdaki ekran görüntüsünde görebilirsiniz. Linux altında yazılım geliştirme çalışması yapmayı düşünmüyorsanız “Development” başlıklı grubu boşuna seçmeyin deriz. Değişik masaüstü denetim yazılımlarını denemek istiyorsanız “Graphical Environment” grubu altında “KDE Workstation” yanı sıra “GNOME Workstation” seçimini de yapmanızı öneririz.



Kim Korkar LINUX'tan?

Unutmayın! Eksikliğini hissettiğiniz paketleri sonradan kolayca yükleyebilirsiniz.

Seçtiğiniz uygulama yazılım grupları arasında bir sunucu yazılım varsa (ki büyük olasılıkla olacaktır), bir uyarı ekranı ile karşılaşacaksınız:



İşletim sistemi ne olursa olsun, sunucu servisleri çalıştırmak tehlikelidir. Sunucu yazılımda yer alabilecek bir hata ya da zayıflık sorun çıkarabilir. Dikkat ederseniz **çıkabilir** diyoruz. Piyasada, bir sunucu yazılım çalışmasına dahi gerek olmadan sorun çıkması garantili işletim sistemleri bulunduğunu belirtmemize herhalde gerek yok. Şaka bir yana, bu uyarı ekranında çekinmeden "Next" seçimini yapabilirsiniz.



Evet... Bir kahveyi hak ettiniz. Şimdi bilgisayarınızın hızına bağlı olarak yarım saat-kırk beş dakika kadar beklemeniz gerekecek. Kurulum CD'lerinin diske kopyalanmasını istemediyseniz arada sırada CD değiştirmeniz istenecektir.



Kim Korkar LINUX'tan?

Yukardaki ekran, LINUX sisteminizin **root** kullanıcısı için bir şifre seçmenizi istiyor. Hatırlarsanız adı **root** olan kullanıcı LINUX açısından özel bir kullanıcıydı. **root** isimli kullanıcı sistemde herşeyi yapmaya yetkilidir. İleride sistem yöneticisi yetkilerine gereksinim duyduğunuzda, **root** kimliğine bürünmeniz gerekecektir.

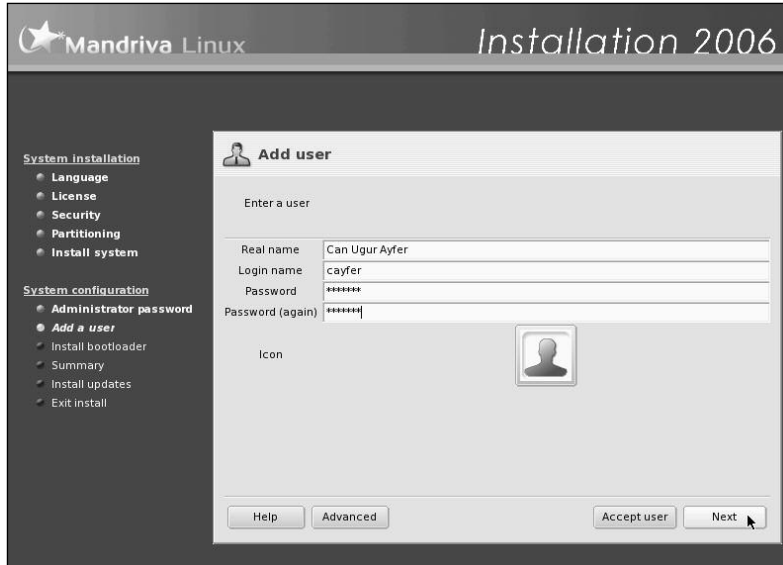


LINUX, “root” kimliğiyle iş yapan kullanıcıların ne yaptığını iyi bildiğini varsayar ve pek çok geri alınamayacak işleme ilişkin komutları sorgusuz sualsiz yerine getirir. Acemiliği üstünüzden atmadan, gerekmedikçe “root” kimliğine bürünmemenizi öneririz. Tek bir komutla kritik dosya ya da izinleri silip tüm sistemi çökterebilirsiniz.

Sisteminiz bir bilgisayar ağına bağlıysa sakın ha “Şifresiz root kullanıcı” (*No password*) seçimini yapmayınız. **root** kullanıcı için şifrenizi seçtikten sonra en az bir tane normal kullanıcı tanıtmalısınız. Doğal olarak bu kullanıcı için en iyi aday kendinizsiniz.



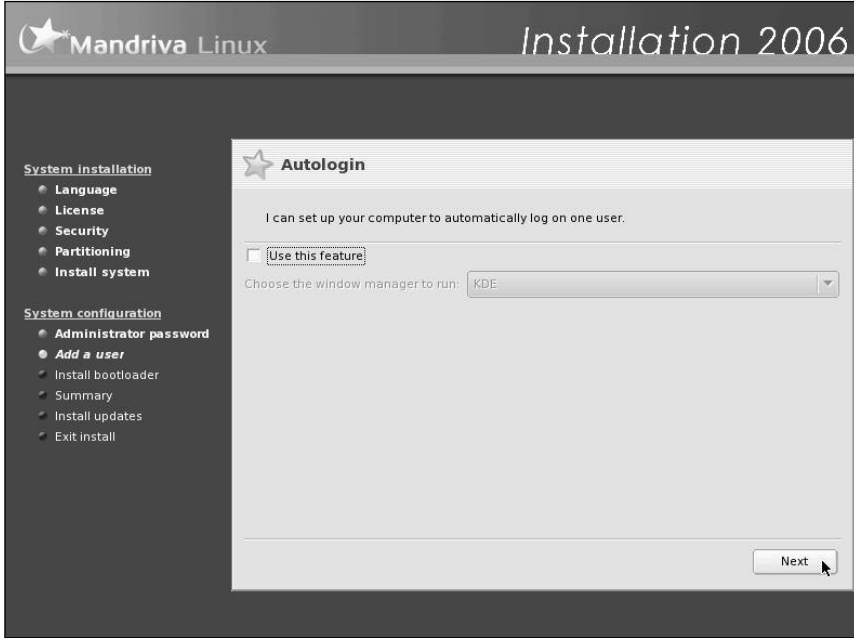
LINUX, güvenlik nedeniyle bilgisayar ağından “root” kimliğiyle gelen istekleri geri çevirir. Bu nedenle sisteminize uzaktan erişerek sistem yönetimiyle ilgili bir iş yapacağınızda, önce normal bir kullanıcı kimliğiyle bağlanıp sonra “root” kimliğine bürünmelisiniz. Bu nedenle kendinize mutlaka sıradan bir kullanıcı hesabı açmalısınız.



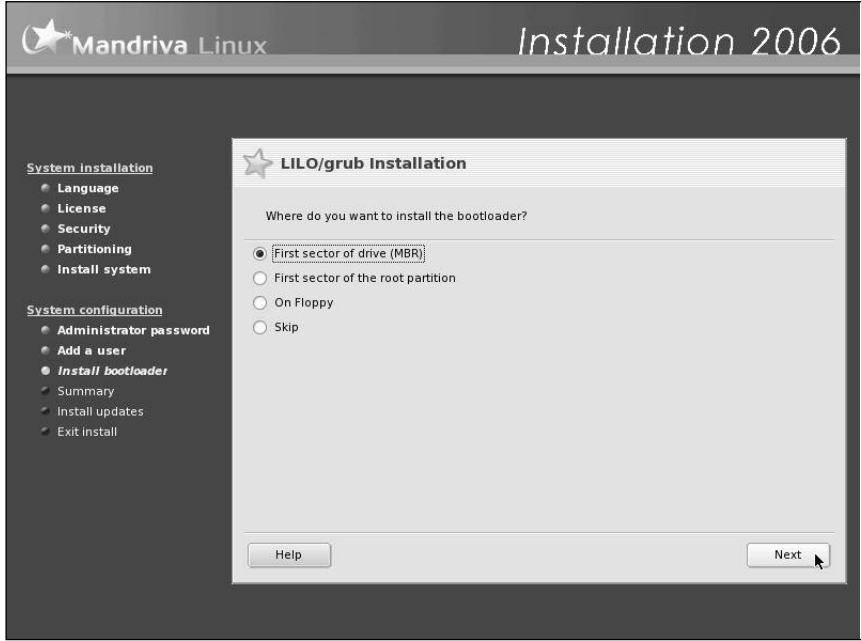
root kullanıcının ardından normal kullanıcıları tanıtmamız istenecektir. Bu aşamada bir tane normal kullanıcı tanıtmamız yeterli olacaktır. Nasılsa sonradan istediğiniz kadar kullanıcı tanıtabilirsiniz.

UNIX geleneklerine göre kullanıcı tanıtmaya kodları (User Name, User ID, Login Name diye de adlandırılır) küçük harflerden oluşturulur. Sizin de aynı geleneğe uymanızı öneririz. Ayrıca kullanıcı tanıtmaya kodu olarak seçeceğiniz karakter dizisinin en fazla 8 karakter uzunluğunda olmasına ve boşluk karakteri, Türkçe'ye özgü karakter ve noktalama işareti içermemesine de dikkat ediniz.

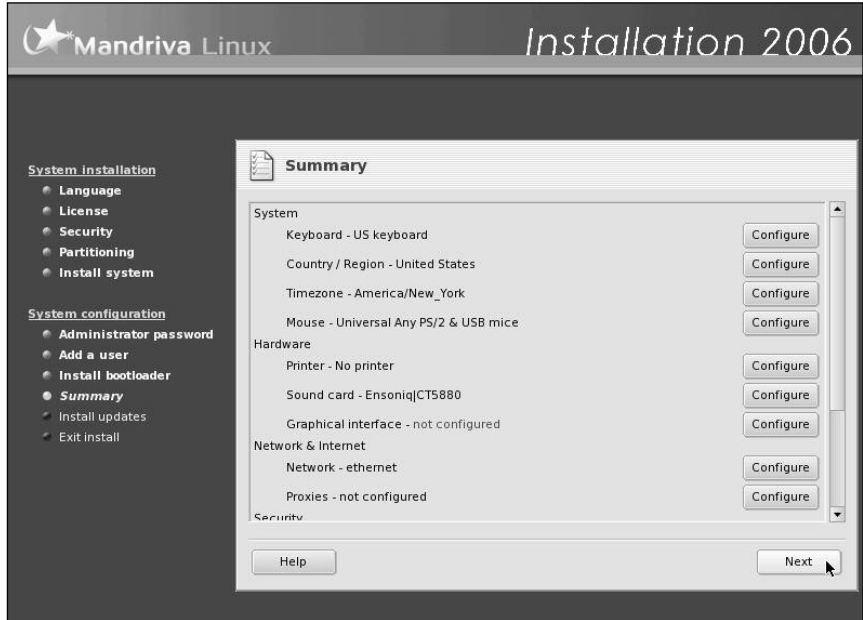
Kullanıcılar için seçtiğiniz kullanıcı tanıtmaya kodları o kişilerin e-posta adresi olacaktır. Bu nedenle **“k1”**, **“u2”** gibi kişisiz kodlar da seçmemelisiniz.



Bu tanıttığınız kullanıcının sisteminiz açıldığında otomatik olarak sisteme giriş yapmasını isteyip istemediğiniz soruluyor. Eğer yanıtınız “Evet” olarsa, bu kullanıcı için başlatılacak pencere yöneticisini de seçebilirsiniz (KDE, GNOME gibi). Önerimiz sistem açıldığında seçilmiş bir kullanıcının sisteme giriş yapmasını seçmemeniz doğrultusundadır. “Do you want to use this feature?” seçimini boş bırakarak “Next” butonunu tıklayınız.



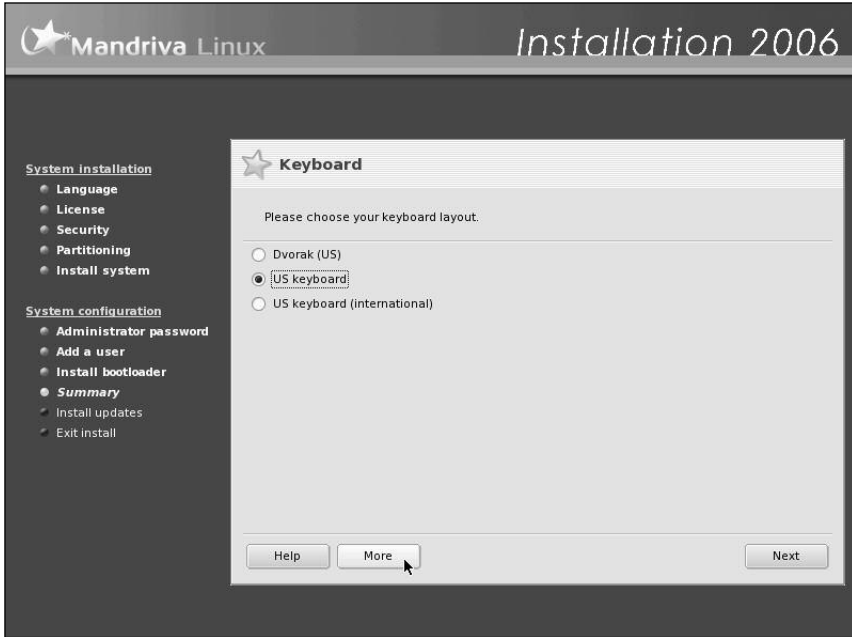
Sisteminizin açılışı sırasında yüklenecek işletim sistemini seçen programın (LILO: *Linux Loader*) yükleneceği disk birimi soruluyor. “First sector of drive (MBR)” seçip devam ediniz.

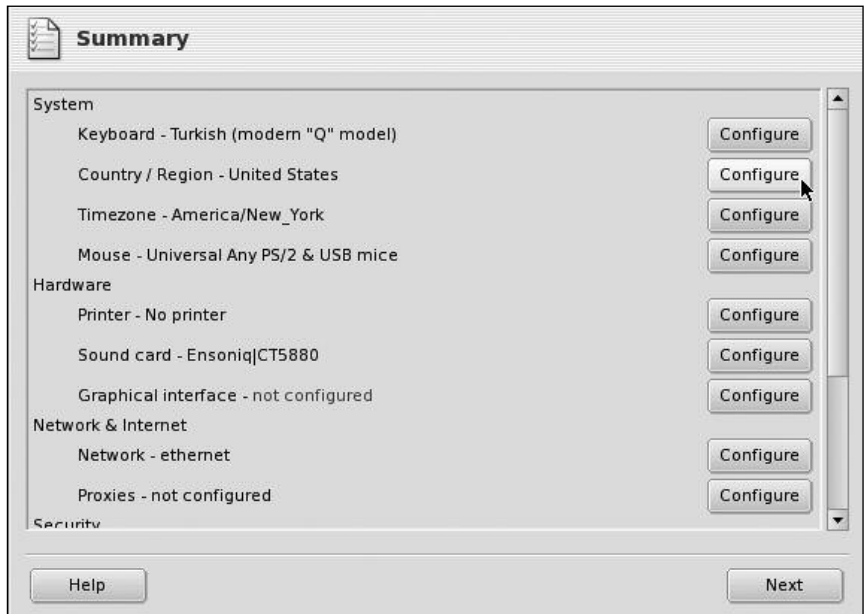
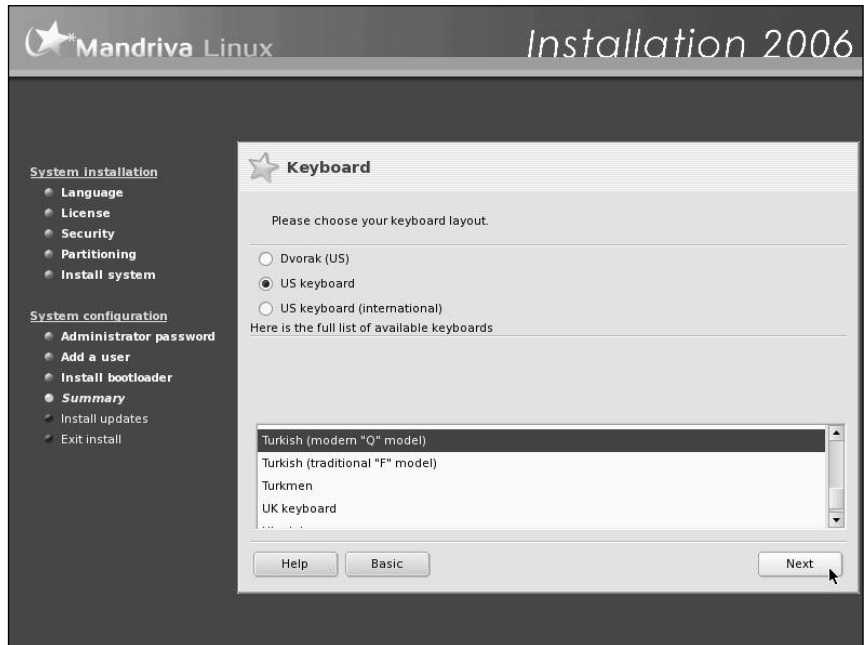


Şimdi de sıra bilgisayarınızın ayrıntılı ayarlarının yapılmasına geldi.

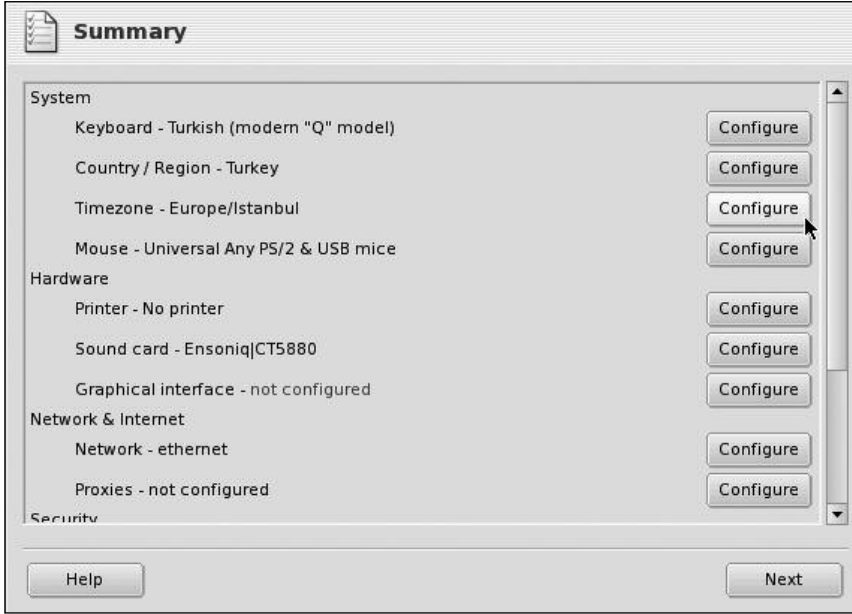
Bu aşamada klavye, arabirim dili (*Country* seçeneği), zaman dilimi (*Timezone*), grafik arabirim kartı (*Graphical Interface*), ağ ayarları (*Network-LAN*), güvenlik düzeyi (*Security Level*), açılış seçeneklerini (*Boot*) seçme ve değiştirme şansı sunuluyor.

Bunlar arasında en önemli olanı “Network-LAN” ayarıdır. Aslında bu seçimlerin hepsini sonradan değiştirebilirsiniz ama hazır yeri gelmişken klavye seçiminde “US keyboard (international)” seçip, “More” butonunu tıklayıp ardından “Turkish (modern “Q” mode)” veya “Turkish (traditional “F” mode)” seçiniz.

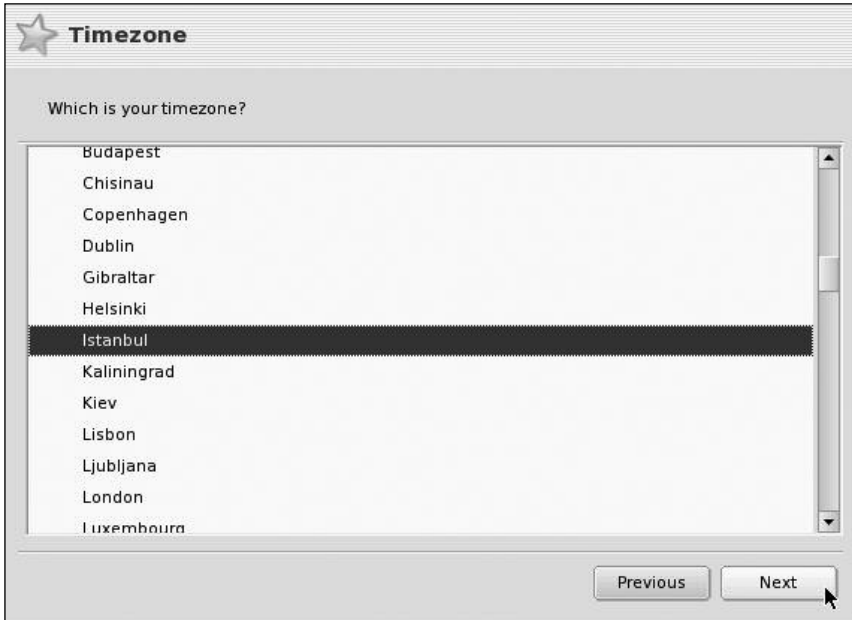




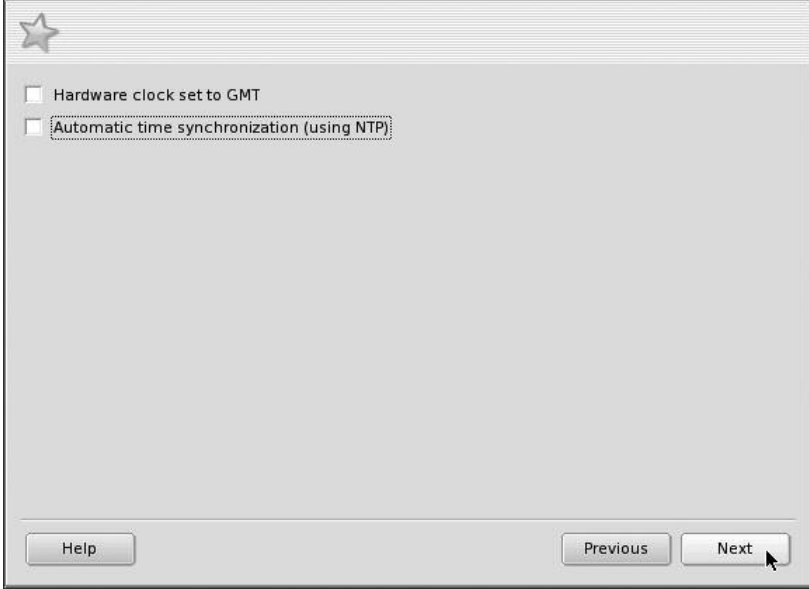
“Country” seçimini “Turkey” olarak yapmanızı öneririz.



Zaman diliminizi (*Timezone*) "Europe/Istanbul" seçmelisiniz. Bu seçimi yaptığınızda iki soru sorulacaktır:



1. Hardware clock set to GMT
2. Automatic time synchronization (using NTP)



“Hardware clock set to GMT” seçimi, bilgisayarınızın saat devresini Greenwich standart saatine ayarlama seçimidir. Boş bırakmanızı öneririz. “Automatic time synchronization” seçimi de sisteminizin saatinin NTP (Network Time Protocol) protokolü ile bir saat sunucusundan alınıp alınmayacağını belirler. Bunu da boş bırakmanızı öneririz.

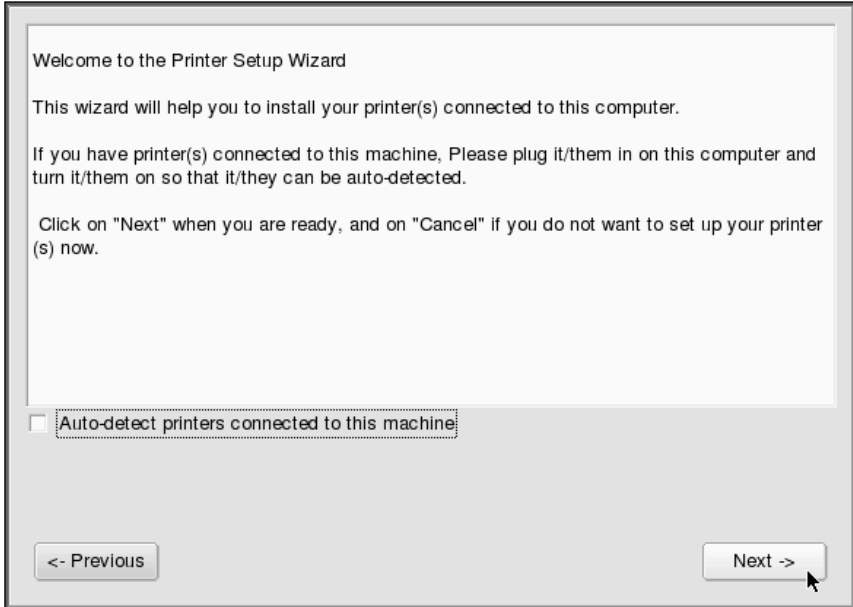
Yazıcı ayarları için “Printer – Remote CUPS Server” seçimini yapınız.



CUPS, **C**ommon **U**NIX **P**rinting **S**ystem sözcüklerinin baş harfleridir (“kaps” diye okunur). Son yıllarda kullanılmakta olan en gelişmiş yazıcı denetim sistemidir. Bir UNIX bilgisayara doğrudan bağlı yazıcılar kadar ağ üzerinden erişilebilen yazıcıları da büyük başarıyla denetleyebilir. Kurulumu da oldukça kolaydır.



Aşağıdaki ekran görüntülerinden de kolaylıkla izleyebileceğiniz gibi yazıcınızın otomatik tanınmasını sağlayabilirsiniz ("Auto-detect printers connected to this machine").



Eğer otomatik ayar seçmediyseniz, yazıcınızın bağlı bulunduğu arabirimi seçmelisiniz:

Please choose the port that your printer is connected to.

☒ Printer on parallel port #0

☐ Printer on parallel port #1

☐ Printer on parallel port #2

☐ USB printer #0

☐ USB printer #1

☐ USB printer #2

<- Previous

Next ->

Yazıcınıza bir isim, belki modelini belirten bir açıklama ve belki de en önemlisi yerini belirten (örneğin hangi odada olduğunu belirten) açıklamalar girmelisiniz. Çok yazıcılı büyük bir bilgisayar ağında yazıcının yerini belirten açıklama notu çok yararlı olmaktadır.

Every printer needs a name (for example "printer"). The Description and Location fields do not need to be filled in. They are comments for the users.

Name of printer	Printer
Description	DeskJet
Location	B316

<- Previous

Next ->

Sıra geldi yazıcı sürücü programları için marka ve model seçmeye...

Which printer model do you have?

Please check whether Printerdrake did the auto-detection of your printer model correctly. Find the correct model in the list when a wrong model or "Raw printer" is highlighted. If your printer is not listed, choose a compatible (see printer manual) or a similar one.

▶ HITACHI

▼ HP

- 2000C
- 2500C
- 2500CM
- 2563
- Business Inkjet 2200

<- Previous

Next ->

Kağıt boyutu vs. ayarları...

Printer default settings

General

Page Size: A4

Printout Mode: Normal (auto-detect paper type)

Media Source: Printer default

Double-Sided Printing: Off

Advanced <- Previous Next ->

Please select the test pages you want to print.
Note: the photo test page can take a rather long time to get printed and on laser printers with too low memory it can even not come out. In most cases it is enough to print the standard test page.

☒ Standard test page
☐ Photo test page
☐ Do not print any test page

<- Previous Next ->

Yazıcı ayarlarının doğru olup olmadığını test etmekte yarar var...

The following printers are configured. Double-click on a printer to change its settings; to make it the default printer; or to view information about it.

Local Printers

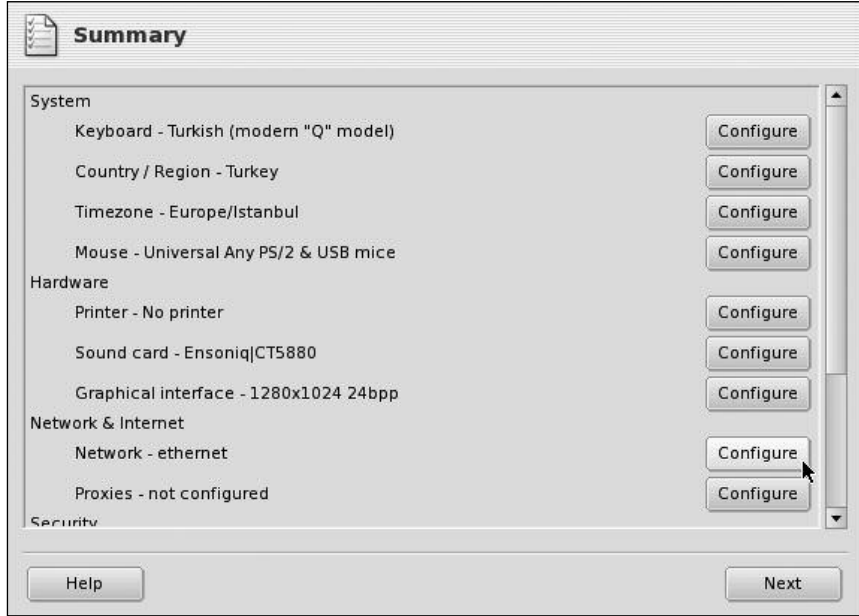
Printer: HP Business Inkjet 2200 on parallel port #0 (Default)

Add a new printer

Expert Mode

Done

Evet! Yazıcı ayarlarınız tamam! Şimdi LAN ayarları önemli!



İnternet bağlantınızı bir ADSL Modem ile yapıyorsanız, yani ADSL Modeminiz bilgisayarınızın içinde takılı ise ya da USB arabirimi ile bağlanıyorsa, bu listeden “ADSL Connection” seçimini yapıp, ADSL Servis sağlayıcınızın size vermiş olması gereken bilgiler aracılığı ile kurulumu devam ediniz.

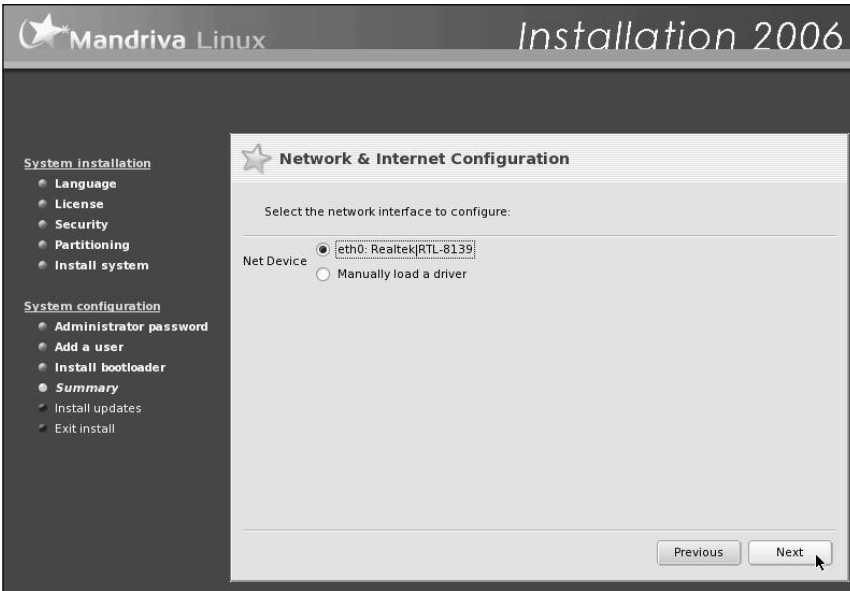
İnternet bağlantınızı ADSL hat üzerinden bir ADSL yönlendirici (ADSL Router) ile yapıyorsanız, büyük olasılıkla bilgisayarınız ADSL yönlendirici cihazına Ethernet arabirimi ile bağlanıyordur. Bu durumda listeden “LAN Connection” seçip kurulumu devam ediniz.

İnternet bağlantınızı bir Kablo Modem ile yapıyorsanız, yani Kablo Modeminiz bilgisayarınızın içinde takılı ise ya da USB arabirimi ile bağlanıyorsa, bu listeden “Cable Connection” seçimini yapıp, Kablo-Net servis sağlayıcınızın size vermiş olması gereken bilgiler aracılığı ile kurulumu devam ediniz.

Bilgisayarınız bir yerel bilgisayar ağına bağlıysa büyük olasılıkla Ethernet arabirimi ile bağlanıyordur. Bu durumda listeden “LAN Connection” seçip kurulumu devam ediniz.

Eğer bilgisayarınızda bir Wi-Fi Telsiz Ethernet arabirimi (Wireless Ethernet) varsa, ve Mandriva kurulum yazılımı bu kartı tanıdıysa, yukarıdaki listede bir de “Wireless Connection” seçeneği yer alacaktır. Eğer ağ bağlantısı için sadece Wi-Fi arabirimi kullanacaksanız “Wireless Connection” seçip devam ediniz. Wi-Fi arabirimi yanısıra Ethernet arabirimi de varsa ve bunu da kullanacaksanız “LAN Connection” seçimiyle devam ediniz.

Şimdi de sıra bulunan Ethernet arabiriminizi seçmeye ve bu arabirim için TCP/IP ayarlarına geldi...



Internet, TCP/IP protokolü denilen bir iletişim protokolleri topluluğu üzerine yapılandırılmıştır. Bu protokol gereği internet veya TCP/IP ağı üzerinde yer alacak her bilgisayarın bir IP numarası olmalıdır. Aynı bir ülkenin telefon şebekesine bağlanacak her telefonun bir abone numarası olması gerektiği gibi... Bu abone numaralarına siz karar veremezsiniz, numaralar telefon hizmetini veren kuruluş tarafından dağıtılır ve her abonenin numarası farklıdır.

Kendi özel telefon şebekeniz varsa, örneğin şirketinizde bir telefon santralınız varsa dahili abonelerin numaralarını istediğiniz gibi seçer ve dağıtırsınız, ancak gene de her abonenin farklı bir numarası olmalıdır. Buna benzer şekilde dış dünyayla doğrudan bağlantısı olmayan bir ağıınız varsa, içerdeki bilgisayarlara, birbirlerinden farklı olmak kaydıyla istediğiniz IP numaralarını verebilirsiniz. Böyle durumlarda genellikle 192.168.0.1 serisi IP numaraları kullanılır.

Bilgisayar ağıınızın dış dünyaya bağlantısı varsa IP numaralarının seçiminde pek özgür değilsiniz demektir. Kullanabileceğiniz IP numaraları için servis sağlayıcınızla veya ağ yöneticinizle görüşmelisiniz.

Aslında TCP/IP protokolüne göre IP numaraları bilgisayarlara değil, bilgisayarların ağ arabirimlerine verilir. Bir başka deyişle, bir bilgisayar üzerinde birden fazla ağ arabirimi varsa (2 Ethernet arabirimi veya bir Ethernet arabirimi + bir çevirmeli ağ arabirimi gibi) o bilgisayar üzerinde tanımlı iki IP numarası olmalıdır. Bu aklınızın bir kenarında bulunsun.



Bir bilgisayarın TCP/IP protokolüyle çalışan bir ağa bağlanabilmesi için dört önemli parametrenin TCP/IP ayarları olarak girilmiş olması gerekir. Bunlar:

- a. IP adresi,
- b. Ağ geçidinin IP adresi (Gateway, Router IP Address),
- c. Alt ağ maskesi (Subnet Mask),
- d. En az bir DNS sunucunun IP adresi.

Bu parametreler, bilgisayarınızın bağlanacağı ağa özgü parametrelerdir ve doğru olarak girilmelidir.

Bilgisayarınızın TCP/IP ayarlarını iki yöntemle yapabilirsiniz:

1. Varsa, bir DHCP servisi üzerinden otomatik olarak,
2. Tüm parametreleri elle kendiniz girerek (manuel).

Eğer ağ bağlantınızı bir ADSL yönlendirici üzerinden veya Wi-Fi arabirimle kuruyorsanız çok büyük olasılıkla yönlendirici veya Wi-Fi erişim noktası cihazınızda (Wireless Access Point) yararlanabileceğiniz bir DHCP sunucu (Dynamic Host Configuration Protocol) vardır. Bu DHCP sunucunun ayarları doğru yapıldıysa, yakınındaki bir bilgisayarın istemesi durumunda kendisine olması gereken TCP/IP ayar parametrelerini gönderecektir. Örneğin Wi-Fi erişim noktaları bu şekilde çalışır. Dizüstü bilgisayarınızla bir Wi-Fi erişim noktasının kapsama alanına girdiğinizde, bilgisayarınız radyo dalgalarıyla DHCP isteği yayınlayarak etrafta kendisine TCP/IP ayar parametreleri verebilecek bir sunucu arar. Erişim noktası da koşullar uygunsa bu istemciye gerekli ayarları yapması için uygun parametreleri gönderir.

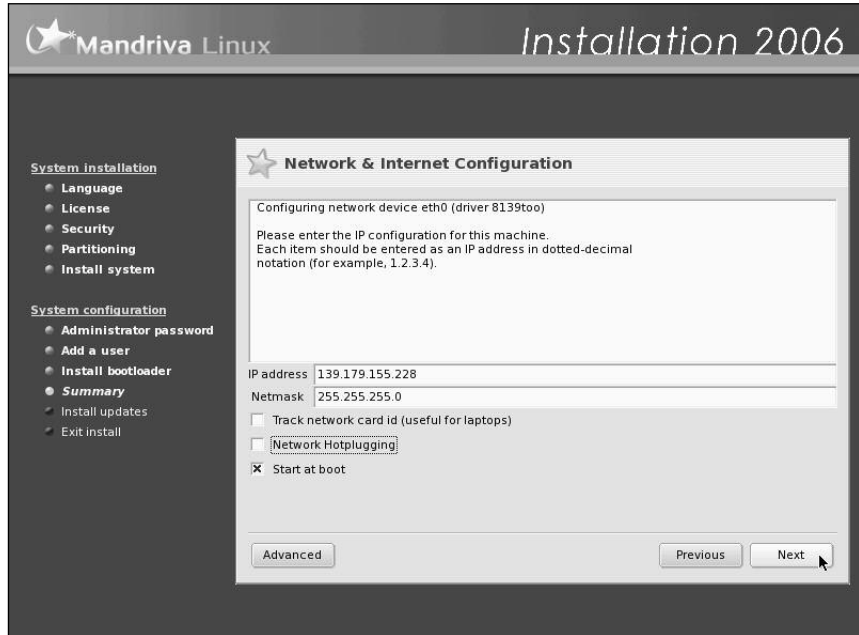
DHCP sunucular sadece Wi-Fi servislerde kullanılmaz. Eğer ağ yöneticisi DHCP ile otomatik ayar dağıtma yolunu seçtiyse, kablolu Ethernet bağlantılarında da DHCP kullanılabilir.

Bir ADSL ya da KabloNet yönlendirici ya da Wi-Fi erişim noktası üzerinden servis alıyorsanız büyük olasılıkla DHCP seçmeniz gerekecektir.

İçinde yer alacağınız ağın özelliklerine göre otomatik ya da manuel kurulum seçimi yaptıktan sonra devam edebilirsiniz.

Kim Korkar LINUX'tan?

Manuel ayarlama yöntemini seçtiğinizde girmeniz gereken bilgiler: IP adresi, alt ağ maskesi (Netmask), DNS sunucuları ve ağ gecidi (Gateway) adresleridir.



The screenshot shows the 'Network & Internet Configuration' window in the Mandriva Linux Installation 2006 installer. The window is titled 'Network & Internet Configuration' and features a star icon. It contains instructions for configuring network device eth0 (driver 8139too). The instructions state: 'Please enter the IP configuration for this machine. Each item should be entered as an IP address in dotted-decimal notation (for example, 1.2.3.4).' Below the instructions, there are input fields for 'IP address' (139.179.155.228) and 'Netmask' (255.255.255.0). There are also checkboxes for 'Track network card id (useful for laptops)' (unchecked), 'Network Hotplugging' (unchecked), and 'Start at boot' (checked). At the bottom, there are buttons for 'Advanced', 'Previous', and 'Next'.

System Installation

- Language
- License
- Security
- Partitioning
- Install system

System configuration

- Administrator password
- Add a user
- Install bootloader
- Summary
- Install updates
- Exit install

Network & Internet Configuration

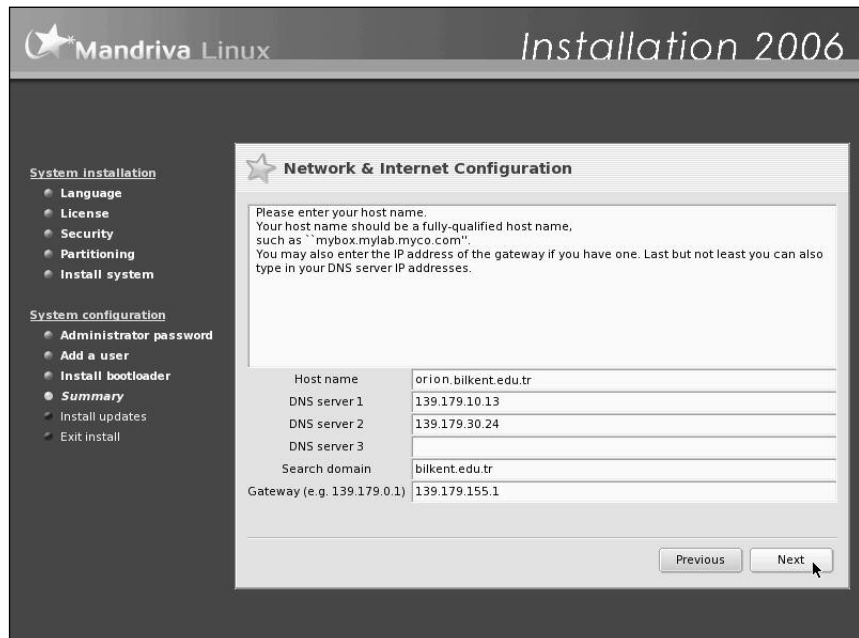
Configuring network device eth0 (driver 8139too)

Please enter the IP configuration for this machine.
Each item should be entered as an IP address in dotted-decimal notation (for example, 1.2.3.4).

IP address: 139.179.155.228
Netmask: 255.255.255.0

☐ Track network card id (useful for laptops)
☐ Network Hotplugging
☒ Start at boot

Advanced Previous Next



The screenshot shows the 'Network & Internet Configuration' window in the Mandriva Linux Installation 2006 installer. The window is titled 'Network & Internet Configuration' and features a star icon. It contains instructions for configuring network device eth0 (driver 8139too). The instructions state: 'Please enter your host name. Your host name should be a fully-qualified host name, such as "mybox.mylab.myc.com". You may also enter the IP address of the gateway if you have one. Last but not least you can also type in your DNS server IP addresses.' Below the instructions, there are input fields for 'Host name' (orion.bilkent.edu.tr), 'DNS server 1' (139.179.10.13), 'DNS server 2' (139.179.30.24), 'DNS server 3' (empty), 'Search domain' (bilkent.edu.tr), and 'Gateway (e.g. 139.179.0.1)' (139.179.155.1). At the bottom, there are buttons for 'Previous' and 'Next'.

System Installation

- Language
- License
- Security
- Partitioning
- Install system

System configuration

- Administrator password
- Add a user
- Install bootloader
- Summary
- Install updates
- Exit install

Network & Internet Configuration

Please enter your host name.
Your host name should be a fully-qualified host name, such as "mybox.mylab.myc.com".
You may also enter the IP address of the gateway if you have one. Last but not least you can also type in your DNS server IP addresses.

Host name: orion.bilkent.edu.tr
DNS server 1: 139.179.10.13
DNS server 2: 139.179.30.24
DNS server 3:
Search domain: bilkent.edu.tr
Gateway (e.g. 139.179.0.1): 139.179.155.1

Previous Next

Bilgisayarınıza vermek üzere bir isim seçmelisiniz. Diyelim ki bilgisayarınıza **orion** ismini uygun gördünüz ve bu bilgisayarı **bilkent.edu.tr** ismiyle kaydedilmiş olan bir ağı bağlıyorsunuz. Bu durumda “Host Name” (Bilgisayar Adı) olarak **orion.bilkent.edu.tr** dizisini girmeli ve ağınızın yöneticisine seçtiğiniz isim ve IP numarasını vererek **DNS** (*Domain Name Services*) kaydı yapılmasını istemelisiniz. Eğer bir makinenin DNS kaydı yapılmazsa diğer bilgisayarlar tarafından isimle bulunamazlar. Aynı telefon rehberinde ismi olmayan abonelere, numaralarını bilenler dışında kimsenin erişemeyeceği gibi.

“DNS Server” satırına parçası bulunduğunuz bilgisayar ağına DNS hizmeti veren bilgisayarın IP numarasını girmelisiniz.

DNS (Domain Name Services) hizmeti veren bilgisayarların temel işlevi sembolik internet adreslerini sayısal IP numaralarına dönüştüren bir çeşit “bilinmeyen numaralar hizmeti” vermektir. İnternet üzerindeki adreslendirme sistemi tamamen sayısal IP adresleri üzerine kurulmuştur. Web tarayıcınızla Hürriyet gazetesinin sayfalarına bakmak için www.hurriyet.com.tr adresini verdiğinizde birilerinin bu sembolik adresi sayısal IP adresine çevirmesi gerekecektir. TCP/IP paketleri sembolik isimlerle hiçbir yetere gidemezler...

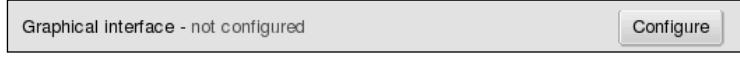


Kullanıcıların bir sürü sayısal adres ezberlemeleri pratik olmayacağı için internet üzerindeki bilgisayar ağlarına ve bilgisayarlara sembolik isimler verilmiştir. Sembolik isimlerin hangi sayısal adreslere karşılık geldiğinin bulunması görevi de DNS sunucularına verilmiştir. Bu sunucu, söz konusu ağ içinde yer alan bir bilgisayar olabileceği gibi bir üst ağda da yer alabilir. DNS sunucunuzun IP numarasının ne olması gerektiğini ağ yöneticinizden veya ağınıza internet hizmeti veren kuruluşun ağ yöneticilerinden öğrenebilirsiniz.

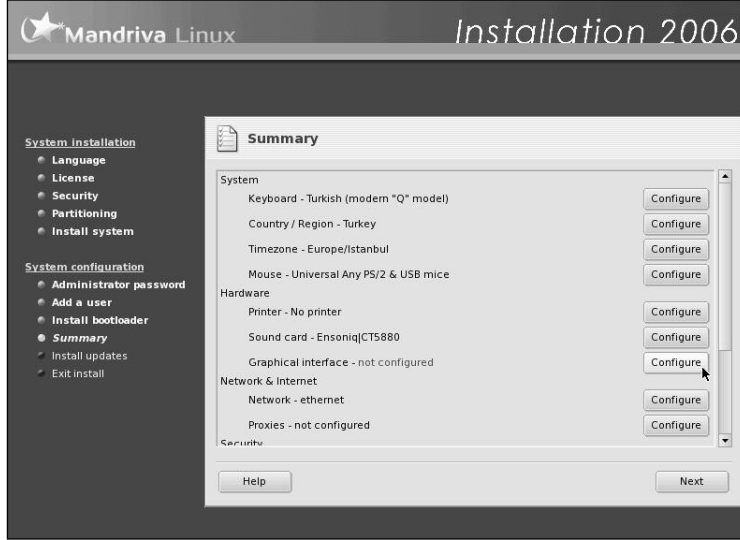
Bu ekrandaki üçüncü önemli parametre bilgisayar ağınızın “Ağ Geçidi” (Gateway, “geytvey” diye okunur, çok yaygın ve yanlış olarak okunduğu şekliyle “getevey” değil!) görevini üstlenmiş olan yönlendiricinin (router) adresidir. Bilgisayar ağınızın “ağ geçidi adresini” servis sağlayıcınızdan öğrenebilirsiniz.

Kim Korkar LINUX'tan?

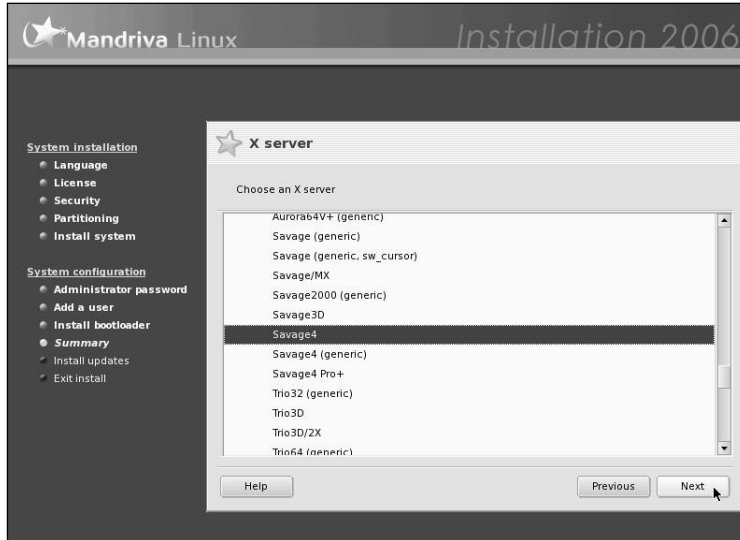
Sıra geldi grafik kartınızın ve ekranınızın ayarlarına.

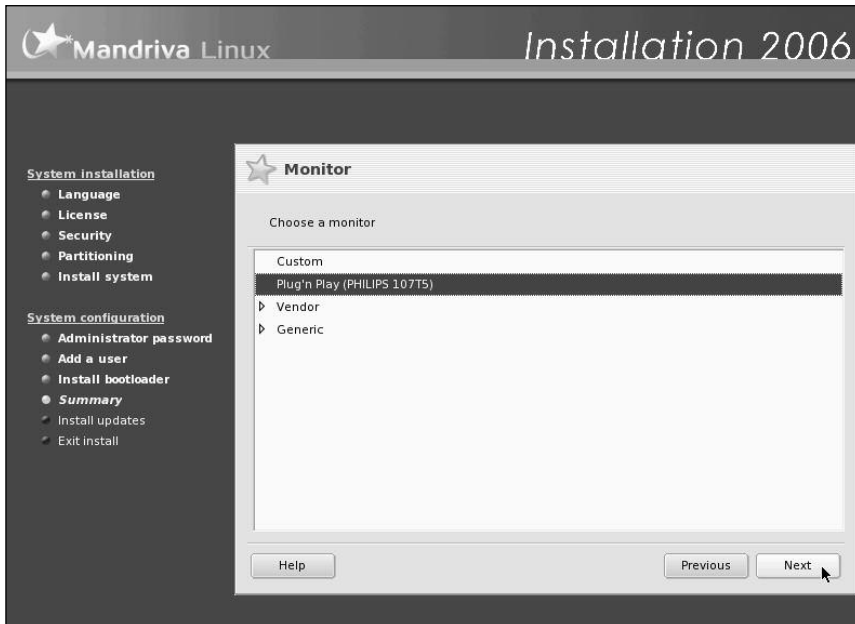
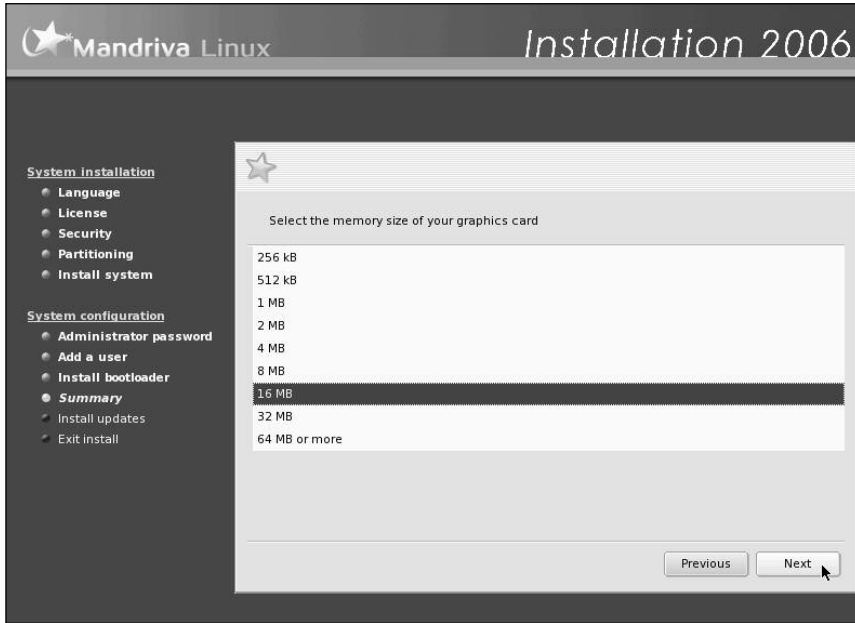


Çok büyük ölçüde otomatik olan bu aşamayı sonuna kadar götürmeli ve özellikle en sondaki “test” işlemini yapmalısınız.



Görüntü kartınız ve monitör tipiniz büyük olasılıkla otomatik tanınarak onaylamanız için size bir öneride bulunulacaktır.

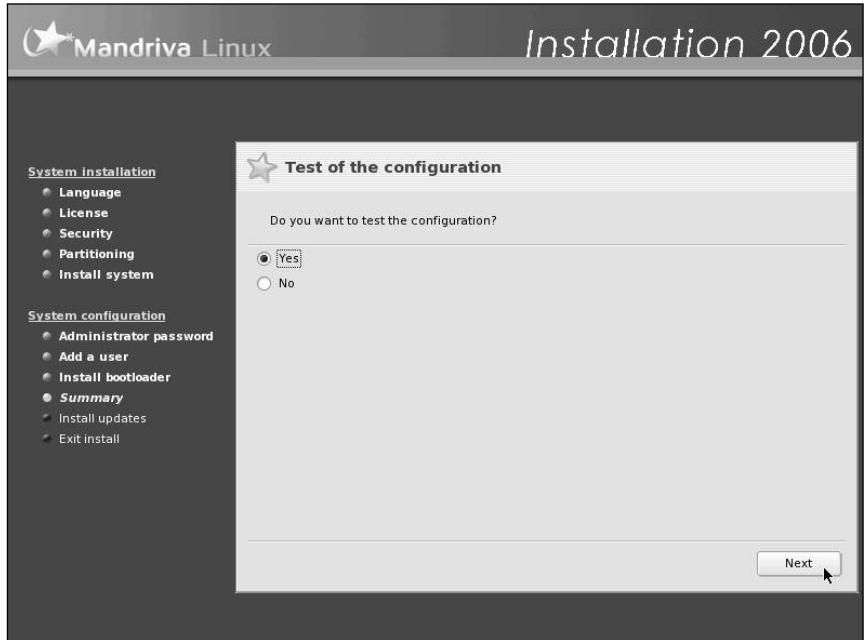


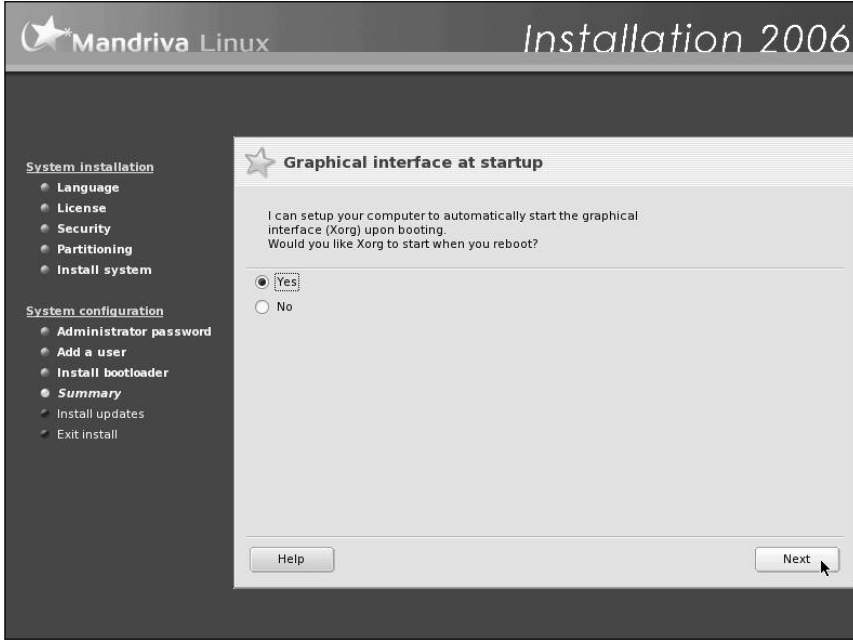


Şimdi de sıra renk zenginliği ile ilgili tercihinizi belirtmekte...

Kim Korkar LINUX'tan?

Doğal olarak, önerilen çözünürlük ve renk sayısını seçmelisiniz. Çözünürlük için önerilenden yüksek değerleri denemeye değebilir.





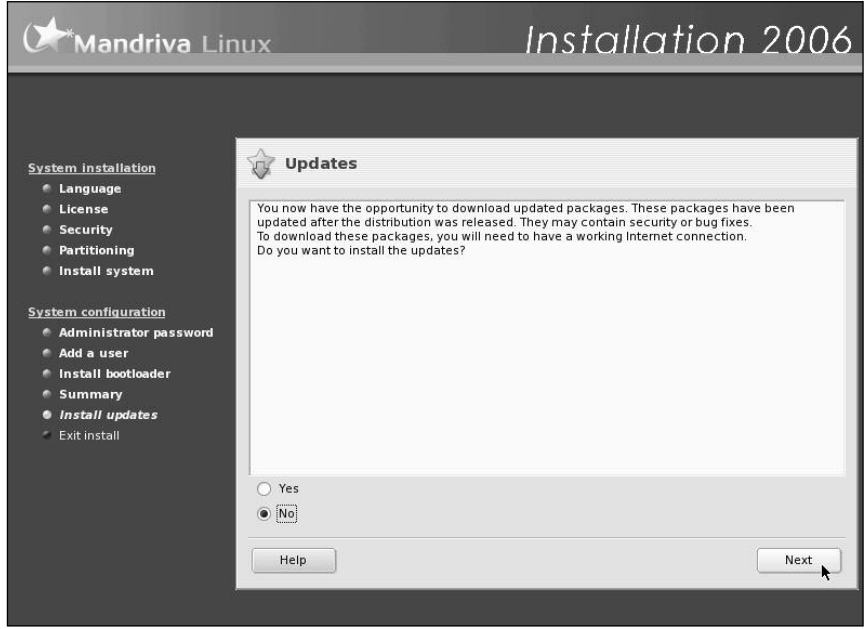
Sistemin açılışlarında X Window sisteminin otomatik olarak başlatılmasını isteyip istemediğiniz soruluyor. Açıkçası, deneyimli kullanıcılar bu soruya “No” yanıtını verip geçer. Grafik ekran gerektiği zaman **startx** komutunu verip Xorg’u kendileri başlatmayı tercih ederler. Ama, estetik nedenlerle, şimdilik de olsa, “Yes” seçmeyi düşünebilirsiniz.

Aslında kurulumla ilgili işleriniz neredeyse bitti. Ayrıntılı kurulum seçenekleri arasında düzenleme yapmanız gereken başka bir bölüm kalmadı. Gene de merak ediyorsanız

- Security-Firewall
- Boot-Bootloader
- Services

gibi seçimleri yapıp oralarda neler varmış bir göz atabilirsiniz.

“Next” tıkladığınızda karşınıza son kurulum ekranı olan “Mandriva LINUX güncelleme” ekranı gelecektir.



Internet bağlantı hızınız yüksekse ve vaktiniz varsa bu güncelleme sürecinden geçmenizi hararetle öneririz. Ancak, LINUX'u yeni öğrenmek üzere kuruyorsanız bizce bu güncelleme süreciyle şimdilik vakit kaybetmeyin.



Evet, bitti... Varsa sürücüden disketi ve CD'yi çıkarıp “Ok” butonunu tıklayınız. Sisteminiz yeniden başlatılacak ve diskteki LINUX belleğe yüklenecektir.

BUNLARI BİLİYOR MUYDUNUZ?

TUX

LINUX işletim sisteminin geliştirilme sürecinin ilk yıllarında projeye katkıda bulunan programcıların üye oldukları internet tartışma listesinde "artık bir logoya gereksinim olduğu" konusunda düşünceler ortaya çıktı. Önceleri şahin, tilki, köpekbalığı, kartal gibi yırtıcı hayvanlar üzerinde duruldu. Kişiliğinin bir özelliğinden olsa gerek, LINUX'un yaratıcısı Linus Torvalds, yırtıcı bir hayvan yerine fikirlerin "penguen" üzerine yönltilmesini önerdi. Tartışmalar ilerledikçe "karnı tok, sırtı pek bir penguen" üzerinde görüş birliği sağlandı.



Sonunda Larry Ewing, halen kullanılmakta olan penguen logosunu geliştirdi. Karnı balıklarla dolu, yeni geçirmiş, mutlu penguen çok beğenildi.

Sıra bu penguene bir isim bulmaya gelmişti. Penguenlerin tüylerinin desenlerinin smokine benzemesi nedeniyle, smokin sözcüğünün İngilizcesi olan "tuxedo"dan esinlenerek "TUX" adı seçildi.

Standart TUX resimleri yanısıra çeşitlemelerini bir arada www.linux.org/info/logos.html adresinde bulabilirsiniz.

LINUX'la İlk Tanışma:

KDE Masaüstü Yöneticisi

03

- İlk Tanışma
 - LINUX Bilgisayarların Açılması
 - LINUX Bilgisayarların Kapatılması
 - İlk Login
- Mandriva LINUX Uygulamaları
 - KDE Ayarları
 - Kullanıcı Yönetimi: UserDrake
 - Dosya Yöneticisi: Konqueror
 - Genel Sistem Yönetimi Araçları: Mandriva Control Center
 - Ofis Uygulamaları
 - Grafik Uygulamaları
 - Kişisel Uygulamalar
 - İnternetle İlgili Uygulamalar
 - Çoklu Ortam (Multimedya) Uygulamaları
- Kopete Anında Mesajlaşma Yazılımı
- K3B
- Gambas
- Eclipse
- Digikam
- Wine
- Konsolda LINUX Kullanımı
 - “Konsol” Kavramı
 - “logout”: Sistemden Çıkış
 - Isınma Hareketleri
- LINUX Dosya-Dizin Yapısı
 - Dosya-Dizin İsimleri
 - Dosyalar ve Dizinler
- LINUX'ta Erişim Yetkileri
 - Yazma, Okuma, Program Çalıştırma Yetkileri
 - Dizinlere Erişim Hakları
 - Suid Biti ve suid Programlar
 - Sticky Bit
 - Dosyaların ve Dizinlerin Sahibini Değiştirmek
 - Dosyaların ve Dizinlerin Gruplarını Değiştirmek

- **Dosyaların ve Dizinlerin Sahibini ve Grubunu Birlikte Deęiřtirmek**
- **bash Kabuk Programı**
- **bash Bařlangıç Dosyaları**
 - **bash History**
 - **! Aracı**
- **Kendi Komutlarınız: Aliases**

İlk Tanıřma

LINUX Bilgisayarların Açılması

Eęer bilgisayarınızda yüklü olan tek iřletim sistemi LINUX ise, bilgisayarınızı açmanız ve KDE (veya GNOME) Login ekranı gelene kadar biraz beklemeniz yeterlidir.

Eęer bilgisayarınızda birden fazla iřletim sistemi yüklüyse, LILO programı hangi disk bölümündeki hangi iřletim sistemini yüklemek istedięinizi soracaktır. LILO menüsündeki seçenekler arasında bir tanesi “varsayılan” yükleme seçeneęidir. LILO'ya siz gerekli yanıtı vermeden bekleme süresi dolarsa bu varsayılan seçim yapılmıř kabul edilir ve ilgili iřletim sistemi yüklenmeye bařlar. Bu davranıř özellikle bilgisayarınızı uzaktan geri yükledięinizde (reboot ettięinizde) ve elektrik kesintilerinden sonra tekrar açılma durumlarında birisinin makinenin bařına gidip seçim yapmasını gerektirmemesi ağısından çok yararlıdır. LILO ile ilgili ayarları “Sistem Yönetimi” bölümünde “LILO Yönetimi” bařlığı altında bulacaksınız.



Açılış Yöneticileri (BOOT Loaders)



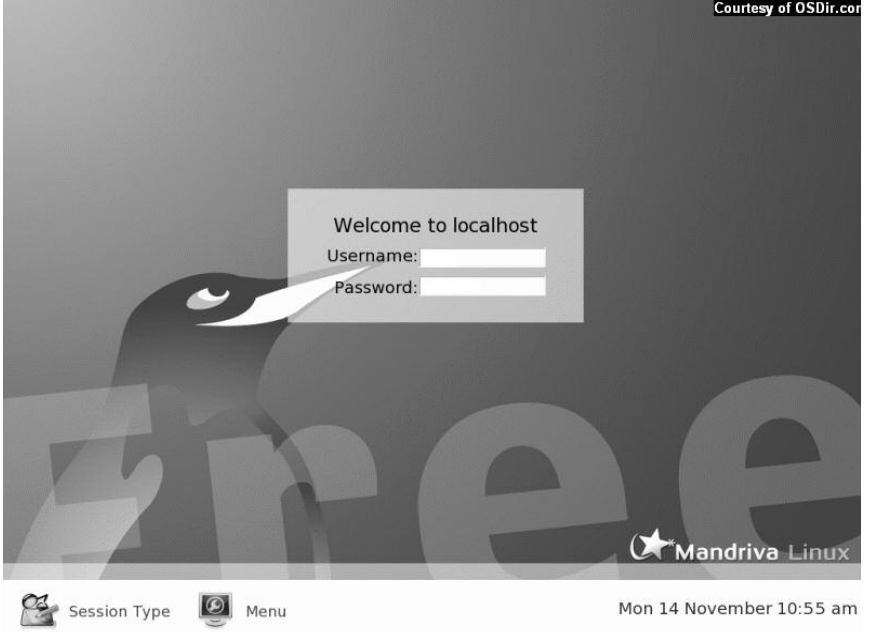
Genel amaçlı bilgisayarların tamamı işletim sistemini belleğe diskten, CD'den, USB bellek çubuğundan ya da ağ üzerinden yükleyerek kullanıma hazır hale gelir. Bu yükleme sürecine bilgisayarların ilk yıllarında "bir insanın kendi kendini ayakkabısının bağlarından çekerek kaldırması" anlamına gelen "bootstrapping" adı verilmiştir. Bu sözcük sonradan "booting", "boot etme" olarak kısaltılmıştır.

Bir bilgisayar açıldığında, PC'lerdeki BIOS gibi ROM'da yer alan bir program parçası uygun bir disk, CD, ağ arabirimi seçerek bu ortamda ön tanımlı bir alandan "Boot loader" yazılımını yüklemeye çalışır. PC'lerde bu ön tanımlı alan genellikle ilk diskin MBR'sidir (Master Boot Record - Temel Boot kaydı). Diskin MBR olarak tanımlı alanında yer alması beklenen, BIOS'dakinden daha yetenekli bir program belleğe yüklenerek çalıştırılır. Bu program, gerek ön tanımlı ayarları gereğince, gerek kullanıcının yapabileceği seçimler doğrultusunda disklerden ya da disk parçalarının birinden ya da bir başka ortamdaki işletim sisteminin çekirdeğini belleğe yükleyerek denetimi işletim sistemine aktarır.

Linux dünyasında, MBR'ye kaydedilerek, işletim sisteminin nereden yükleneneceğinin belirtilmesini sağlayan birden fazla yazılım vardır. Bunlardan en popüler iki tanesi LILO ve GRUB yazılımlarıdır. Mandriva aksi belirtilmedikçe bu iş için LILO'yu kullanır.

LINUX'un açılış sırasında ekrana listelediği mesajlar ilk başta size ürkütücü gelebilir. Normal koşullarda bunlara bakmanız bile gerekmez ama sorunlu bilgisayarlarda sorunun kaynağını bulmak ve düzeltmek için bu mesajlar çok yararlı olmaktadır. Bu mesajlar sayesinde açılış sürecinin hangi aşamasında, hangi iş yapılırken sorun çıktığını kolayca görebilir ve sorunu yaratan yazılım ya da donanım unsurunu belirleyebilirsiniz. Mavi ekrandan daha iyi değil mi?

İşletim sistemini yükleme süreci tamamlandığında grafik login ekranı görünecektir.



LINUX Bilgisayarların Kapatılması

Hemen belirtelim: LINUX bilgisayarları kapatmanız gerektiğinde kesinlikle, ama kesinlikle “törenselle kapatma” sürecini başlatmalı ve sonuna kadar beklemelisiniz.

İşler sarpa sarınca makinenizi “kütüt” diye kapatmayın sakın. Elbette mecbur olduğunuz durumlar hariç...

LINUX, donanım sorunları olmadığı sürece kolay kolay çakılmaz, dolayısıyla bilgisayarınızı sık sık anahtarından kapatmak zorunda kalacağınızı sanmıyoruz. Enerji sorunu olmadığı sürece LINUX bilgisayarların tipik açık kalma ve çalışma süreleri aylarla ölçülür. Hatta, ABD’de bir üniversitede (galiba MIT) yanlışlıkla bulunduğu odanın kapısı duvarla örülen bir bilgisayarın yıllarca sorun çıkarmadan çalıştığı ve bu durumun bir tesisat onarımı sırasında duvarın yıkılmasıyla ortaya çıktığı anlatılır. Donanım eklemek, taşımak ya da çekirdek (**kernel**) değişikliği yapmak gerektiğinde bilgisayarınızı kapatmalısınız elbette ama tabii ki kuralına göre.

Öncelikle bütün kullanıcıların sisteminizle bağlantılarını kesmelerini istemeniz gerekir. Kullanıcılara çeşitli yollarla sistemi kapatacağınızı haber verebi-

lirsiniz. **wall** komutuyla herkesin ekranına bir mesaj yollayabileceğiniz gibi telefon da edebilirsiniz.

Ardından masa üstü denetleyicinizden **Logout** seçmelisiniz. (Bilgisayarı kapatmaya niyetlendiğinizde işe **Start** diye başlamaktan daha mantıklı.)

Logout menüsünde üç seçenek göreceksiniz:

- Login as a different user (Sistemden çık ve başka bir kullanıcı olarak tekrar login ol)
- Turn off computer (Sistemi kapat)
- Restart computer (Sistemi yeniden başlat)

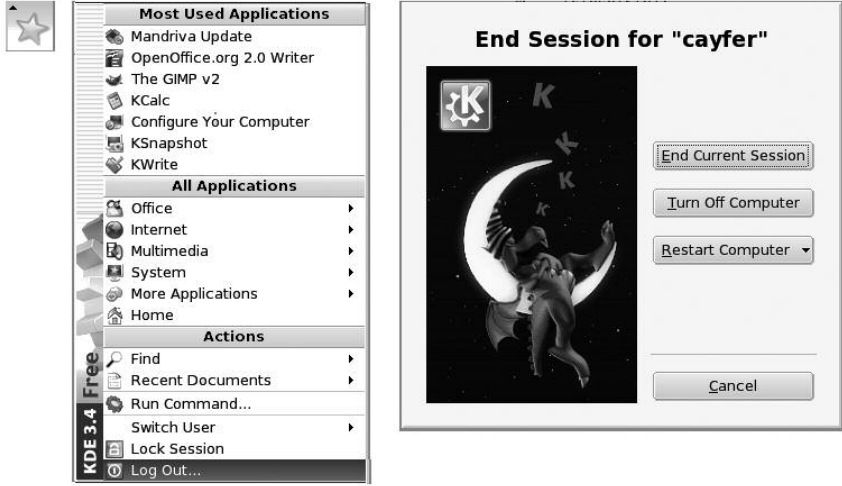


Bunlardan “Login as a different user” sistemle sizin, şahıs olarak bağlantınızı kesecektir. Masa üstünüz kapanacak ve sisteminiz yeni bir kullanıcının konsoldan sisteme girmesine ve kendi istediği masa üstüyle çalışmasına olanak verecek şekilde “**Login**” ekranına dönecektir.

Sistemden çıktığınızda yalnızca size ait işler durdurulur. Arka plandaki servisler (web sunucusu, veritabanı sunucusu gibi servisler) olduğu gibi işlerine devam eder. **Halt** adı üstünde sistemi kapatma törenini başlatır. Bu seçim yapıldığında

- Önce kullanıcıya ait masa üstünde çalışan programlar birer birer kapatılır,
- Sonra masa üstü yöneticisi (*KDE*, *GNOME* gibi) kapatılır,
- X sunucusu kapatılır,
- Geri planda çalışan sunucu programlar (veritabanı yönetim sistemleri, web ve ftp sunucuları, e-posta ve diğer internet servisleri gibi) birer birer kapatılır,
- Bilgisayarınızda çok kullanıcılu düzen için servis veren programlar kapatılarak sistem tek kullanıcılu duruma getirilir,
- Tek kullanıcılu durumda çalışan programlar kapatılır,

- En son olarak da çekirdek program durdurulur. Eğer bilgisayarınızın kasası ATX veya benzeri özellikte bir kasaysa, bilgisayarınızın güç kaynağı da kapatılır. Yok eğer kasanızın güç kaynağı yazılımla denetlenemiyorsa ekranda artık bilgisayarı anahtarından kapatabileceğinize ilişkin bir mesaj görüntülenir. (*Power off*) Bu mesajı görünce sisteminizi anahtarından kapatabilirsiniz.



Peki bilgisayarı “küüüt” diye kapattığınızda ya da elektrik kesildiğinde ne olur?

Bu biraz da bilgisayarda hangi servislerin çalıştığına bağlıdır. Bir kere doğal olarak üzerinde çalıştığınız ve henüz kaydedilmemiş dosyalarda yaptığınız değişiklikler kaybolur. MySQL gibi veritabanı sistemleri çalışıyorsa son güncellenen birkaç kayıttaki güncellemeler de kaybolabilir. Bunlar enerji kayıplarının doğal sonuçlarıdır.

Eğer disklerinizi “ext2” dosya sistemiyle formatladıysanız kayıplarınız yukarıda sözü edilen dosya güncelleme kayıplarının ötesine gidebilir; bazı dizinleri ve dosyaları da kaybedebilirsiniz. Eğer önerilerimize kulak verip disklerinizi “ext3” veya “reiser fs” dosya sistemiyle formatladıysanız pek bir şey olmaz.

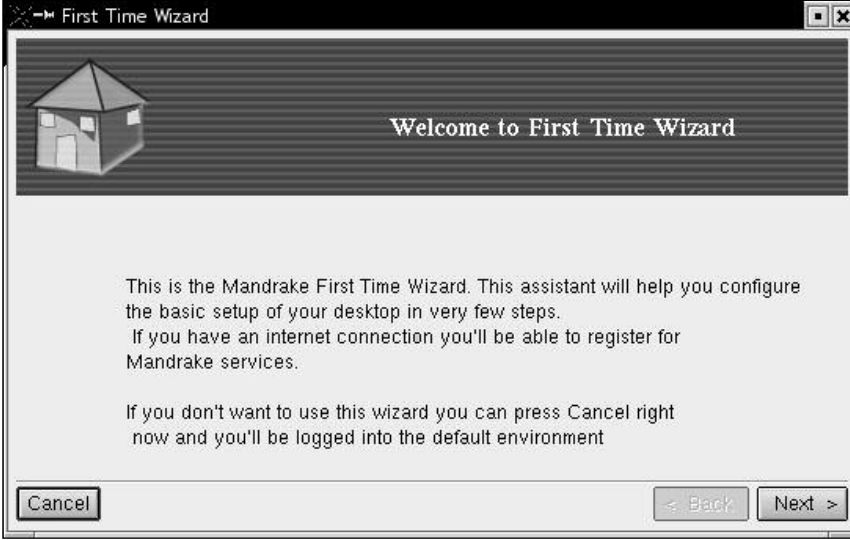
Ancak, her durumda, sisteminiz bir dahaki açılışında düzgün kapatılmamış olduğunu farkedip dosya sistemlerini kontrol etmek üzere **fsck** (*file system check*) yazılımını tüm disk bölümlerini kontrol edecek şekilde otomatik ola-

rak çalıştıracaktır. Bu program içinden çıkamadığı bir durumla karşılaşırsa bazı düzenlemeleri yapmadan önce sistem yöneticisinden izin ister. Açıkçası böyle bir durumla karşılaşırsanız tüm sorulara “yes” demekten başka seçeneğiniz de pek yoktur.

Sisteminizde önemli işler yapacaksınız; hele hele başkalarına hizmet verecek servisleriniz çalışacaksa, kesintisiz güç ve yedekleme işini çok ciddiye almanız gerekir.

İlk Login

Şimdi login penceresinde sistemin kurulumu sırasında tanıttığınız normal kullanıcıyı seçip şifrenizi giriniz. Ağ bağlantınız çalışıyorsa



ekranı görünecektir. Eh! Bu kadarcık reklam olacak elbette...

Evet! LINUX işletim sistemi altında KDE masa üstünüz (*desktop*) karşınızda... (Kolay görebilmeniz için ekranın tamamı yerine sol alt köşesi gösterilmiştir.)



KDE, GNOME gibi masa üstü yöneticileri, bir LINUX bilgisayarın kolay kullanımına yöneliktir. Pek çok işinizi hiç UNIX komutu kullanmadan yapabilirsiniz. Eğer herhangi bir MS-Windows işletim sistemine alışkınsanız, seçtiğiniz masa üstü yöneticisini kullanmak çok kolay gelecektir. Neler yapılabildiğini keşfetmek için menülerde biraz dolaşmanız yeterlidir.

Bu kitapta masa üstü olarak hep KDE yazılımından örnekler göreceksiniz. GNOME yerine KDE seçmemizin teknik bir nedeni yok. KDE bize daha sevimli geliyor, o kadar.



KDE'nin herşeyini anlatmak için bu kitap kadar bir kitap daha yazmak gerekiyor. Bu nedenle KDE'ye alışmanıza yönelik olarak yalnızca temel bazı özelliklerine değineceğiz. Gerisini kendiniz keşfedebilirsiniz.

KDE altında bir uygulama programı başlatmak için:

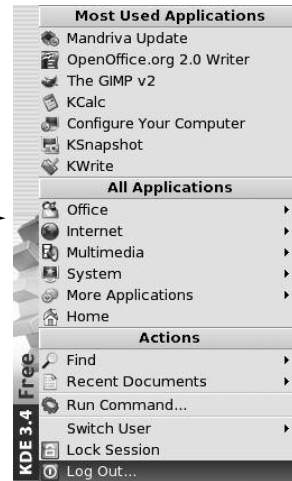
1. Uygulamanın masa üstünde ikonunu varsa o ikona tıklamalı veya
2. Uygulamanın “sembolik menü”de (*kicker*) ikonunu varsa o ikona tıklamalı veya
3. Uygulamayı KDE menüsünden seçmeli veya
4. KDE menüsünden “Run Command” seçip istediğiniz programı başlatacak komutu açıkça yazmalısınız.



KDE Masa Üstü Denetim Elemanları		
1	Masa Üstü İkonları: Kişisel dizininiz için “Konqueror Dosya Yöneticisi” ni çalıştırmak için “Home” ikonunu tıklayabilirsiniz.	
2	Sembolik Menü (Kicker): Sık kullanılan programların birer ikon olarak yer aldığı menü bölgesidir. Sık sık kullandığınız, örneğin web tarayıcı, e-posta yazılımı gibi yazılımlar için bu alana birer ikon yerleştirerek söz konusu programları KDE menüsünü açmaya gerek kalmaksızın çalıştırabilirsiniz.	
3	Ekran Seçici: KDE, sanki iki ekranınız varmış gibi çalışabilmeniz için bir olanak sunar. (Sanal ekran sayısını istediğiniz kadar arttırabilirsiniz.) Ekran seçicideki iki kareden istediğinizi tıklayıp o ekrana geçebilirsiniz.	
4	Görev Çubuğu (Task Bar): Yukardaki örnek ekranda, çalışmakta olan “The Gimp” ve “Konsol” yazılımlarına ilişkin program ikonları görünüyor.	

	Bu uygulama programlarına ilişkin pencereleri göremiyor olsanız bile görev çubuğundaki ikonlarını tıklayarak o pencereleri öne çıkarabilirsiniz.	
5	KDE Menüsü: Masa üstüne veya sembolik menüye yerleştirilmemiş uygulamaları başlatmak için KDE Menü ikonunu tıklayarak çeşitli kategorilerdeki programlar arasından seçim yapabilirsiniz	
6	Hızlı Menü: Linux için geliştirilmiş birçok program geri planda sürekli çalışır durumda bekleyecek şekilde geliştirilmiştir. Örneğin, ses çıkış şiddetini ayarlamayı sağlayacak yazılım, günlük iş planlarınızı izleyebileceğiniz KOrganizer yazılımı sürekli arka planda çalışır ve hazır durumda olabilir. Bu tip yazılımlar çalıştıkları sürece “Hızlı menü” alanında birer ikonla temsil edilirler. Ön plana çıkarmak istediğiniz yazılım için ilgili ikonu tıklamanız yeterlidir. Bu menünün sembolik menüden farkı zaten çalışmakta olan bir yazılımı ön plana çıkarmasıdır; oysa sembolik menüden bir ikon tıkladığınızda, ilgili program diskten yüklenerek başlatılır. Sembolik menü ile bir yazılımın birden fazla kopyasını başlatabilirsiniz; oysa hızlı menüden bir programın çalışmakta olan kopyasını ön plana getirirsiniz.	

Belki de en çok kullanacağınız KDE butonu “Menü Butonu”dur. Bu butonu tıkladığınız anda karşınızda bir menü belirecektir. Bu menüden yapamayacağınız iş yok gibidir. Gerek uygulama programlarını başlatmak, gerekse sistemin her türlü davranışını bu menüdeki yazılımları kullanarak denetlemek olasıdır. Kolay kullanım için menüdeki yazılımlar konularına göre gruplanmıştır. Örneğin ofis yazılımları (*OpenOffice* gibi), ağ yönetimine



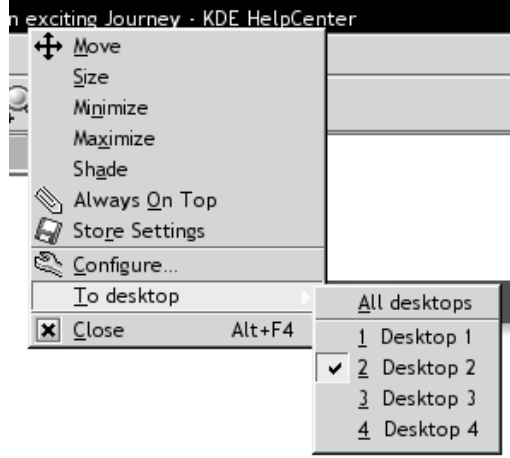
ilişkin yazılımların hepsi bu menüden başlatılabilir. KDE menüsündeki grupların varsa, alt menüleri fare imlecini menü maddesi üzerine getirdiğinizde açılır.



Her KDE penceresinin sağ ve sol üst köşelerinde pencere denetim butonları vardır.

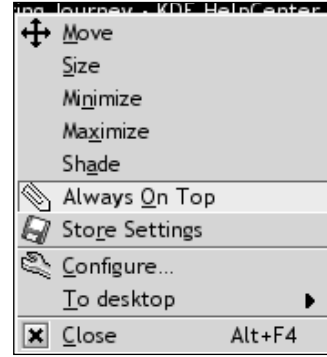
KDE Pencere Denetim Butonları	
	Uygulamayı ikonlaştırır. (Minimize eder.)
	<p>Tüm ekranı uygulamaya verir. (Maksimize eder.)</p> <p>Bu butonu sağ fare düğmesiyle tıklarsanız pencere ekranı yanlamasına dolduracak şekilde yatay olarak genişletilir.</p> <p>Orta düğmeyle (varsa) tıklarsanız pencere ekranı yukardan aşağı dolduracak şekilde düşey olarak büyütülür.</p>
	Uygulama penceresini eski boyutlarına getirir.
	Uygulamayı kapatır.
<p>Bu butonların görünüşleri, pencere içindeki yerleri ve hatta bazen işlevleri kullandığınız masa üstü temasına göre değişebilir, fakat temelde hepsi burada anlatılan örneğe çok benzemektedir.</p>	



Bazen bir uygulamanızın çalıştığı sanal ekranı değiştirmek isteyebilirsiniz. Bu durumda söz konusu uygulamanın başlık çubuğuna (*title bar*) fare ile sağ tıklayıp önce “To Desktop” seçip ardından gelen listeden sanal ekranınızı seçin.


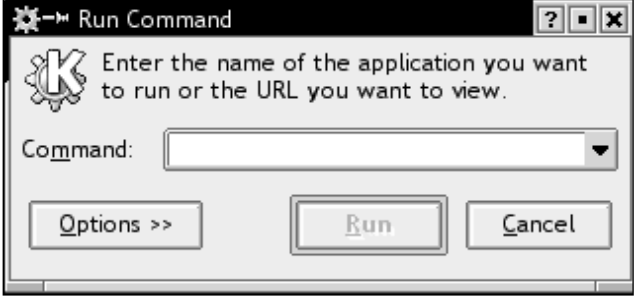


Gene bazen bir uygulama pencerenizin her koşulda üstte kalmasını ve başka uygulamalara ait pencereler tarafından örtülmemesini isteyebilirsiniz. Böyle bir durumda söz konusu uygulamanın başlık çubuğuna (*title bar*) fare ile sağ tıklayıp “Always on Top” seçiniz.

KDE'nin bazı çok önemli klavye kısa yollarından söz etmeden geçemeyeceğiz.



KDE Masaüstü Klavye Komutları	
Alt-Tab	<p>Çalışmakta olan işlere ait pencereler arasında dolaşmakta kullanılır. Alt tuşunu basılı tutarken Tab'a her basışınızda çalışmakta olan işlerin pencere listesinin ilerlediğini göreceksiniz. İşaretili iş, ilgilendiğiniz uygulama olarak karşınıza geldiğinde Alt tuşunu bırakırsanız o pencere en üste gelecektir. “Ben bu işi fareyle de yapıyorum zaten!” diyor olabilirsiniz. Doğru tabi! Ama bir gün faresiz çalışmak zorunda da kalabilirsiniz. Bu özellikler aklınızın bir kenarında bulunsun.</p> <div> (The GIMP) </div>

<p>Ctrl-Tab</p>	<p>Sanal ekranlar arasında dolaşmak için kullanılır. Ctrl tuşunu basılı tutarken her Tab basışınızda sanal ekran göstergesi bir ilerler. İlgilendiğiniz sanal ekrana gelince Ctrl tuşunu bırakırsanız monitörünüzde o sanal ekran görüntülenecektir.</p> 
<p>Ctrl-F1 Ctrl-F2 ...</p>	<p>Fonksiyon tuşunun numarasına karşılık gelen sanal pencerenin seçilmesini sağlar. Örneğin üçüncü sanal pencereye geçmek istediğinizde Sanal Pencere Seçicisinde üçüncü sanal pencerenin üzerine fare ile tıklayabileceğiniz gibi Ctrl-F3 tuşlarını da kullanabilirsiniz.</p>
<p>Alt-F2</p>	<p>Menüde olmayan uygulamaları başlatmak üzere komut vermek için kullanılır.</p> 
<p>Alt-Ctrl-Esc</p>	<p>Pencere belirterek uygulama öldürmek için kullanılır. Alt-Ctrl-Esc tuşlarına birlikte bastığınıza fare imleciniz bir kuru kafaya dönüşür. Fare imleci kuru kafa iken üzerine tıklayacağınız pencerenin sahibi olan program öldürülür. Özellikle kontrolden çıkan programlardan kurtulmak için çok kullanışlı bir araçtır. Aynı işi “xkill” programını kullanarak da yapabilirsiniz.</p>
<p>Alt-F4</p>	<p>Aktif penceredeki uygulamayı sonlandırır.</p>

Eh! Artık pencerelere nasıl hükmedeceğinizi öğrendiğinize göre yeni Mandriva LINUX'unuz altında ne gibi uygulama yazılımlarınız varmış bir göz atalım isterseniz...

Mandriva LINUX Uygulamaları

Kitabın yazması belki de en zor kısmına geldik. Ne de olsa üç CD dolusu yazılım yükledik! Şimdi hangi birinden başlamalı?

Gerek Mandriva gerekse KDE'ye özgü uygulama yazılımlarını ve çeşitli sistem ayarlarını tüm ayrıntılarıyla anlatmamız olanaksız. Bunların neredeyse herbiri ayrı bir kitap konusu olabilecek kadar ayrıntılı ve zengin konulardır. Bu kitapta yalnızca oynamak isteyebileceğiniz ya da sistem yönetimi açısından oynamanız gereken yazılımlar hakkında, temel işleri yapmaya yetecek kadar açıklamalar bulacaksınız. Daha fazla ayrıntı için Mandriva'nın yüklenmesi sırasında diskinize yerleştirilmiş olan çevrim içi dökümantasyona ya da İnternet'teki kaynaklara başvurabilirsiniz.

Bu yeni ortamınızda mutlu olmanız bizim için çok önemli olduğundan biraz KDE konfigürasyonunu anlatalım.

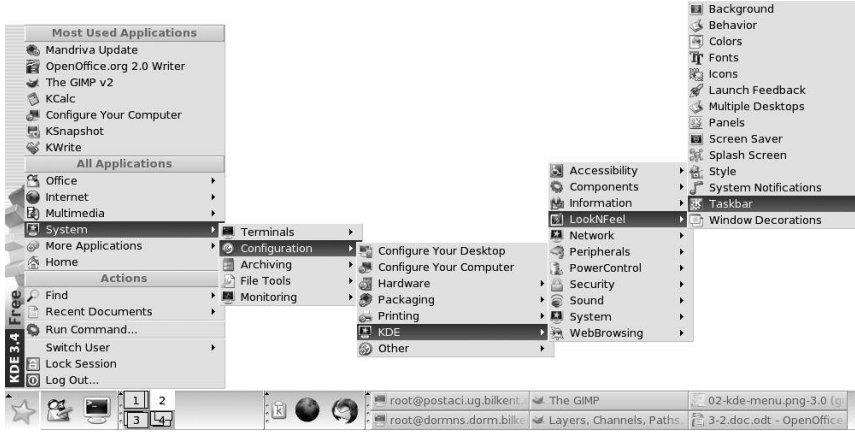
KDE Ayarları

Görsel Ayarlar (Look and Feel)

Masa üstü yönetici olarak KDE de seçmiş olsanız, GNOME da seçmiş olsanız işlevsel açıdan pek fazla bir fark olmayacaktır. Bu nedenle masa üstü ayarlarından söz ederken yalnızca KDE'den örnekler vereceğiz.

Masa üstü ayarları için birçok değişik yol olmasına rağmen yeni başlayanlar için en kolayı KDE menüsünden yapacağınız

Configuration – KDE – LookNFeel seçimleridir.

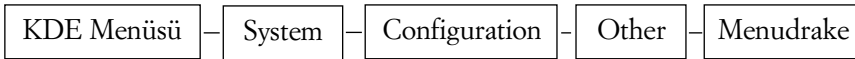


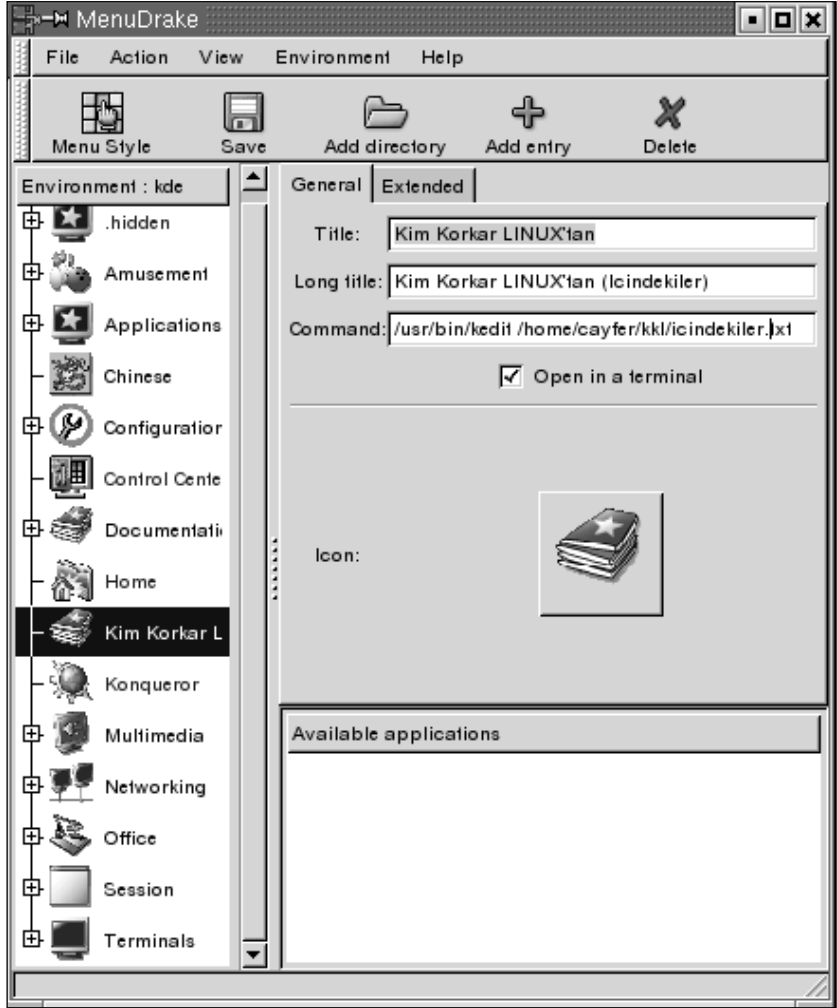
Bu menüden yapabileceğiniz seçimleri ve bunların masa üstünüzün görsel özelliklerinde neden olacağı değişikliklerin hepsini anlatmamıza olanak yok. İstedığınız gibi kurcalayıp zevkinize en uygun şekli bulunca seçimlerinizi kaydedebilirsiniz. Yeri gelmişken; KDE masa üstü düzenlemelerinde bir “tema” kavramı vardır. KDE için gerek dağıtım içinde gelen, gelexse internet’ten indirebileceğiniz çok hoş renk-desen-düzen temaları vardır. Bunları kde.themes.org ve www.kde-looks.org adreslerinde bulabilirsiniz.

Menü Ayarları

KDE menüsünde değişiklik yapmak istediğinizde; örneğin menüye yeni bir program eklemek istediğinizde, kullanmanız gereken program MenuDrake’dir.

MenuDrake programını KDE menüsünden başlatabilirsiniz:





- Önce eklemek istediğiniz menü maddesi için soldaki listeden uygun bir ana başlık (directory) seçin ya da “Add directory” ile yeni bir ana başlık yarattın.
- Daha sonra da “Title” ve “Long title” kutularına menüye yeni eklediğiniz programla ilgili açıklamaları girin.
- Sonra ilgili ana başlığı seçip “Add Entry” seçin.
- “Command” kutusuna da bu programı başlatmak için verilmesi gereken komutu yazın.
- Son olarak da programınıza uygun bir ikon seçip “Save” butonunu tıklayın.

İleri Düzey KDE Ayarları

Son derece kapsamlı olan “KDE ayar seçenekleri için menüden “system-Configuration-KDE” seçin.

Karşınıza kategorilere ayrılmış birçok seçenek çıkacaktır:

Her bir kategori isminin yanındaki ok işaretine tıklayarak daha ayrıntılı başlıklara ulaşabilirsiniz. Merak etmeyin, bütün bu seçeneklere bir yanıt vermek zorunda değilsiniz. Seçeneklerin varsayılan değerleri sisteminizin şu andaki ayarlarını yansıtır ve bunların arasından sadece değiştirmek istediklerinizi kurcalamanız yeterlidir.



KDE Ayarları menüsündeki kategorilere kısaca göz atmak isterseniz:

KDE Ayarları Seçenek Kategorileri	
Accessibility	Engelli kullanıcılara sunulan seçeneklerin yanısıra KDE’de kullanılacak dil, klavye yayılım ayarları gibi ayarların bulunduğu seçimdir.
Components	Web tarayıcı, e-posta istemci programı gibi yazılımların ön tanımlarının yapılması için gereken bileşen ayarlarını yapmak için kullanılır. Örneğin, bir KDE uygulaması içinde bir URL tıkladığınızda, hangi web tarayıcısının başlatılacağı; bir e-posta adresi tıklanığında hangi e-posta istemcisinin başlatılacağını ayarları buradan yapılır.

Information	<p>Aslında buradan yapabileceğiniz bir ayar değişikliği yoktur. Adından da anlaşılacağı gibi sisteminizle ilgili ayrıntılı bilgi alabileceğiniz bir seçimdir.</p> <p>Bilgisayardan iyi anlayan bir arkadaşınız size sisteminizle ilgili tuhaf bir ayrıntı sorarsa “bir dakika” deyip buraya dalaabilirsiniz. Örneğin 600 MHz sandığınız merkezi işlem biriminizin aslında 601.377 MHz frekansında çalıştığını görüp mutlu olabilirsiniz. (Bu sizi gerçekten mutlu ediyorsa, bilgisayarlara kendinizi biraz fazla kaptırmışsınız demektir; dikkatli olmanızı öneririz...)</p>
LookNFeel	<p>Birçok okuyucunun en çok zaman harcayacağını tahmin ettiğimiz bölüm burasıdır. Masa üstünün, pencerelerin, menülerin, kısacası çalışma ortamınızın görünüşünü ve davranışını buradan değiştirebilirsiniz. Sisteminizi, ilk kurduğunuzda karşınıza gelen standart KDE görünümünden kurtarıp, fonda beğendiğiniz bir fotoğraf olan, sevdiğiniz renklerle donanmış, hatta isterseniz pencerelerin değişik yerlerine farenin belli düğmelerinin tıklanmasının tuhaf sonuçlar doğurduğu bir ortam haline getirebilirsiniz. Bu kategoriye biraz kurcaladıktan sonra göreceksiniz ki size belki de gereğinden fazla seçenek sunulmuştur.</p>
Network	<p>İnternet ve diğer ağ konularıyla ilgili, özellikle KDE paketiyle beraber gelen programları etkileyen çeşitli ayarları buradan yapabilirsiniz. Yeni başlayanlar için bu seçim altında oynayacak fazla bir şey olmadığını belirtmekte yarar var.</p>
Peripherals	<p>Bilgisayarınıza bağlı çeşitli aygıtların ayarları bu başlık altındadır. Özellikle klavye ve fareyle ilgili seçenekleri burada bulabilirsiniz.</p>
PowerControl	<p>Özellikle dizüstü bilgisayar kullanan okuyucuların ilgisini çekecek, enerji tasarrufuyla ilgili ayarlar buradadır.</p>
Sound	<p>Ses kartınız ve sisteminizin yapacağı sesli uyarılarla ilgili ayarlar.</p>

System	Sistemin bazı temel ayarlarını, örneğin tarih ve saat ayarlarını buradan yapabilirsiniz. Ayrıca sistem yöneticisi olarak ilgilenebileceğiniz “Login Yöneticisi” gibi seçenekler de buradadır. Bu kategoride değişiklikler yapabilmek için “root” kullanıcı şifresini kullanmanız gerekecektir.
WebBrowsing	Bu başlık altında, İnternet’te dolaşmayla ilgili, özellikle de KDE’nin standart tarayıcısı olan Konqueror programını ilgilendiren ayarları bulabilirsiniz.

Bütün bu seçenekler arasında kendinizi kaybolmuş hissetmeniz normaldir. LINUX’un bunun gibi her konuda çok, ama çok seçim sunmasına alışmanız iyi olur.

Bu bilgisayar dünyasında aslında çok tuhaf terimler var. “Desktop wallpaper” (masa üstü duvar kağıdı) teriminin ne kadar saçma olduğunu hiç düşündünüz mü?



KDE ayar seçenekleri ile yapabileceğiniz herşeyi anlatmaya kalkmak mantıksız olur. Gerisini size bırakıyoruz. Lütfen değişik seçeneklerle oynamaya çekinmeyin. “Defaults” butonunu kullanarak herşeyi eski haline döndürebileceğinizi unutmayın.

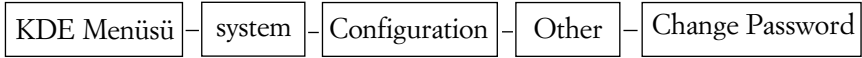
Dikkatli bir kullanıcıysanız, bunca ayar arasında ekran çözünürlüğü ve renk zenginliğiyle ilgili bir ayar kategorisi olmadığını farketmişsinizdir. Bunun nedeni, ekranın aslında KDE masa üstü yöneticisi tarafından değil “X Window” tarafından yönetiliyor olmasıdır. Hatırlarsanız daha önceki bölümlerde LINUX ve UNIX işletim sistemlerinde grafik ekranların denetiminin X tarafından yapıldığını, X tarafından çalıştırılan bir pencere yöneticisinin bu grafik ekranın uygulama programları arasında paylaşılmasını denetlediğini, pencere yöneticisinin çalıştırdığı bir masa üstü denetleyicisinin de kullanıcıya çalışma ortamı hazırladığını anlatmıştık. Aksi belirtilmedikçe KDE masa üstü yöneticisi “kwm” pencere yöneticisi altında çalışır. Aksini nasıl belirteceğinizi ya da aksini belirtmenin ne gibi bir anlamı olduğunu merak etmek için henüz çok erken...



Kullanıcı Yönetimi

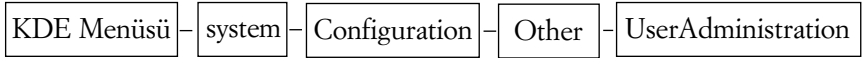
Eğer LINUX bilgisayarınızı sizden başka kimse kullanmayacaksa yalnızca iki kullanıcı tanımlamak yeterlidir: Biri **root** kullanıcı diğeri ise siz! Her ne kadar sisteminize her zaman **root** olarak bağlanmanıza teknik bir engel yoksa da bunu hiç, ama hiç tavsiye etmeyiz. LINUX, **root** kullanıcının ne yaptığını iyi bildiğini varsayıp verilen komutları pek bir uyarıda bulunmadan, aynen yerine getirir. Bu da oldukça tehlikelidir. En küçük hatanızda sistem için son derece kritik bir dosya ya da dizini silebilirsiniz.

Sistemi yalnızca siz kullanıyor olsanız bile (özellikle internet'e bağlı bir bilgisayarda) şifrenizi en geç birkaç ayda bir değiştirmenizi öneririz. Şifrenizi değiştirmek için KDE menüsünden



seçebilirsiniz. Doğal olarak önce geçerli şifreyi vermeniz sonra da yeni şifreyi iki kez girmeniz istenecektir. Bu şekilde şifresi değiştirilecek olan kullanıcı, sistemde o sırada KDE'yi çalıştırmakta olan kullanıcıdır.

Yeni bir kullanıcı tanıtmak gerektiğinde

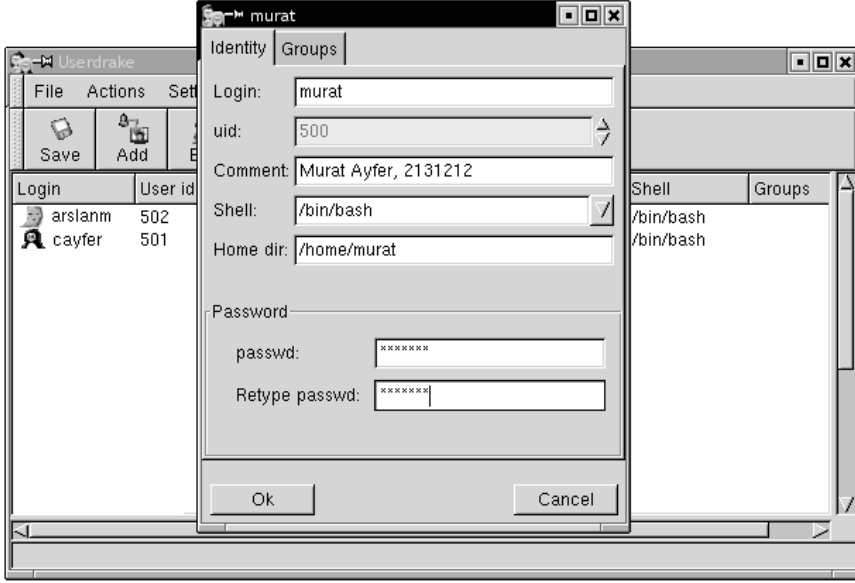


Seçimlerini yaptığınızda “UserDrake” yazılımı başlatılacak ve kullanıcı hesaplarının yönetimi için seçenekler sunulacaktır.

Sisteme kullanıcı eklemek ve çıkarmak doğal olarak sistem yöneticisinin (yani “**root**” kullanıcının) görev tanımına giren bir iştir, bu nedenle önce sistemin “**root**” kullanıcı şifresini girmeniz istenir.



Daha sonra sistemde kayıtlı kullanıcıların da listelendiği ve yeni kullanıcı tanıtmayı yapmak için gerekli araçlarla donatılmış bir pencere görüntülenir:



LINUX'ta aslında her kullanıcı ismiyle değil **uid** (User ID) denilen bir kullanıcı numarasıyla tanınır. İsimler yalnızca bir kolaylıktır. Bu nedenle yeni hesap açılışı sırasında bu yeni kullanıcıya bir de kullanıcı numarası verilmektedir. UserDrake programı, yeni eklediğiniz kullanıcı için daha önce kullanılmamış bir numarayla sıradan atayacaktır. Bu nedenle özel bir nedeni olmadıkça **uid** kutusuna veri girmeniz söz konusu değildir. Kullanıcının **Login** adı olarak tipik olarak 3-8 karakter uzunluğunda ve özel karakterler (örneğin Türkçe'ye özgü karakterler) içermeyen bir isim seçmelisiniz. **Shell** kutusuna ise bu kullanıcı sisteme bağlandığında kendisi için başlatılacak kabuk programını belirtmelisiniz. Varsayılan değer olarak gelen **"/bin/bash"** değerini değiştirmeniz için hiçbir neden göremiyoruz. Aslında bu **Command** değerini kullanarak kullanıcının sisteminize bağlandığında yapabileceği işleri sınırlamak ya da sistemde tanımlı olmasına rağmen sistemi doğrudan kullanmasını önlemek olasıdır ama bu ayrıntılara şimdilik girmeyi çok gereksiz buluyoruz. Şimdi soracaksınız: *"Sistemde tanımlı ama sistemi doğrudan kullanamayan kullanıcı da neden gereksin?"*

Gerekebilir... Diyelim kullanıcılarınıza e-posta adresleri vermeniz gerekiyor ama bilgisayarınızı doğrudan kullanmalarını istemiyorsunuz. O zaman bu

tip kullanıcılar için kabuk programı olarak “**/bin/false**” tanımlarsınız. Webmail ya da POP3 üzerinden kendilerine gelen e-postaları okuyabilirler ve kendileri de e-posta gönderebilirler ama bilgisayarınızın başına bile otursalar, sisteminizi kullanamazlar.

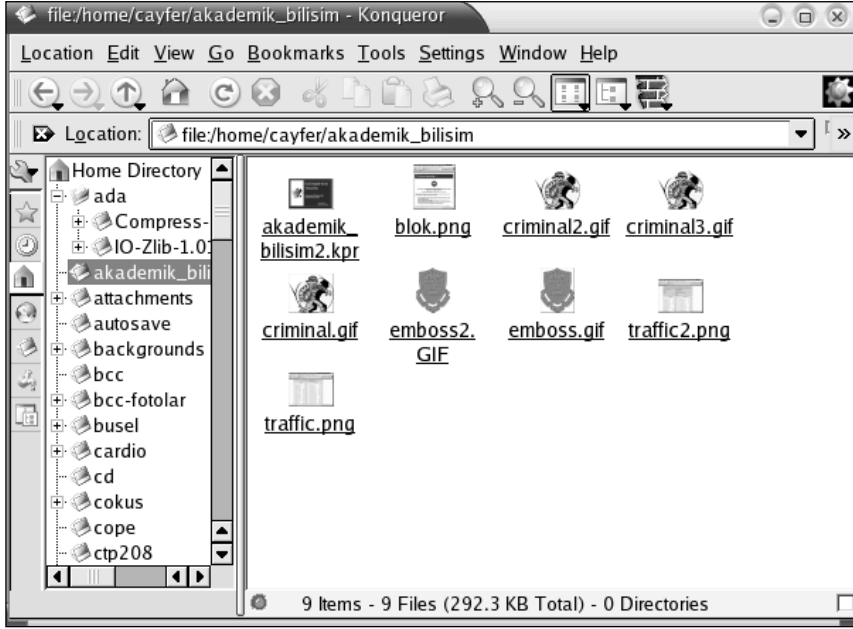
Kullanıcı tanıtımı yaparken verilmesi gereken bir diğer önemli parametre de kullanıcının kişisel dizinidir (*Home Directory*). Bu dizin kullanıcının kişisel dosyalarını saklayabileceği, üzerinde her türlü yazma, okuma ve çalıştırma yetkilerinin bulunduğu bir dizindir. Kısacası bu dizinin sahibi, tanıttığınız kullanıcı olacaktır. Kullanıcıların kişisel dizinleri için kapasite sınırlaması da yapılabilir. Örneğin bir kullanıcının kişisel dizinine yerleştirebileceği dosyaların toplam büyüklüğünü 100 Mbyte ile sınırlamak mümkündür. Ancak bu sınırlamayı yapabilmeniz için sisteminizde kota (*quota*) denetim yazılımının kurulu olması gerekir. Kota denetim sisteminin kurulması ve denetlenmesini “Sistem Yönetimi” bölümünde bulabilirsiniz.

Dosya Yöneticisi: Konqueror

KDE dosya yöneticisi aslında bir web tarayıcısı olan **Konqueror** programıdır. Kullanımı son derece kolaydır, bu nedenle yazılımın nasıl kullanıldığını anlatmayacağız. İçgüdüleriniz ve Windows deneyimleriniz doğrultusunda çek-çekiştir, kes-yapıştır gibi işlemleri deneyin çalıştığını göreceksiniz.

Yalnız unutmayın, KDE dosya yöneticisiyle dizinleri ve dosyaları ancak yetkileriniz elverdiğince görebilir, düzenleyebilirsiniz.





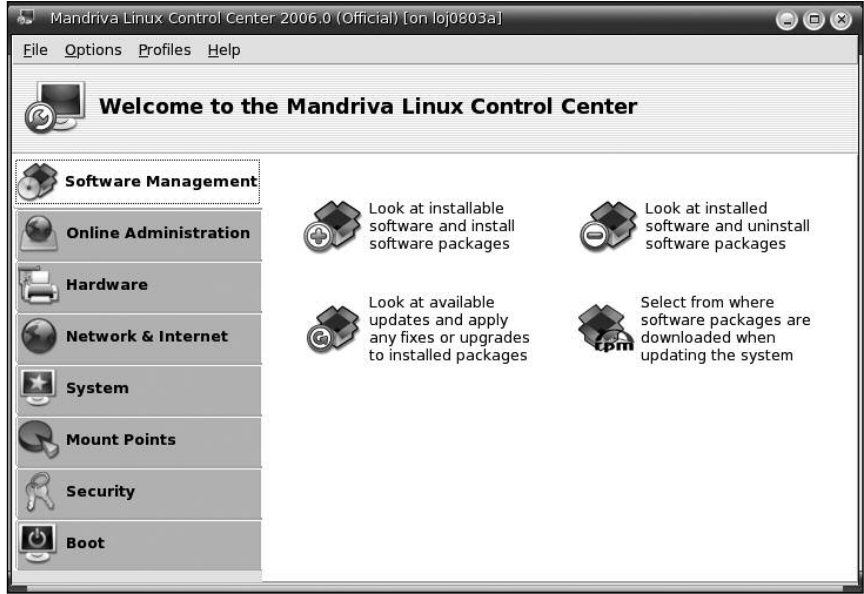
Genel Sistem Yönetimi Araçları: Mandriva Linux Control Center

Sistemin genel yönetimi ile ilgili işlerin bir çoğunu Mandriva Linux Control Center yazılımı ile yapabilirsiniz. KDE menüsünden

KDE Menüsü – System – Configuration – Configure Your Computer

seçimlerini yaptığınızda sisteminizin **root** kullanıcı şifresini girmeniz istenecek, ardından MCC menüsü görünecektir.

Kim Korkar LINUX'tan?



Yeri gelmişken bir kolaylıktan bahsetmek istiyoruz:

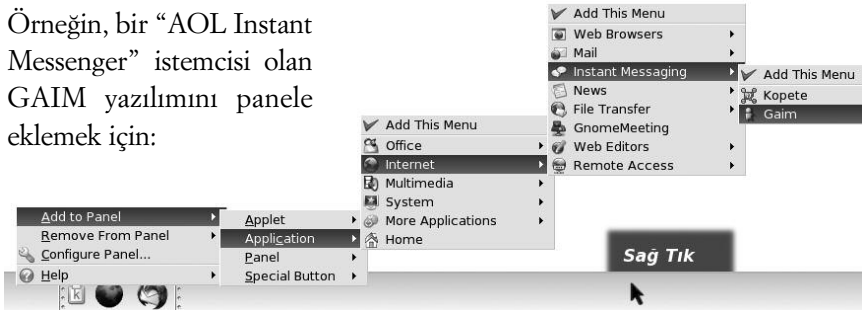
Yukarda gösterdiğimiz gibi, birkaç menüden geçerek ulaşılan bir programı eğer sık sık kullanacaksanız, bu programın ikonunu KDE menüsünün sağındaki “panel” butonları arasına ekleyebilirsiniz.





Bunun için görev çubuğunun boş bir noktasını sağ tuşla tıklayın. Karşınıza çıkan menüden




















seçimlerini yapın ve ardından hangi menüdeki hangi programı panele almak istiyorsanız o programı tıklayın.







Örneğin, bir “AOL Instant Messenger” istemcisi olan GAIM yazılımını panele eklemek için:

















Mandriva Linux Control Center (MCC) Mandriva Linux Denetim Merkezi	
 Software Management	<p>Yazılım Yönetimi: Bilgisayara yazılım yüklemek, yüklü yazılımlar arasında gerek duymayanları kaldırmak ve en önemlisi işletim sistemi/yazılım güncellemelerini yapmak için kullanılır.</p>
 <p>Look at installable software and install software packages</p>	<p>Linux dağıtım CD'leri binlerce yazılım içerir. Kurulum sırasında bunların birçoğu bilgisayara yüklenmez. Örneğin, kurulum sırasında özellikle seçmediyseniz, MySQL sunucu ve istemci programları yüklenmeyecektir.</p> <p>Bu yazılımları sonradan yüklemek için “install software packages” seçimini yapabilirsiniz.</p>
 <p>Look at installed software and uninstall software packages</p>	<p>Kullanmadığınız yazılımları gereksiz güvenlik riski oluşturmamak için ya da diskte yer kazanmak için silmenizi öneririz. “Uninstall software packages” bu iş için idealdir.</p>
 <p>Look at available updates and apply any fixes or upgrades to installed packages</p>	<p>MCC “Software Management” işlemlerinin en önemlisidir. İşletim sisteminin ve Mandriva dağıtımındaki yazılımların güncellemeleri bu seçimle yapılır.</p> <p>Tüm işletim sistemlerinde olduğu gibi güncelleme Linux'ta da çok önemlidir. Özellikle güvenlikle ilgili olanlar...</p>

 <p>Select from where software packages are downloaded when updating the system</p>	<p>MCC, işletim sistemi ve uygulama programları için güncelleme yayınlanıp yayınlanmadığını, yayınlandıysa bunların hangilerinin sizi ilgilendirdiğini öntanımlı bazı sunucularla internet üzerinden bağlantı kurarak araştırır. Bu ön tanımlı sunucuları değiştirmek ihtiyacı duyarsanız kullanmanız gereken MCC seçimidir.</p>
 <p>Online Administration</p>	<p>Çevrim içi sistem yönetimi: VNC (Virtual Network Connection: Sanal Ağ Bilgisayarı) tekniğini kullanarak başka bir bilgisayarın masaüstünü kendi masaüstünüze taşıma işini “Online Administration” ile yapabilirsiniz.</p> <p>Açıkçası bir başka Linux/UNIX bilgisayarını yönetmek için buna hiç gerek yok ama, gene de yaptıklarınızın karşıdaki bilgisayarın başında oturan kişi tarafından da görünebilmesi, eğitim açısından çok yararlı olabilmektedir.</p> <p>Bu işlevin asıl önemi uzaktaki bir Windows bilgisayarını yönetmek gündeme geldiğinde ortaya çıkmaktadır. Evet, yanlış okumadınız! Uzaktaki bir Windows bilgisayarını yönetmekten söz ediyoruz. Eğer bir Windows bilgisayara “Remote Desktop” yazılımı yüklü ve çalışır durumdaysa, Linux bilgisayarınızdan Windows masaüstünü kendi masaüstünüze getirip uzaktaki Windows bilgisayarını kullanabilirsiniz.</p>
 <p>Hardware</p>	<p>Donanım: Bilgisayarınızın çeşitli donanım elemanlarını yönetmenize yardımcı olarak işlevler MCC'nin “Hardware” başlığı altında toplanmıştır.</p>

	<p>Yazıcıdan tarayıcıya klavyeden ekran çözünürlüğüne kadar değişik ayar ve seçimleri burada yapabilirsiniz:</p> <div>  Look at and configure the hardware  Configure your monitor </div> <div>  Change the screen resolution  Set up TV card </div> <div>  Set up the keyboard layout  Set up the pointer device (mouse, touchpad) </div> <div>  Set up the printer(s), the print job queues, ...  Set up scanner </div> <div>  Set up a UPS for power monitoring </div>
 Network & Internet	<p>Bilgisayarın ağ bağlantılarını düzenleme, ayarları yapma ve değiştirme işlemleri “Network & Internet” başlığı altında toplanmıştır.</p>
 Set up a new network interface (LAN, ISDN, ADSL, ...)	<p>Yeni bir ağ bağlantısı tanımlamak için kullanılır.</p>
 Alter miscellaneous internet settings	<p>Bilgisayarınızın adını (hostname) ve DNS ayarlarını yapabileceğiniz veya değiştirebileceğiniz seçenektir.</p>
 Reconfigure a network interface	<p>Ağ arabiriminizin ayarlarını (IP adresi, ağ geçidi, alt ağ maskesi gibi) değiştirmenizi sağlar.</p>
 Monitor the network connections	<p>Ağ bağlantılarındaki trafik akışlarını izlemenizi sağlar. Bu işlevle, gelen giden paket sayıları, geliş ve gidiş hızları gibi parametreleri ölçebilir ve grafik üzerinde izleyebilirsiniz.</p>

 Delete a network interface	Bir ağ arabirimindeki ayarları iptal etmek ve arabirimi devreden çıkarmak için kullanılır.
 Set up a proxy server for files and web browsing	Web taramalarınız için “Proxy” sunucu (Vekil sunucu) ayarı yapmak için kullanılır. Bu olanağı ancak İnternet Servis Sağlayıcı'nız proxy servisi veriyorsa kullanmanızı öneririz.
 Share the Internet connection with other local machines	<p>Eğer bu bilgisayarın ADSL modemle veya Kablo-Modem'le sağlanan internet bağlantısını başka bilgisayarlarla paylaşmak isterseniz gerekli ayarları burada yapabilirsiniz.</p> <p>Aslında bağlantı paylaşma özelliği LINUX işletim sisteminin doğal bir özelliğidir. Yönlendirici (router) olarak çalışabilen her bilgisayar bağlantı paylaşımı da yapıyor demektir. Eh, yönlendirici olarak çalışabilme özelliği LINUX'un doğasında olduğuna göre...</p>
 Wireless connection	Varsa, bilgisayarınızdaki kablosuz ağ arabiriminin (Wi-Fi) ayarları için kullanılır.
 Manage hosts definitions	Varsa, sık sık eriştiğiniz bilgisayarları daha kısa veya başka isimlerle anmanızı sağlayan /etc/hosts dosyasının düzenlenmesini sağlar. Başlangıç düzeyinde pek işinize yaramayacaktır.
 System	Sistem: İşletim sisteminizin çeşitli işlevlerini denetleyebileceğiniz başlıkları içerir.

	<div>  Select the application menu layout and change which programs are shown on the menu </div> <div>  Choose the display manager that enables to select which user to log in </div> <div>  Enable or disable the system services </div> <div>  Manage, add and remove fonts. Import Windows(TM) fonts </div> <div>  Adjust the date and the time </div> <div>  View and search system logs </div> <div>  Open a console </div> <div>  Add, remove or change users of the system </div> <div>  Configure backups of the system and of the users' data </div>
 Select the application menu layout and change which programs are shown on the menu	KDE menüsünde düzenlemeler yapmanızı sağlar.
 Choose the display manager that enables to select which user to log in	Bilgisayarınızın grafik ekranını denetleyecek olan pencere yöneticisini seçmenizi sağlar. Örneğin, (eğer her ikisini de yüklediyseniz) KDE ve GNOME pencereleri arasında seçim yapmanızı sağlar.
 Enable or disable the system services	<p>Sisteminizde kurulu servislerin içinde hangilerinin çalışmakta olduğunu görebileceğiniz bir seçimdir. Bu listede kurulu ama çalışmayan servisler “Stopped” sözcüğüyle belirtilmiştir. “Stop” ve “Start” butonlarına basarak söz konusu servisin hangi komutla çalıştırılabileceğini ya da durdurulabileceğini öğrenebilirsiniz. “On boot” butonu basılı servisler sistemin açılış sırasında otomatik olarak başlatılan servislerdir.</p> <p>Biz kendi LINUX bilgisayarlarımızdaki servislerin denetimini buradan yapmıyoruz. Bu kitabın daha sonraki “Açılış Denetimi” başlıklı bölümde açıkladığımız kavram ve yöntemleri kullanıyoruz.</p>

	<p>Size de böyle yapmanızı öneririz. Böylece sistemin denetimi konusunda hem daha fazla şey öğrenmiş olursunuz hem de sistemi daha rahat denetlersiniz.</p>
 <p>Manage, add and remove fonts. Import Windows(TM) fonts</p>	<p>Yazıtipi (font) konusu LINUX'un daha doğrusu grafik ortamda kullanılan LINUX'un en berbat konusudur. Xorg sisteminin genel amaçlı olma gereklerinden dolayı yazı tipi yönetimi son derece karmaşık bir hale gelmiştir. LINUX ve X ile ileri düzey deneyiminiz olmadıkça bu ayarlarla oynamamanızı öneririz. Bilgisayarınıza yeni bir yazıtipi yükleme gereği duyduğunuzda yazıtipi dosyalarının ekindeki kurulum notlarını okumanızı öneririz.</p> <p>Bu yazıtipi ayarları menüsündeki "Get Windows Fonts" butonuna basarak varsa bilgisayarınızda yüklü bulunabilecek MS-Windows yazıtiplerini X altında kullanılabilir hale getirebilirsiniz. Windows'u satın aldığınızda, içindeki yazıtiplerini de kullanma hakkını satın almış oldunuz nasılsa... Windows yazıtiplerini bu menüden sisteme yükleyebilmek için TTF yazıtipi dosyalarını /usr/share/fonts altında bir dizine yerleştirmeniz gerekir.</p>
 <p>Adjust the date and the time</p>	<p>Bilgisayarınızın tarih ve saat ayarlarını yapmak içindir. Eğer çok hassas bir ayar istiyorsanız ya da bilgisayarınızın saat donanımı yeteri kadar hassas çalışmıyorsa, "Network Time Protocol (NTP)" kullanarak bilgisayarınızın saat ayarının bir saat sunucusu ile yapılmasını sağlayabilirsiniz.</p>







View and search system logs





LINUX'un seyir defterlerini izlemeniz için bir araçtır. Tüm UNIX'ler gibi LINUX da, sistemde olup biten neredeyse her şeyin kaydını tutar. Bilgisayarınızın sunucu hizmetleriyle ilgili log kayıtları genellikle **/var/log** dizini altında, ilgili oldukları programın adından oluşturulmuş dosyalarda ve dizinlere saklanır.





Örneğin, e-posta sunucusu olarak çalışan bilgisayarlarda, e-posta alışverişleriyle ilgili log kayıtları **/var/log/mail** isimli dizinde yer alan dosyalarda biriktirilir. Bu dosyaya bilgisayarınıza gelen ve bilgisayarınızdan gönderilen her e-posta mesajı için bir kayıt düşülür.




Zaman zaman Mandriva Linux Denetim Merkezi'nin bu seçeneğini kullanarak, ya da başka yöntemlerle sisteminizin log dosyalarını incelemelisiniz. Önceleri bu log kayıtları çok anlamsız, daha doğrusu Çince gibi gelecektir. Zamanla bu log satırlarını yorumlamayı ve aralarında bir önlem almanızı gerektiren bir olay kaydı olup olmadığının görmeyi öğreneceksiniz.


Bu log dosyaları doğal olarak kısa sürede hızla büyüyüp diskinizi, daha doğrusu **/var** dizinin bağlandığı (mount edildiği) disk bölümünü doldurma eğilimindedir. Sürekli büyüyen bu log dosyalarına "dosya dönüşümü" (log rotation) işlemi uygulanır. Genellikle sabaha karşı otomatik olarak çalıştırılan logrotate isimli bir yazılım, log dosyalarına bu dönüşüm işlemini uygular. logrotate çalıştığında **/etc/logrotate.conf** dosyasında belirtilmiş esaslar

	<p>doğrultusunda birikmiş log dosyalarının isimlerini değiştirip en eskisini atmaya başlar. Örneğin dönüşüm yapma zamanı geldiğinde e-posta loglarının saklandığı /var/log/mail dizinindeki info, info.1, info.2, info.3, info.4, info.5 dosyaları varsa info.5 dosyası silinir; info.4'ün adı info.5 olarak değiştirilir. Bu şekilde tüm info.* dosyalarının isimleri kaydırılır ve info isimli dosya info.1 yapılır. O andan itibaren de e-posta servisinin yeni log kayıtları info isimli dosyada biriktirilmeye devam eder. Böylece log dosyalarının diskin tamamını kaplayacak şekilde büyümeleri önlenmiş olur.</p>
 Open a console	<p>Bir sistem konsol penceresi açmak için kullanılır.</p>
 Add, remove or change users of the system	<p>Bilgisayarınızı kullanma yetkisi olan kullanıcıların hesaplarını yönetmek için kullanılır.</p>
 Configure backups of the system and of the users' data	<p>Sisteminizin yedeklenmesi için kullanılacak işlemdir. İsterseniz sistemin tamamını, isterseniz sadece kullanıcı dizinlerini yedekleyebilirsiniz.</p>
 Mount Points	<p>Bağlantı Noktaları: Bilgisayarınızın disk ya da disklerindeki bölümleri yönetmek için kullanılan işlemdir.</p> <p>“NFS” (Network File Services) ve “Samba” (Linux kaynaklarının Windows işletim sistemi kullanan bilgisayarlar tarafından kullanılmasını sağlayan servis) servislerini iyice öğrenmeden bu seçenekleri kullanmamanızı öneririz.</p>

 <p>Create, delete and resize hard disk partitions</p>	<p>Disk bölümlemesini değiştirmek, yeni disk taktığınızda bu diskin bölümlemesini yapmak için bu işlevden yararlanabilirsiniz. (Bkz. Bölüm 5 - Dosya Sistemleri).</p>
 <p>Set where your CD/DVD burner is mounted</p>	<p>DVD veya CD-ROM sürücünüzün hangi dizine bağlanmasını istediğinizi belirttiğiniz ayar seçeneğidir. Normal koşullarda bu ayarları değiştirmeniz için hiç bir neden olmayacaktır. Varsayılan ayarlarla tüm kullanıcıların sürücüde takılı olan CD'yi kullanmaya yetkisi olacaktır. Sürücüye bir CD takıldığında ve bu CD'yi kullanan bir yazılım başlatıldığında sürücü <code>/mnt/cdrom</code> dizinine otomatik olarak bağlanır.</p>
 <p>Set where your floppy drive is mounted</p>	<p>Disket sürücünüzün hangi dizine bağlanmasını istediğinizi belirttiğiniz ayar seçeneğidir. Normal koşullarda bu ayarları değiştirmeniz için hiç bir neden olmayacaktır. Varsayılan ayarlarla tüm kullanıcının sürücüde takılı olan disketi kullanmaya yetkisi olacaktır.</p> <p>Sürücüye bir disket takıldığında ve bu disketi kullanan bir yazılım başlatıldığında sürücü <code>/mnt/floppy</code> dizinine otomatik olarak bağlanır.</p>
 <p>Set NFS mount points</p>	<p>UNIX ve türevi işletim sistemlerinde bilgisayarlar arası dizin paylaşımı NFS (Network File System) standardında yapılır. NFS, SUN Microsystems firmasının UNIX dünyasına bir armağanıdır ve uzun yıllardır başarıyla kullanılmaktadır. NFS kullanarak çevrenizdeki UNIX bilgisayarlarda yer alan paylaşıma açılmış dizinlere erişmek için gerekli ayarları burada yapabilirsiniz.</p>

 <p>Manage NFS shares</p>	<p>Kendi bilgisayarınızdaki disklerin başka UNIX/Linux makineler tarafından paylaşılmasını istediğinizde siz de bilgisayarınıza NFS sunucu yazılımı yüklemelisiniz. Yeni başlayanlar için çok gerekli olmayan bir özellik olduğu için kısa kesiyoruz.</p>
 <p>Set Samba mount points</p>  <p>Manage configuration of Samba</p>	<p>Linux bilgisayarınızdaki disk/dizin ve yazıcıları ağ üzerinden MS-Windows bilgisayarlarla paylaşmanız mümkündür. Bunun için Linux bilgisayarınızda “samba” sunucu yazılımı kurulu ve çalışıyor olmalıdır. Windows işletim sistemi, dünyada kendinden başka işletim sistemi olabileceği düşünülmediğinden olsa gerek, kaynak paylaşımını sadece kendi protokolleri ile yapabilmektedir. Samba yazılımı, Linux işletim sisteminde, Microsoft'un paylaşım için geliştirdiği SMB protokolünün kullanılabilmesini, böylece iki yönlü olarak kaynakların paylaşılabilmesini sağlar. Ancak, samba yazılımı başlı başına bir kitap yazılmasını gerektirecek kadar kapsamlı ve yetenekli olduğu için burada ayrıntıya giremiyoruz. Kaldı ki yeni başlayanlar için çok da gerekli değil diye düşünüyoruz.</p>
 <p>Set WebDAV mount points</p>	<p>WebDAV oldukça yeni bir dosya sistemidir. Amacı, web sitelerine ilişkin dosya ve dizinlerin http protokolünün yeni özelliklerini kullanarak uzaktan düzenlemektir. WebDAV hakkında daha ayrıntılı bilgi için www.webdav.org adresine göz atabilirsiniz.</p>

 <p>Set up sharing of your hard disk partitions</p>	<p>NFS ve Samba servisleriyle dizinlerin paylaşımına açılması ve bunlarla ilgili güvenlik ayarları sistem yöneticisinin sorumluluğundadır ve tüm ayarlar bu iki sunucu yazılımının konfigürasyon dosyaları üzerinde yapılan değişikliklerle yapılır. Bazı durumlarda kullanıcıların kendi kişisel dizinlerini paylaşımına açıp kapatmaya yetkileri olmasını isteyebilirsiniz. Güvenlik açısından çok tehlikeli olmasına rağmen kullanıcılarınıza bu yetkiyi vermek istiyorsanız “Partition Sharing” seçimi ile tercihlerinizi belirtebilirsiniz. Bu yetkiyi tüm kullanıcılara verebileceğiniz gibi yalnızca seçilmiş kullanıcılara da verebilirsiniz.</p>
 <p>Security</p>	<p>Bilgisayarınızı dışarıdan gelebilecek saldırılara karşı koruyan “iptables” ateş duvarı (firewall) yazılımının temel ayarlarını burada yapabilirsiniz.</p>
 <p>Set up a personal firewall</p>	<p>Bu ayarları değiştirmedığınız sürece bilgisayarınız dışardan gelebilecek her türlü protokol paketine karşı açıktır. İnanmayacaksınız ama bunun pek bir zararını görmezsiniz. Solucan (worm) ve virüs bulaşma riski Linux kurmanızla birlikte artık çok büyük ölçüde geride kaldı.</p> <p>Ancak Linux kullanıyor olmanız kendinizi savunmamanız için bir neden olamaz. Zaman içinde “iptables” ateş duvarı programını öğrenmek ve uygun ayarları yapmak durumundasınız. Eğer dışarıya hiçbir servis vermiyorsanız bu işlevi seçip “Everything (No firewall)” seçimini kaldırmanızı, böylece dışardan gelebilecek beklenmedik tüm paketleri bloke etmenizi öneririz.</p>

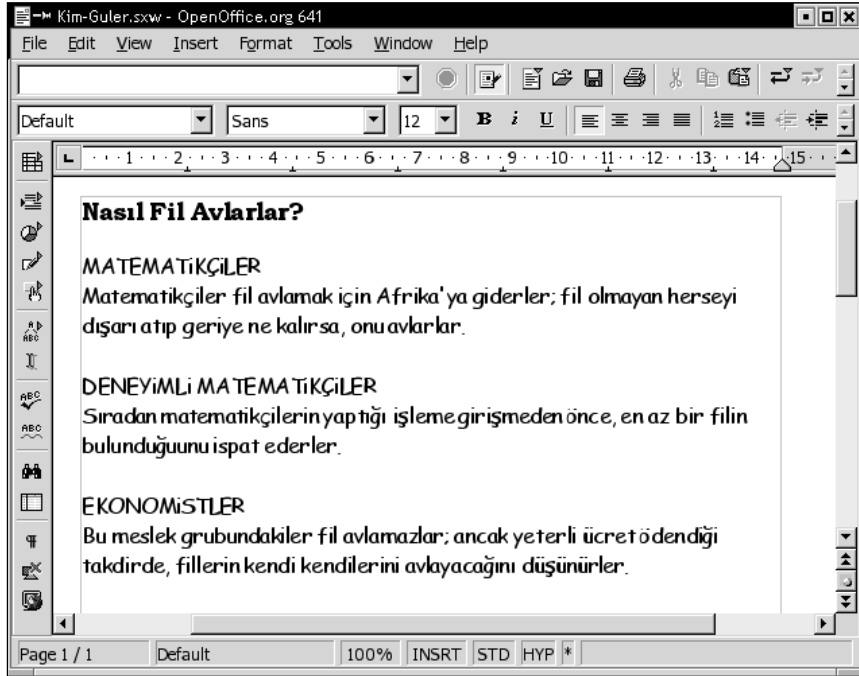
 Boot	Bilgisayarınız açıldığında işletim sisteminin yüklenmesi süreci ile ilgili ayarları yapabileceğiniz işlevler bu başlık altında toplanmıştır. Başlangıç düzeyindeki okuyucuların bu ayarları değiştirmemesini öneririz.
---	--

Ofis Uygulamaları

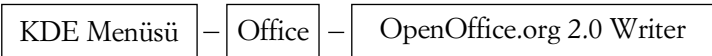
OpenOffice.Org

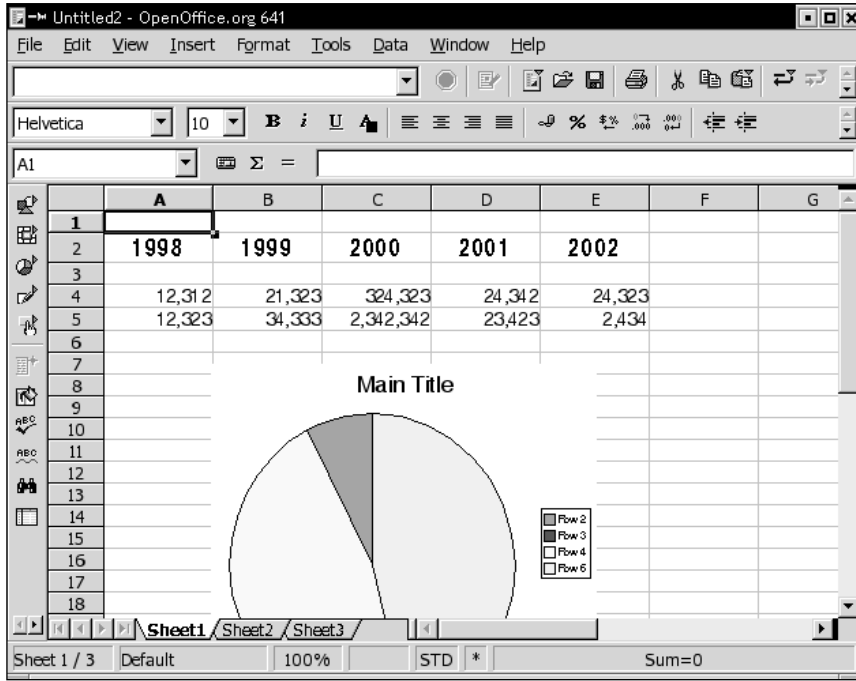
Yakın zamana kadar insanlar Windows işletim sistemini terkedip Linux'a geçmemek için ms-office uygulamalarını mazeret olarak gösteriyorlardı. "Efendim, endüstri standardı!" diyorlardı.

OpenOffice.org ile MS-Office dosyalarını açabileceğiniz gibi dosyalarınızı bu format ta da kaydedebilirsiniz. Buyrun, OpenOffice.org...



OpenOffice Writer: Kelime İşlemci.



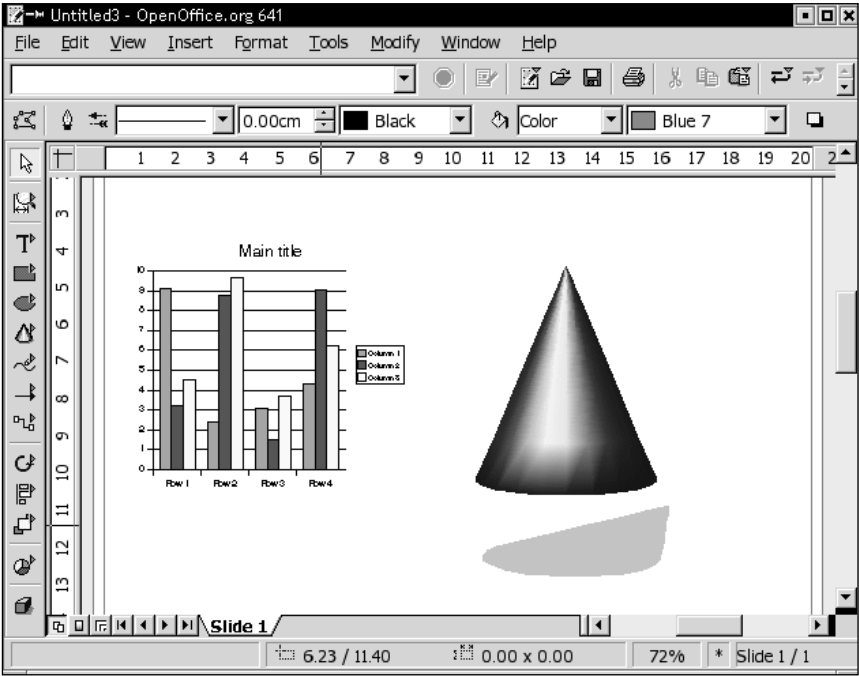


OpenOffice Writer: Kelime İşlemci.

KDE Menüsü

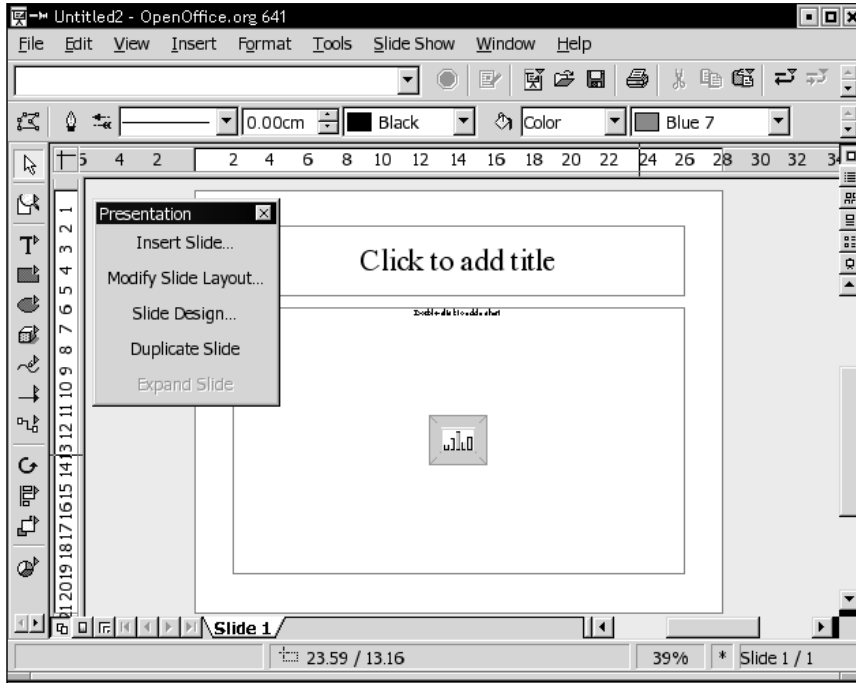
– Office

– OpenOffice.org 2.0 Calc



OpenOffice Draw: Grafik Aracı.

KDE Menüsü – Office – OpenOffice.org 2.0 Draw



OpenOffice Impress: Sunum Hazırlama.

KDE Menüsü

– Office

– OpenOffice.org 2.0 Impress

Grafik Uygulamaları

The Gimp

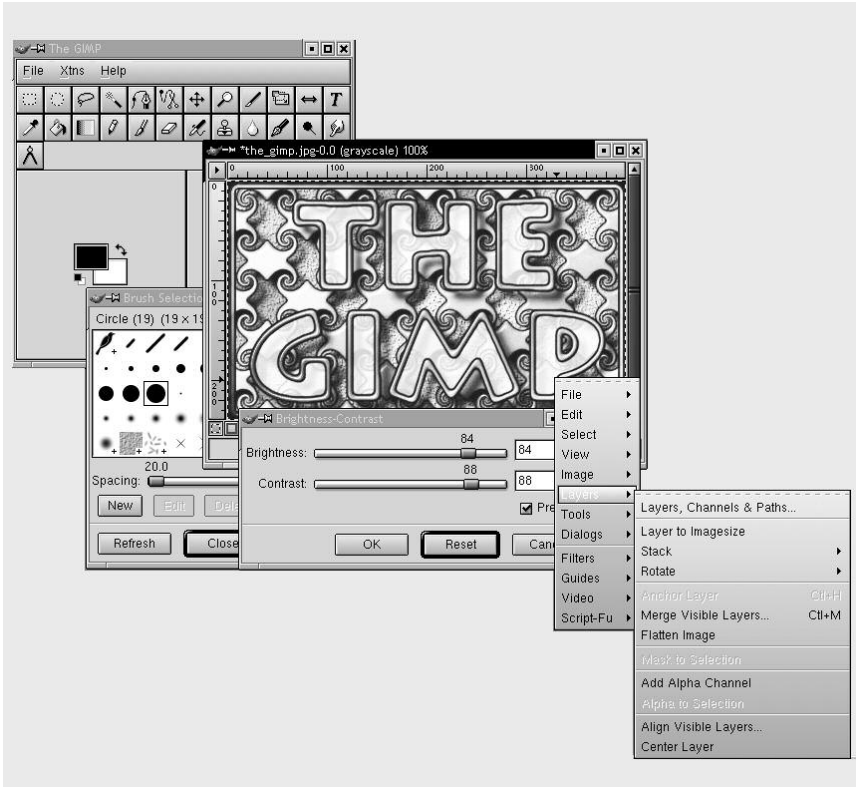
Adobe firmasının Photoshop isimli grafik editörüne çok benzer ve onun kadar yetenekli, başarılı bir yazılımdır. PhotoShop kullanmayı bilen birisinin “The Gimp” paketini öğrenmek için en fazla 10-15 dakikaya gereksinimi olacaktır. Eğer PhotoShop öğrenmeye fırsatınız olmadıysa üzülmeyin, artık gerek de kalmadı! Özgür yazılım kavramı size herkesin 600 dolar ödeyerek satın aldığı programın yaptığı herşeyi yapabilen bir yazılımı ücretsiz olarak sunuyor... Haydi iş başına...

KDE Menüsü

– Multi Media

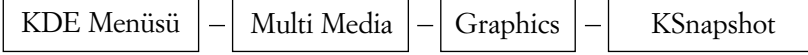
– Graphics

– The Gimp



Diğerleri

Mandriva ile birlikte sisteminize yüklenmiş olan 20'den fazla grafik işleme yazılımı vardır. Kimi yalnızca ikon çizmek için, kimi fotoğraf albümü oluşturmak için kimi de ekrandaki görüntüleri kaydetmek içindir. Örneğin bu kitaptaki ekran görüntüleri



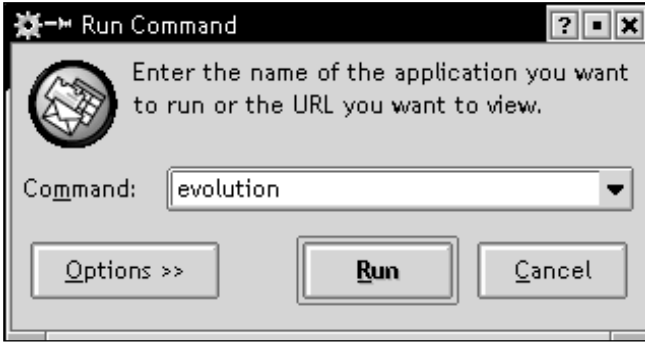
diye başlatılabilen “**Screen Capture**” programıyla kaydedilmiştir.

Kişisel Uygulamalar

Ximian-Evolution

Adres defteriniz, ajandanız ve isterseniz de e-posta yazılımınız. Outlook alışkanlığı olup da bu yüzden MS-Windows'dan vazgeçemeyenler için kusursuz bir uygulama yazılımıdır.

İsterseniz bu kez Evolution programını KDE menüsünden değil, bir başka yöntemle başlatalım; çeşit olsun: Klavyeden Alt-F2 tuşlarına basın (“KDE Menüsü – Run Command” için kısa yol) karşınıza gelen kutuya “evolution” yazıp gönderin.



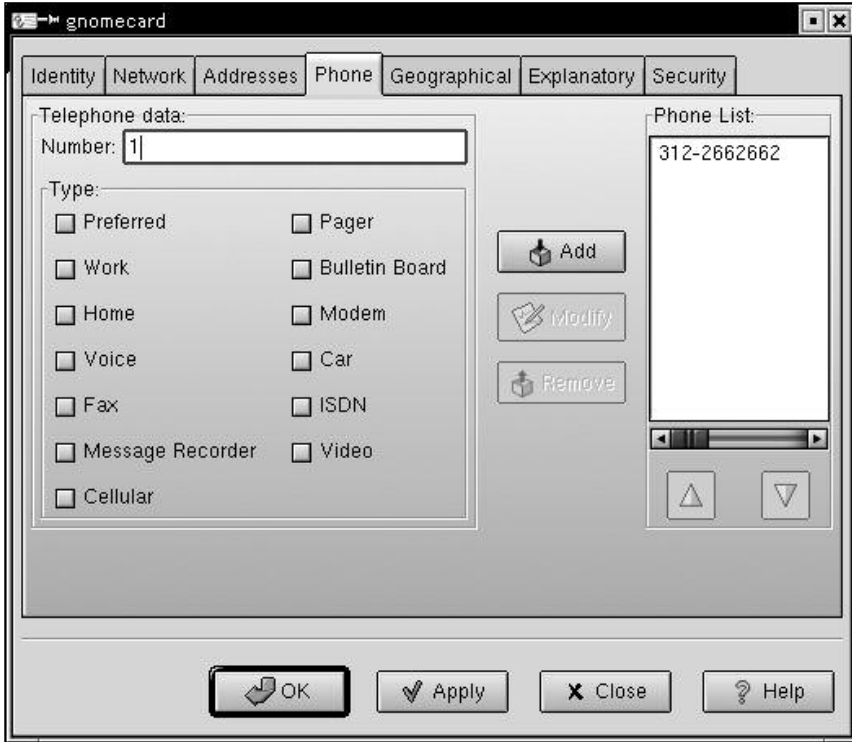
Evolution yazılımını ilk kullanışınızda kişisel ayarlar için sorulacak birkaç sorudan sonra e-posta, adres defteri, ajanda gibi uygulamalara erişebileceğiniz ekran karşınızda belirecektir.

Kim Korkar LINUX'tan?



Gnome-Card

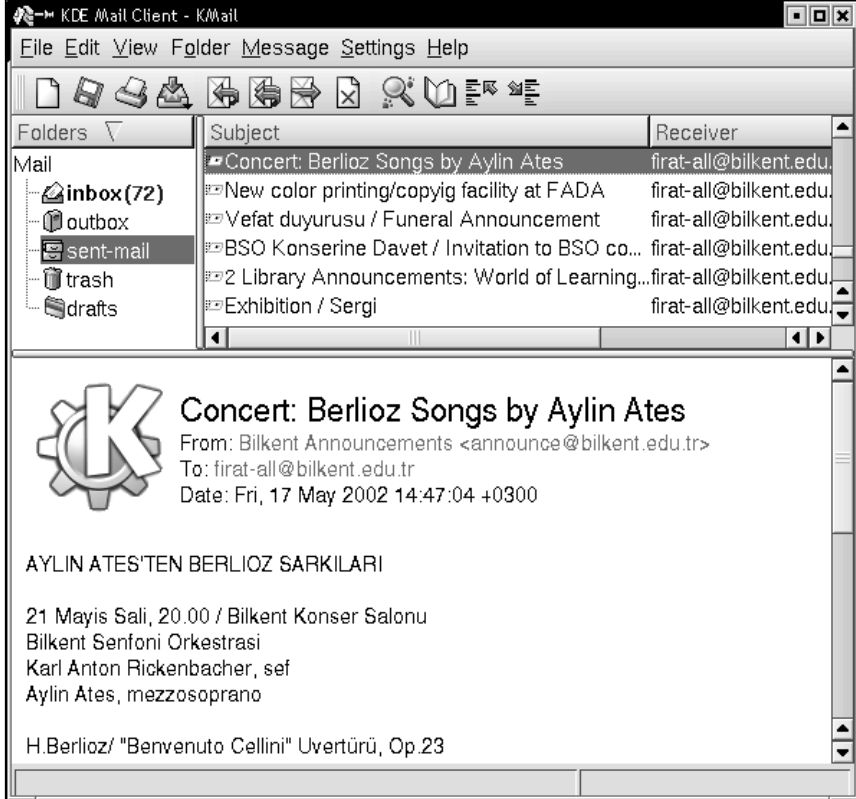
Yalnızca adres defterine gereksinim duyanlar için.



Internetle İlgili Uygulamalar

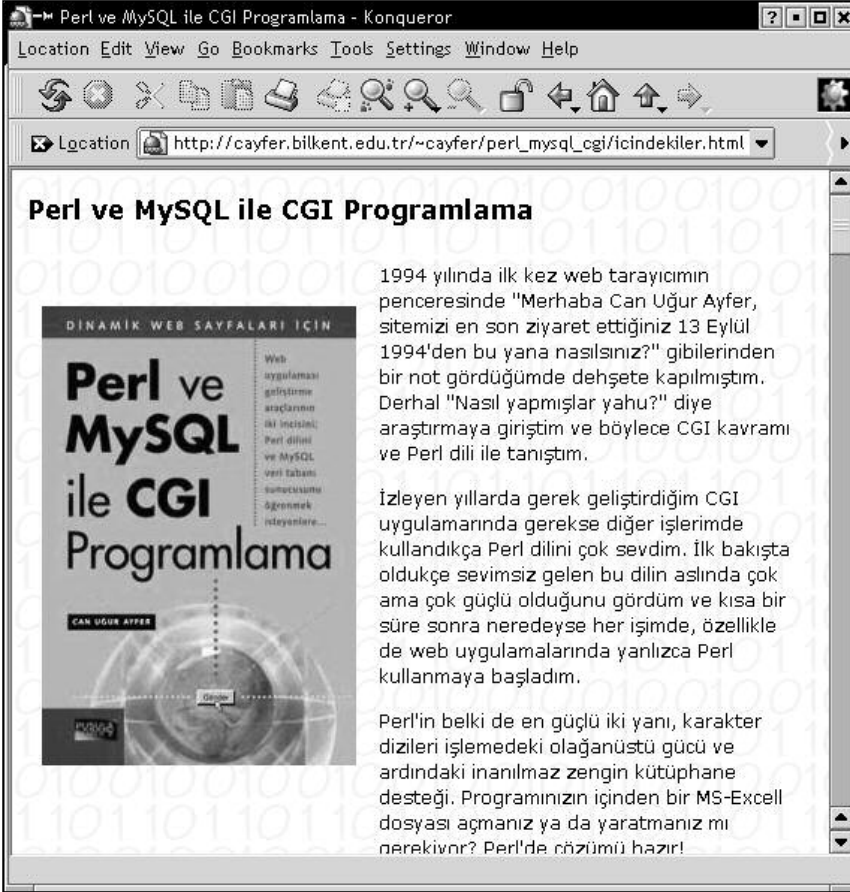
Kmail

Yetenekli bir e-posta istemcisidir. Hem POP3 hem IMAP desteği vardır (IMAP'de neyin nesi diyorsanız boş verin, o kadar önemli değil. En azından sistem yöneticisi olma yolunda ilerlemeye başlayınca kadar).



Konqueror

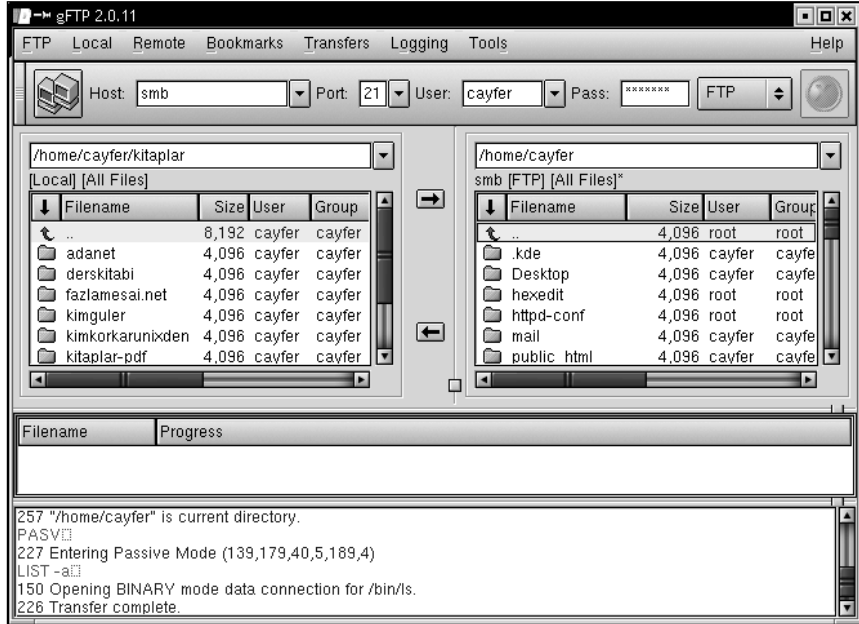
Netscape ve Internet Explorer alternatifi bir web tarayıcısıdır. En önemli özelliği çok hızlı olmasıdır. Javascript ve Java desteği de bulunan bu tarayıcının olsa olsa tek olumsuz tarafı “ile de MS-IE” gerektiren web sayfalarında başarılı olmamasıdır.



FTP İstemcisi (gFTP)

Son derece kolay kullanılan, dosya yöneticisi görünümünde bir FTP istemcisidir. FTP sunuculardan dosya çekmek ve bu sunuculara dosya göndermek için keyifle kullanacağınıza inanıyoruz. Dosya transferi yapmak istediğiniz FTP sunucusunun adını ya da IP adresini "Host" kutusuna yazdıktan sonra "Port" olarak 21 seçin, kullanıcı kimliğinizi ve şifrenizi girip en soldaki ikonu tıklayıp bağlanın. Gerisi dosyaları seçip transfer yönüne ilişkin ok tuşuna basmaktan ibarettir...

KDE Menüsü – Internet – File Transfer – gFTP



Web Sayfası Hazırlama Araçları

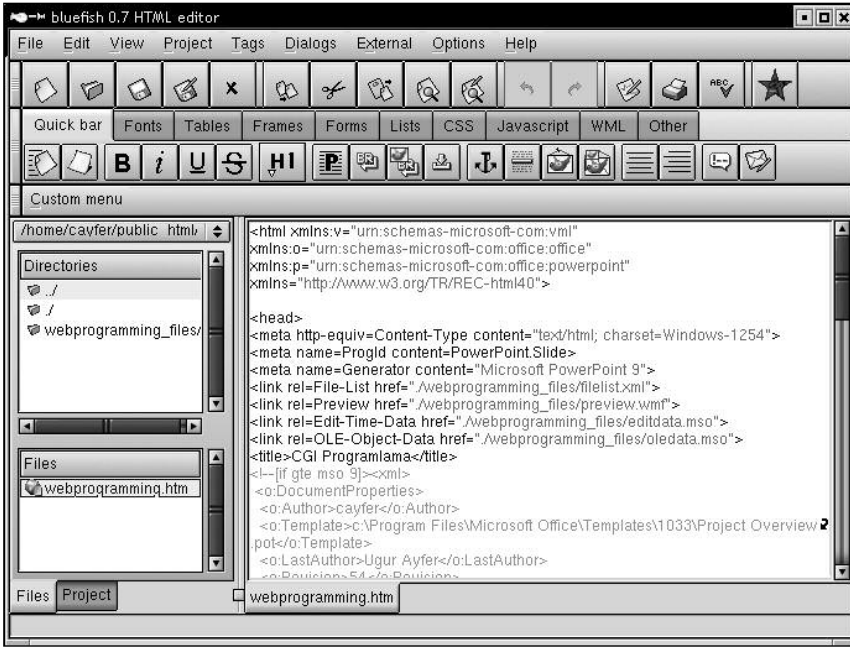
KDE Menüsü

– Internet

– Web Editors

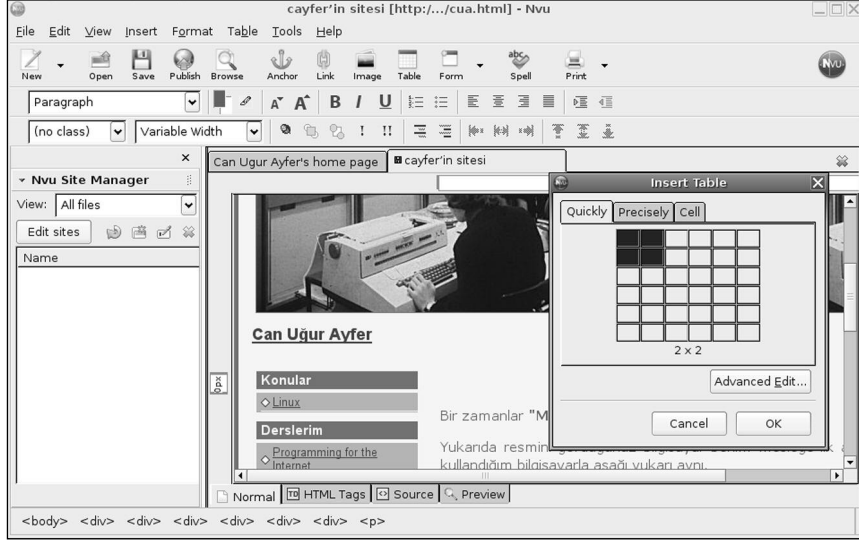
– Bluefish

altında birçok web sitesi hazırlama aracı bulacaksınız. Bunlardan en kullanışlı olanları “Bluefish” ve “Screen” programlarıdır. Mozilla’nın web editörünü özellikle WYSIWYG (What you see is what you get: Ne görüyorsan o) web araçlarından hoşlanana öneririz. Screen, web işine biraz profesyonelce yaklaşanlar içindir.



Kim Korkar LINUX'tan?

Bir de **nvu** web tasarım aracı var: WYSIWYG (What you see is what you get: "Ne görüyorsanız aynen öyle")bir web tasarım aracıdır. "Linux'ta FrontPage yok!" diyenlere yanıtır.



Çoklu Ortam (Multimedya) Uygulamaları

XMMS

Mükemmel bir mp3 çalma yazılımıdır. Ses kartınız varsa mutlaka kurmanız gereken yazılımlardan biridir.

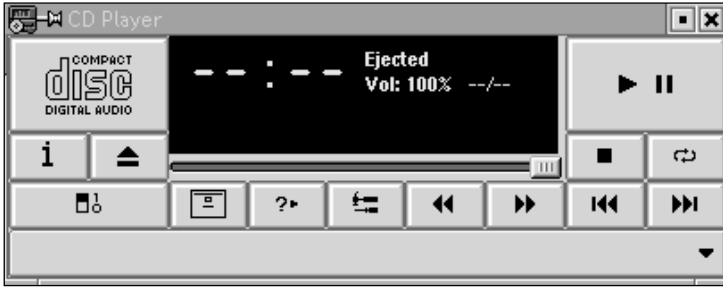


Elbette yegâne MP3 çalarınız XMMS değildir. Totem, nine, Amarok, Juk gibi çok sayıda seçeneğiniz var.



CD Player

Adı üstünde... Bilgisayarınızda Audio CD dinlemek isterseniz kullanabileceğiniz son derece başarılı bir yazılımdır. Ses ayarları için ise “Sound Mixer” yazılımını başlatmanızı öneririz.



Ve Yüzlerce, Binlerce Uygulama Yazılımı...

Mandrake ile birlikte bu kitapta anlatmakla bitmeyecek kadar çok uygulama programı yüklenmiş durumda. Artık bunları keşfetmek size düşüyor.

Kopete Anında Mesajlaşma Yazılımı

Linux altında hayatın çok daha kolay olduğunu ispatlayan, belki de en önemli yazılımlardan biri Kopete'dir. MSN, AIM, Yahoo Messenger, ICQ gibi önemli mesajlaşma servislerinin hepsine birden tek bir yazılımla bağlanmanızı sağlar:

Kim Korkar LINUX'tan?



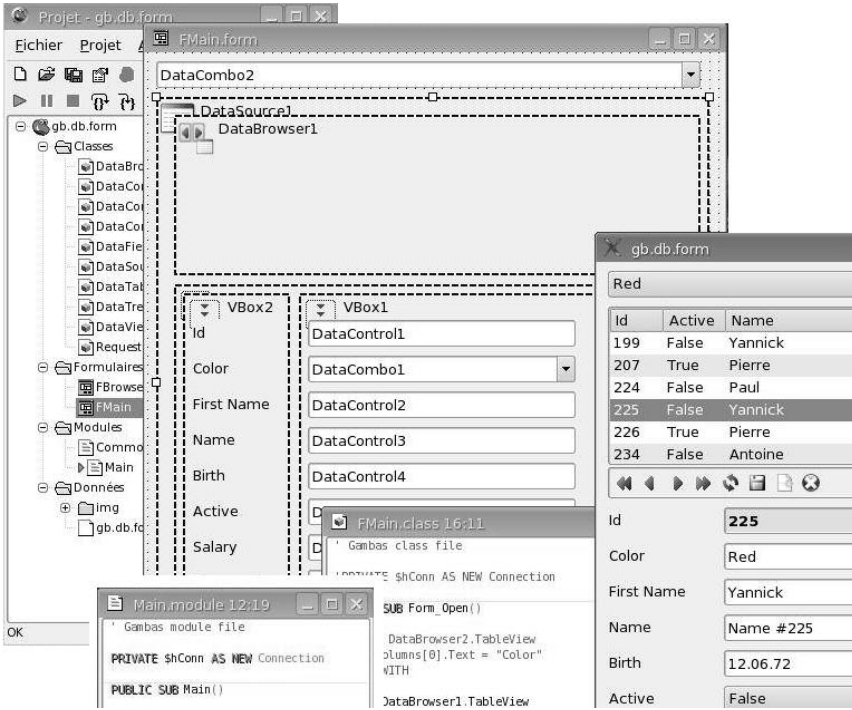
K3B

CD/DVD yazmak için **k3b** var:



Gambas

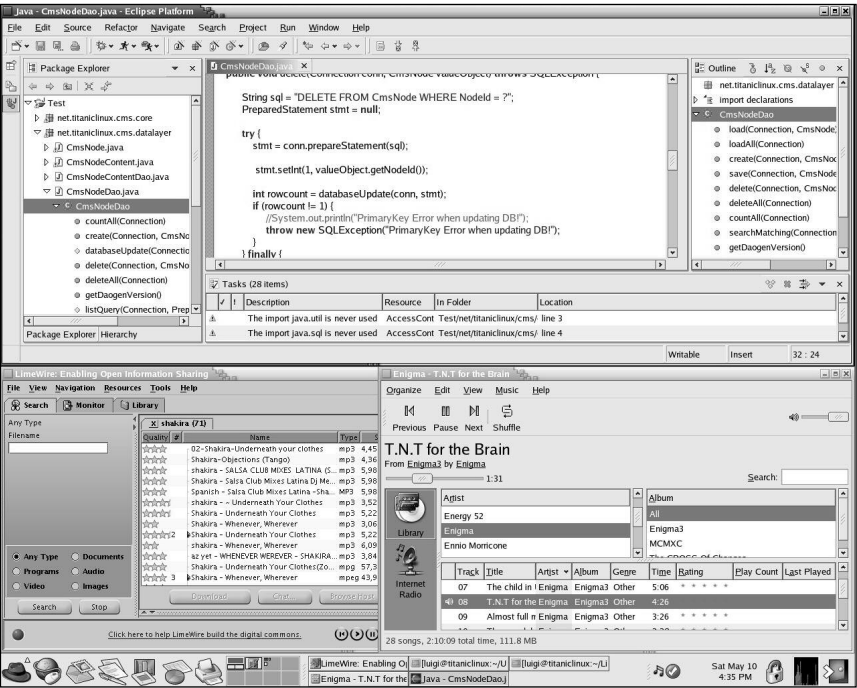
Visual Basic meraklısı programcılar için **Gambas** var:



Kim Korkar LINUX'tan?

Eclipse

Java programcıları için onlarca Java geliştirme ortamı var; en çok beğenilenlerinden birisi de Eclipse:



Digikam

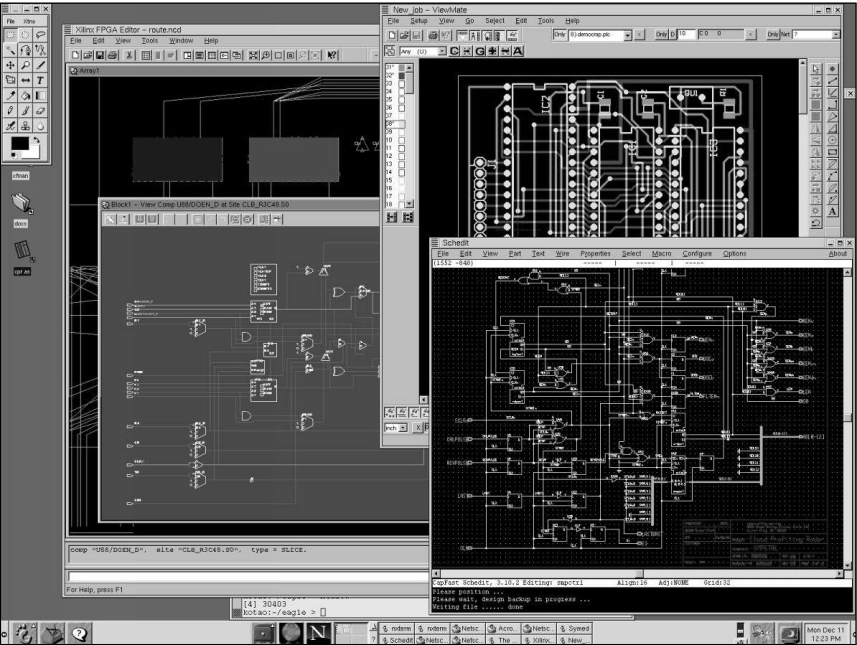
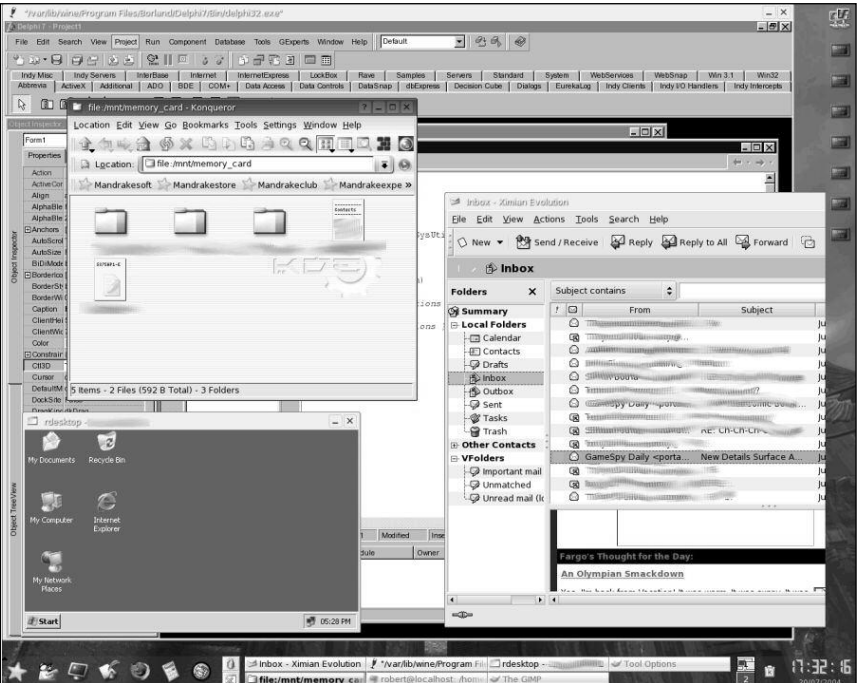
Sayısal fotoğraf makinenizle çektiğiniz resimleri düzenlemek; kırmızı çıkan gözlerin rengini düzeltmek için **digikam** var:



Wine

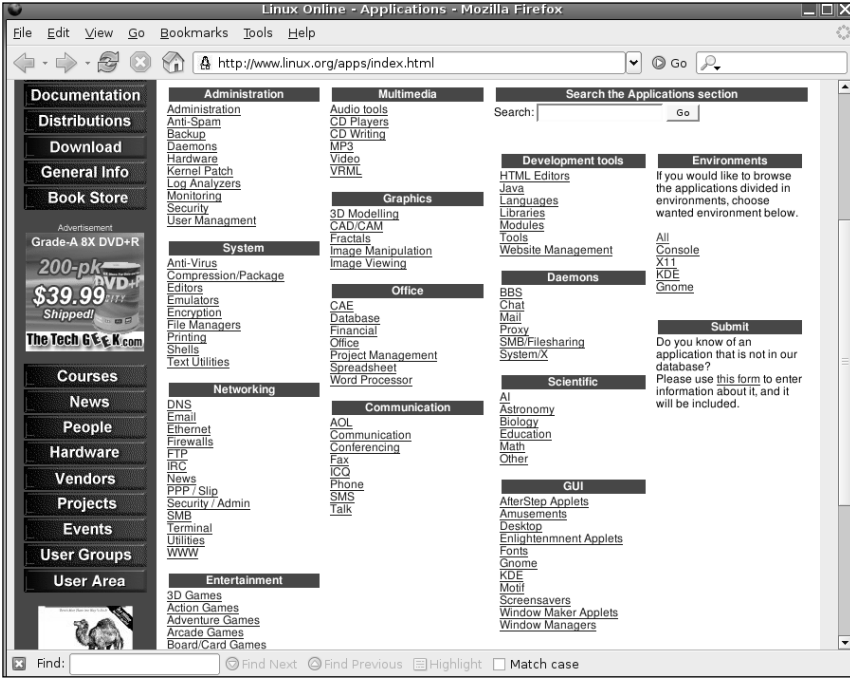
Linux işletim sistemi altında MS Windows yazılımlarını çalıştırmanızı sağlar. Örneğin Delphi ile program yazmak zorundaysanız; ya da elektronik devrelerinizi Xilinx ile tasarlamak zorundaysanız; bir başka deyişle, işiniz gereği bir Windows yazılımı kullanmanız gerekiyorsa, **wine** işinize çok yarayacaktır:

Kim Korkar LINUX'tan?



Oyun meraklısı mısınız? Bilgisayarınızı çoğunlukla bir oyun makinesi olarak mı kullanıyorsunuz? O zaman Linux size göre değil! Her ne kadar çok sayıda Linux oyunu varsa da ve birçok Windows oyunu **wine** altına çalışıyorsa da, Linux sizi mutlu etmeyecektir. Oyun meraklılarına önerimiz, işletim sistemi ve antivirüs vergilerini muntazam ödeyip, virüs ve solucanlarla dolu bir dünyada yaşamaya devam etmeleri ve günde 2-3 kez güncelleme yapmaları, haftada bir kez bilgisayarlarına işletim sistemini yeniden kurmaya devam etmelidir.

Açık kaynak kodlu Linux uygulamalarının tamamını ya da çoğunu bu kitapta listelememiz olası değil. En iyisi sizin arada sırada <http://www.linux.org/apps> adresini ziyaret ederek önemli uygulamaların izlemenizdir:



Bundan Sonrası

LINUX'un grafik araçlarını kullanarak işlerinizin neredeyse hemen hemen hepsini yapabilirsiniz; özellikle kişisel çalışmalarınıza yönelik kelime işlem, elektronik tablolar gibi işler için geliştirilmiş uygulama yazılımlarının neredeyse tamamı X altında çalışacak şekilde yazılmıştır.

Bazı temel LINUX yönetim işlemleri içinse, grafik uygulamalarla uğraşmaya hiç gerek yoktur. Ustalar birçok işi doğrudan LINUX komutlarıyla yapmayı tercih ederler. Bu tür çalışma hem daha fazla esneklik sağlar hem de yapılan işin etkilerinin hemen gözlenmesini sağlar. Bu nedenle **konsol** pencereleri LINUX ortamının vazgeçilmez unsurlarındandır.

Eh, **konsol** deyince LINUX'u doğrudan kullanabilmek için LINUX, daha doğrusu UNIX işletim sisteminin komutlarını öğrenmek gerekecektir. Bundan sonraki bölümlerde KDE ve benzeri masa üstü yöneticilerini kenara bırakıp **konsol** pencerelerinden verilebilecek LINUX komutlarına yoğunlaşacağız. Aslında, LINUX öğrenmeye şimdi başlıyorsunuz.

Konsolda LINUX Kullanımı

“konsol” Kavramı

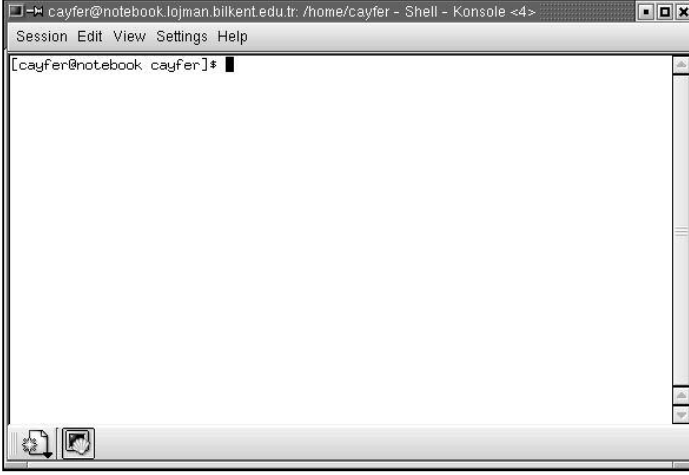
LINUX kullanırken “telnet” ve “konsol” sözcüklerini çok sık duyacaksınız. Bu iki benzer kavramı baştan iyi anlarsanız ileride rahat edersiniz.

Bildiğiniz gibi LINUX çok kullanıcı bir işletim sistemidir. Çok kullanıcı olması aynı zamanda “çok iş düzeni”ni de desteklemesi anlamına gelmektedir. Bir başka deyişle LINUX işletim sistemi altında çalışan bir bilgisayarı bir anda birden fazla kullanıcı kullanabilir ve her bir kullanıcı da birden fazla işi aynı anda yapabilir. Tek merkezi işlem birimine (CPU) sahip bilgisayarlarda bu “aynı anda” sözü azıcık anlamını değiştirse de her merkezi işlem biriminin her kullanıcıya ve işe kısa da olsa biraz zaman ayırarak sırayla hizmet vermesi sayesinde sistem aynı anda birden fazla iş yapıyormuş hissini uyandırır. Özellikle sistem yönetimine yönelik birçok iş yaparken doğrudan LINUX işletim sistemine konut vermeyi tercih edeceksiniz. Bunun için de **konsol** işlevi gören bir uygulama penceresine gereksinim duyacaksınız.

“Konsol”, eskiden büyük bilgisayarları denetlemek için kullanılan, grafik özellikleri olmayan, doğrudan sisteme bağlı, genellikle de sistemin üzerinde bulunan, bir ekran ve bir klavyeden oluşan bir cihazdı. Artık “terminal donanımları” pek kalmadı ama “konsol” kavramı aynen kullanılmaya devam ediyor.

Bilgisayarınızda bir konsol penceresi açmak için KDE ekranının sol alt tarafındaki ikonunu bir kez tıklayınız. Karşınıza gelecek ekran, LINUX işletim sistemiyle doğrudan konuşabileceğiniz terminal ekranı olacaktır:





İstedığınız kadar konsol penceresi açabilirsiniz. Herbiri tamamen bağımsız “terminal”ler gibi çalışacaktır. Üstelik herbirinin görsel özellikleri de farklı olabilir.

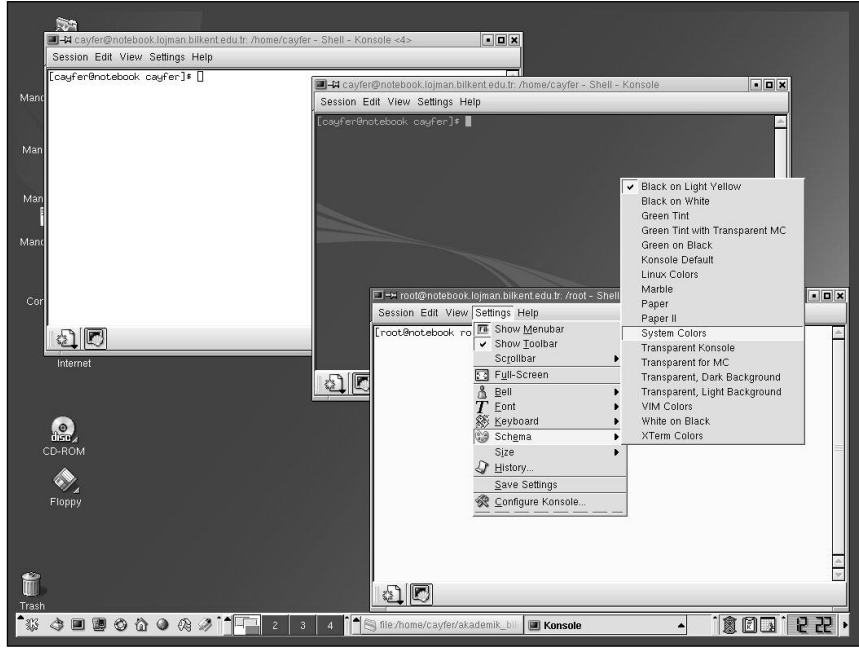
telnet ve ssh

Güvenliğin bilgisayar dünyasında önemli bir sorun olmadığı zamanlarda bir UNIX bilgisayara uzaktan erişmek için kullanılan en önemli ve yaygın araç, TCP/IP protokolleri arasında da önemli bir yeri olan “telnet” uygulamasıydı. “telnet”, temel olarak, bir bilgisayarın yönetim konsolunu uzağa taşımak işini hallederdi (hala da halleder). Ağ üzerinden erişebildiğiniz herhangi bir UNIX bilgisayara bir telnet istemci programıyla bağlandığınızda kullanıcı kodu ve şifresi sorulur; doğru yanıtlar verilirse kullanıcının karşısına, sisteme konsol komutları verebileceği bir terminal açılırdı. “telnet” yazılımları, protokol standartları gereği açık iletişimde bulunur; yani iletişim hattını dinleyenler, bir telnet seansında karşılıklı gelip giden tüm verileri görebilir. Bu zayıflığı ortadan kaldırmak için ssh (Secure Shell) sunucu ve istemci yazılımları geliştirildi.



“ssh” ile iletişime başlarken istemci ve sunucu arasında önce bir kriptografi anahtar takımı oluşturulur (açık anahtarlar şifreleme tekniği ile - Public Key Encryption). Bu adımdan sonra gelip giden tüm veri paketleri artık şifrelidir; hattı dinleyen varsa bir şey anlaması mümkün olmayacaktır. Güvenlik kaygılarıyla “telnet” artık neredeyse tamamen terk edildi ve yerini “ssh” aldı. Ancak, telnet ya da ssh istemcisi ile bir kez bağlantı sağlandıktan sonra terminal/konsol penceresinin görünüşü ve işlevleri tıpa tıp aynıdır.

Kim Korkar LINUX'tan?



LINUX altında çalışan bir bilgisayara sık sık uzaktan erişmek isteyeceksiniz. Uzaktan erişmekten kastettiğimiz, yerel ağ ya da internet üzerinden bilgisayarınıza bağlanmaktır. Örneğin, bir nedenle bir işi ya da servisi durdurup yeniden başlatmanız gerekebilir; sisteminizi kapatıp açmanız gerekebilir. Bu gibi durumlarda bir “ssh istemci” yazılımıyla sisteminizin “ssh” sunucusu servisine bağlanıp sanki konsoldaymış gibi çalışabilirsiniz. Bir başka deyişle sisteminizin “ssh” servisi, uzaktaki bilgisayarda çalışan “ssh sunucu” yazılımının kendi penceresi içinde sizin bilgisayarınıza ait bir terminal çalıştırılmasını sağlayacaktır. Şu anda size çok anlamlı gelmeyebilir ama bu “ssh” servisi olağanüstü değerli bir servistir. Hiç yerinizden kalkmadan yüzlerce LINUX/UNIX bilgisayarı denetleyebilir, yönetebilir ve kullanabilirsiniz. Oysa NT ve türevi işletim sistemi ile çalışan bilgisayarlarda sistem yönetimi ile ilgili işlerde çoğunlukla bilgisayarın başına gitmek gerekmektedir.

KDE masa üstü yöneticisinin standart “terminal” uygulama yazılımı olan **kterm** yazılımını başlattığınızda, zaten sistemi kullanmakta olan bir kullanıcı olduğunuz için size kullanıcı kodu ve şifre sorulmaz. KDE’yi başlatmış olan kullanıcının kimliğiyle çalışan bir terminal penceresi açılır. Bu penceredeki **[cayfer@notebook cayfer]\$** satırı, LINUX’in sizden komut almaya hazır olduğunu belirten **hazır işareti**’dir. (*prompt*)

Bu hazır işaretilinde, LINUX'un sizden komut almaya hazır olduğundan başka çok önemli bir bilgi daha vardır. Bu bilgi, **\$** karakteridir. Şimdi sıkı durun, hazır işaretinizde **\$** görüyorsanız kullanmakta olduğunuz kabuk programı (*shell*) **Bourne Shell** veya onun bir türeği olan **Bourne Again Shell**'dir. (*bash*) (Sabırlı olmanız gerektiği konusunda uyarılmıştık...)

Kabuk (Shell) kavramı, UNIX kullanıcılarının iyi anlaması gereken bir kavramdır.



"[cayfer@notebook tmp]\$" benzeri bir hazır işaretinin karşısına yazacağınız komutu irdeleyen, yapılmasını istediğiniz işe ait programı belleğe yükleyen, gerekli parametreleri bu programa aktaran programlara kabuk programı denir.

Birçok işletim sisteminden (Windows gibi) farklı olarak, UNIX'te, kullanıcının tercihiine bağlı olarak kullanılabileceği birden fazla komut yorumlayıcısı (kabuk = shell) vardır. Bu kabuklara örnek olarak.

sh	Bourne Shell
csh	C-Shell
ksh	Korn Shell
bash	Bourne Again Shell
tcsh	Geliştirilmiş csh

gösterilebilir. LINUX dünyasının en çok beğenilen ve aksi belirtilmedikçe kullanıcıların her terminal bağlantısında varsayılan (default) kabuk olarak başlatılan "bash" kabuk programıdır.

"Yeni kullanıcılar için şimdilik bu kadar bilgi yeter" deyip devam edelim.

Eğer kullanmakta olduğunuz kabuğun (sistem yöneticisinin sizin için uygun gördüğü kabuk) hangisi olduğunu öğrenmek istiyorsanız, hazır işaretinin karşısına **echo \$SHELL** komutunu yazınız. Göreceğiniz, **/bin/bash** benzeri bir satır, kabuk programınız olarak çalışmakta olan programı gösterecektir. "**/bin/bash**", kabuk programı olarak "**/bin**" dizini altındaki "**bash**" dosyasındaki programın çalışmakta olduğunu belirtmektedir.

"[cayfer@notebook cayfer]\$" görünümündeki "hazır" işaretindeki diğer yararlı bilgiler ise şunlardır:

- Kullanıcı kodunuzun “**cayfer**” olduğu,
- Bu terminal penceresinin “**notebook**” isimli bilgisayara ait olduğu,
- Çalışma dizininizin “son bölümünün” “**cayfer**” olduğu; yani çalışma dizininizin “**/home/cayfer**” benzeri bir yapıda olduğudur. (Bu “çalışma dizini” meselesi anlaşılmaz geldiyse hiç üzülmeyin, birkaç bölüm sonra bu konuya ayrıntılı olarak değineceğiz. Şimdilik olduğu gibi kabul edip devam edebilirsiniz.)

Hazır işaretinde yer alan bu bilgiler ilk bakışta size yararsız ve anlamsız gelmiş olabilir. Ancak her bir terminal penceresinde farklı bilgisayarlara farklı kullanıcı kodlarıyla bağlı bulunabileceğinizi düşünürseniz bu bilgilerin ne denli değerli olduğunu kabul edersiniz.

Standart LINUX sistemlerinde kullanılabilecek kabuk programları ve bu programların dosya isimleri şunlardır:

Kabuk Programının Dosya Adı	Kabuk Programının Adı
/bin/bash	Bourne Again Shell
/bin/csh	C-Shell
/bin/sh	Bourne Shell
/bin/ksh	Korn Shell
/bin/tcsh	T C-Shell

Kullandığınız kabuk programı hangisi olursa olsun, temel UNIX kuralları değişmeksizin geçerli olacaktır. Yeni başlayanların, eğer mümkünse, **bash** kabuk programını kullanmalarını öneririz. Bu kitapta göreceğiniz örneklerin büyük çoğunluğu **bash** için verilecektir.

Hangisi olursa olsun, UNIX kabuk programları, oldukça gelişmiş yeteneklerle donatılmışlardır, tabii bir o kadar da karmaşık! Kabuk programlarıyla uygulama programları bile yazılabilir. Gelişmiş programlama dillerinde yer alan “**while**”, “**if**”, “**case**”, “**call**” gibi programlama araçlarının hepsi şu veya bu şekilde kabuk programlarında da bulunmaktadır. Örneğin “bash

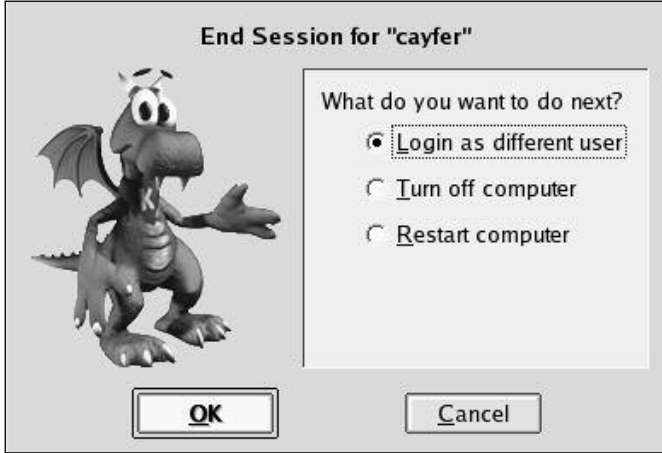
Programlama”, tek başına 400-500 sayfalık bir kitabın konusu olacak kadar geniş bir konudur.

“logout”: Sistemden Çıkış

LINUX işletim sistemi ile yapmakta olduğunuz işleri tamamladığınızda “sistemden logout ederek” LINUX’la kişisel bağlantınızı kesmenizi öneririz. Aslında bu gereklilik yalnızca bir güvenlik önlemidir. Amacı, bilgisayarınızın başına geçebilecek birisinin açık terminal ekranlarından sizin kimliğinizi kullanarak bir şeyler yapmasını önlemektir.

Dikkatinizi çekmek istediğimiz önemli bir konu, logout işlemiyle sistemin kapatılması işlemlerinin birbirlerinden farklı işlemler olmasıdır. Tüm kullanıcıları logout etmiş olan bir LINUX bilgisayarı, arka planda yapması için verilmiş görevlerini yerine getirmeye devam edecektir. Örneğin web, e-posta alıp verme, veritabanı sunum servisi ve ftp servisleri gibi servisler, hiçbir kullanıcı sisteme bağlı olmasa bile kesintisiz devam edecektir.

Sistemin başında grafik ekranla çalışıyorsanız (örneğin KDE altında) sistemle bağlantınızı kesmek için masa üstü yöneticinizin menüsünden “Login as different user” seçimini kullanmalısınız.



Sisteminize bir telnet yazılımıyla uzaktan bağlandıysanız işiniz bittiğinde terminalinizde veya terminallerinizde “**exit**” veya “**logout**” komutlarından birini kullanarak kişisel bağlantınızı kesebilirsiniz.

```
cayfer@notebook.ijman.bilkent.edu.tr: /home/cayfer - Shell - Konsolu <2>
[cayfer@notebook cayfer]$ ls
akademik_bilisin  ndk_configure_sources.png  snapshot2.png  tnp
Desktop          ndk_update.png            snapshot3.png  win-fonts
it               snapshot1.png              snapshot4.png
[cayfer@notebook cayfer]$
[cayfer@notebook cayfer]$ df -h
Filesystem      Size  Used Avail Use% Mounted on
/dev/hda6       8.2G  2.0G  5.8G  26% /
none            125M   0 125M   0% /dev/shm
/dev/hda1       9.8G  3.7G  6.1G  38% /mnt/windows
[cayfer@notebook cayfer]$
[cayfer@notebook cayfer]$ df -k
Filesystem      1K-blocks      Used Available Use% Mounted on
/dev/hda6       8538852    2093704    6011396   26% /
none            127712      0     127712    0% /dev/shm
/dev/hda1      10231392   3873392    6358000   38% /mnt/windows
[cayfer@notebook cayfer]$
[cayfer@notebook cayfer]$ exit
```



LINUX işletim sisteminde bir bilgisayarı paylaşan kullanıcılar söz konusudur. Bu durumda kullanıcıların kayıtlı bilgilerini birbirlerine karşı korumak gerekir. Bir sabah işe geldiğinizde tüm kayıtlı bilgilerinizin kaybolduğunu düşünebiliyor musunuz?

Kullanıcıların kayıtlı bilgilerinin yanı sıra, işletim sistemi, kendisini de hatalı komutlara ve kötü niyetli kullanıcılara karşı korumak zorundadır. Bu koruma mekanizmasının temelinde kullanıcı adı ve şifresi yer almaktadır. Her LINUX kullanıcısı şifresini iyi korumak zorundadır. Şifrenizi belki iyi koruyabilirsiniz; ancak logout komutunu vermeden terminalinizin başından kalkarsanız, arkanızdan terminalin önüne gelip oturan birisi, sizin kişiliğinizi LINUX'a vereceği komutlarla, bilerek ya da bilmeyerek kayıtlı dosyalarınıza zarar verebilir.


Logout etmeniz, bilgisayarı da kapatabileceğiniz anlamına gelmez. Lütfen; ama lütfen, LINUX işletim sistemi ile çalışan bir bilgisayarı işiniz bittiğinde küt diye kapatmayınız. Bir LINUX bilgisayarının sağlıklı bir şekilde kapatılabilmesi için bir dizi törenselleştirme işlemi yapılması gerekir.

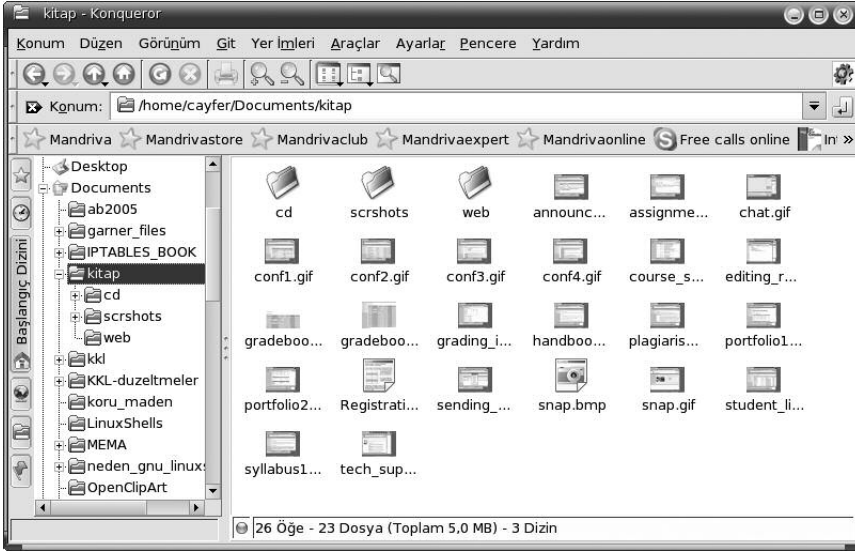
Isınma Hareketleri

Ne Var Ne Yok?

Bir bilgisayarın başına ilk oturduğunuzda genellikle elinizin altında hangi dosyaların ve dizinlerin bulunduğunu görmek istersiniz. LINUX'ta disk(ler)de bulunan dosya ve dizinlerin listesini görmek için uygulayabileceğiniz birkaç yöntem var. En basiti doğal olarak KDE'nin dosya yöneticisini kullanmaktır.



KDE dosya yöneticisini çalıştırmak için masa üstündeki Home ikonunu veya KDE panelindeki  ikonunu tıklayabilir veya KDE menüsünden “Home (Personal Files)” seçimini yapabilirsiniz.



KDE dosya yöneticisi (Konqueror) aslında bir web tarayıcısıdır. Dosya ve dizinleri düzenlemek, kopyalamak, silmek, taşımak için kullanılacağı gibi “Location:” kutusuna geçerli URL adresleri vererek internet web sayfalarında dolaşmakta da kullanabileceğiniz bir yazılımdır.



“Ne var; ne yok?” sorusuna grafik kullanıcı arabirimi kullanarak yanıt aramayı anlatmak istemiyoruz. Bu kitabın asıl amacı LINUX/UNIX öğretmek

Kim Korkar LINUX'tan?

olduğu için aynı soruya LINUX komutlarıyla yanıt aramanın yolunu göstermenin daha yararlı olacağını düşünüyoruz.

Bir terminal penceresi açıp **ls** komutunu verdiğinizde yukardaki grafik arayüzde görüntülenen listenin işlevsel olarak aynısını terminal pencerenizde göreceksiniz:

```
cayfer@lt: ~  
File Edit View Terminal Tabs Help  
-bash-3.00$ pwd  
/home/cayfer/Documents/kitap  
-bash-3.00$  
-bash-3.00$ ls  
announcement.gif      editing_resources.gif  scrshots/  
assignment_detail_1.gif gradebook2.gif         sending_a_message.gif  
cd/                    gradebook.gif         snap.bmp  
chat.gif               grading_info.gif       snap.gif  
conf1.gif              handbook.gif           student_list.gif  
conf2.gif              plagiarism.gif          syllabus1.gif  
conf3.gif              portfolio1.gif         tech_support.gif  
conf4.gif              portfolio2.gif         web/  
course_schedule.gif    Registration_form  
-bash-3.00$
```

Bu şekilde verilmiş bir **ls** komutu, kişisel dizininizdeki dosya ve dizinlerin bir listesini üretecektir. Bu listedeki alt dizinler isimlerinin sonunda bir “/” işaretiyle, çalıştırılabilir program içeren dosyalar da isimlerinin sonunda bir “*” işaretiyle gösterilecektir. Basit dosyaların (veri, yazı vs. içeren dosyalar) isimlerinin sonunda bir tanıtım işareti olmayacaktır. Daha kalabalık bir liste görmek için “**ls /etc**” komutunu deneyiniz.



“home” Dizini

LINUX çok kullanıcı bir işletim sistemi olduğu için her kullanıcının kişisel dosyalarını saklayabileceği bir düzeni de sağlamak zorundadır. Bu nedenle, kullanıcı hesaplarının açılması sırasında her kullanıcının kendi dosya ve dizinlerini yerleştirebileceği bir dizin açılır. Her kullanıcı kendi “home” dizininde her türlü hakka sahip olur; ancak bir başka kullanıcıya ait kişisel dizinde neredeyse hiçbir hakkı olmaz. Apartman hayatındaki “daire”ler gibi bir şey...

Ben Kimim?

İlk bakışta çok anlamlı değilmiş gibi görünen bu soru, UNIX dünyasında zaman zaman sorulması gereken önemli bir sorudur. Eğer kullandığınız UNIX bilgisayar büyük bir bilgisayar ağının bir parçasıysa ve siz bu ağ üzerinden birçok bilgisayara ulaşabiliyorsanız ve bu değişik bilgisayarlardaki kullanıcı isimleriniz (*user-id*) farklıysa, uzun çalışma seansları sırasında herhangi bir pencere içinde geçerli olan kullanıcı kimliğinizi şaşırtabilirsiniz. Korkmayın, UNIX kimlik bunalımına yol açan bir işletim sistemi değildir. O başka bir işletim sisteminin özelliğidir. Hemen,

whoami ve **who am i**

komutlarını vererek, LINUX'un sizi o anda hangi kimlikle tanıdığını öğrenebilirsiniz. Özellikle sistem yöneticileri, zaman zaman başka kullanıcıların kimliğine bürünme gereksinimi duyarlar. Bu işi **su** (*switch user*) komutuyla yaparlar. Bir o, bir bu kullanıcı kimliğine büründüklerinde de bazen şaşırmalar olur. Böyle bir durumda hemen **whoami** komutunu vererek o penceredeki kimliklerini öğrenebilirler.

Şifrenizi Değiştirmek İsteddiğinizde...

UNIX altında çalışan bir bilgisayara sizin adınızı (yani kullanıcı kodunuzu demek istiyoruz) kullanarak ulaşabilen herkes, size gelen elektronik postaları okuyabilir, tüm dosyalarınıza ve dizinlerinize erişebilir, hatta silebilir ve değiştirebilir. Belki daha da kötüsü sizin adınızı kullanarak İnternet'te suç olarak nitelendirilen işler yapabilir.

LINUX erişim şifrenizi sık sık değiştirmelisiniz. "*Ne kadar sık?*" diyorsanız "*altı ayda bir*" deriz. Elbette şifrenizin ortaya çıktığını düşündüğünüz olursa hemen değiştirmelisiniz. Nitekim, bazı sistem yöneticileri, kullanıcılarını belirli sıklıklarda şifrelerini değiştirmeye otomatik olarak zorlayan önlemler alırlar. (**Password aging**: *Şifre eskitme*.)

Bilgisayar dünyasında geçirdiğimiz süre içinde şifresini küçük sarı kağıtlara yazıp ekranın üzerine yapıştıranından, salonun öbür tarafından "*yahu senin şifren neydi?*" diye bağırarak sorulan soruya aynı şekilde bağırarak yanıt veren kullanıcılar bile gördük.

Şifre seçmek ve korumak ciddi bir iştir. Seçtiğiniz şifre, sizin tarafınızdan kolayca hatırlanacak, ancak başkaları tarafından kolayca tahmin edilemeyecek bir karakter dizisi olmalıdır. Eşinizin veya çocuğunuzun adı, soyadınız, arabanızın plakası, doğum tarihiniz, şifre olarak kullanılması sakıncalı olan dizilerdir. Şifre olarak çok karmaşık diziler seçip, sonra da bu şifreyi unutmamak için bir kenara yazmak da çok tehlikelidir.

Şifrenizi seçerken, mümkün olduğunca harf ve sayıları karıştırınız. Daha iyi-si, hem büyük, hem küçük harfleri birarada kullanınız.

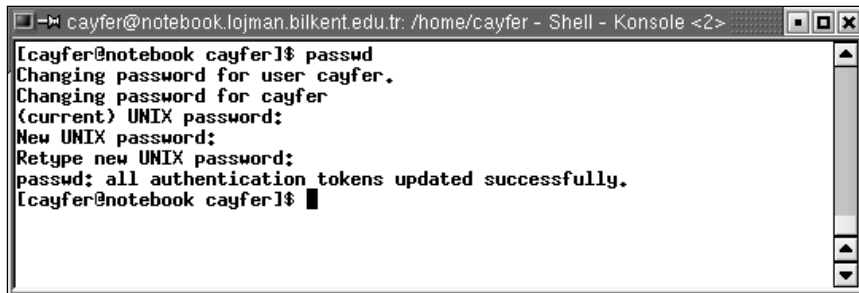
Şifreniz ne çok uzun, ne de çok kısa olsun. 6-8 karakterlik diziler hem kolay hatırlanır, hem de klavyeden yazılırken pek hata yapılmaz.

ayfer	Çok kötü bir şifre, hemen tahmin edilir.
AyfeR-1995	Eh.. Fena değil ama şifre kırma programları hemen çözecektir.
123456	Çok ciddiyetsiz, üstelik klavyeden yazarken kolayca izlenir.
Helmi.	“Hayatta en hakiki mürşit ilimdir” sözünün baş harfleri. Hem kolay hatırlanır, hem de tahmin etmesi çok güçtür. İyi bir şifre!
x1e34TQ?w/&1+	Harika bir şifre, ama siz hatırlayabilecek misiniz bakalım?

Bilgisayara erişim şifrenizi (*password*) değiştirmek istediğinizde,

passwd

komutunu kullanmalısınız. Şifreyi değiştirebilmek için o anda geçerli olan şifreyi bilmeniz gerekecektir.



```
cayfer@notebook.ijman.bilkent.edu.tr: /home/cayfer - Shell - Konsole <2>
[cayfer@notebook cayfer]$ passwd
Changing password for user cayfer.
Changing password for cayfer
(current) UNIX password:
New UNIX password:
Retype new UNIX password:
passwd: all authentication tokens updated successfully.
[cayfer@notebook cayfer]$
```

Yeni şifreyi iki kez vermeniz istenmesi oldukça mantıklı değil mi? Klavyeden yazarken ekranda göremeyeceğiniz bir şifreyi hatalı yazarsanız, bir daha bu sisteme login etmeniz olanaksız hale gelecektir.

Şifrenizi unutursanız **root** kullanıcıdan (sistem yöneticisi) yardım alabilirsiniz. LINUX işletim sisteminde kullanıcı şifreleri tek yönlü bir kriptolama tekniğiyle “şifrelenip” saklanırlar. “Tek yönlü” dememizin nedeni, bu kriptolama tekniğinin geri dönüşü, yani şifrelenmiş bir diziye bakıp bunun neyin şifrelenmiş hali olduğunu bulmanın deneme-yanılma dışında bir yolu olmamasıdır. Ancak **root** kullanıcı sisteminde her şeyi yapmaya, her dosyayı ve dizini okumaya, silmeye, değiştirmeye yetkili olduğu için istediği kullanıcının şifresini istediği zaman değiştirebilir. Üstelik eski şifresini de bilmesine gerek yoktur. Bu nedenle şifrenizi kaybettiğinizde **root** kullanıcı size yeni bir şifre verebilir.

Sisteminizin **root** şifrenizi unutursanız işiniz biraz zorlaşacaktır. Unutulmuş **root** şifrelerinin yenilenmesi konusunu kitabın “Sistem Yönetimi” bölümünde bulabilirsiniz.

İmdaaaaat!..

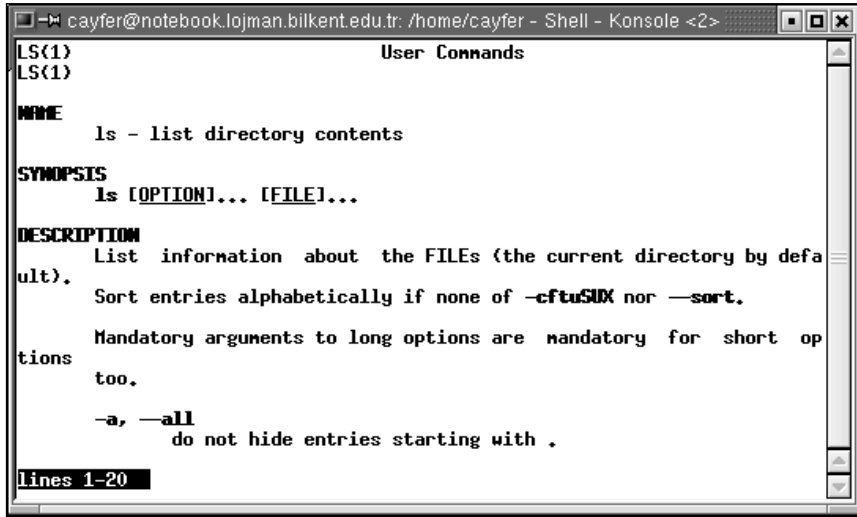
LINUX işletim sisteminde kullanılabilecek binlerce komut vardır. Seyrek kullanılan komutların genel yapılarını ve parametrelerinin hepsini hatırlamak pek kolay olmadığı için, LINUX tüm komutlarının kullanım kılavuzlarını standart ve çevrim içi (*online*) olarak size sunmaktadır. Bir komutun nasıl kullanılacağını öğrenmek ya da hatırlamak istediğinizde,

man komut-adı

komutunu vermeniz, “komut-adı” adlı komutun kullanım kılavuzu sayfalarının ekranınızda görüntülenmesini sağlayacaktır. Örneğin, **ls** komutunun nasıl kullanılacağını, ne gibi seçenekleriniz olduğunu merak ederseniz,

man ls

komutunu kullanabilirsiniz.



```

cayfer@notebook.ijman.bilkent.edu.tr: /home/cayfer - Shell - Konsole <2>
LS{1}
LS{1}
NAME
    ls - list directory contents

SYNOPSIS
    ls [OPTION]... [FILE]...

DESCRIPTION
    List information about the FILES (the current directory by default).
    Sort entries alphabetically if none of -cftuSUX nor --sort.
    Mandatory arguments to long options are mandatory for short options too.

    -a, --all
        do not hide entries starting with .

lines 1-20

```

Güç... Daha Fazla Güç...

Yapmayı düşündüğünüz iş için yetkileriniz mi yetmiyor? **root** olun!

Elbette **root** kullanıcı kimliğine bürünmek için **root** şifresini bilmeniz gerekiyor; ama sistemi siz kurduğunuza göre biliyorsunuzdur.

root kimliğine büründüğünüzde yetkileriniz sınırsızdır. Ancak bu sınırsız yetki, yanında büyük de sorumluluk getirir.

Bir kere ne yaptığınızı iyi bilmelisiniz, zira LINUX, kendisine **root** tarafından verilen komutları sorgusuz sualsiz yerine getirir. Tüm izinleri silme komutunu vererseniz siler! Diski formatlama komutunu vererseniz formatlar!

Öte yandan **root** kullanıcı için sistemde şifreyle korunan hiçbir şey yoktur, **root**, istediği kullanıcının istediği dosyasını açıp değiştirebilir, **root** istediği kullanıcının şifresini değiştirebilir; üstelik eski şifresini bilmesine de gerek yoktur.


Bu kadar yetki açıkçası ürkütücüdür. Geri dönüşü olmayan hatalar yapma olasılığını çok arttırdığı için gerekmedikçe **root** kimliğine bürünmeyin. Gerktiğinde **root** olun ve işiniz bitince hemen normal kullanıcı kimliğinize dönün.

LINUX'ta **root** kimliğine geçmek için terminal penceresinden

su -

komutunu verip arkasından sorulduğunda **root** şifresini girmelisiniz. İşiniz bittiğinde de **exit** komutuyla root kimliğinizi terkedin. (Aranızda politika-cılar varsa bundan hoşlanmayacaktır. **root** kimliğine sahipken **exit** komutu vermek bakanlık koltuğunu bırakmak gibi bir şey olsa gerek).

Aşağıdaki ekranda “**cayfer**” isimli kullanıcı root kimliğine bürünmek için “**su -**” komutunu verip “**mayfer**” isimli kullanıcının şifresini değiştiriyor ve ardından **/var/www/html/mayfer** dizininin ve altındaki tüm dosya ve alt dizinlerin sahibini “**mayfer**” yapıyor.



```

-> root@notebook.loyman.bilkent.edu.tr: /root - Shell - Konsole
[cayfer@notebook cayfer]$ su -
Password:
[root@notebook root]#

[root@notebook root]# passwd mayfer
Changing password for user mayfer.
New UNIX password:
Retype new UNIX password:
passwd: all authentication tokens updated successfully.
[root@notebook root]#

[root@notebook root]# chown -R mayfer /var/www/html/mayfer
[root@notebook root]#
    
```

su komutu yalnızca root kimliğine bürünmek için kullanılmaz. Kullanıcı kimliğinizi “**hasan**” yapmak istediğinizde “**su - hasan**” komutunu kullanabilirsiniz. Normal kullanıcılar **su** ile kimlik değiştirmek istediğinde yeni kimliğin şifresi sorulacaktır. Eğer **root** bir başka kullanıcı olmak isterse şifre sorulmaz.



su komutunu “-” parametresiyle kullanmanın özel bir anlamı vardır:

“**su mayfer**” komutunu verirsiniz kimliğiniz “**mayfer**” olur. Aynı komutu “**su - mayfer**” olarak verirsiniz gene kimliğiniz “**mayfer**” olur ama bu sefer mayfer’in kabuğuna bürünürsünüz; yani mayfer’in kişisel dizinindeki kabuk başlangıç programları çalıştırılır. Bir başka deyişle tam “**mayfer**” olursunuz.

Sisteminizde “root” kimliğine bürünme gereksinimi duyduğunuzda vermeniz gereken “su” komutunu “**su -**” şeklinde vermeye alışmanızı öneririz. (Hatırlıyorsanız **su** komutuna kimlik belirtmezseniz “root” kimliği kabul ediliyordu.)

LINUX Dosya-Dizin Yapısı

Tüm bilgisayar işletim sistemlerinin olduğu gibi, LINUX’un da en temel amaçlarından biri, kullanıcıların verilerini ve programlarını bilgisayar ortamında düzenli bir şekilde saklamalarına yardımcı olmaktır. LINUX işletim sisteminde tüm veriler, programlar ve her biri aslında bir program olan komutlar dosyalarda (*file*); dosyalarsa dizinlerde (*directory*) gruplanmış olarak saklanır.

LINUX dosya yapısını anlatırken okuyucunun MS-DOS/Windows işletim sistemlerine aşina olduğunu varsayacağız ve bu nedenle sık sık Windows’la karşılaştırmalar yapacağız.

Dosya-Dizin İsimleri

Tüm UNIX’lerde olduğu gibi LINUX’ta da dosya-dizin isimlerinde kullanılacak harflerin büyük ve küçük olması farklıdır ve önemlidir. Örneğin;

veriler.2002.dat ve

Veriler.2002.dat

apayrı iki dosya olarak nitelendirilir. Olası karışıklıkları önlemek için önerimiz mümkün olduğunca küçük harflerden oluşan dosya ve dizin isimleri kullanmanızdır.

LINUX'ta dosya ve dizin isimlerinde **uzantı** (*extension*) kavramı yoktur. Yani bir dosyanın adının uzantısına bakıp o dosyanın bir program dosyası olup olmadığına karar veremezsiniz. Bir başka deyişle LINUX'ta çalıştırılabilir program dosyalarının EXE, COM, BAT gibi uzantıları olmayacaktır. “**hesapla.dat**” isimli bir dosya pekala bir program dosyası olabilir ve çalıştırılabilir.

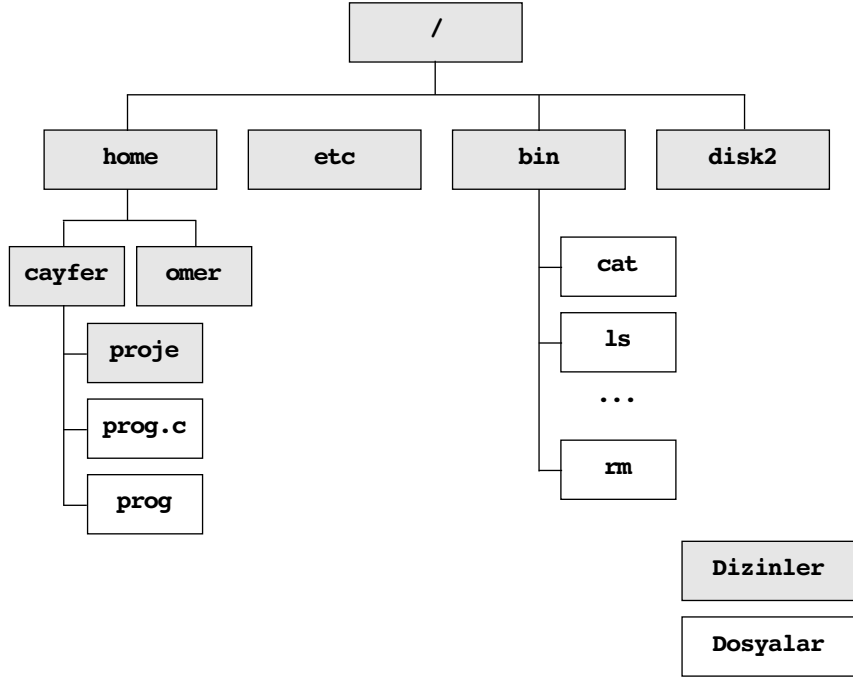
LINUX'ta dosya uzantısının anlamı olmadığı, dosya uzantısı kullanılamayacağı anlamına gelmez. Örneğin Acrobat doküman dosyalarının uzantıları genellikle “**.pdf**” olacaktır; C dili ile yazılmış programların kaynak kodlarının yer aldığı dosyaların uzantıları “**.c**” olacaktır. Bu uzantılar aynı Windows dünyasındaki gibi gelenekselleşmiş uzantılardır. LINUX'taki en önemli fark program dosyalarının “**.EXE**”, “**.COM**” gibi uzantıları olması gereğinin olmamasıdır. Bunun dışında kullanıcılara hayatı kolaylaştıran dosya uzantıları her zaman yararlı olmaktadır.

LINUX dosya-dizin isimlerinde nokta (.) karakterinin özel bir anlamı yoktur. Dosya adı içinde istediğiniz kadar nokta kullanabilirsiniz. Ancak, nokta ile başlayan dosya isimleri bir anlamda özeldir; adı nokta ile başlayan dosyalar yarı gizli dosyalardır. Özellikle belirtmedikçe, dosya isimleri listelerinde bu tür dosyaları göremezsiniz.

ls komutunu “**-a**” parametresiyle kullanmadığınız sürece, adı noktayla başlayan dosyalar listede görünmez. Varlığı kullanıcıları pek ilgilendirmeyen, genellikle çeşitli uygulamalara ilişkin kişisel tercihlerin saklandığı dosyalara, **ls** listelerinde kalabalık yapmasınlar diye noktayla başlayan isimler verilir.

LINUX, Windows'dan da tanıdığınız hiyerarşik dosya-dizin yapısını kullanmaktadır. En üst düzeyde bir **root dizini** ve bunun altında gerektiği gibi yerleştirilmiş olan dosya ve alt dizinler ile gene bu alt dizinler altında yerleştirilmiş dosyalar ve gene alt dizinler...

Şematik olarak göstermek gerekirse:



Dikkat ederseniz, Windows dosya yapısından farklı olarak root dizininin adı “\” (*back-slash*) değil, normal “/” (*slash*) karakteridir. Aynı şekilde, bir dosyanın dizinler arasındaki yerini tanımlarken, Windows’taki “\” karakteri yerine “/” karakteri kullanılır. Bunu örneklerle göstermek gerekirse, yukarıdaki dosya-dizin yapısında yer alan bazı dizin ve dosyaların tam isimleri şöyle yazılır:

```
/home/cayfer/proje  
/home/cayfer/prog.c  
/bin/cat
```

Her Horoz Kendi Çöplüğünde...

LINUX işletim sisteminde, her kullanıcının kendisine ait bir “kullanıcı dizini” ya da “kişisel dizini” (UNIX terminolojisinde: **home directory**) vardır. Bu dizin, kullanıcının sisteme tanıtımı sırasında, sistem yöneticisi tarafından yaratılır. Her kullanıcının kendi “kullanıcı dizini”nde ve bu dizin altında yer alan dosya ve alt dizinler üzerinde sınırsız yetkileri vardır. Bu dizin altında istediği gibi dosya ve alt dizinler yaratır, bunları siler, isimlerini ve içeriklerini değiştirir vs. vs.

Her kullanıcının kendi dizinindeki bu yetkileri, başka kullanıcıların dizinleri üzerinde yoktur. Bir başka deyişle, “**ayfer**” isimli kullanıcı, “**omer**” isimli kullanıcının dizinindeki dosyaları silemez, değiştiremez, “**omer**” izin vermedikçe okuyamaz; hatta varlığından bile haberdar olamaz.

Sisteme bağlanan her kullanıcı, çalışma dizini, kendisine ait kullanıcı dizini olacak şekilde çalışmaya başlar. Sistem yöneticileri, kullanıcı dizinlerini, genellikle “/home” dizini altına açtıkları dizinler olarak düzenlediklerinden (tipik bir UNIX geleneği), **cayfer** isimli kullanıcının login ettiğinde kendini “/home/cayfer” gibi bir dizinde bulması doğaldır.

login : cayfer

Password :

[cayfer@notebook cayfer]\$

Bu örnekteki **notebook**, kullandığınız LINUX bilgisayarının adıdır. Eğer bilgisayarınız bir bilgisayar ağına bağlıysa, bu adın sistemin hazır işaretinde (*prompt*) görünüyorsa terminal pencerenizin hangi bilgisayara bağlı olduğunu gösterdiği için çok işinize yarayacaktır.

Özel bir kullanıcı olan “**root**” kullanıcısının kişisel dizini “/home” dizini altında değil, “/” dizini altında yer alır (“/root”).

Her ne kadar sisteme bağlandığınızda LINUX sizi kendi kullanıcı dizininize yerleştirirse de, bu yerleşim mutlak değildir. İsterseniz **cd** komutu ile çalışma dizininizi (*default directory* veya *current directory*) değiştirebilirsiniz.



```

cayfer@notebook.ijman.bilkent.edu.tr: /usr/bin - Shell - Konsol <2>
[cayfer@notebook cayfer]$ pwd
/home/cayfer
[cayfer@notebook cayfer]$
[cayfer@notebook cayfer]$ cd /usr/bin
[cayfer@notebook bin]$
[cayfer@notebook bin]$ pwd
/usr/bin
[cayfer@notebook bin]$ █

```

Neredeyim?

Çalışma dizininizin hazır işaretinde gösterilmesini sağlayan kullandığınız kabuk programıdır. (Buradaki örnekler **bash** kabuğu için verilmiştir.) Çalışma

Kim Korkar LINUX'tan?

dizininizin tamamını hazır işareti içinde göremeyebilirsiniz. Komut satırının çoğunu harcamamak için **bash** kabuğu hazır işareti içinde çalışma dizininin sadece son bölümünü görüntüler. Nerede olduğunuzu hatırlatmaya yardımcı olsun diye...

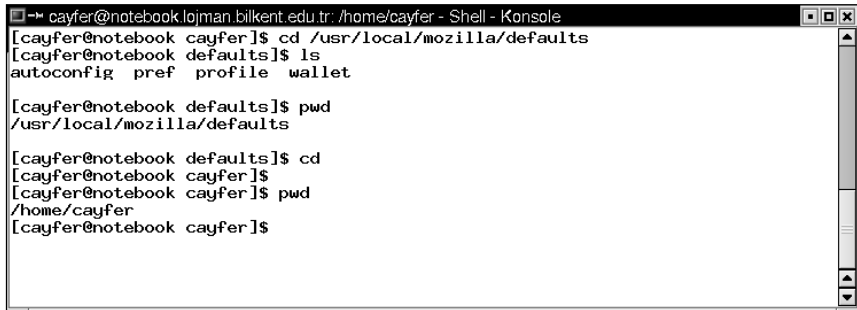
Eğer çalışma dizininizin tam yerini görmek istiyorsanız, yukardaki örnek ekranda da görebileceğiniz gibi

pwd (*print working directory*)

komutunu kullanabilirsiniz.

Yuvaya Dönüş

Çalışma dizininiz neresi olursa olsun **cd** komutunu parametresiz olarak verirseniz, kişisel dizininize dönersiniz.



```

cayfer@notebook lojman.bilkent.edu.tr: /home/cayfer - Shell - Konsole
[cayfer@notebook cayfer]$ cd /usr/local/mozilla/defaults
[cayfer@notebook defaults]$ ls
autoconfig  pref  profile  wallet

[cayfer@notebook defaults]$ pwd
/usr/local/mozilla/defaults

[cayfer@notebook defaults]$ cd
[cayfer@notebook cayfer]$
[cayfer@notebook cayfer]$ pwd
/home/cayfer
[cayfer@notebook cayfer]$
```

Ne Var Ne Yok? (Ama Bu Kez Daha Bir Merakla...)

Çok doğal olarak, bulunduğunuz dizinde yer alan dosya ve alt dizinlerin bir listesini görmek isteyeceksiniz.

Kullanacağınız komut en basit haliyle:

ls

komutudur.

```

cayfer@notebook.lojman.bilkent.edu.tr: /home/cayfer - Shell - Konsolu <2>
[cayfer@notebook cayfer]$ ls
addrbook.tar.gz      evolution/           mp3selection
akadenik_bilisin/   foto1.jpg           ok11
akadenik_bilisin.kpr hyperhelp*          oner-resume-doc
amavis*             java/              oner-resume.html
anti-span           Kim-Guler.sxw       oner-resume.pdf
attachments/        kitaplar/           onlisans.pdf
backgrounds/        kkk.html           playlist
bashrc.txt          lisans.pdf          post/
blackbox-menu       login1.cgi          public_html/
bnp                 login.cgi           res.txt
cgi.doc             loj-net.kwd         Say
ctp208/            lsx.sh*             Sent
cua.JPG             Mail/              snapshot1.png
Desktop/            ndk_configure_sources.png Trash
dorm-mail           ndk_update.png
Drafts              niata_linux.jpg
[cayfer@notebook cayfer]$

```

Bu listede dizinler, isimlerinin sonuna yerleştirilen “/” karakterleriyle; program veya komut dosyalarıysa “*” ile belirtilmiş olarak karşınıza çıkacaktır. Herhangi bir eki olmayan isimlerse, program dosyası veya dizin olmayan sıradan dosyalara aittir.

Bazı isimlerin sonunda “@” işareti göreceksiniz. Bu işaretin anlamını açıklamak için henüz biraz erken; ama şimdilik şu kadarını söyleyebiliriz: “@” işaretli dosya veya dizinler, aslında orada olmayan dosya ve dizinleri belirler. Bu liste her zaman alfabetik sırada ve dosya isimlerinin izin verdiği ölçüde birden fazla sütun halinde dökülecektir. Bu listeye önce ilk sütunu, sonra diğer sütunları göreceğiz şekilde bakmaya alışmalısınız.

Dosyalar ve dizinler hakkında daha detaylı bilgi istiyorsanız “**ls -l**” komutunu kullanmalısınız. **ls** komutunun bu formunu mutlaka deneyiniz ve bu form ile alacağınız listenin nasıl yorumlandığını lütfen çok çok iyi anlayınız. LINUX dosya sisteminin mantığını iyi kavrayabilmeniz açısından buradan başlayarak anlatılanlar oldukça önemlidir.




```
cayfer@notebook.ijman.bilkent.edu.tr: /home/cayfer - Shell - Konsol <2>
[cayfer@notebook cayfer]$ ls -l
total 119732
-rw-r--r-- 1 cayfer cayfer 16267 Sep 26 16:25 addrbook.tar.gz
drwxr-xr-x 5 cayfer cayfer 4096 Feb 9 10:11 akademik_bilisin
-rw-rw-r-- 1 cayfer cayfer 18528 Jan 31 18:03 akademik_bilisin.kpr
-rwxr-xr-x 1 cayfer cayfer 30704 Sep 8 2001 anavis
-rw-rw-r-- 1 cayfer cayfer 663451 Sep 13 18:50 anti-span
drwxr-xr-x 5 cayfer cayfer 4096 Feb 9 10:11 attachments
drwxr-xr-x 3 cayfer cayfer 4096 Feb 9 10:11 backgrounds
-rw-r--r-- 1 cayfer cayfer 736 Jul 8 2002 bashrc.txt
-rw-r--r-- 1 cayfer cayfer 18779 Dec 31 07:53 blackbox-menu
-rw-r--r-- 1 cayfer cayfer 1134 Sep 8 2001 bmp
-rw-r--r-- 1 cayfer cayfer 791552 Sep 8 2001 cgi.doc
drwxr-xr-x 2 cayfer cayfer 4096 Feb 9 10:11 ctp208
-rw-r--r-- 1 cayfer cayfer 444386 Jul 29 2002 cua.JPG
drwxrwxr-x 3 cayfer cayfer 4096 Feb 9 00:06 Desktop
-rw-r--r-- 1 cayfer cayfer 5432 Feb 6 2002 dorn-mail
-rw----- 1 cayfer cayfer 1757 Jan 17 07:17 Drafts
drwxr-xr-x 5 cayfer cayfer 4096 Feb 9 10:11 evolution
-rw-r--r-- 1 cayfer cayfer 768941 Dec 31 2001 foto1.jpg
-rwxr-xr-x 1 cayfer cayfer 197 Sep 8 2001 hyperhelp
```

ls -l (*long list*)

Bu ayrıntılı liste, inanamayacağınız kadar çok bilgi içermektedir. Bu aşamada bütün detaylara girmeyeceğiz; sadece satırlardan birkaçını örnek olarak ele alıp, bir fikir verecek şekilde kısaca açıklayacağız.

Yukarıdaki ekran görüntüsünde “d” ile başlayan satırlar **dizinlere**; “-” ile başlayan satırlar ise dosyalara ilişkin bilgi satırlarıdır.

Her satırdaki “**rwxr-xr-x**” benzeri kalıplarda gördüğünüz kodlar, kullanıcıların dosya (ya da izin) üzerindeki erişim yetkilerini tanımlamaktadır. “LINUX’ta Erişim Yetkileri” başlıklı bölümde bu “**rwxr-x---**” gibi kodların ne anlama geldiği daha ayrıntılı bir şekilde açıklanacaktır. Şimdilik kısaca açıklayalım:

Dosyalara Ait "ls -l" Satırları	
-rw-r--r-- 1 cayfer cayfer 736 Jul 8 2002 bashrc.txt	
-rw-r--r--	Bu satırın bir dosyayla ilgili olduğunu (en baştaki “-” işareti- retinden anlıyoruz); bu dosyanın sahibinin bu dosyada okuma (r : <i>read</i>), yazma (w : <i>write</i>) yetkilerinin olduğunu; diğer kullanıcıların sadece okuma yetkilerinin bulunduğunu belirtiyor.
1	Bu satırın basit bir dosya hakkında bilgi içerdiğini belir- tiyor.
cayfer	Bu dosyanın sahibinin “ cayfer ” isimli kullanıcı olduğunu belirtiyor.
cayfer	Bu dosyanın sahibinin “ cayfer ” grubunda olduğunu be- lirtiyor.
736	Dosyanın uzunluğunun 736 byte olduğunu belirtiyor.
Jul 8 2002	Dosyanın en son 8 Temmuz 2002’de değişikliğe uğradığını belirtiyor. Eğer dosya en son içinde bulunduğunuz yılda (2003) değiştirilmiş olsaydı, bilgi satırında yıl yerine saat gö- rünecekti. (addrbook.tar.gz dosyasında olduğu gibi.)
bashrc.txt	Dosyanın adının bashrc.txt olduğunu gösteriyor.

Dizinler için **ls -l** satırının görünümü ve yorumlanması biraz farklıdır:

Dizinlere Ait "ls -l" Satırı	
drwxr-xr-x 2 cayfer cayfer 4096 Feb 9 10:11 ctp208	
drwxr-xr-x	Bu satırın bir dizinle ilgili olduğunu (en başta “ d ” har- fi var); bu dizinin sahibinin bu dizinde okuma (r : <i>re- ad</i>), yazma (w : <i>write</i>) ve çalıştırma (x : <i>execute</i>) yetkile- rinin olduğunu; diğer kullanıcıların sadece okuma ve çalıştırma (dizin için “çalıştırma yetkisi”nin özel bir an- lamı vardır) yetkilerinin bulunduğunu belirtiyor.

2	Bu satırın bir dizin hakkında bilgi içerdiğini ve bu dizinin altında başka bir dizin olmadığını belirtiyor. ("link sayısı" adı verilen bu parametre dizinler için kendi altındaki alt dizinlerin 2 fazlasını gösterir; daha doğrusu bu sayının içinde dizinin kendisi ve bir üstteki dizin de vardır.) Eğer bu " ctp208 " dizinin altında üç tane alt dizin olsaydı, bu sayı 5 olurdu.
cayfer	Bu dizinin sahibinin " cayfer " isimli kullanıcı olduğunu belirtiyor.
cayfer	Bu dizinin sahibinin " cayfer " grubundan bir kullanıcı olduğunu belirtiyor.
4096	Bu sayının dizinler için pek işe yarayan bir anlamı yoktur. Dosya sisteminden sistemine farklılıklar gösterir. Dikkate almayabilirsiniz. Meraklısı için şunu söyleyebiliriz: Dizindeki dosyaların listesini, bu dosyaların diskin hangi bloklarında yer aldığını gösteren bilgileri saklamak için kullanılan disk alanı miktarını byte olarak gösterir.
Feb 9 10:11	Dizinin en son 9 Şubat (içinde bulunduğunuz yılın 9 Şubat'ı), saat 10:11'de değişikliğe uğradığını, örneğin bu saatte bir dosya/alt-dizin eklendiğini, ya da dosya/alt-dizin silindiğini belirtiyor. Bir dizinin altındaki bir dosya da yapılan değişiklikler dizinin son değişiklik saat ve tarihini değiştirmez. Bu değişiklik için dosya/alt dizin silinmesi ya da eklenmesi gerekir.
ctp208	Dizinin adının ctp208 olduğunu göstermektedir.

ls komutunun diğer yeteneklerini; daha doğrusu seçeneklerini öğrenmek istiyorsanız

man ls

komutunu kullanabilirsiniz. Bu komutu verdiğinizde, aşağıda bir kısmı gösterilen uzun açıklamalar ekranınıza listelenecektir. Bu açıklamalar, kullandığınız LINUX işletim sistemine ait kullanım kılavuzunun **ls** komutu ile ilgi-

li bölümleridir. **man** komutunu verdiğinizde, listelenecek satırlar bir ekran sayfasından fazlaysa, birinci sayfanın listelenmesi tamamlandığında, ekranın sol alt tarafında,

--- more ---

işareti göreceksiniz. Bu mesaj, listelenen açıklamaların devamı olduğunu, bu sayfayı okumayı tamamlayınca klavyeden bir komut vererek listenin devamını görmeyi mümkün olduğunu belirtmektedir.

--- more --- görüldüğünde

Boşluk tuşuna (<i>Space Bar</i>)	basarsanız, bir sonraki sayfa,
Return tuşuna	basarsanız, bir sonraki satır,
b (küçük b) tuşuna	basarsanız, bir önceki sayfa listelenir.

Şimdi, **ls** komutunun ayrıntılarını öğrenmek için “**man ls**” komutunu bir deneyiniz.

```

cayfer@notebook.lojman.bilkent.edu.tr: /home/cayfer - Shell - Konsol <2>
LS(1)                                User Commands
LS(1)
NAME
    ls - list directory contents

SYNOPSIS
    ls [OPTION]... [FILE]...

DESCRIPTION
    List information about the FILES (the current directory by default).
    Sort entries alphabetically if none of -cftuSIX nor --sort.
    Mandatory arguments to long options are mandatory for short options too.
    -a, --all
        do not hide entries starting with .

lines 1-20
    
```

ls komutunun man sayfalarına bakarak birkaç yararlı seçenek bulunabilir.

Örneğin:

ls -al Tüm dosya/dizinleri ve varsa gizli dosyaları da (isimleri noktayla başlayan dosyalar ve dizinler) listeler.

- ls -hl** Dosyaların uzunluklarını daha okunaklı bir biçimde (Kbyte ve MByte olarak) listeler. (**h**: *human readable*)
- ls -Rl** Dosyaları listelerken varsa alt dizinlerin içeriklerini de listeler.
- ls -tl** Dosyaların ayrıntılı listesini üretirken dosya ve dizinleri son güncellenme tarihlerine göre sıralar. Son çalıştırdığınız komutun hangi dosyaları değiştirdiğini bulmak için çok yararlı bir özellik...



man

UNIX dünyasının önemli geleneklerinden biri de çevrim içi komut dökümantasyonudur.

“man sayfaları” diye anılan (İng.: Manual kısaltması) dosyalarda tüm UNIX komutlarının nasıl kullanılacaklarını, olası parametrelerini, seçeneklerini ve örneklerini bulabilirsiniz.

Örneğin “**fsck**” komutunun nasıl kullanılacağını hatırlayamadığınızda “**man fsck**” komutunu verip, **fsck** komutu hakkında çok ayrıntılı ve gerçekten işe yarayan açıklamalar alabilirsiniz. Hatta man komutunun ayrıntıları için “**man man**” komutunu bile kullanabilirsiniz.

Dosyalar ve Dizinler

Kullanıcıların bir bilgisayarla yaptıkları çalışmaların meyveleri üretilen dosyalardır. Çizim programları kullanarak hazırladığınız çizimleri bir sonraki çalışma adımı için dosyalarda (disk, disket veya teypte) saklamalısınız. Yazdığınız programları da aynı şekilde... Ya da sistemin davranışını düzenlemek üzere değiştirdiğiniz konfigürasyon dosyalarını...

Kendinize ait kişisel dosyalarınızı veya sistem denetim dosyalarını düzenlemek, kopyalarını çıkarmak, yedeklerini almak, günlük işlerinizin önemli bir parçası olacaktır; bu nedenle LINUX'un dosya kavramını uzun uzun anlatmak istiyoruz.

LINUX'ta dosyalar, aynı Windows'da olduğu gibi, ilgili oldukları uygulamaya göre düzenlenmiş ve isimlendirilmiş dizinlerde (*directory*) saklanır. LINUX'un çok kullanıcı bir işletim sistemi olmasından dolayı, diskin veya disklerin kullanıcılar arasında paylaşılması, dosyaların konuları yanı sıra,

kullanıcıların kimliklerine göre de gruplanmasını gerektirir. Hatırlarsanız, her kullanıcının bir “kişisel dizini” ya da “kullanıcı dizini” olduğundan daha önce söz etmiştik. (*home directory*)

Dosya ve dizinlerin kullanıcılar arasında paylaşılmasından dolayı, bir kullanıcıya ait dosyaların ve dizinlerin bir “erişim yetkisi” mekanizması ile diğer kullanıcılara karşı korunması gerekmektedir. LINUX'ta bu koruma mekanizması, kullanıcıların sisteme tanıtılması sırasında verilen “kullanıcı ismi”, “kullanıcı numarası” ve kullanıcının ait olduğu “çalışma grubunun numarası”na dayandırılır. Normal şartlar altında, kullanıcıların kendilerine ait “kullanıcı numarası”nı ve “çalışma grubu numarası”nı bilmelerine gerek yoktur. Kullanıcı ismi ve kullanıcı numarası arasındaki bağlantı, işletim sistemi tarafından otomatik olarak sağlanır.

Dosya Yaratmak

Dosya yaratmanın birçok yöntemi vardır:

- Grafik kullanıcı arabirimini kullanarak başlatacağınız herhangi bir uygulama programıyla,
- Terminal veya konsoldan başlatacağınız herhangi bir uygulama programı ya da LINUX komutuyla,
- Yazdığınız bir C programını derleyerek,
- Eski bir dosyanın kopyasını çıkararak,
- CD'den veya teypten diske dosya kopyalayarak,
- Program çıktılarını yönlendirerek (I/O Redirection) (Sabırlı olun! Ne anlama geldiğini anlatacağız) dosya yaratabilirsiniz.

Biz en basitinden başlayalım...

cat Komutu

UNIX'te çok sık kullanılan, çok işlevli bir komuttur. Bu işlevlerden biri düz yazı içeren dosyaların içeriğini görüntülemektir.

En basit kullanım formu:

cat dosya_adı (*catenate*)

şeklinde. Bu formda kullanıldığı zaman, **dosya_adı** adlı dosyayı ekrana, daha doğrusu, LINUX diliyle “standart çıktı birimi”ne (*Standard Output*)

gönderir. Standart çıktı birimi genellikle ekran olduğu için, **cat** komutu bir dosyayı ekrana listelemek için kullanılır.

Denemek için,

```
cat /etc/issue
```

komutunu verebilirsiniz. Eee... Hani dosya yaratıyorduk?

cat komutu dosya yaratmak için kullanıldığında verilmesi gereken komutun formu ise biraz daha farklıdır...

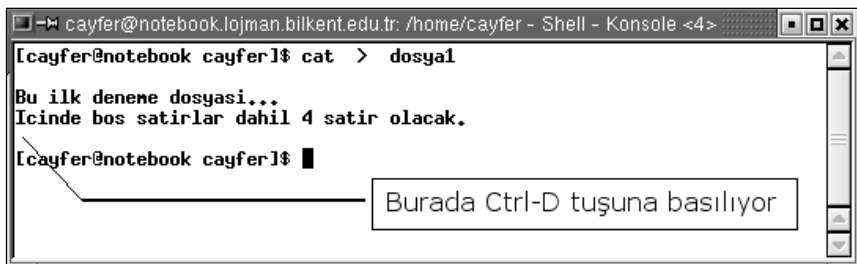
```
cat > yenisdosya
```

Bu formda kullanıldığında, **cat** komutu, standart girdi birimi'nden (klavyeden, *Standard Input*) aldığı bilgileri, **yenisdosya** isimli bir dosyaya yönlendirecektir. (Bir başka deyişle kopyalayacaktır.)

Şimdi isterseniz **dosya1** isimli ilk küçük dosyanızı yaratmak üzere

```
cat > dosya1
```

komutunu veriniz, daha sonra **imleç** (*cursor*) yeni satırın başına geldiğinde, dosyanın içinde yer almasını istediğiniz satırları giriniz. Örneğin,



Girdiğiniz satırlar tamamlanınca, imleç satır başındayken Ctrl ve D tuşlarına birlikte basarak (EOF: “End of File” karakteri) standart giriş biriminizde “dosyanın sonuna” geldiğinizi belirtin.

Dosya adı verirken dizin adı belirtmediğiniz için, **dosya1** adlı dosya çalışma dizininizde yaratılacaktır. Herhangi bir hata mesajı almadıysanız, dosya problemsiz yaratıldı demektir. Eğer **dosya1**'in yaratılıp yaratılmadığını kontrol etmek isterseniz iki yöntem önerebiliriz:

Birinci yöntem, **cat** komutuyla **dosya1** dosyasındaki satırları ekrana görüntülemek olabilir. İkinci yöntemse, “**ls -l**” komutuyla dosyanın adını, uzunluğunu, ne zaman yaratıldığını ve sahibinin kim olduğunu görebilirsiniz.

İki yöntemi de deneyiniz.

Eğer **cat** komutunu parametresiz olarak vererseniz, komut pek de anlamlı olmayan bir iş yapmaya, standart giriş biriminden okuyup standart çıkış birimine kopyalamaya başlayacaktır. Yani, klavyeden (standart giriş birimi) bastığınız her tuş, standart çıkış birimine (ekran) kopyalanacaktır.



Yanlışlıkla düşebileceğiniz bu durumdan kurtulmak için, imleç satır başındayken Ctrl-D tuşuna basınız. Bu hareketiniz kopyalama işini sona erdirecektir. Bu işlem, diskteki dosyalarınızı hiçbir şekilde etkilemez.

Küçük bir varyasyonla **cat** komutunu ilginç bir iş yapmak için de kullanabilirsiniz:

```

cayfer@notebook.lojman.bilkent.edu.tr: /home/cayfer - Shell - Konsol <4>
[cayfer@notebook cayfer]$ cat >> dosya1
BU SATIR dosya1'IN SONUNA EKLENECEK
[cayfer@notebook cayfer]$ cat dosya1
Bu ilk deneme dosyası...
İçinde boş satırlar dahil 4 satır olacak.
BU SATIR dosya1'IN SONUNA EKLENECEK
[cayfer@notebook cayfer]$

```

Burada Ctrl-D tuşuna basılıyor

cat komutunu bir de şu şekilde deneyiniz:

```

cayfer@notebook.lojman.bilkent.edu.tr: /home/cayfer - Shell - Konsol <4>
[cayfer@notebook cayfer]$ cat > /dosya1
bash: /dosya1: Permission denied
[cayfer@notebook cayfer]$

```

Olmadı değil mi?. (**Permission denied**: Bu işi yapmaya yetkiniz yok!)

Nedeni açık... **dosya1** isimli dosyayı root (/) dizininin hemen altında yarat-

Kim Korkar LINUX'tan?

maya çalıştınız ve sizin bu dizine kayıt yapmaya yetkiniz olmaması da çok doğal.

Bu komutu root kullanıcı olarak vermiş olsaydınız böyle bir mesajla karşılaşmayacaktınız ve dosyanız yaratılacaktı.

cp Komutu (copy)

Bu komutun ne işe yaradığını söylemeye gerek olduğunu sanmıyoruz; ama nasıl kullanıldığı önemli...

En basit formuyla,

cp dosya1 dosya2 (*copy*)

dosya1 isimli dosyayı **dosya2** isimli dosyaya kopyalayacaktır.



Eğer **dosya2** isimli dosya yoksa, yaratılacaktır. (Tabii bu dosyanın yer alacağı dizinde dosya yaratmaya yetkiniz varsa...) Eğer bu isimde bir dosya eskiden varsa, üzerine kopyalama yapılacak ve eski içeriği kaybolacaktır. Böyle bir durumda, eski bir dosyanın üzerine kayıt yapmak üzere olduğunuz konusunda uyarılmayabilirsiniz! Dikkatli olmanız gerekir.

Eğer dikkatinize güvenmiyorsanız, **cp** komutunu,

cp -i dosya1 dosya2

formunda kullanın. **-i** parametresi (*interactive*), eski bir dosyanın üzerine kayıt yapılması durumunda kullanıcının **Overwrite?** mesajı ile uyarılmasını ve ancak **y** yanıtı verilirse devam edilmesini sağlar.



-i parametresini kullanmayı unutmaktan korkuyorsanız, kitabın **alias** komutu ile ilgili bölümünü okuyunuz. Bu bölümde çeşitli UNIX komutlarını kalıcı olarak değiştirmenin, hatta kendinize özgü UNIX komutları yaratmanın yollarını bulacaksınız.

Bir başka form:

cp dosya_adı dizin_adı

dosya_adi isimli dosyayı, **dizin_adi** isimli dizinin altına kopyalar. İsterse-
niz **-i** seçeneğini gene kullanabilirsiniz.

Bu formla ilk form arasında görünüş olarak hiçbir fark yoktur. İkinci para-
metreyle verilen isim bir dizine aitse, verdiğiniz komut ikinci formda kabul
edilir ve birinci dosya bu dizinin altına kopyalanır. İkinci parametreyle be-
lirtilen isimde bir dosya varsa, ya da bu isimde hiçbir şey (dosya veya dizin)
yoksa; komut ilk formda kabul edilerek ilk parametredeki dosyanın kopya-
sı çıkarılır.

Bir başka form:

cp dosya1 dosya2 dosya3 dizin_adi

dosya1, **dosya2** ve **dosya3** isimli dosyaları, **dizin_adi** isimli dizinin altına
kopyalar. İsterseniz **-i** parametresini kullanabilirsiniz.

UNIX **cp** komutunda, kopyalamanın nereden nereye yapılacağını mutlaka
açıkça belirtmelisiniz. Yani,

cp /dizin1/dosya1

şeklinde bir komut kullanamazsınız. Bu şekilde yazacağınız bir komut, **/di-
zin1**'in altındaki **dosya1** isimli dosyayı, çalışma dizinine kopyala anlamına
gelmez; hatalı daha doğrusu eksik bir komuttur.



Dizin Kopyalamak

LINUX'ta dizin kopyalamak için gene **cp** komutu kullanılır; ancak özel bir
parametreyle birlikte...

Dizin kopyalamak için kullanılan form:

cp -r dizin1 dizin2

şeklindedir. Bu formda verilen kopyalama komutu; varsa, dizinlerin alt di-
zinlerinin de kopyalanmasını sağlar. (**r**: *recursive*; *özyinelemeli*)

cp komutu üzerine birkaç ileri düzey örnek vererek bu konuyu geçmek isti-
yoruz:

Dosya ve Dizin Kopyalama Örnekleri	
cp /etc/issue /tmp/issue2	/etc isimli dizinin altındaki issue isimli dosyayı, /tmp isimli dizinin altına, adını issue2 olarak değiştirerek kopyalar.
cp /etc/issue /tmp	/etc isimli dizinin altındaki issue isimli dosyayı, /tmp isimli dizinin altına, adını değiştirmeden kopyalar.
cp /etc/issue .	/etc isimli dizinin altındaki issue isimli dosyayı çalışma dizinine (bir başka deyişle, “buraya”) kopyalar.
cp /home/cayfer/prg1 ..	/home isimli dizinin altındaki cayfer dizinin altındaki prg1 isimli dosyayı çalışma dizininin bir üstündeki dizine kopyalar.
cp -r /home/cayfer/dizin1 ..	/home isimli dizinin altındaki cayfer dizinin altındaki dizin1 isimli dizini çalışma dizininin bir üstündeki dizine kopyalar.

rcp /etc/issue gunes:/home/cayfer (**rcp**: *remote copy*)

Çarpıcı bir örnek: **/etc** dizindeki **issue** isimli dosyayı “**gunes**” isimli bir başka bilgisayardaki **/home** dizinin altındaki **cayfer** dizinine kopyalar.

Bu komutu verebilmeniz için komutun verildiği bilgisayarla “**gunes**” isimli bilgisayarların bir bilgisayar ağı ile birbirlerine bağlı olmaları ve sizin ikisine de erişim hakkınız olması gerekmektedir. Bu ve buna benzer komutlara daha sonra ayrıntılı olarak değineceğiz.

Dizin Yaratmak

UNIX'te dizin yaratmak için **mkdir** komutu kullanılır.

Formu basittir:

```
mkdir dizin  
mkdir eski_dizin/yeni_dizin (make directory)
```

gibi...

Doğal olarak, yalnızca yetkiniz olan yerlerde dizin yaratabilirsiniz.

Çalışma Dizinini Değiştirmek

LINUX komutlarında bir dosyadan söz ederken; örneğin **cat** komutuyla bir dosyanın içine bakmak üzere dosyanın adını verirken, dosyanın yer aldığı dizini tam olarak belirtmezseniz, dosyanın o andaki çalışma dizininizde (*working directory, current directory*) bulunduğu varsayılır. Çalışma dizinini değiştirmek için kullanılan komut,

```
cd yeni_çalışma_dizini (change directory)
```

Örnekler vermek gerekirse:

Çalışma Dizini Değiştirme Komutu (cd) Örnekleri	
cd /home/ayfer/proje1	Pek açıklama gerektirmiyor...
cd ../proje2	Bir üstteki dizinin altındaki proje2 isimli dizine geçer.
cd ../../mektuplar	İki üst düzeydeki dizinin altındaki mektuplar isimli dizine geçer.
cd ~omer	Kullanıcı adı “omer” olan kullanıcının kişisel dizinine geçer (omer’in home dizini).
cd	Her nerede olursa olsun, komutu veren kullanıcıya ait dizine geçer. (Yuvaya dönüş!)



Dizinler arasında gidip gelirken, zaman zaman kaybolmanız doğaldır. Özellikle hazır işaretniz (prompt) çalışma dizininiz hakkında bilgi vermiyorsa...

Kaybolduğunuzda, **pwd** komutu ile (print working directory) o andaki çalışma dizininizin hangi dizin olduğunu öğrenebilirsiniz.

Dosya Silmek

Artık diskte yer almasını istemediğiniz dosyaları silmek için kullanacağınız komut,

```
rm dosyarm dosya1 dosya2 ... dosyaN (remove)
```

formlarındadır.

Bir seferde (tek komutta), farklı dizinlerde yer alan dosyaları da silebilirsiniz.

```
rm /dizin1/dosya1 /baska_dizin/dosya2 ...
```

Eğer dosyalar silinmeden önce onaylamak istiyorsanız **-i** parametresini kullanabilirsiniz:

```
rm -i /dizin1/dosya1 /baska_dizin/dosya2 ...
```

Bu komut formunu kullandığınızda, silinecek her dosya için teker teker,

```
rm: remove `/dizin1/dosya1'?
```

sorusu sorulacak ve sadece “**y**” yanıtını verdiğiniz dosyalar silinecektir.

Dizin Silmek

Artık diskte yer almasını istemediğiniz dizinleri, altlarındaki dosya ve alt dizinleriyle birlikte silmek için kullanacağınız komut,

```
rm -r dizin  
rm -r dizin1 dizin2 ... dizinN (remove)
```

formlarındadır. (**-r**: *recursive*; özyinelemeli)

Bir seferde, farklı dizinlerde yer alan dizinleri de silebilirsiniz.

```
rm -r /dizin1/alt_dizin1 /baska_dizin/dizin2 ...
```

Eğer dizinler silinmeden önce onaylamak istiyorsanız **-i** parametresini kullanabilirsiniz:

```
rm -ir /dizin1/alt_dizin1 /baska_dizin/dizin2 ...
```

Bu komut formunu kullandığınızda, silinecek her dizin ve altlarındaki dosyalar/dizinler için teker teker,

```
rm: remove '/dizin1/alt_dizin1'?
```

gibi sorular sorulacak ve sadece “**y**” yanıtını verdiğiniz dizinler silinecektir.

LINUX işletim sisteminde “undelete” (silinen dosyaları geri getirme) görevini yerine getirecek bir program ya da komut yoktur. Sildiğiniz dosya ve dizinler, bir daha geri getirilemeyecek şekilde silinir. Bu nedenle **rm** komutunu kullanmadan önce iyi düşünmelisiniz.



Dosya / Dizin Adı Değiştirmek

```
mv eski_dosya_ismi yeni_dosya_ismi  
mv eski_dizin_ismi yeni_dizin_ismi (move)
```

Bir isim değişikliği yapmak istediğinizde, doğal olarak, söz konusu dosya veya dizinin yer aldığı dizinde, yeni isimde bir dosya ya da dizin bulunmamalıdır.

Dosya / Dizin Yeri Değiştirmek

Bu iş için kullanacağınız komut gene **mv (move)** komutudur.

```
mv eskiyeri/dosya yeniyeri/dosya (move)
```

Bir yer değişikliği yapmak istediğinizde, doğal olarak, söz konusu dosya veya dizinin yer alacağı yeni dizinde, aynı isimde bir dosya ya da dizin bulunmamalıdır.

Buraya kadar temel birkaç LINUX komutundan, sık sık da yetkilerden söz ettik. Artık şu yetki meselesini biraz daha açmanın zamanı geldi...

LINUX'ta Erişim Yetkileri

LINUX işletim sistemi, kendisini ve denetlediği kaynakları, acemi veya kötü niyetli kullanıcılara karşı korumak zorundadır. Öte yandan, kullanıcıların dosya ve dizinlerini de birbirlerine karşı korumak gerekmektedir. Bir üniversitenin bilgisayarındaki öğrenci işleri müdürlüğünün kayıtlarına herkesin erişebildiğini hayal edebiliyor musunuz? Hele siz öğrenciyken...

LINUX işletim sistemi, oldukça kuvvetli bir güvenlik sistemine sahiptir ve bu güvenlik sisteminin temelinde, kullanıcıların sisteme tanıtımı sırasında yapılan düzenlemeler yatar. Sistemin yönetiminden sorumlu olan kişi(ler), genellikle kullanıcıları sistemi kullanım amaçlarına göre sınıflandırır(lar). Örneğin; öğrenci işleri, kütüphane, satın alma, mühendislik fakültesi, edebiyat fakültesi gibi... Bu sınıflara “**kullanıcı grupları**” (*user group*) adı verilir ve her kullanıcı grubunun bir numarası olur.

Sonra, sıra her bir kullanıcı için bir isim ve kullanıcı numarası vermeye ve bu kullanıcıların ait oldukları grupları belirlemeye gelir.

Özetlemek gerekirse, LINUX işletim sistemi ile çalışan bir bilgisayar kullanmak için herkesin bir kullanıcı hesabına ve dolayısıyla bir kullanıcı numarasına sahip olması gerekir. Kullanıcılarınıza sistemde daha önce kullanılmamış ve onun kimliğini hatırlatan bir kullanıcı ismi seçmek ve kullanıcının görev tanımına uygun bir gruba yerleştirmek size düşecektir. Kullanıcının kişisel dizininin yerine de karar verdikten sonra artık kullanıcının hesabını açabilirsiniz. Şimdi dönelim dosya ve dizinlerin erişim haklarına...

Yazma, Okuma, Program Çalıştırma Yetkileri

Hatırlarsanız, “**ls -l**” komutu ile bir dizinde yer alan dosyaların (ve dizinlerin) ayrıntılı listesini aldığınızda,

```
-rwxr-xr-x 1 cayfer cayfer 30704 Sep 8 2001 amavis
```

benzeri satırlar görmekteydiniz.

Dosya ve dizinlere erişim yetki tanımları **rwxr-xr-x**'e benzeyen kod dizileriyle gösterilir. Bu dokuz karakterden oluşan dizi aslında üçer karakterlik üç parçadan oluşmaktadır. (Bu örnekte “**rwx**”, “**r-x**” ve “**r-x**”)

İlk üç karakter dosyanın sahibinin yetkilerini, ikinci üçlü, dosyanın sahibiy-le aynı kullanıcı grubunda yer alan kullanıcıların yetkilerini, son üçlü ise di-ğer kullanıcıların bu dosya üzerindeki yetkilerini tanımlamaktadır.

r w x	r w x	r w x
Sahibinin Yetkileri	Aynı Gruptakilerin Yetkileri	Diğer Kullanıcıların Yetkileri

Her üçlü de aynı kalıptadır. Her üçlünün ilk pozisyonunda bir **r** harfinin varlığı, ilgili kullanıcının dosyayı okuma yetkisinin bulunduğunu gösterir. Bu pozisyonda bir eksi işareti varsa, söz konusu kullanıcı tipi için okuma yetkisi olmadığı anlaşılır.

Bu mantıkla,

r: Okuma yetkisi (*read access*),

w: Yazma yetkisi (*write access*),

x: Dosya bir program dosyasıysa, programı çalıştırma yetkisini gösterir (*execute access*).

Birkaç örnek konuya iyice açıklık getirecektir:

Dosya Erişim Yetki Kodları Örnekleri	
Dosya Yetki Kodu	Anlamı
rwXrwxrwx	Bu dosyayı herkes okuyabilir, herkes bu dosyaya kayıt yapabilir, dosyanın adını değiştirebilir; hat-ta dosyayı silebilir. Eğer bu bir program dosyasıysa, herkes bu programı çalıştırabilir.
rwXr-Xr-X	Bu dosyayı herkes okuyabilir ve program dosya-sıysa çalıştırabilir; ancak sadece sahibi bu dosya-da bir değişiklik yapabilir.
rwX-----	Bu dosya üzerinde sahibi istediği tüm işlemleri yapabilir; ancak dosya, diğer kullanıcılara tama-men kapalıdır.

rw-r--r--	Bu dosya bir program dosyası değil, çünkü hiç kimsenin çalıştırma (<i>execute</i>) yetkisi yok! Sahibi dosyayı okuyup yazabilir ancak diğer kullanıcılar sadece okuyabilir.
rw-rw----	Bu dosya da bir program dosyası değildir. Dosyanın sahibi ve kendisiyle aynı grupta olan kullanıcıların okuyup yazma yetkileri vardır, ancak diğer kullanıcıların hiçbir şekilde erişimleri mümkün değildir.
rwX--X--X	Sahibi dışında kalan kullanıcılar, bu program dosyasını sadece çalıştırabilirler.

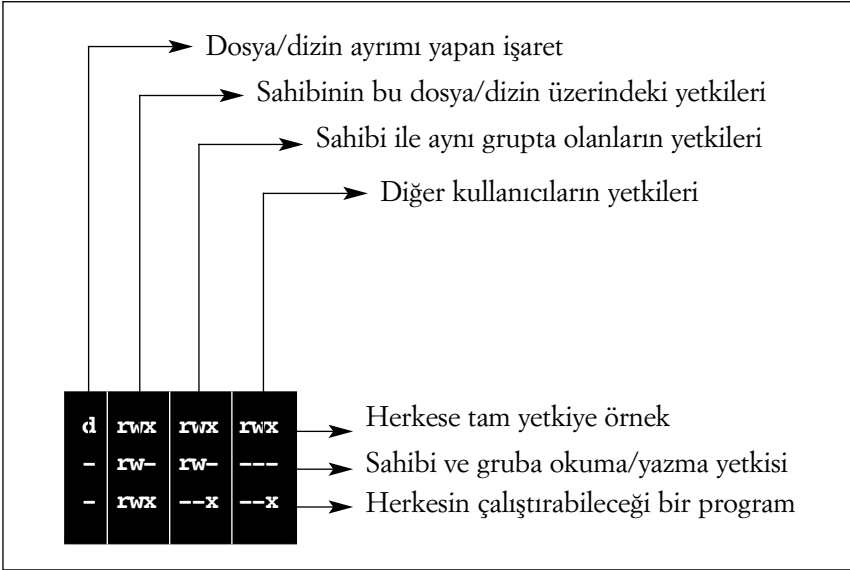


Dizinler için de “**rwXr-Xr--**” yetki kodları söz konusudur. Dosya yetki kodlarına çok benzemekle beraber, detaylarda bazı önemli farklılıklar vardır. Bu farkları daha sonra açıklayacağız.

Diskinizdeki dosya ve dizinlerin bazılarının yetki kodlarında “**r**”, “**w**” ve “**x**” harflerinden farklı olarak “**s**”, “**S**” ve “**t**” gibi kodlar da görebilirsiniz. Şimdilik bunlara pek aldırmayın.

Bir kez daha özetlemek gerekirse:

“**ls -l**” komutu verdiğinizde alacağınız **dosya-dizin** listesinde göreceğiniz yetki kalıpları yan sayfadaki şemaya göre yorumlanmalıdır:



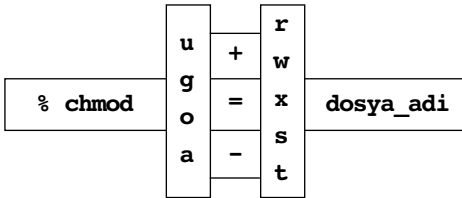
Doğal olarak dosya ve dizinler üzerindeki yetkileri değiştirmek mümkündür; ancak erişim yetkilerini değiştirmeye yetkili olmanız gerekmektedir. Bu yetki sadece dosyanın veya dizinin sahibi ile “**root**” kullanıcısında vardır.

Dosya ve dizinlerin erişim yetkilerini değiştirmek için,

chmod (*change mode*)

komutu kullanılır.

Bu komut iki değişik formda kullanılabilir. Kullanımı göreceli olarak kolay olan formu:



veya UNIX notasyonunda yazıldığında

chmod [ugoa][+--][rwxst] dosya_adi

formudur.

Köşeli parantezler, “aralarındaki karakterlerden biri seçilmelidir” diye okunur.

Bu formda,

- u:** Dosya/dizinin sahibi (*user*)
- g:** Dosya/dizinin sahibiyle aynı grupta olanlar (*group*)
- o:** Diğer kullanıcılar (*others*)
- a:** Herkes (*all*)

- +:** Yetki ekleme
- =:** Yetki eşitleme
- :** Yetki çıkarma

- r:** Okuma yetkisi (*read*)
- w:** Yazma yetkisi (*write*)
- x:** Çalıştırma yetkisi (*execute*)
- s:** suid biti (daha sonra anlatacağız)
- t:** sticky bit (daha sonra anlatacağız)

Birkaç örnek vermek gerekirse:

chmod a+x adres komutu, **adres** isimli program dosyasına herkes için çalıştırma yetkisi verir.

chmod o-w mhsb.z mhsb.z dosyasından, diğer kullanıcıların yazma yetkisini kaldırır.

chmod go=rx adres adres isimli dosyanın grup ve diğerleri için erişim yetkisini **r-x** kalıbına eşitler.

chmod komutunun bir diğer formu da (ki LINUX ustaları tarafından genellikle tercih edilen formdur), yetkilerin sayısal olarak gösterildiği formdur. Yetki tanım grupları aşağıdaki tabloya göre sayısal birer değerle eşleştirilir:

4	2	1
r	w	x

4	2	1
r	w	x

4	2	1
r	w	x

| Sahibi (Owner) | Aynı Gruptakiler (Group) | Diğerleri (Others) |

Diyelim ki **adresler** dosyasının erişim yetkilerinin **rwxr-xr-x** olmasını istiyorsunuz. Bu yetki kalıbını üçer üçer ayrılmış olarak düşünüp (**rw** **r-x** **r-x**), yukarıdaki tabloya göre verilmek istenen yetkilere karşılık gelen sayıları üçlü gruplar halinde toplayınız ve elde edeceğiniz üç tane sayıyı yan yana getirip üç haneli bir sayı elde ediniz. Bir başka deyişle:

4	2	1	4		1	4		1
r	w	x	r	-	x	r	-	x
7			5			5		
755								

chmod komutunda bu şekilde elde edilen sayıyı kullanarak dosya ya da dizinlerinizin erişim yetkilerini tanımlayabilirsiniz;

chmod 755 adresler

Bir **chmod** komutu ile birden fazla dosyanın erişim yetkilerini aynı anda değiştirebilirsiniz:

chmod 755 *.pl *cgi

Bir dizindeki tüm alt dizin ve onların altındaki tüm dosyaların erişim yetkilerini tek harekette değiştirmek istediğinizde

chmod -R 755 dizin1

komutundaki gibi “**-R**” parametresini kullanabilirsiniz.

Dizinlere Erişim Hakları

Dosyalar için **r**, **w** ve **x** yetkileri yeteri kadar açık olmasına karşın dizinler için bu yetkilerin anlamları biraz daha karmaşıktır. Şöyle ki:

Bir dizin için **r** (*read*) yetkiniz varsa: O dizindeki dosyaların isimlerini **ls** komutu ile görebilirsiniz. [Eğer **x** (*execute*) yetkiniz yoksa, bir dizin için **ls** komutunu kullanabilmenize karşın, “**ls -l**” komutunu kullanamazsınız].

Bir dizin için **w** (*write*) yetkiniz varsa:

O dizindeki dosyaların yerleşiminde değişiklikler yapabilirsiniz. Örneğin, dosyaların adını değiştirebilirsiniz veya dosyaları silebilirsiniz. Eğer bir dizinde **w** yetkiniz varsa, fakat o dizin içindeki bir dosyaya **w** yetkiniz yoksa, o dosyanın içeriğini değiştiremezsiniz, silemezsiniz veya adını değiştiremezsiniz.

Bir dizin için **x** (*execute*) yetkiniz varsa: Çalışma dizinizi bu dizin olarak değiştirebilirsiniz. (**cd** komutunu bu dizin için kullanabilirsiniz). Bir dizini çalışma dizini olarak kullanmak için **r** (*read*) yetkisi yeterli değildir; **x** yetkisi de gerekir. İçinde gizli bilgiler olmayan ama gene de diğer kullanıcılar tarafından kurcalanmasını istemediğiniz dizinler için en uygun yetki düzenlemesi **rwxr-xr-x** (755) olarak kabul edilir.

Eğer bir dizininizi sizden başka kimsenin kullanmasını ve içine bakmasını istemiyorsanız, "**chmod go-rwx dizin_adi**" komutuyla o dizinin erişim yetkilerini "**rwX-----**" (700) olarak ayarlayabilirsiniz.

Bir dizin için **w** (*write*) ve **x** (*execute*) yetkiniz varsa: O dizinde yer alan bir dosyaya yazma yetkiniz olmasa bile o dosyayı silebilir veya adını değiştirebilirsiniz. Evet! Bu birlikte yaşaması zor bir özellik! O yüzden değerli dosyalarınızı içeren dizinler için kendinizden başka kimse için yazma yetkisi vermemelisiniz. Bir başka deyişle bir dizin için verilmiş olabilecek "**rwXrw-rw-**" ya da "**rwX---rwX**" gibi bir yetki kalıbı çok tehlikelidir. Bir nedenle bu tip bir yetki kalıbı kullanmak zorunda kalırsanız o dizinin "**sticky bit**"ini de ayarlamalısınız. Bunun nasıl yapılacağını öğrenmek için bir iki sayfa daha sabretmelisiniz.

umask : Dosya ve Dizinlerin Varsayılan Erişim Yetkileri

Bu bölüm şu aşamada size fazla karışık gelirse hiç düşünmeden atlayabilirsiniz. Aklınızda kalması gereken tek şey, bir LINUX sistemde yeni yaratılan dosya ve dizinler için atanacak varsayılan (*default*) erişim yetkilerinin **umask** değeriyle belirleniyor olduğudur.

umask komutunu parametresiz kullanırsanız, o anda geçerli olan **umask** değerini öğrenirsiniz.

```
[cayfer@notebook cayfer]$ umask
```

022

umask değerini değiştirmek için;

```
[cayfer@notebook cayfer]$ umask 033
```

gibi bir komut kullanmanız gerekir.

umask değerinin yorumlanması biraz gariptir. **umask** değeri, verilen yetkileri değil, kaldırılan yetkileri belirtir.

Bir örnekle anlatmak daha kolay olacak galiba:

Şimdi; 022 sayısını 0 2 2 şeklinde 3 ayrı sayı olarak düşünün ve her sayıyı üçer haneli ikilik sayılara (*binary*) çevirin.

0 2 2: 000 010 010 gibi...

Bu diziyi, “**rw- rw- rw-**” ve “**rxw rxw rxw**” şablon yetki kodları ile alt alta yazın.

umask Değerinden Varsayılan Erişim Yetkilerinin Hesaplanması		
	Dosyalar İçin	Dizinler İçin
Şablon Yetki	rw- rw- rw-	rxw rxw rxw
umask (022)	000 010 010	000 010 010
Atanacak Yetki	rw- r-- r--	rxw r-x r-x

Bu düzenlemede 0'ların altına gelen yetkilere dokunulmamakta, ancak 1'lerin altına gelen yetkiler kaldırılmaktadır.

Yani, **umask** değerinin 022 olarak atanmasından sonra yeni yaratılan bir dosyanın erişim yetkileri “**rw-r--r--**”; bir dizinin erişim yetkileri ise “**rxw-r-xr-x**” olarak belirlenecektir.

Kararsız kullanıcılara önerimiz, umask değerini değiştirmemeleri ve böylece varsayılan **umask** değeri olan 022'yi kullanmalarıdır.

Suid Biti ve suid Programlar

Birkaç sayfa önce, **chmod** komutundan söz ederken **suid bit** kavramından bahsetmiştik. Bir program dosyasının suid bitini aktif hale getirmek (yani “**chmod +s prog**” gibi bir komut vermek), bu **prog** programını çalıştıran kullanıcıların, program çalıştığı sürece ve sadece bu program ile ilgili dosyalar açısından, program dosyasının sahibinin yetkilerine sahip olmalarını sağlar. Biraz karışık oldu ama, aşağıdaki örnek açıklayıcı olacaktır. (Bu aşamada suid programların yararı size bir şey ifade etmiyorsa hiç dert etmeyin).



Şifresini değiştirmek isteyen bir kullanıcı **passwd** komutunu kullanacaktır. Bu program kullanıldığında, şifre değişikliği, sahibi root olan **/etc/shadow** dosyasında bir kayıt değişikliği yapılmasını gerektirecektir. Ancak bu dosya, LINUX'un en önemli dosyalarından birisi olduğu için çok iyi korunmakta ve sahibi (yani root) dışında kimsenin bu dosyaya ne okuma, ne de yazma yetkisi bulunmamaktadır. İşte suid kavramı bu soruna bir çözüm getirmektedir. **passwd** programının yer aldığı **/usr/bin/passwd** dosyasının suid biti set olduğu için, **passwd** komutunu veren kullanıcılar bu program çalıştığı sürece ve **/etc/shadow** dosyasına erişim söz konusu olduğunda geçici olarak root yetkilerine sahip olacaklardır.

suid programlar önemli birer emniyet gediği olabilirler. Bir programa suid özelliği vermeden önce iki kez, hatta üç kez düşününüz. Eğer, suid özelliği vermek istediğiniz program, kullanıcıya bir şekilde UNIX komutu verme olanağı sağlıyorsa, bu programa kesinlikle suid özelliği vermeyiniz.

“suid” özelliğine sahip dosyalar, ayrıntılı **ls** listelerinde bir **s** harfiyle gösterilir.

```
-r-s--x--x 2 root root 512 Feb 24 09:56 passwd* gibi.
```

Bir dosyayı suid dosya yapmak için;

```
chmod u+s dosya_adi
```

komutunu kullanabilirsiniz. Bu komutu kullanabilmek için ya dosyanın sahibi ya da “**root**” olmanız gerekir.

Sticky Bit

Eski UNIX uyarlamalarında, disklerin ortalama erişim sürelerinin ve veri transfer hızlarının düşük olduğu zamanlarda, program dosyalarının disklerden belleğe yüklenebilmeleri için geçen süreler kullanıcıları rahatsız etmekteydi. Bu yüzden, sık sık kullanılan komutları oluşturan programların disk dosyalarına “**sticky**” özelliği verildi. Bu özellik sayesinde, bu tip programlar, bir kez belleğe yüklendikten sonra, programın çalışması sona erdiğinde bile bellekten atılmazlardı; böylece, komutun bir sonraki kullanımı için program bellekte hazır olurdu. “**sticky**” özelliğine sahip dosyalar, ayrıntılı **ls** listelerinde bir **t** harfiyle gösterilir.

```
-rwxr-xr-t 2 root root 512 Feb 24 09:56 ls* gibi.
```

Artık, günümüz UNIX'lerinde **sticky bit** kavramı bu anlamda kullanılmamaktadır.

Eğer “Dizinlere Erişim Yetkileri” bölümünü dikkatli okuduysanız başkaları için yazma ve okuma yetkisi vereceğiniz bir dizinde yer alan dosyaların tehlikede olduğunu farketmiş olmalısınız. LINUX'ta bu tehlikeyi ortadan kaldırmak için “sticky bit” kullanılır. Bir dizinin “sticky bit”ini kullanarak o dizine yazma-okuma yetkileri ne olursa olsun, söz konusu dosya üzerindeki asıl yetkilerin, dosyanın kendi yetki kalıbı tarafından belirlenmesini sağlayabilirsiniz.

Bir dizinin “sticky bit”ini devreye sokmak için;

```
chmod +t dizin_adi
```

komutunu kullanabilirsiniz. Bu komutu kullanabilmek için ya dizinin sahibi ya da “**root**” olmanız gerekir.

Yukardaki paragraf pek fazla bir şey ifade etmediyse hiç dert etmeyin! Sticky Bit kavramı bilgisayarınızı LINUX altında çalıştırmanız için mutlaka bilmeniz gereken kavramlardan biri değildir.



Dosyaların ve Dizinlerin Sahibini Değiştirmek

chown (*change owner*)

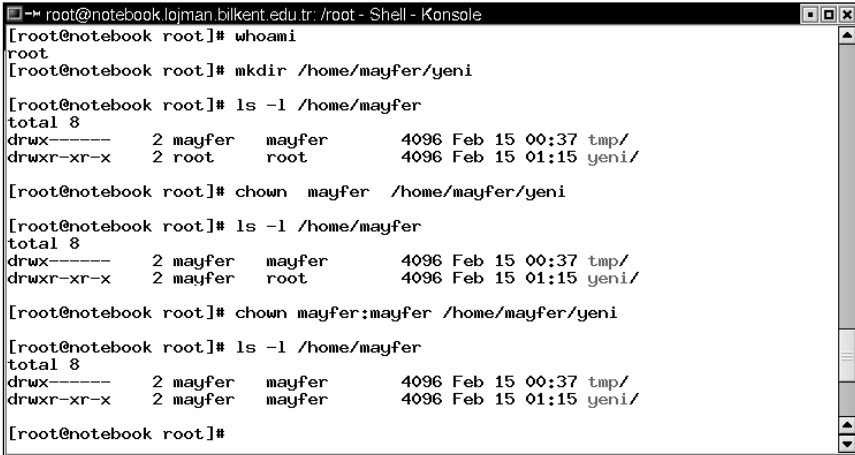
Bu komutu sadece “root” kullanabilir!

Erişim yetkileriyle ilgili olarak, zaman zaman dosya ve dizinlerin sahiplerinin değiştirilmesi gerekebilmektedir. Örneğin, root kullanıcı bir nedenle, bir kullanıcı dizininde bir dosya ya da dizin yaratırsa ve bu yeni yaratılan dosya/dizin o kullanıcı tarafından tam yetkiyle kullanılmasını isterse, bunu sağlamanın yolu, bu yeni yaratılan dosya/dizin sahibini o kullanıcı yapmaktır.

Bir dizindeki tüm alt dizin ve onların altındaki tüm dosyaların sahiplerini tek harekette değiştirmek istediğinizde;

chown -R cayfer dizin1

komutundaki gibi “-R” parametresini kullanabilirsiniz.



```
-- root@notebook.lojman.bilkent.edu.tr: /root - Shell - Konsole
[root@notebook root]# whoami
root
[root@notebook root]# mkdir /home/mayfer/yeni

[root@notebook root]# ls -l /home/mayfer
total 8
drwx----- 2 mayfer mayfer 4096 Feb 15 00:37 tmp/
drwxr-xr-x  2 root   root   4096 Feb 15 01:15 yeni/

[root@notebook root]# chown mayfer /home/mayfer/yeni

[root@notebook root]# ls -l /home/mayfer
total 8
drwx----- 2 mayfer mayfer 4096 Feb 15 00:37 tmp/
drwxr-xr-x  2 mayfer root   4096 Feb 15 01:15 yeni/

[root@notebook root]# chown mayfer:mayfer /home/mayfer/yeni

[root@notebook root]# ls -l /home/mayfer
total 8
drwx----- 2 mayfer mayfer 4096 Feb 15 00:37 tmp/
drwxr-xr-x  2 mayfer mayfer 4096 Feb 15 01:15 yeni/

[root@notebook root]#
```

Dosyaların ve Dizinlerin Gruplarını Değiştirmek

chgrp (*change group*)

Bu komutu sadece “root” kullanabilir!

Diyelim ki **/home/cayfer/stok** dosyasına “**pazarlama**” grubundaki tüm kullanıcıların erişebilmelerini istiyorsunuz. Ama diğer kullanıcıların bu dosyanın içeriğini görmelerini istemiyorsunuz. Bu durumda **/home/cayfer/stok** isimli dosyanın erişim yetki kalıbı için en uygun seçimin

rw-r-----

olduğu konusunda sanıyoruz aynı görüştesiniz.

Bir de dosyanın ait olduğu grubu “**pazarlama**” yaparsanız iş tamamdır. Bu işi de,

chgrp pazarlama /home/cayfer/stok

gibi bir komutla halledebilirsiniz. Ancak, bu komutun çalışabilmesi için sistemde “**pazarlama**” isimli bir grup tanımlı olmalıdır. Bir LINUX sistemde tanımlı olan gruplar **/etc/group** isimli bir dosyada saklanır. Bu dosyanın sahibi “root” kullanıcısıdır. “**more /etc/group**” komutuyla sisteminizde tanımlı grupları görebilirsiniz. Yeni bir grup eklenmesi gerektiğinde “**root**” bu dosyayı basit bir editörle açıp istediği düzenlemeleri yapabilir.

Bir dizindeki tüm alt izin ve onların altındaki tüm dosyaların gruplarını tek harekette değiştirmek istediğinizde;

chgrp -R pazarlama dizin1

komutundaki gibi “**-R**” parametresini kullanabilirsiniz.

Dosyaların ve Dizinlerin Sahibini ve Grubunu Birlikte Değiştirmek

Bir dosya ya da dizinin hem sahibini hem de grubunu tek komutla değiştirebilirsiniz. Tabii ki “**root**” iseniz...

chown cayfer:pazarlama dosya_adi

chown cayfer:pazarlama dizin_adi

chown -R cayfer:pazarlama dizin_adi

gibi komutlar işinizi görecektir.

bash Kabuk Programı

UNIX “Kabuk programı” (*shell*) kavramını iyi anlamanız çok önemlidir. Kabuk programları, sistemde çalışan programları saran, bu programların çalışması için gereken ortamı yaratan ve kucakladığı programların işletim sistemiyle iletişim kurmasını sağlayan yazılımlardır. Sistemde çalışan neredeyse her program bir kabuk içinde çalışır.

Kabuk programlarının üç önemli işlevi vardır:

1. Yazılımların çalışma ortamını belirlemek.
2. Varsa, kullanıcılarla işletim sisteminin iletişimini sağlamak. (Bu cümle biraz garip kaçtı galiba... Evet, kullanıcısı olmayan programlar olabilir. Örneğin **crond** geri planda sessizce çalışan, sistem saatini ve **crontab** dosyalarını gözleyen, zamanı geldiğinde planlanmış işleri başlatan bir yazılımdır. Bu yazılımın herhangi bir kullanıcısı olmadığı için kullanıcı-işletim sistemi iletişimi anlamlı değildir.)
3. Yazılımların çalışmasını denetleyebilecek kodların yazılabileceği bir programlama dili ortamı sağlamak.

Büyük olasılıkla daha iyi tanıdığınız Windows-MSDOS dünyasından bir benzetme yapmak gerekirse, COMMAND.COM yazılımı aslında bir kabuk programıdır.

UNIX kabuk programları da aynen bu işleri yapar. Ancak, UNIX işletim sistemi çok kullanıcılı olduğu için her program için ayrı bir kabuk gerekir.

Önemli bir fark da, UNIX dünyasında çok çeşitli kabuk programları olması ve kullanıcıların kendi kabuk programlarını seçebilmesidir.

Nitekim, bir terminal penceresi açtığınızda, sizi, kullanıcı hesabınız açılırken tanımlanmış olan kabuk programı karşılayacaktır. LINUX dünyasının en popüler kabuk programı **bash** olduğu için eğer değiştirmediyerseniz sisteme yaptığınız her bağlantı için başlatılacak kabuk **bash** olacaktır. Eğer bir nedenle **bash** kabuk programını beğenmiyorsanız, ya da teknik bir nedenle örneğin **csh** kabuğuna gereksinim duyuyorsanız, **bash** kabuğunuz içinde **/bin/csh** komutunu vererek **C-Shell** kabuğuna bürünebilirsiniz.

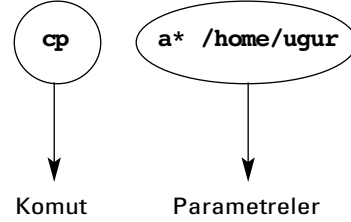
UNIX Kabuk programlarının hepsi son derece güçlü bir programlama dili desteğine sahiptir. O kadar ki, sadece kabuk komutları kullanarak bir bord-ro programı bile yazmak olasıdır.

Bir UNIX sisteme telnet ile bağlandığınızda ya da X ekranınızda yeni bir ter-minal penceresi açtığınızda o terminal için hemen bir kabuk başlatılacak ve başlatılan bu kabuk programı size bir hazır işareti gösterecektir. Terminal pencerenizde verdiğiniz tüm LINUX komut satırlarını bu kabuk irdeleye-cek, verdiğiniz komuta uygun programı diskten belleğe yükleyecek ve var-sa, verdiğiniz parametreleri bu programa aktaracaktır.

Şimdi kullanıcının bir terminalden yazdığı komutun **bash** tarafından irde-lenmesi ve yerine yerine getirilmesi aşamalarını bir örnekle açıklamaya çalışalım:

cp ./a* /home/ugur

komutu verilip de Return tuşuna bastığınızda, komut satırı önce parçalanarak bi-leşenlerine ayrılacaktır. Parçalama için boşluk, virgül gibi noktalama işaretleri ve tab karakterleri ayraç olarak kullanılacaktır. Satırbaşındaki ilk bileşenin bir ko-mut adı olduğu; komut satırının geri ka-lanının da bu komutun parametreleri ol-duğu kabul edilerek komut satırı çözüm-lenmeye çalışılacaktır.



Kabuk programı “a*” parametresini ayırırken * (*asterisk*) karakterini gö-rünce şöyle bir duralayıp çalışma dizininde yer alan ve adı a harfi ile başla-yan tüm dosya ve dizinlerin isimlerini bulacak ve sanki herbiri klavyeden ya-zılmışçasına komut satırına yerleştirecektir. Sonra da /home/ugur karakter-lerini son parametre olarak değerlendirecektir. Sonuçta, ilk verdiğiniz ko-mut,

cp abc aktifler acik.doc /home/ugur

şekline dönüşecektir. (Tabii ki çalışma dizininizde **abc**, **aktifler** ve **acik.doc** isimli dosyalar olduğunu varsayarsak.)

Sonra da, **cp** komutuna ait program dosyası **PATH** ("**path**" de olabilir) isimli kabuk değişkeninde (*shell variable*) yer alan dizinlerde aranacak ve bulunursa belleğe yüklenerek çalıştırılacaktır. (**PATH** ve **path** kabuk değişkenleri daha sonra anlatılacaktır.)

Komut satırının geri kalan kısmıysa, yani

"abc aktifler acik.doc /home/ugur", cp

komutuna parametre olarak gönderilecektir.



Tüm kabuk programlarında dikkat edilmesi gereken önemli bir nokta var: Dosya isim kalıpları (wildcards) kullanılan komutlarda, bu kalıpların açılması işlemi komut çalıştırılmadan önce yapılır ve bu açılmış halleri, ilgili komut programına parametre olarak aktarılır. Bir başka deyişle; "**cp *dat /tmp**" gibi bir komut verdiğinizde "***dat**" parametresi komut çalıştırılmadan önce kabuk tarafından açılabilirdiği kadar açılır; ondan sonra komut çalıştırılarak bu açılmış dosya isimleri parametre olarak gönderilir. Yani, sizin "***dat**" ve "**/tmp**" diye iki parametre ile çalıştırmayı düşünerek girdiğiniz bir komut, çalışma dizininizdeki dosyalar arasında kaç tane adı "**dat**" ile biten dosya bulunduğuyla ilgili olarak değişebilecek sayıda parametre ile çalıştırılır.

Dosya isim kalıbını, çalıştırmak istediğiniz programa kabuk tarafından dokunulmadan göndermek istediğinizde (**find** komutunda bu konuyu daha ayrıntılı olarak anlatacağız) dosya kalıbınızı tırnaklar arasına yerleştirmelisiniz.

komut "***dat**" **/tmp** gibi.

Dosya İsim Kalıpları

Kullanacağınız bir LINUX komutunda birden fazla dosyanın adından söz etmek mantıklıysa (örneğin **cp** komutu gibi) ve bu dosyaların isimleri belli bir kalıba uyuyorsa (örneğin söz konusu dosyaların isimleri "**a**" ile başlıyorsa, veya bu isimlerin hepsinin sonu "**dat**" ise, veya dosyaların isimleri içinde "**cayfer**" geçiyorsa) bu dosyaların isimlerini teker teker yazmak yerine dosya isim kalıplarını kullanabilirsiniz.

```
cp a* /tmp
cp *dat /home/cayfer
cp *cayfer* /var/yedek
```

gibi...

Kabuk programlarında dosya kalıplarında kullanılabilecek **joker** (*wilcard*) karakterleri özet olarak şunlardır:

Dosya İsim Kalıbı Karakterleri	
*	Her türlü karaktere uyar. Tüm dosyaları seçen bir kalıp kullanmanız gerekirse tek bir * karakteri yeterli olacaktır. MS-DOS'taki gibi "*.*" yazarsanız, adının içinde en az bir nokta olan dosyaları seçmiş olursunuz; yani adının içinde nokta olmayan dosyaları seçmemiş olursunuz!
?	Herhangi bir tek karaktere uyar.
[a,b,c]	a veya b veya c karakterlerinden birine uyar.
[0-9]	0'dan 9'a kadar rakamlara uyar.

Örnekler	
cat kitap[1-3] > hepsi	kitap1, kitap2 ve kitap3 dosyalarını peşpeşe ekleyerek hepsi isimli dosyaya kopyalar.
chmod a=x *. [o,sh]	Adının son karakterleri .o veya .sh olan tüm dosyaların erişim yetkilerini herkes için "--x" olarak değiştirir.
chmod a=x * [.o, .sh]	Yukardakinin aynıdır.
cp ?? /tmp	Adı iki karakterden oluşan dosyaları /tmp dizinine kopyalar.



* kalıp karakteri, adı . (nokta) ile başlayan dosyalar hariç tüm dosya isimlerine uyar. Örneğin,

```
cp * /home/ayfer/sakla
```

komutu, çalışma dizinindeki, adı nokta ile başlayanlar dışındaki tüm dosyaları /home/ayfer/sakla dizinine kopyalayacaktır. Eğer adı nokta ile başlayan dosyaları da (.login, .bashrc gibi) kopyalamak istiyorsanız,

```
cp .* * /home/ayfer/sakla
```

veya

```
cp .!* .b* * /home/ayfer/sakla
```

gibi komutlar kullanmalısınız.

Ortam Değişkenleri

Bir LINUX bilgisayarına login ettiğinizde, bir **shell** (*kabuk*) çalışma seansı başlatmış olursunuz. Bu seans boyunca kullanacağınız birtakım programlar çalışmalarını düzenleyen bazı değişkenlerin (kabuk değişkenleri) belirli değerlere sahip olmasını isteyebilirler. Örneğin, **crontab** programı (sisteminizde belirli aralıklarla, örneğin her Pazartesi saat 23:30'da, çalıştırılmasını istediğiniz programları düzenler) **EDITOR** isimli bir kabuk değişkeninde (ortam değişkeni, shell değişkeni) kullanacağı editorü görmek isteyecektir. Benzeri şekilde bash kabuğu **PATH** isimli bir ortam değişkeninde tam yeri belirtilmeyen komutlara ilişkin program dosyalarını araması gereken dizinlerin sıralı listesini bulmak isteyecektir.

Geleneksel olarak ortam değişkenlerinin isimleri büyük harflerle yazılır.

Ortam değişkeni yaratma, değer verme, değer değiştirme için kullanılan komutlar kullandığınız kabuğa göre değişecektir.

bash kabuğunda bir ortam değişkeni yaratmak ya da değerini değiştirmek için komut satırına

```
DEGISKEN=2003
```

veya

export DEGISKEN=2003

gibi bir komut yazmanız yeterlidir.

PATH Ortam Değişkeni

Vereceğiniz komutların program dosyalarının aranmasını istediğiniz dizinlerin listesini vermek istediğinizde

PATH=/bin:/usr/bin:/usr/local/bin:.

gibi bir komut verebilirsiniz. O andan itibaren terminalden vereceğiniz tüm komutlara ait program dosyaları öncelikle **/bin** dizininde; orada bulunamazsa **/usr/bin** dizininde; orada da bulunamazsa **/usr/local/bin** dizininde; orada da bulunamazsa çalışma dizininizde (sondaki noktaya dikkat) aranacaktır. Söz konusu program dosyası ilk olarak bu dizinlerden hangisinde bulunursa o program dosyası belleğe yüklenerek çalıştırılacaktır. Eğer program dosyası bu dizinlerden hiçbirinde bulunamazsa;

bash: xyz: command not found

hata mesajıyla ödüllendirilirsiniz.

Eğer **PATH** değişkenine yeni bir dizin ekleme gereksinimini duyarsanız;

PATH=\$PATH:/usr/local/prog

gibi bir komutla eski **PATH** değişkeninin değerine **"/usr/local/prog"** dizinini ekleyebilirsiniz.

Dikkat ettiyseniz bir ortam değişkeninin değerini kullanmanız gerektiğinde, o ortam değişkeninin adının başına bir **\$** işareti koymanız gerekiyor.

Diğer Önemli Ortam Değişkenleri

SHELL: Herhangi bir anda, içinde çalıştığınız kabuk programının adını ve bu kabuğun program dosyasının yerini ve adını içerir.

echo \$SHELL

komutunu verirsiniz, **/bin/bash** gibi bir yanıt alırsınız.

Kim Korkar LINUX'tan?

TERM: Kullandığınız terminalin tipini belirler. En çok kullanılan **TERM** değerleri “**xterm**” ve “**vt100**” dır.



Uygulama programlarının imlecin ekrandaki yerini denetleyebilmesi, ekrana yazılacak metinlerin renk, aydınlık olma, alt çizgili görüntülenme gibi özelliklerini kullanabilmesi için kullandığınız terminalin tipini bilmesi gerekir. Unutmayın, LINUX çok kullanıcıli bir işletim sistemidir ve kullanıcıların bir LINUX bilgisayarını kullanabilmek için onun önüne gelip oturması gerekmez. Uzaktan, kimbilir hangi yazılım ve/veya donanımı kullanarak bağlanabilen kullanıcıların ekran tiplerini LINUX'a bildirmeleri gerekmektedir. **TERM** ortam değişkeni işte bu amaçla kullanılır. X-Window ortamında da çalışan terminal programları için bu **TERM** değişkenine genellikle “**xterm**” değeri verilir bu iş genellikle otomatik olarak yapılır. Windows ve Mac gibi platformlarda çalışan terminal programlarıyla bir LINUX bilgisayara bağlandığınızda ise genellikle **TERM** değişkenine “**vt100**” değerini vermek iş görecektir. Makul terminallerin hepsi 20 yıl öncesinin “mükemmel” terminali olarak tarihe geçmiş olan DEC marka VT100 modeli terminal cihazları gibi davranacaktır.

Herhangi bir anda çalışma ortamınız için tanımlı olan ortam değişkenlerini ve değerlerini görmek isterseniz

env

komutunu kullanabilirsiniz.

Ortam Değişkenlerinin Kalıcı Olarak Değiştirilmesi

bash kabuğunda çalışırken terminalden vereceğiniz

PATH=\$PATH:/yeni/path

gibi bir komutun etkisi yalnızca o terminal içinde çalışmakta olan kabuk kopyasında görülecektir ve o kabuk programı sona erdirildiğinde (örneğin pencere kapatıldığında) söz konusu ortam değişkeni ayarları kaybolacaktır. Bu değişikliklerin kalıcı olmasını ve bundan sonra sisteme her bağlandığınızda ve her açtığınız terminal penceresinde geçerli olmasını istiyorsanız bunu bir yerlere kaydetmelisiniz.



Unutmayın! Bir terminal penceresi açtığınızda o pencere için yeni bir kabuk programı başlatılır. Her bir kabuk da sanki siz bilgisayara yeni bağlanmışsınız gibi davranır. Yaratacağınız veya yeni değer vereceğiniz ortam değişkenlerinin her terminal bağlantınızda karşınıza çıkmasını istiyorsanız ortam değişkeni atama deyimini kişisel dizininizdeki `.bashrc` dosyasına eklemeniz gerekir. Bu dosyadaki **bash** komutları her bash kabuğu başlatıldığında otomatik olarak çalıştırılır.

bash Başlangıç Dosyaları

bash kabuk programı her çalışmaya başladığında, önce kendisini çalıştıran kullanıcının kişisel dizinindeki `.bashrc` isimli dosyanın içindeki kabuk komutlarını çalıştırır. Bu iş, kabuk programı her başladığında yapılır. Açtığınız her pencere yeni bir kabuğun çalıştırılmasına neden olduğu için kişisel dizininizdeki `.bashrc` dosyasının içindeki komutlar her terminal penceresi açtığınızda çalıştırılacaktır.

bash kabuk programı bir şekilde durdurulduğunda (örneğin penceresi kapatıldığında ya da **exit** komutu verildiğinde) o kullanıcının kişisel dizinindeki (varsa) `.bash_logout` isimli dosyada yer alan komutları çalıştırır. Her kabuk programının kendine göre başlangıç dosyaları vardır. Örneğin **cs**h için bu dosyalar `.cshrc`, **ksh** için `.kshrc` gibi isimler alırlar.

Bir kabuk programının çalışmaya son vermesi için ilgili pencere kapatılabilir; kullanıcı o kabuk programına **exit** veya **logout** komutunu verebilir. Kullanıcı kabuk programının komut satırı yorumlayıcısına dosya sonuna geldiği anlamında **^D** (Ctrl-D) karakteri gönderebilir; ya da bir başka terminalde çalışan yetkili birisi (ki genellikle ancak **root** kullanıcı veya işin sahibidir) **kill** komutuyla kabuk programına kendini öldürmesi emrini gönderebilir.

Bu kabuk başlangıç dosyaları işi size biraz karışık geldiyse hiç dert etmeyin. LINUX bilgisayarınızı standart işler için kullandığınız sürece bu dosyalara bakmanız bile gerekmeyecektir. Ama gene de merak ettiyseniz kişisel dizinize geçip (**cd** komutu parametresiz olarak vermeniz bunu sağlayacaktır)

```
cat .bashrc
```

komutunu veriniz.



Aklınızın bir köşesinde bulunsun; Windows ve MS-DOS işletim sistemlerine benzetmek gerekirse bu kabuk başlangıç dosyaları kişisel StartUp veya AUTOEXEC.BAT dosyaları gibidir; sadece çok daha gelişmiş yeteneklerle donatılmışlardır.

Sistem yöneticisi iseniz genellikle **bashrc** başlangıç dosyasında yapacağınız bir değişikliğin tüm kullanıcılar için geçerli olmasını istersiniz. Her kullanıcının kişisel dizinindeki **.bashrc** dosyalarını değiştirmek ya da değiştirmelerini istemek yerine sistemin her kullanıcı için otomatik olarak çalıştıracağı komutları içeren **/etc/bashrc** dosyasında gerekli değişiklikleri yapmak yeterli olacaktır.

```
cayfer@notebook.lojman.bilkent.edu.tr: /home/cayfer - Shell - Konsole <2>
[cayfer@notebook cayfer]$ more .bashrc
# .bashrc

# Kişiyi özel bashrc:

# Varsa, sistemin bashrc'sini çalıştır
if [ -f /etc/bashrc ]; then
    . /etc/bashrc
fi

# ls komutunu temel formunda kullanmak için
unalias ls
#
export ORACLE_HOME=/usr/lib/oracle
export PATH=$PATH:/usr/local/bin/netscape

[cayfer@notebook cayfer]$
```

Kabuk programı sona erdiğinde çalıştırılan kişisel **.bash_logout** dosyasına ise gereksiz dosyaları temizleyen komutlar yerleştirilebilir. Tüm kullanıcılar için geçerli olan bir **bash_logout** dosyası ise gene **/etc** dizininde yer alabilir.

bash History

bash kabuğunun çok kullanışlı bir özelliği vardır: Son verdiğiniz bir sürü komutu saklar ve ekranda hazır işareti varken klavyenizde her yukarı ok tuşuna bastığınızda bir önce verdiğiniz komut getirilir. “bash history” özelliikle son komutu tekrarlamak gerektiğinde çok kullanışlıdır. Diyelim ki oldukça uzun bir komutu yanlış yazıp gönderdiniz:

```
co /home/cayfer/tez/arastirma/bulgular1 /tmp/cayfer/gecici
```

Yukarı ok tuşuna basınca son komut olan bu hatalı komut ekrana gelecektir. Sola okla da en baştaki “o” harfinin sağına gelip önce “backspace” tuşu, sonrada hatayı düzeltmek üzere “p” tuşuna basıp komutu tekrar gönderebilirsiniz.

En son kaç komutun saklanacağı **HISTSIZE** ortam değişkeninde belirtilir. Varsayılan değer son 1000 satırın saklanacağı anlamında 1000’dir. Bunu değiştirmek istediğinizde kişisel dizininizdeki **.bashrc** dosyasındaki ilgili satırı değiştirmeniz ya da dosyanın içine

```
HISTSIZE=750
```

gibi bir satır eklemeniz yeterli olacaktır.

Bir kullanıcının son kullandığı 1000 (veya **HISTSIZE** değişkeninde belirtildiği kadar) komut o kullanıcının kişisel dizinindeki **.bash_history** isimli dosyada saklanır.

! Aracı

Daha önce verilmiş komutlar arasında, **c** harfi ile başlayan en son komutu tekrarlamak için “**!c**” komutunu kullanabilirsiniz. Daha kesin tanımlamalar gerekirse, “**!ca**” gibi daha uzun diziler kullanabilirsiniz.

Son vermiş olduğunuz **1000** komutu görmek için **history** komutunu kullanabilirsiniz. Bu komutu verdiğinizde, daha önce vermiş olduğunuz komutlar, birer sıra numarasıyla ekrana listelenir. Bu listedeki komutlardan herhangi birini tekrarlamak istediğinizde **!** işareti ve hemen yanına tekrarlamak istediğiniz komutun sıra numarasını yazıp Enter tuşuna basmanız yeterlidir. (**!14** gibi..). **bash** kabuğu, verilen komutların tamamını değil; **\$HISTSIZE** ortam değişkeninin değeri kadar komutu saklar. **\$HISTSIZE** ortam değişkeninin varsayılan değeri 400’dir.



Birçok komutu yazarken gerek komut adını, gerekse komutun parametresi olabilecek dosya isimlerinin tamamını klavyede yazmanız gerekmez.

Örneğin, OpenOffice paketindeki kelime işlemci olan “oowriter” programını komut satırından başlatmak istediğinizde klavyeden “oowriter” sözcüğünün tamamını yazmanıza gerek yoktur.

“oo” yazıp <tab> tuşuna bastığınızda **PATH** ortam değişkeninizde belirtilmiş dizinlerde adı “oo” ile başlayan çalıştırılabilir program dosyaları aranır. Bu şartı sağlayan tek bir dosya bulunursa sanki siz klavyeden “oowriter” yazmışsınız gibi komut otomatik tamamlanır. Yok eğer adı “oo” ile başlayan birden fazla çalıştırılabilir program varsa, karşınıza bu kalıba uyan dosyaların bir listesi çıkarılır. Bir harf daha; örneğin “w” ekleyip gene <tab> tuşuna bastığınızdaysa bu kez adı “oow” ile başlayan komutlar aranır. “oow” ile başlayan tek komut büyük olasılıkla “oowriter” olacağı için komutunuz bash tarafından tamamlanacaktır.

Aynı kolaylığı dosya isimlerinde de kullanabilirsiniz. Örneğin:

```
cp prozgui-3.04.source.tar.gz sil.gz
```



yazmanız gerektiğinde klavyeden “cp pr” yazdıktan sonra <tab> tuşuna bastığınızda çalışma dizininizde adı “pr” ile başlayan dosyalar taranacak ve adı bu kalıba uyan dosyanın adı otomatik olarak tamamlanacaktır.

Kendi Komutlarınız: Aliases

Daha önce LINUX'ta kendi komutlarınızı yaratmanın mümkün olduğunu belirtmiştik. İşte bu amaçla kullanılan **alias** komutuna birkaç örnek...

```
alias dir=ls
alias copy="cp -i"
alias ll="ls -l"
alias mroe=more
alias h=history
alias ls="ls -F"
```

“alias dir=ls” komutu, MS-DOS alışkanlıklarından kolay vazgeçemeyen kullanıcılar için yararlı olabilir. Bu komutu verdiğinizde, artık, dosya listesi

almak için isterseniz **dir** isterseniz **ls** komutunu kullanabilirsiniz. “**dir**” ve “**ls**” komutları artık eşanlamlı olur.

Aynı mantıkla **cp** komutunun **-i** parametresiyle kullanılan şeklini “**copy**” diye yeni bir komut olarak tanımlayabilirsiniz.

Çok ilginç bir şekilde klavyesi hızlı olan kullanıcılar sıkça “**more**” yerine “**mroe**” yazarlar. Tuşların yerleşiminden olsa gerek... Siz de bu hatayı çok yapanlardansanız oldukça sık kullanılan “**more**” komutunu yanlış yazma olasılığına karşı önceden önlem alıp, eşanlamı “**more**” olan bir “**mroe**” komutu tanımlayabilirsiniz. Artık klavyeden “**more**” da yazsanız, “**mroe**” da yazsanız aynı komut çalışacaktır.

alias komutunu kullanarak kendi komutlarınızı yaratmak ya da mevcut komutların sizin istediğiniz parametrelerle çalışmasını istediğinizde gerekli alias komutlarını kişisel dizininizdeki **.bashrc** dosyasının sonuna ekleyebilirsiniz. Eğer bu **alias** tanımlarının tüm kullanıcılar için geçerli olmasını istiyorsanız bu eklemeleri **/etc/bashrc** dosyasına yapmalısınız.

Şimdi sorduğunuzu duyar gibiyiz: *“Peki! Bir dosyada nasıl değişiklik yapacağım?”*

Bunun yanıtı da bir sonraki bölümde...

BUNLARI BİLİYOR MUYDUNUZ?

CD Tabanlı Linux Dağıtımları

Bir bilgisayarda LINUX kullanabilmek için o bilgisayarın diskine LINUX kurmak zorunda olmadığınızı biliyor muydunuz?

CD tabanlı LINUX dağıtımlarından birini kullanırsanız, diskinizi hiç bozmadan bilgisayarınızı LINUX işletim sistemiyle açıp kullanabilirsiniz. Elbette ki bu şekilde kullanılan bir işletim sisteminden çok fazla iş beklemek doğru değildir ama denemek ve öğrenmek için pek fena da sayılmaz.

Şu sıralar oldukça popüler olan, CD tabanlı LINUX dağıtımı Knoppix'i deneyebilirsiniz. Eğer İnternet bağlantı hızınız bir CD dolusu dosyayı indirmeye uygunsa www.knoppix.org adresini bir ziyaret etmenizi öneririz. Eğer bağlantı hızınız CD indirmeye uygun değilse, Knoppix CD'lerini www.linux34.com adresinden son derece makul bir fiyata ısmarlatabilirsiniz.

Editörler: Kedit vs. ve En Önemlisi: vi

04

- **vi: Visual Editor**
 - **vi Bir Kelime İşlemci Değildir!**
 - **Beğenseniz de Beğenmeseniz de, vi Öğrenmelisiniz!**
 - **Dosya Açmak / Yaratmak**
 - **vi Editörünün “Durum”ları**
 - **gvim**

LINUX işletim sistemiyle birlikte kullanılabilecek pek çok editör vardır. Bunlar KDE, GNOME gibi masa üstü yöneticileri altında kullanılabilecek grafik arabirimli editörler ve grafik olmayan ekranlarda kullanılabilecek konsol editörleridir.

KDE ile beraber yüklenen dosya editörlerini

KDE Menüsü

–

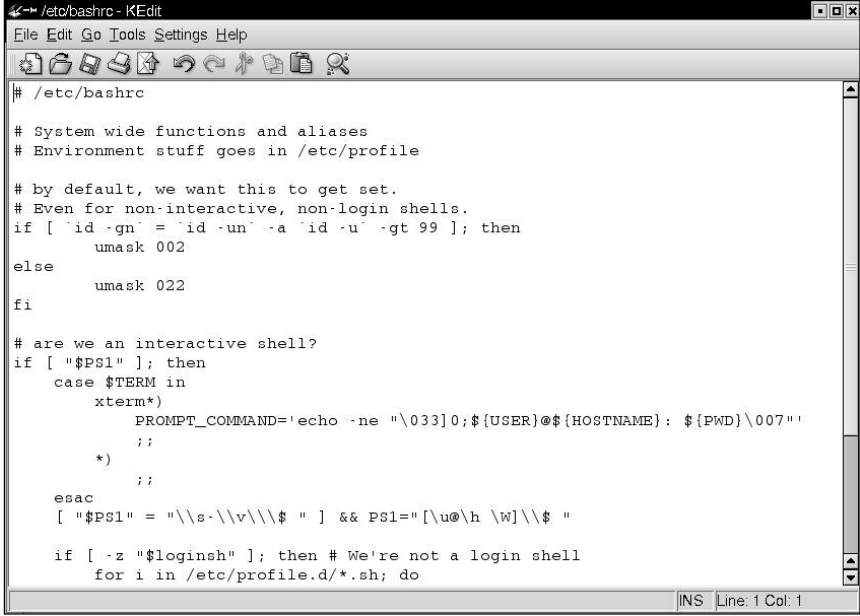
More Applications

–

Editors

altında bulabilirsiniz.

Bunları deneyip zevkinize uygun birini seçebilirsiniz. Bu menüdeki grafik kullanıcı arabirimli (GUI) editörlerin kullanımı son derece kolaydır.



```
# /etc/bashrc

# System wide functions and aliases
# Environment stuff goes in /etc/profile

# by default, we want this to get set.
# Even for non-interactive, non-login shells.
if [ `id -gn` = `id -un` -a `id -u` -gt 99 ]; then
    umask 002
else
    umask 022
fi

# are we an interactive shell?
if [ "$PS1" ]; then
    case $TERM in
        xterm*)
            PROMPT_COMMAND='echo -ne "\033]0;${USER}@${HOSTNAME}: ${PWD}\007"'
            ;;
        *)
            ;;
    esac
    [ "$PS1" = "\s-\v\\\$ " ] && PS1="\u@\h \W\\\$ "

    if [ -z "$loginsh" ]; then # We're not a login shell
        for i in /etc/profile.d/*.sh; do
```

Konsol editörlerine gelince durum biraz değişir. Bunlar grafik ekranlar daha bilgisayar dünyasında ortaya çıkmadan yazılmış editörler olup her türlü bilgisayar terminalinde çalışacak şekilde geliştirilmiştir. Şimdi diyeceksiniz ki “grafik ekranı olmayan bilgisayar mı kaldı ki?” Kaldı tabii.. Kalmaya da devam edecek gibi görünüyor. Niye bilgisayar denince yalnızca PC aklınıza geliyor? Hiç ekranı olmayan bilgisayarlar bile var. Bu nedenle konsol editörleri hala profesyonel bilgisayarçıların; daha doğrusu UNIX’cilerin eli ayağıdır.

Konsol editörleri arasında en çok sevilenler **pico**, **emacs** ve **vi** editörleridir. Bunlar arasında tüm UNIX uyarlamalarında standart olanı **vi** editörüdür. Bu nedenle UNIX dünyasına adımını atan herkesin **vi** öğrenmesi oldukça önemlidir.

vi: Visual Editor

vi Bir Kelime İşlemci Değildir!

vi bir editördür! Yalnızca basit metin dosyaları üzerinde kullanıldığında işe yarar. Font, kalın yazı, italik yazı gibi kavramlar **vi**'da yoktur. Yani mektuplarınızı, hele tekliflerinizi ve raporlarınızı **vi** ile yazmaya çalışmak anlamsız olacaktır. Bu gibi işler için OpenOffice, KOffice gibi yazılımlar daha uygundur.

Eğer kelime işlemci gereksiniminiz bilimsel makalelere yönelik ve özellikle matematiksel sembollerle doluysa o zaman TeX, LaTeX, Lyx üçlüsünü öğrenmenizi öneririz. Düzleme çizilebilecek ve yazılabilecek her türlü dökümanı uluslararası standartlarda geliştirebilirsiniz.

Beğenseniz de Beğenmeseniz de, vi Öğrenmelisiniz!

vi UNIX dünyasının standart editörüdür. UNIX veya herhangi bir türeviyle çalışan her bilgisayarda bulunur.

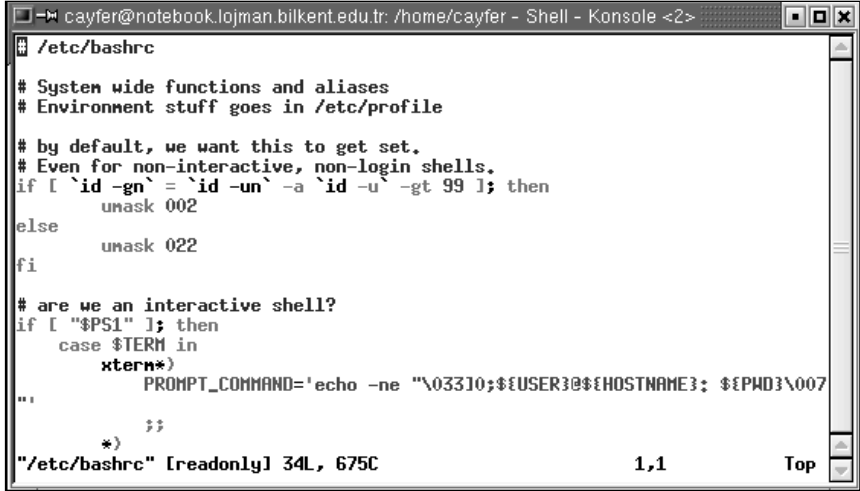
vi ile ilk karşılaşan kullanıcılar genellikle “*bu ne ya..*” diye tepki gösteriyorlar. Bunlar arasında sabırlı olanlar bir süre sonra **vi** kullanmayanlara hayretle bakmaya başlıyorlar. Çok güçlü bir editör olmasına rağmen bu kitapta **vi** hakkındaki herşeyi anlatmayacağız. Zaten pek anlatamayız da.. **vi** için ayrıca bir kitap yazmak gerekir. Önce temel editör işlemlerinden başlayalım isterseniz...

Dosya Açmak / Yaratmak

Bir dosyayı düzenlemek üzere açmak için

vi dosya-adı

komutunu verebilirsiniz. Örneğin “**vi /etc/bashrc**” komutunu verdiğinizde karşınıza



```

cayfer@notebook.lojman.bilkent.edu.tr: /home/cayfer - Shell - Konsole <2>
/etc/bashrc
# System wide functions and aliases
# Environment stuff goes in /etc/profile

# by default, we want this to get set.
# Even for non-interactive, non-login shells.
if [ `id -gn` = `id -un` -a `id -u` -gt 99 ]; then
    unmask 002
else
    unmask 022
fi

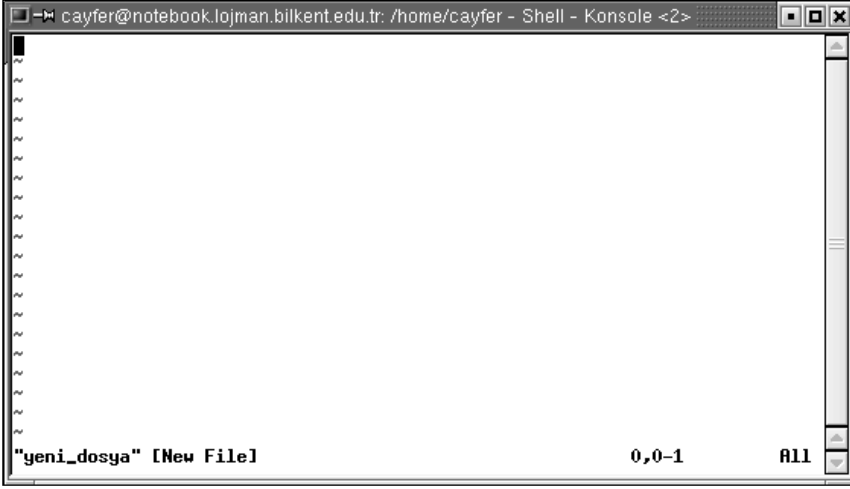
# are we an interactive shell?
if [ "$PS1" ]; then
    case $TERM in
        xterm*)
            PROMPT_COMMAND='echo -ne "\033I0;${USER}@${HOSTNAME}: ${PWD}\007"'
            ;;
        *)
            ;;
    esac
fi

"/etc/bashrc" [readonly] 34L, 675C      1,1      Top
```

gibi bir ekran gelecektir.

Ekranın sol alt tarafında düzenlenen (edit edilen) dosyanın adı, yalnızca okuma için ([readonly]) açıldığı, satır sayısı ve kaç byte uzunluğunda olduğu belirecektir. Bu örnekte **/etc/bashrc** dosyasının yalnızca okuma için açılmasının nedeni “**root**” kullanıcısı dışındakilerin bu dosya üzerinde yazma yetkisi olmamasındandır.

Eğer **vi** komutuna parametre olarak verdiğiniz dosya diskte yoksa, yeni ve boş bir dosya yaratılarak karşınıza



ekranı gelecektir. Bu ekrandaki ~ işaretleri, boş satırlar ile olmayan satırlar karışmasın diye görüntülenir.

vi Editörünün “Durum”ları

vi editörü herhangi bir anda iki durumdan birindedir:

1. Ekleme durumu
2. Komut durumu

“Ekleme Durumu”nda, klavyeden bastığınız herşey imlecin bulunduğu noktadan başlayarak dosyaya eklenecektir.

“Komut Durumu”nda klavyeden bastığınız herşey bir **vi** komutu olarak değerlendirilecektir.

vi editörünü ekleme durumuna sokmak için “**i**” tuşuna (**küçük i: insert**) bir kez basmanız yeterlidir.

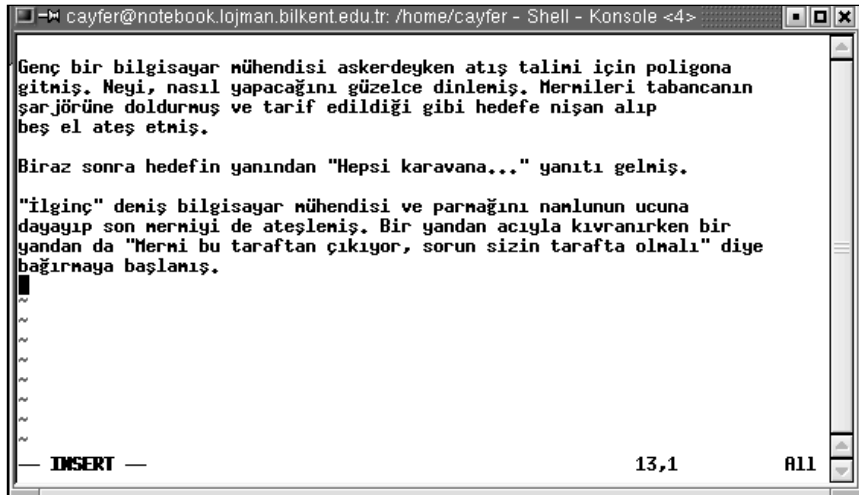
Ekleme durumundan komut durumuna geçmek için ise Esc tuşuna bir kez basmanız gerekmektedir. Esc tuşuna birden fazla kez basmanız bir şey değiştirmeyecektir. Yani komut durumundayken Esc tuşuna basarsanız gene komut durumunda kalırsınız.

vi ekleme durumundayken (*insert mode*) ekranın sol alt tarafında “-- **INSERT** --” işaretini görürsünüz.

Sağ tarafta ise dosyanın kaçınıcı satırında ve o satırın hangi yatay pozisyonunda olduğunuzu izleyebilirsiniz.

Ekleme Durumu

Dosyanız ekleme durumundayken (bu duruma geçmek için bir kez “**i**” tuşuna basmalısınız) birkaç satır veri giriniz. Satırların sonuna geldiğinizde Enter tuşuyla satır başı yapabilirsiniz.



```
cayfer@notebook.lojman.bilkent.edu.tr: /home/cayfer - Shell - Konsol <4>
Genç bir bilgisayar mühendisi askerdeyken atış talini için poligona
gitmiş. Meyi, nasıl yapacağını güzelce dinleniş. Mernileri tabancanın
şarjörüne doldurmuş ve tarif edildiği gibi hedefe nişan alıp
beş el ateş etmiş.

Biraz sonra hedefin yanından "Hepsi karavana..." yanıtı gelmiş.

"İlginç" demiş bilgisayar mühendisi ve parmağını namlunun ucuna
dayayıp son merniyi de ateşleniş. Bir yandan acıyla kıvrılırken bir
yandan da "Merni bu taraftan çıkıyor, sorun sizin tarafta olmalı" diye
bağırılmaya başlanış.

-- INSERT -- 13,1 All
```

Komut Durumu

Dosyayı kaydetmek için yani editöre “**kaydet**” (*write*) komutunu verebilmek için ekleme durumundan (hatırlarsanız en başta bir küçük **i** harfi ile bu duruma geçmiştiniz) çıkıp komut durumuna geçmeniz gerekecektir. Ekleme durumundan çıkmak için bir kez Esc tuşuna basmanız yeterlidir. (Fazla Esc basmanın bir zararı olmaz. Sisteminiz her fazla basış için bir kez düdük çalacaktır; o kadar.)

Ekranda komut durumunda bulunduğunuza ilişkin bir işaret aramayın. Yoktur! Ekleme durumunda olmamanız komut durumunda olduğunuzu gösterir. Ancak, hangi durumda bulunduğunuzdan emin olmak istiyorsanız, bir kez Esc tuşuna basıp komut durumunda olmayı garantileyebilirsiniz.

de değildir. Eğer dizin belirtmediyseniz o dosya çalışma dizininizde de olmayabilir. Bir başka olasılık da, dosya adını yazarken büyük harf-küçük harf konusuna dikkat etmemiş olabilirsiniz; hatırlarsanız LINUX işletim sisteminde **deneme** ile **Deneme** farklı dosya adlarıdır!

Aşağıdaki mesajlardan birini görürseniz:

```
[Read Only]          veya
File is Read only     veya
Permission denied
```

söz konusu dosyaya ya da bulunduğu dizine yazma yetkiniz yok demektir. Normal olarak yapabileceğiniz bir şey olmadığından **vi** seansını bir an önce durdurup, dosyanın sahibi ile görüşmeniz ya da uygun bir kullanıcı kimliğine bürünmeniz gerekmektedir.

File System full mesajını görürseniz, çalıştığınız disk ya da kotanız dolmuş demektir; gereksiz dosyaları silerek yer açmanız gerekmektedir.



UNIX işletim sisteminde, kullanıcıların disklerde kullanabilecekleri alanların toplam büyüklüğünü sınırlama (kota koyma) olanağı vardır. Böyle bir sınırlamanın olup olmayacağına, olacaksa her kullanıcı için kotanın kaç megabyte olacağına sistem yöneticisi karar verir.

Kullanıcılar kotalarını doldurduklarında bir mesajla uyarılır ve genellikle kotalarını biraz aşmalarına izin verilir; ancak birkaç gün içinde tekrar kota limitlerinin altına inmezlerse, sistem daha fazla yeni dosya yaratmalarına veya eski dosyalarını büyütmelerine izin vermeyecektir.

vi editörü, terminal penceresinin tamamını kullanan **tam-ekran** (*full screen*) bir editördür. Bu nedenle temel işlevler için en iyi hakim olunması gereken komutlar, imleci ekranda dolaştırma komutlarıdır.

İmleç Dolaştırma Komutları

Bu komutların verilmesi sırasında, editör komut konumunda bulunmalıdır. (Yani editör yeni başlatılmış olmalı veya ekleme konumundan çıkmak için Esc tuşuna basılmış olmalıdır. Hatırlayacaksınız, fazla Esc basmanın bir zararı yoktu; dolayısıyla bulunduğunuz konumla ilgili bir şüpheniz varsa, komut durumuna geçmek için hiç çekinmeden Esc tuşuna basınız.)

Normal koşullarda imleci ekranda dolaştırmak için klavyedeki ok tuşlarını kullanabilirsiniz. Ama UNIX dünyasında ne zaman nasıl bir terminal veya terminal emülasyon programı kullanacağınız belli olmaz; bu nedenle ok tuşları bulunmayan bir terminali kullanırken gerekebilecek imleç dolaştırma komutlarından söz etmekte yarar var:

vi İmleç Yönlendirme Komutları	
Tuş	İmleç Hareket Yönü
Oklar	...
h	Sola bir karakter.
j	Aşağı bir satır.
k	Yukarı bir satır.
l	Sağa bir karakter. (Küçük L harfi)

Tabii bu arada Return (veya Enter) tuşu ile BackSpace tuşunun da sırasıyla satırbaşı ve bir geri anlamına geldiğini hatırlatmakta fayda var.

Birçok **vi** komutu gibi, imleç hareket komutlarının da başına bir sayı koyarak, bu sayı kadar sağa, sola, aşağı ve yukarı hareket sağlanabilir.

3h	3 karakter sola.
2k	2 satır yukarı, gibi.

Birerli adımlar dışında hareket sağlayan ve çok kullanılan imleç dolaştırma komutlarından bazılarıysa:

o	(Sıfır) İmlecin bulunduğu satırın başına.
\$	İmlecin bulunduğu satırın sonuna.
w	Bir sonraki sözcüğün başına.
b	Sözcüğün başına. (Ya da bir önceki sözcüğün başına)

Ekleme Komutları

Editör kullanırken en çok yapılan işlerden biri, eski bir metnin başına, sonuna ve araya satır/kelime/harf eklemektir. **vi** editöründe ekleme konumuna geçmenin yöntemlerinden birini (**i** komutu) daha önce belirtmiştik. Yine de tekrarlamak istiyoruz:

i	İmlecin, üzerinde bulunduğu karakterin hemen solundan başlayarak, Esc tuşuna basıncaya kadar basılan her karakteri metne ekler. Varsa, eski metin sağa doğru itelenir.
----------	--

Diğer ekleme komutları:

a	İmlecin, üzerinde bulunduğu karakterin hemen sağından başlayarak, Esc tuşuna basıncaya kadar, basılan her karakteri metne ekler. Varsa, eski metin sağa doğru itelenir.
A	İmlecin, üzerinde bulunduğu satırın sonundan başlayarak, Esc tuşuna basıncaya kadar, basılan her karakteri metne ekler.



Bu komutların büyük veya küçük harflerle verilmesi farklı ve önemlidir. Dikkatli olun.

Yazı Silmek

Daha önce yazılmış metin parçalarını silmek için kullanılan komutları aşağıdaki tabloda örnekleriyle özetlenmiş olarak bulacaksınız:

Metin Silme Komutları	
x	İmlecin üzerinde bulunduğu tek karakteri siler.
3x	İmlecin üzerinde bulunduğu karakter dahil, sağa doğru 3 karakter siler.
dw	İmlecin bulunduğu yerden kelime sonuna kadar siler. Eğer imleç sözcüğün başındaysa, sözcüğü siler.

2dw	İmlecin bulunduğu yerden başlayarak 2 sözcük siler.
dd	İmlecin bulunduğu satırı siler.
2dd	İmlecin bulunduğu satır dahil, aşağı doğru iki satır siler.
D	İmlecin bulunduğu yerden satır sonuna kadar siler. (dş komutuna eşdeğerdir)
dş	Satırın sonuna kadar siler. (D komutuna eşdeğerdir)

Yanlışlıkla Silerseniz...

Eğer hatanızı hemen fark ederseniz, **u** komutuyla (*undo*) son silme işlemini geri döndürebilirsiniz. Undo aslında yalnızca son silme işlemini değil, son değişikliği geri döndürür.

Undo komutunu **U** harfiyle (büyük U) verdiğinizde en son değişiklik yapılmış olan satırdaki tüm değişikliklerin hepsi birden geri alınır.

Metin Bloklarının Yerini Değiştirmek

vi editörüyle metin bloklarının yerini değiştirmek istediğinizde kullanacağınız yöntem kes-yapıştır yöntemidir. Yerini değiştireceğiniz metin bloğunu önce bulunduğu yerden, uygun bir komutla (örneğin tek satır için **dd** komutu gibi) silmeli (kesme işlemi), daha sonra **p** komutuyla (*put*) yeni yerine yapıştırmalısınız. Eğer yapmak istediğiniz işlem bir metin bloğunu silmekse, kesme işleminden sonra başka bir yere yapıştırmamanız yeterlidir. Bu işlemleri bir örnekle anlatmak daha kolay olacak:

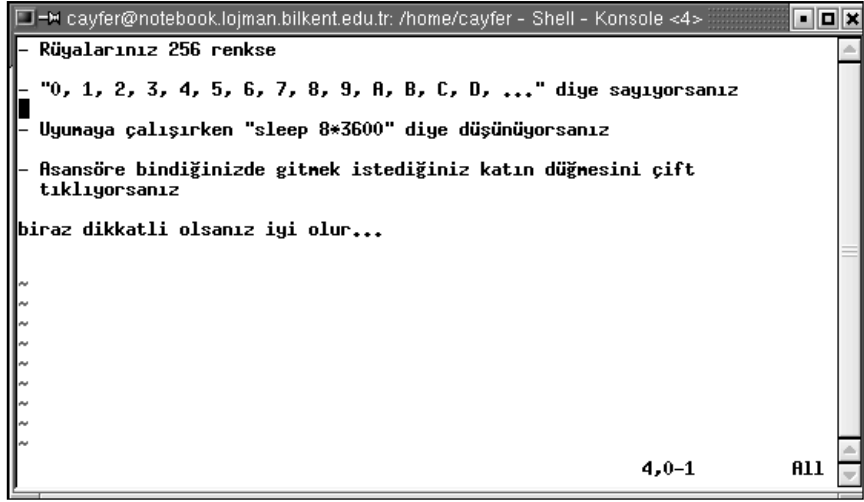
Aşağıdaki örnek metindeki, “- **Rüyalarınız**” sözcüğüyle başlayan üçüncü satırı en üst satıra taşımak istediğinizi varsayalım. Önce imleci bu satırın üzerine getirmelisiniz. (**dd** komutuna hazırlık.)

[illegible]

Dikkat ederseniz küçük bir bozukluk oldu. Yazının estetik bütünlüğünü korumak için “**Rüyalarınız...**” satırını altındaki boş satırla birlikte taşımak gerekiyordu.

dd komutuyla tek satır kesmek yerine **2dd** komutuyla iki satır kesseydik ve **P** ile bu iki satırı yapıştırsaydık böyle olmayacaktı. Neyse çok önemli değil. Oldu artık. Şimdi en kolayı birinci satırın altına bir boş satır eklemek ve “**0**, **1**, ...” satırından sonraki çift boş satırdan birini silmek.

Bunu yapmak için imleç birinci satırın üstündeyken “o” (küçük o) (open line) komutunu deneyin. İmlecin altına yeni bir boş satır eklenecektir. Tekrar “Esc” tuşuyla “ekleme” durumundan çıkın. Sonra imleci birini silmek istediğiniz çift boş satırdan birinin üzerine götürüp **dd** ile bir satır kesin ama hiçbir yere yapıştırmayın. Oldu işte!



İsterseniz şimdi de **vi** komutlarının birarada kullanılmasıyla elde edilebilecek ilginç bir etkiyi gösterelim:

Metnin herhangi bir yerinde, **x** ve **p** komutlarını peşpeşe veya birarada kullanırsanız (**xp**) imlecin altındaki karakterle sağındaki karakterin yerini değiştirmiş olursunuz. Lütfen deneyiniz!

Metin Bloklarını Kopyalamak

vi editörü ile metin bloklarını kopyalamak istediğinizde kullanacağınız yöntem kopyala-yapıştır yöntemidir. Kopyalayacağınız metin bloğunu önce bulunduğu yerde, **y** (*yank*) komutuyla (tüm satır için **yy** komutu veya **y** gibi) geçici belleğe aktarmalı ve daha sonra kopyalanacağı yere **p** komutu (*put*) ile yapıştırmalısınız.

Bu işlemleri de bir örnekle anlatmak daha kolay olacak galiba... Yukarıdaki örnek metindeki, “**Rüyalarınız...**” sözcüğüyle başlayan ilk satırı, “**biraz dikkatli...**” sözcükleriyle başlayan satırın altına kopyalamak istediğinizi varsayalım:

Önce imleci ilk satırın üzerine getirin. Sonra **yy** komutuyla bu satırı geçici bellek sahasına alın. Ekranda bir değişiklik olmayacaktır.

Daha sonra imleci “**biraz dikkatli...**” satırına indirerek, **p** (küçük p) komutu ile yeni yerine yapıştırın.

Kopyalama ve yer deęiřtirmede kullanılan **p** (küçük p) yapıřtırmayı imlecin bulunduęu satırın altına, **P** (büyük p) ise üstüne yapar.



Son Komutu Tekrarlamak

Bir nedenle, son deęiřiklik komutunuzu başka yerlerde de tekrarlamamız gerekirse, “.” (nokta) komutu ile bunu yapabilirsiniz.

Metin Eklemenin / Deęiřtirmenin Birkaç Deęiřik Yolu

Komut durumundayken basılacak tuřlarla bazı önemli metin deęiřtirme işlevleri yerine getirilebilir. Bunlar arasında sıkça kullanılanları ařaęıdaki tabloda bulabilirsiniz.

Komutların hangi İngilizce sözcüklerin kısaltmaları olduęunu da İngilizce bilen okurların hatırlamalarını kolaylařtırmak için tabloya ekledik.

Bazı Önemli Metin Deęiřtirme Komutları		
Komut	İngilizcesi	Açıklama
I	Insert	(Büyük I) İmlecin bulunduęu satırın BAřINA eklemeye başla. (Esc’e kadar)
o	Open Line	(Küçük o) İmlecin bulunduęu satırın ALTINA bir boş satır aç ve oraya eklemeye başla.
O	Open above	(Büyük O) İmlecin bulunduęu satırın ÜSTÜNE bir boş satır aç ve eklemeye başla.
s	Substitute Char	İmlecin bulunduęu yerdeki karakteri sil ve yerine yeni metni eklemeye başla. (Esc’e basıncaya kadar)
S	Substitute Line	İmlecin bulunduęu satırını sil ve yerine yeni metni eklemeye başla.

r	Replace Char	İmlecin bulunduğu karakteri bir sonra basılacak karakterle değiştir.
R	Replace Text	İmlecin bulunduğu noktadan itibaren, yeni metni eski metnin üzerine yerleştir. (Esc'e basıncaya kadar)
J	Join	(Büyük j) İmlecin bulunduğu satırla arkasındaki satırı birleştir.
cw	Change Word	İmlecin bulunduğu sözcüğü, yeni girilecek sözcükle değiştir.



Birçok **vi** komutu, başına bir çarpan yerleştirerek birden fazla kez tekrarlanabilir. (Aynı cebirdeki gibi...)

Örneğin, satır silmek için kullanılan **dd** komutu, **5dd** şeklinde verildiğinde "5 satır sil" anlamına gelir. Komutların bu özelliğini daha önceki örneklerde fark etmiş olmalıydınız.

Metnin İçinde Dolaşmak

Şu ana kadar hep birkaç satırdan oluşan dosyalarda çalıştık. Hayat her zaman bu kadar kolay değildir. Özellikle elektronik posta için metin yazarken ya da program geliştirirken, sık sık metin içinde sayfa sayfa (bir başka deyişle ekran ekran) ileri-geri gitmek gerekir. Bu işlemler için kullanılan komutları burada kısaca bir tablo halinde sıralayacağız. Birer kere denemenizi öneririz. Üzerinde çalışmak için uzun bir dosya hazırlamak yerine, sisteminizin **/etc** dizinindeki **termcap** dosyasını kendi dizininize kopyalayıp bu dosya üzerinde çalışabilirsiniz. ("**cp /etc/termcap ~**" sonra "**vi termcap**")

Metin İçinde Dolaşma Komutları		
Komut	İngilizcesi	Açıklama
Ctrl-F	Forward	Bir ekran ileri.
Ctrl-B	Backward	Bir ekran geri.
Ctrl-D	Down	Yarım ekran ileri.
Ctrl-U	Up	Yarım ekran geri.
Ctrl-R	Redraw	Ekranı yeniden düzenle. (Çalışırken sistemden gelebilecek mesajlar ekranınızdaki görüntüyü bozarsa bu komut, vi ekranını ekranı silip yeniden oluşturarak düzenleyecektir.)
Ctrl-Y		Ekranı bir satır aşağı kaydır, imleç yerinde kalsın.
Ctrl-E		Ekranı bir satır yukarı kaydır, imleç yerinde kalsın.
z RETURN		(Küçük z) İmlecin bulunduğu satır ekranın en üstüne gelecek şekilde ekranı düzenle.
z.		(Küçük z ve nokta) İmlecin bulunduğu satır ekranın ortasına gelecek şekilde ekranı düzenle.
z-		(Küçük z ve eksi) İmlecin bulunduğu satır ekranın en altına gelecek şekilde ekranı düzenle.
H	Home	Ekranın en üst satırına git.
M	Mid Screen	Ekranın orta satırına git.
L	Lower Screen	Ekranın en alt satırına git.
RETURN		Bir sonraki satırın ilk karakterine git.

Yukarıda açıklanan komutların bazı özel kullanımları vardır. Bunlar ender kullanılan özellikler olup, burada sadece bir fikir vermek amacıyla birkaç örnek vereceğiz:

200z RETURN	200. satırı ekranın en üstüne getir.
4H	Ekranın en üst satırının 4 altındaki satıra git.
5L	Ekranın en alt satırının 5 üstündeki satıra git.

Metnin İçinde Arayarak Dolaşmak

Diyelim ki, program yazıyorsunuz ve değişiklik yapmak istediğiniz satırın yeri hakkında bir fikriniz yok. Tek hatırladığınız, değiştirmek istediğiniz satırda **'kayit_sayısı++'** diye bir karakter dizisi var. Bu diziyi içeren bir satır bulmak için komut durumunda,

/kayit RETURN

yazmanız yeterli olacaktır. İmleç, içinde bu dizi geçen ilk satırda duracaktır. Arama, imlecin bulunduğu yerden ileriye doğru yapılır. Eğer geriye doğru arama yapmak isterseniz **"/** yerine **"?"** karakterini kullanmanız gerekir.

Diyelim ki, programın ilk rastladığı **kayit** sözcüğü, sizin ilgilendiğiniz değil ve aramaya devam etmek istiyorsunuz. Bu durumda **n** (*next*) tuşuna basmanız yeterlidir. Tüm arama komutunu yeniden yazmanız gerekmez. Eğer aramanın yönünü değiştirmek isterseniz **n** yerine **N** komutunu veriniz. Eğer aramanın halen hangi yönde olduğunu hatırlamıyor, fakat aramayı ileriye doğru yönleltmek istiyorsanız **/** tuşuna, tam tersi içinse **?** tuşuna basabilirsiniz.

Bulup Değiştirmek

Diyelim ki, bulunduğunuz satırda **"Linux"** diye bir sözcük var ve bu sözcüğü **"LINUX"** olarak değiştirmek istiyorsunuz. Bu işi yapmak için imleci **"i"** harfinin üzerine götürerek **4rINUX** komutunu vermeniz yeterli olacaktır. Bu yöntemde bulma görevini siz; değiştirme görevini ise **vi** üstlenmiş oluyor. Her iki işi de **vi**'in yapmasını istiyorsanız, değişikliğin yapılmasını istediğiniz satırın üzerine gelip,

:s/Linux/LINUX veya
:s/inux/INUX

komutlarını verebilirsiniz. Bu komutlar yalnızca ilk rastladıkları **"Linux"** ka-

rakter dizisini “**LINUX**” dizisiyle ve “**linux**” dizisini “**INUX**” dizisiyle değiştirecektir.

Bir satırdaki tüm “**Linux**” karakter dizilerini “**LINUX**” olarak değiştirmek için,

:s/Linux/LINUX/g

komutunu kullanmalısınız.

Şimdi, **s** komutuna birkaç ileri düzey örnek vererek bu komutun başka yeteneklerini de göstermek istiyoruz:

İleri Düzey “s” Komutu Örnekleri	
Komut	Açıklama
:1,100s/Unix/UNIX/g	Dosyanın 1. ve 100. satırları arasında rastlanan tüm “ Unix ” dizilerini “ UNIX ” olarak değiştir.
:1,\$s/Ugur/Ugur Ayfer/g	Dosyanın 1. ve sonuncu satırları arasında rastlanan tüm “ Ugur ”ları “ Ugur Ayfer ” olarak değiştir.
:%s/Ugur/Ugur Ayfer/g	% işareti tüm dosya anlamındadır.
:s/teh/the/gc	Dosyadaki tüm “ teh ”leri “ the ” olarak değiştirir; ancak her bir değişiklik için kullanıcı onayı ister. (c : <i>confirmation</i>)
:g/Ayfer/s/Ugur/U./g	Tüm dosyada “ Ayfer ” dizisini arar (g/Ayfer/); “ Ayfer ” bulunduğu her satırdaki tüm “ Ugur ” dizilerini “ U. ” ile değiştirir.
:%s/Perl/\U&/g	Dosyadaki tüm “ Perl ” dizilerini “ PERL ” ile değiştirir. (\U büyük harfe dönüştürür; & ise, aranan diziye baştan yazmamak için kullanılan bir kısaltmadır.)

:g/^\$/d	Tüm boş satırları bulur ve siler. (^ işaretini satır başı anlamına; \$ ise satır sonu anlamına gelir. Bu örnekte içinde hiçbir karakter olmayan (başı ve sonu yan yana olan) satırlar silinecektir; boşluk karakterleri içeren satırlar bu kalıba uymayacağından silinmeyecektir. Eğer içinde boşluk karakterleri olan satırları da silmek istiyorsanız, :g/^ *\$/d komutunu kullanmalısınız. (^ işaretinden sonraki boşluğa dikkat!)
:g/^/mo0	Çılgın bir örnek! Bir dosyadaki satırların sırasını ters çevirir... (Son satırı birinci satır, sondan ikinci satırı ikinci satır,...). Nasıl mı? Başlangıcı olan her satırı (^) (zaten her satırın bir başlangıcı vardır) sıfıncı satıra taşır. (mo , move anlamındadır) Bu kısacık komutun bu işi yapacağına inanmıyorsanız bir deneyin.

Metnin İçinde Satır Numaralarını Kullanarak Dolaşmak

vi editörü, normalde ekranda satır numaralarını göstermez. Eğer metnin içinde satır numaralarını kullanarak dolaşmanız gerekiyorsa (40. satıra git, sonra 75. satıra git, vs.), ekranda satır numaralarını görmek çok yararlı olacaktır.

Satır numaralarını ekranda görebilmek için:

:set number

Sıra numarasını bildiğiniz bir satıra gitmek için kullanacağınız komut:

nnG (nn numaralı satıra git, örneğin **55G**, **134G**) olacaktır.

“:55” ve “:134” komutları da imleci sırasıyla **55.** ve **134.** satırların başına götürecektir.

```

1 - Rüyalarınız 256 renkse
2
3 - "0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, ..." diye sayıyorsanız
4
5 - Uyunaya çalışırken "sleep 8*3600" diye düşünüyorsanız
6
7 - Asansöre bindiğinizde gitmek istediğiniz katın düğmesini çift
8   tıklıyorsanız
9
10 biraz dikkatli olmanız iyi olur...
11
12 █

```

Tuş Kısaltmaları

Metninizi yazarken bazı kelime ya da kalıpları çok sık tekrarlamamız gerekebilir. Örneğin, metninizin birçok yerinde “Aircraft Owners and Pilots Association” adlı organizasyonun adını yazmanız gerekecekse,

:ab aopa Aircraft Owners and Pilots Association

komutuyla bir kısaltma (*abbreviation*) tanımlayabilirsiniz. Artık, klavyeden her **“aopa”** yazdığınızda, sanki açık açık “Aircraft Owners and Pilots Association” yazmışsınız gibi kabul edilecektir. Bu kısaltmanın iptal edilmesini istediğiniz zaman;

:unab aopa (*unabbreviate*)

komutu yetecektir. (**vi**'dan çıktığınızda kısaltmalarınız yok olur.)

ab komutu ile yapılan tuş tanımlamaları, sadece “insert” modunda, yani araya metin girme konumunda anlamlıdır. (Tanımlamış olduğunuz bir kısaltmayı kullanmadan önce **i**, **a**, **A**, **o** veya **O** komutlarından birini vermiş olmanız gerekir).



Kim Korkar LINUX'tan?

Eğer sık kullandığınız **vi** komutlarına ilişkin bir kısaltma tanımlamak istiyorsanız, bu tanımlamanızı **map** komutuyla yapmanız gerekir. Örneğin,

map ^Y dd

tanımlaması, Ctrl-Y tuşunun, bulunduğunuz satırın silinmesini sağlayan **dd** komutu ile eş anlamlı olarak kullanılmasını sağlar.

Eğer **map** komutu ile F2 fonksiyon tuşuna “tüm boş satırları silme” komutunu tanımlamak isterseniz,

map #2 :g/^\$/d

komutunu kullanabilirsiniz. **map** ile yapılmış tanımlamaları iptal etmek için **unmap** komutunu kullanmalısınız. (**unmap #2** gibi)

Bu tip kısaltmalarınızın kalıcı olmasını istiyorsanız **.vimrc** dosyası ile ilgili bölümü okuyunuz.

vi Başlatırken Verebileceğiniz Komutlar

Çok önemli olmamakla birlikte, editörü başlatırken komut satırından verebileceğiniz birkaç **vi** komutu vardır.

vi +230 mektup1.mail	mektup1.mail dosyasını aç ve imleci 230. satıra götür.
vi + telefonlar	telefonlar dosyasını aç ve imleci son satıra götür.

Dosya İşlemleriyle İlgili Komutlar

vi editörü ile yarattığınız veya üzerinde çalışarak değişiklikler yaptığınız dosyaları diske kaydetmek ve buna benzer işlemler için kullanılan komutlar aşağıda sıralanmıştır.

Bu komutları kullanmadan önce komut durumuna geçmiş olmanız gerekmektedir. Komut durumuna geçmek için en az bir kez Esc tuşuna basmanız gerektiğini hatırlıyorsunuz, değil mi?

Dosya İşlemleriyle İlgili Komutlar	
Komut	Açıklama
zz	Dosyayı son haliyle diske kaydet ve vi 'dan çık. (:wq komut dizisine eşdeğerdir.)
:q	Dosyada değişiklik yapılmayacak, vi 'dan çık. (<i>quit</i>)
:q!	Yapılan değişikliklerden vazgeçildi, dosyayı değiştirmeden vi 'dan çık. (<i>quit</i>)
:w	Dosyayı diske kaydet. (vi 'da kal) (<i>write</i>)
:wq	Dosyayı diske kaydet ve vi 'dan çık. (<i>write-quit</i>)
:x	vi 'dan çık, değişmişse dosyayı diske kaydet. (<i>exit</i>)
:w dosya2	Üzerinde çalışılmakta olan dosyayı, " dosya2 " adıyla diske kaydet. (<i>write</i>)
:1,100w bolum1	Üzerinde çalışılmakta olan dosyanın ilk 100 satırını " bolum1 " isimli bir dosyaya kaydet.
:r dosya3	" dosya3 " adlı dosyayı oku ve imlecin bulunduğu noktadan başlayarak araya ekle. (<i>read</i>)

vi İçinden UNIX Komutu Vermek

Bazen, **vi** programıyla bir dosya üzerinde çalışırken, geçici olarak kabuğunuza dönüp, başka bir UNIX komutu çalıştırmanız gerekebilir. Diyelim ki, üzerinde çalışmakta olduğunuz dosyanın içine, bir başka dosyayı kopyalamanız gerekti, ama bu dosyanın tam adını hatırlayamadınız. **ls** komutunu bir kullanabilerseniz, bu dosyanın adını hemen hatırlayacaksınız. Böyle durumlar için, **vi**, size kabuk programınıza bir çıkış olanağı vermektedir. Bu olanaktan yararlanabilmek için Esc tuşu ile komut durumuna geçip,

!ls

komutunu veriniz.

UNIX'in zerafetine bir örnek: Diyelim ki, bir program için kullanım kılavuzu yazıyorsunuz ve kılavuzunuzun bir bölümüne, söz konusu programın bir çıktısını eklemek istiyorsunuz.

Kullanabileceğiniz komut,

:r !prog

Bu komutu verdiğinizde, **prog** isimli program çalıştırılacak ve çıktısı **vi** ile düzenlenmekte olduğunuz dosyada, imlecin bulunduğu noktaya yerleştirilecektir. (Sanki diskten bir dosya okumuşsunuz gibi...)

Birden Fazla Dosyayı Peşpeşe İşlemek

vi programını başlatırken, dosya adı olarak birden fazla parametre verebilirsiniz. Örneğin,

vi dosya1 dosya2 veya

vi dosya*

Bu durumda, **vi** önce **dosya1** isimli dosyayı düzenlenmek üzere ekrana getirecektir. Bu dosyayla işiniz bitip de,

w komutuyla birinci dosyayı (**dosya1**) kaydettikten sonra (write)

n komutuyla ikinci dosyaya (**dosya2**) geçebilirsiniz. (next)

.vimrc Dosyası

.vimrc dosyası, **vi** programıyla ilgili özel tercihlerinizi belirttiğiniz dosyadır. Eğer kişisel dizininizde **.vimrc** isimli bir dosya (dosya adının başındaki noktaya dikkat ediniz) varsa, **vi** programını her başlattığınızda, bu dosyanın içindeki **vi** komutları otomatik olarak çalıştırılacak ve böylece tercihleriniz ayarlanmış olacaktır. **.vimrc** dosyası basit bir text dosyası olup, **vi** dahil her türlü editörle yaratılabilir. Bu dosyada yer alabilecek bazı **vi** komutlarına örnekler vermek istiyoruz.

map ^Y dd	Ctrl-Y “satır sil” anlamında.
map #2 :g/^\$/d	F2 tuşu, “boş satırları sil” anlamında.
ab cua Can Uğur Ayfer	“cua” bir kısaltma olarak kullanılmış.

Daha Ayrıntılı Bilgi İçin...

vi programı hakkında daha yazılabilecek çok şey var. Eğer ilginizi çekiyorsa, O'Reilly Media'nın *Learning the vi Editör* (yazarı Linda Lamb; ISBN 0-937175-67-6) isimli kitabını hararetle tavsiye ederiz.

vi Komutları Özeti			
Komut	Açıklama	Komut	Açıklama
A	Satır sonuna eklemeye başla	x	İmlecin bulunduğu karakteri sil
a	İmlecin sağına eklemeye başla	dd	İmlecin olduğu satırı sil
I	Satır başına eklemeye başla	d3	İmlecin olduğu yerden 3 karakter sil
i	İmlecin soluna eklemeye başla	d\$	İmleçten sonrasını sil
O	Bu satırın üstüne satır ekle	u	Son değişikliği iptal et (Undo)
o	Bu satırın altına satır ekle	U	Satır için Undo
b	Bir önceki kelimenin başına git	r	Tek karakter değiştir
w	Bir sonraki kelimenin başına git	cw	Kelime değiştir
e	Bir sonraki kelimenin sonuna git	cc	Satırın tamamını değiştir
\$	Satır sonuna git	^	Satır başına git
H	Ekranın başına git	L	Ekranın sonuna git
^F	Sayfa ileri (Ctrl-F)	^B	Sayfa geri (Ctrl-B)
S	Satırın tamamını değiştir	:w	Dosyayı kaydet (w yeni-isim de olabilir)
/dizi	Dosyada " dizi "yi bul (ileriye doğru)	:q	Çık (Quit)
?dizi	Dosyada " dizi "yi bul (geriye doğru)	zz	Kaydet ve çık
:nu	Satırları numarala	:q!	Değişikliklerden vazgeç ve çık
	Blok taşımak için: <ul style="list-style-type: none"> • Blok başına gidiniz. • Örneğin 8dd ile sekiz satır siliniz • Taşıyacağınız yere gidip, p komutunu veriniz. 		Blok kopyalamak için: <ul style="list-style-type: none"> • Blok başına gidiniz. • Örneğin 8yy ile tampon belleğe sekiz satır alınız • Kopyalayacağınız yere gidip, p komutunu veriniz.

Bu sayfanın fotokopisini çekinmeden çekebilirsiniz. Dava açmayız...

gvim


Eğer **vi**'in gücünden yararlanmak, ama grafik ortamda çalışıyor olmanın avantajlarını da kaybetmemek istiyorsanız, **gvim** isimli mükemmel bir editör daha var. Bu editör **vi**'in bütün özelliklerini korurken aynı zamanda fare kullanımını da destekleyerek görsel birçok yenilik getiriyor.

gvim, “Graphical vi Improved” (görsel ve geliştirilmiş vi) sözcüklerinden elde edilmiş bir kısaltmadır. Terminal ekranınızdan **gvim** yazarak çalıştırabileceğiniz gibi bu editörü KDE menüsünde Applications – Editors altında da bulabilirsiniz.

Bütün **vi** komutları **gvim**'de de aynen çalışır. Ek olarak farenizi kullanarak imlecin yerini değiştirebilir ve metin üzerinde değişiklikler yapabilirsiniz. Kes-yapıştır türü hareketler bu sayede daha rahat olur.

Eğer 3 tuşlu bir fareniz varsa X kullanırken pencereler içinde veya arasında metin kopyalamak çok kolaydır. Fareyle seçtiğiniz her metin parçası otomatik olarak kopyalanır ve farenizin orta butonunu tıkladığınızda imlecin bulunduğu yere yapıştırılır. Ayrıca Ctrl-C, Ctrl-V gerekmez. Eğer KDE kullanıyorsanız Klipper programı da kes-yapıştır işlemlerine bir sürü ek özellik ekler. Eğer üç tuşlu bir fareniz yoksa, genellikle iki tuşa birden basarak aynı sonuca ulaşabilirsiniz. Fare ayarlarındaki “emulate 3 buttons” seçeneği bunun içindir.



Ekranınızın sağ altındaki  ikonuna tıkladığınızda önceden kopyaladığınız 10 metin parçasına ulaşabilirsiniz. (Tabii ki bu sayıyı ve Klipper'in daha birçok özelliğini değiştirebilirsiniz.)

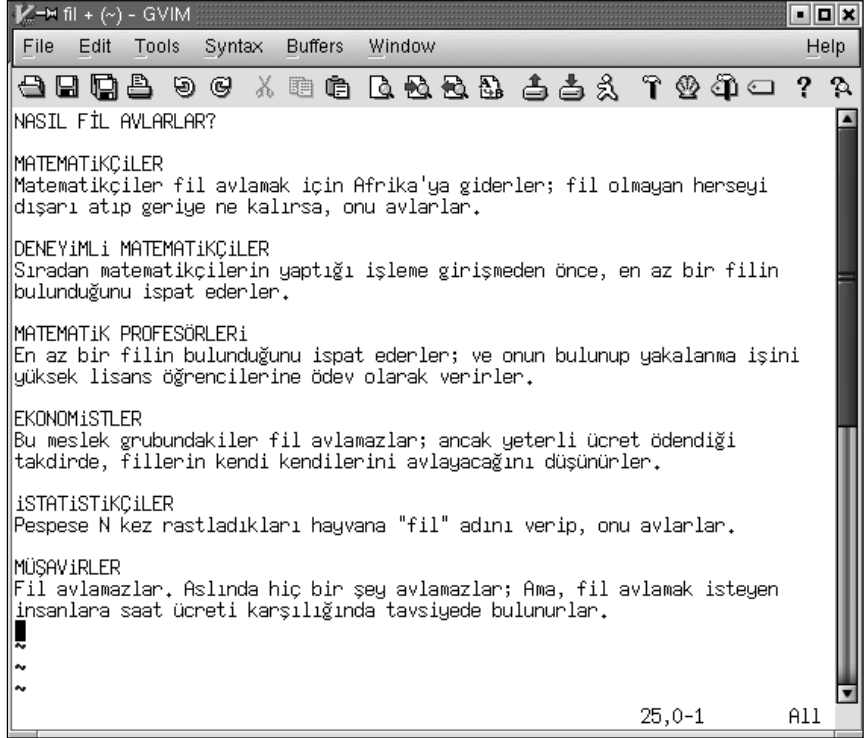
gvim'in en hoş özelliklerinden biri, aklınıza gelebilecek her türlü programlama diline uygun yazım kuralı seçilebilme özelliğidir. Eğer geleneksel dosya isimleri kullanıyorsanız **gvim** kullanarak açtığınız dosyalar zaten belirtilen dosya uzantısına uygun dilin yazım kurallarına göre düzenlemeler yapacak şekilde açılacaktır.

Örneğin adı “.c” diye biten bir dosyayı açtığınızda **gvim** dosyanızı otomatik olarak C dilinin yazım kurallarına göre renklendirecektir. Böylece çeşitli parantez, tırnak gibi açıldıysa kapatılması gereken gramer unsurları kolay al-

Kim Korkar LINUX'tan?

gılanabilecek şekilde ekranınıza gelecektir. **gvim** programının "Syntax" menüsünü kullanarak bu renklendirmenin hangi programlama diline göre yapılacağını seçebilirsiniz.

vi'dan farklı olarak gvim kullanırken karşınızda bir yardım menüsü vardır. **gvim** ile ilgili herşeyi bu yardım menüsünden ulaşacağınız dökümanların içinde bulabilirsiniz.



Önemli LINUX Kavram ve Komutları

05

- Önemli LINUX Kavramları
 - Standart Giriş ve Standart Çıkış
 - Çekirdek: Kernel
 - Dosya Sistemleri
 - Süreçler
 - Link Kavramı ve ln Komutu
 - “Pipe” Kavramı
- Biraz Nefes Alalım
 - Kullanışlı LINUX Komutları

Önemli LINUX Kavramları

Standart Giriş ve Standart Çıkış

“Standart Giriş” ve “Standart Çıkış”, LINUX işletim sisteminin çok önemli iki kavramıdır.

LINUX konsol komutlarının yüzde doksanı, işlevlerini standart giriş biriminden okuyacakları veriler üzerinde yerine getirip, varsa sonuçlarını standart çıkış birimine gönderir. Bir başka deyişle, LINUX komutlarının yüzde doksanı, görevlerini klavyeden okuyacakları veriler üzerinde yerine getirip, varsa sonuçlarını ekrana gönderir.

Standart giriş birimine LINUX terminolojisinde “**STDIN**”, standart çıkış birimine de “**STDOUT**” denir.

Örneğin **wc** komutu (*word count*) standart girişten okuyacağı verilerdeki karakter, sözcük ve satırları sayar; bu sonuçları da standart çıkış birimine görüntüler.

```
cayfer@notebook.lojman.bilkent.edu.tr: /home/cayfer - Shell - Konsol <4>
[cayfer@notebook cayfer]$ wc
Birçok insan için bilgisayar apandisit gibidir; sorun çıkarıncaya
kadar varlığını bile farketmezler.
Edward A. Feigenbaum, 1983
6      16      131
[cayfer@notebook cayfer]$
```

wc komutunun çıktısı olan satırda 3 sayı göreceksiniz. Birincisi, programa girdi olarak verilen dosyadaki satırların sayısı; ikincisi sözcüklerin sayısı; sonuncusu da karakterlerin sayısıdır. **wc** komutunun biraz daha ayrıntılı olarak anlatımını birkaç sayfa ileride bulacaksınız.



Bir programa standart girişten girilen verilerin sonunu belirtmek için Ctrl-D tuşuna basılır. Bir başka deyişle klavyenin “dosya sonu” Ctrl-D ile belirtilir. Bu nedenle yukardaki örneği denerken **wc** programına karakterlerini, sözcüklerini ve satırlarını sayması için girdiğiniz satırlar bitince imleç satır başındayken Ctrl-D tuşuna bir kez basmalısınız. Sadece bir kez basmanız önemlidir. İki kez basarsanız kabuk programınıza da standart girişin sonuna geldiğinizi belirtmiş olursunuz ki bu kabuk programınızın işini bitirdiğini zannedip sona ermesine ve telnet penceresinin kapanmasına yol açar. Eğer yanlışlıkla basılan Ctrl-D tuşunun **bash** kabuğunu sonlandırmamasını; onun yerine açıkça **exit** komutunun kullanılması gerekmesini istiyorsanız örneğin

```
export IGNOREEOF=8
```

gibi bir komutla **bash** kabuğunun ancak peşpeşe sekiz tane Ctrl-D basılması durumunda kendini öldürmesini belirtebilirsiniz.

Elbette bu komutu kişisel dizininizdeki **.bashrc** dosyasına yerleştirerek sisteme her bağlanışınızda, bir başka deyişle sizin için **bash** kabuğunun her başlatılışında bu komutun otomatik olarak çalışmasını sağlayabilirsiniz. Eğer **IGNOREEOF** ortam değişkeninin tüm kullanıcılar için “8” değerini almasını istiyorsanız “**export IGNOREEOF=8**” komutunu **/etc/bashrc** dosyasına yerleştirebilirsiniz.

Giriş ve Çıkışı Yönlendirmek

Çalıştırıldığında ürettiği çıktıları standart çıktıya (ekrana) yazan bir komut veya uygulama programının çıktılarını saklamak isterseniz standart çıktıyı bir disk dosyasına yönlendirebilirsiniz. Bunun için programı çalıştıran komut satırının sonuna küçük bir ekleme yapmanız gerekecektir:

```
ls -al /tmp > dosya_adi
```

Yukardaki komutta “**ls -al /tmp**” komutu çalıştırılmadan önce standart çıktı, diskteki çalışma dizininde yer alacak **dosya_adi** isimli dosyaya yönlendirilecektir. Bu durumda **ls** komutunun üreteceği tüm çıktılar **dosya_adi** isimli dosyaya kaydedilecektir. Çalışma dizininde **dosya_adi** isimli bir dosya zaten varsa ve erişim yetkileri uygunsa **ls** programının çıktısı bu dosyadaki kayıtların üzerine kaydedilecektir, yani dosyanın eski içeriği kaybolacaktır.

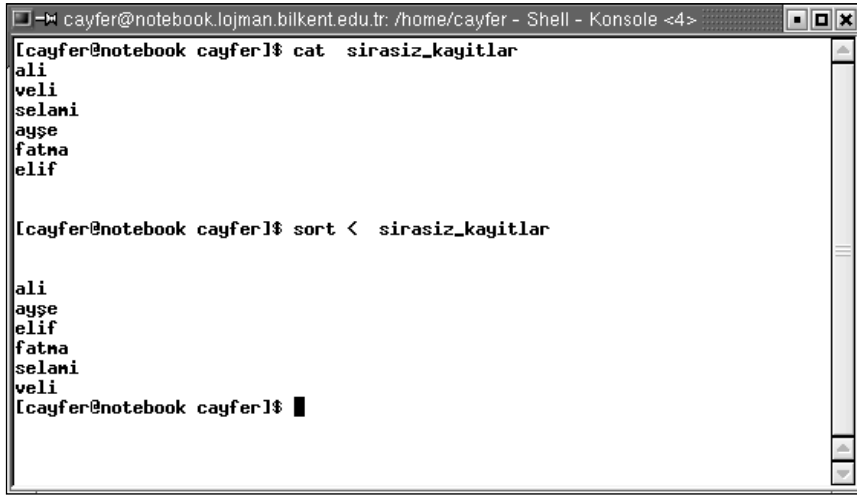
Bazı durumlarda, standart çıktının yönlendirileceği dosyada bulunan eski kayıtları bozmadan yenilerini bunların arkasına eklemek isteyebilirsiniz. Bu durumda yönlendirmeyi “**>>**” ile yapmanız yeterli olacaktır.

```
ls -al /tmp >> dosya_adi
```

Çalışmak için gereksinim duyduğu verileri standart girişten (klavyeden) okumak üzere yazılmış bir programın, söz konusu verileri diskteki bir dosyadan almasını sağlamak için standart giriş yönlendirmesi yapmalısınız. Örneğin,

```
sort < sirasiz_kayitlar
```

komutu, sıralanacak satırları diskteki “**sirasiz_kayitlar**” isimli dosyadan alacaktır. Verilerini standart girişten okuyup, çıktısını standart çıkış birimine gönderen programlara **filtre** programlar denir. **sort** komutu bu filtre programlara güzel bir örnek oluşturur çünkü program standart girişten gelen kayıtları sıralayıp standart çıkışa gönderir.



```
cayfer@notebook.lojman.bilkent.edu.tr: /home/cayfer - Shell - Konsole <4>
[cayfer@notebook cayfer]$ cat sirasiz_kayitlar
ali
veli
selami
ayşe
fatma
elif

[cayfer@notebook cayfer]$ sort < sirasiz_kayitlar

ali
ayşe
elif
fatma
selami
veli
[cayfer@notebook cayfer]$
```

Giriş yönlendirme ve çıkış yönlendirmeyi birlikte kullanabilirsiniz. Örneğin **sirasiz** isimli dosyadaki satırları sıralayıp, sıralanmış satırları **sirali** isimli bir dosyaya kaydetmek için aşağıdaki komutlardan birini kullanabilirsiniz:

```
sort < /tmp/sirasiz > /tmp/sirali
sort > /tmp/sirali < /tmp/sirasiz
```

STDERR

UNIX programcılarının geleneksel yaklaşımlarından biri, yazdıkları programların hata mesajlarının standart çıktıda değil, “standart hata”da (yani STDERR’de) belirecek şekilde kod geliştirmeleridir. Aksi belirtilmedikçe STDERR, “ekran” ortamıdır; işte bu yüzden hata mesajlarını da ekranda görürsünüz.

STDOUT ve STDERR ayırımı, gerektiğinde hata mesajlarının ayrı bir ortama yönlendirilmesini sağlar. Örneğin normal bir kullanıcı olarak (yani “**root**” olmadan)

```
du -s /
```

komutunu vererseniz, ekranda

```
[cayfer@cayfer /]$ du -s /
du: cannot change to directory `/bcc/mail': Permission denied
du: cannot change to directory `/bcc/bcc2': Permission denied
du: cannot change to directory `/etc/cups/ssl': Permission denied
du: `/etc/cups/certs': Permission denied
du: cannot change to directory `/etc/skel/tmp': Permission denied
du: cannot change to directory `/etc/uucp': Permission denied
```

gibi hata mesajları alırsınız. Bu mesajlar aslında STDOUT'a değil, STDERR'e gönderilmektedir. Nitekim **du** komutunu, çıktısını **/tmp/du_raporu** diye bir dosyaya yönlendirmek üzere

```
du -s / > /tmp/du_raporu
```

şeklinde verseniz bile bu hata mesajları **du_raporu** dosyasına değil, gene ekrana görüntülenecektir. **/tmp/du_raporu** dosyasına yalnızca diskte kapladığı alan başarıyla hesaplanabilen dizinlere ilişkin satırlar yönlendirilecektir.

Böylece; özellikle geri planda çalışan işlerin hata mesajlarıyla terminal penceresini kirlitmelerini önlemek ve kaybolmasın diye hata mesajlarını ayrı bir dosyada biriktirmek için STDERR'i bir dosyaya yönlendirmek mümkündür:

```
du -s / > /tmp/rapor 2> /tmp/hatalar &
```

Yukarıdaki komuttaki ">" karakteri STDOUT'u yönlendirmek istediğinizi; "2>" karakterleri ise STDERR'i yönlendirmek istediğinizi; en sondaki "&" ise programı geri planda çalıştırmak istediğinizi belirtmektedir.

Standart Giriş ve Standart Çıkış kavramlarının yanı sıra, LINUX işletim sisteminde, iyi kavranması gereken birkaç önemli kavram daha var. Bu kavramlar, ilk okuduğunuzda çok karışık ya da anlaşılmaz gelebilir; ama LINUX sistem yöneticisi olma yolunda ilerlemeyi düşünüyorsanız lütfen dikkatlice okuyunuz, gerek duyarsanız başka kaynaklara başvurunuz ama bu kavramları anlamadan geçmeyiniz.

Bu kavramlar:

- Çekirdek... İngilizce, daha doğrusu UNIX'çesiyle "kernel",
- Dosya Sistemleri; UNIX'çesiyle "file systems",
- Süreçler; UNIX'çesiyle "process",

- Bağlantılar; UNIX'çesiyle “links”,
- “pipe” kavramı.

Çekirdek: Kernel

Modern işletim sistemleri, tüm işlevlerini yerine getirecek modülleriyle birlikte belleğe yüklenmezler. Bunun en önemli nedeni bellekten tasarruf edebilmektir. İşletim sistemi ne kadar az bellek kullanırsa uygulama programlarına o kadar fazla boş bellek kalacaktır. Bu nedenle modern işletim sistemleri, bellekte kalması kaçınılmaz ve “kernel” adı verilen “çekirdek” modüller ve bunun etrafında gerektikçe belleğe yüklenen yan modüller topluluğu olarak geliştirilir.

LINUX çekirdeğinin temel görevleri bellek ve süreçleri denetlemek ve donanım birimlerini yönetmektir. Örneğin, çekirdek, bilgisayara takılı olan IDE disk sürücülerin tüm teknik ayrıntılarının farkındadır. Dönme hızından yazıcı kafanın nasıl hareket ettirileceğine kadar ince ayrıntıları bilir. Benzeri şekilde ses arabirimlerini, Ethernet arabirimlerini tanıyıp bunları denetleyebilen yazılımlar birer çekirdek modülüdür. UNIX ve dolayısıyla LINUX çekirdeklerinin en önemli özelliği, tanıdıkları bu çevre birimlerini diğer işletim sistemi modüllerine ve uygulama programlarına birer “özel dosyaymış” gibi göstermeleridir. UNIX'in 30 yılı aşkın bir süredir bilgisayar dünyasında başarıyla yer alabilmesinin en önemli nedeni çekirdeğin bu özelliğidir.

Çekirdek için IDE elektronik arabirim standardında üretilmiş, 7200 rpm hızında dönen, 6 plakası ve 12 okuma-yazma kafası olan bir disk, çekirdek dışında kalan modüller için **/dev** dizini altında yer alan **hda** isimli bir özel dosyadır (*node*). Benzer şekilde bilgisayarın birinci disket sürücüsü **/dev/fd0**, birinci Ethernet kartı **/dev/eth0**'dır.

Sistemin açılışı sırasında ilk yapılan işlerden biri çekirdeğin belleğe yüklenmesidir. Çekirdek belleğe yüklenince sistemin denetimini üzerine alacak, açılışı tamamlamak üzere gereken modülleri kendisi yükleyecek; işi biten ya da bekleyebilecek modülleri bellekten atacaktır.

Bazı işletim sistemi işlevlerini yerine getiren yazılım modülleri çekirdekte yer almalıdır. Örneğin, belleğin yetmemesi durumunda disk takas alanlarının bellek gibi kullanılmasını sağlayan modüller her zaman gerçek bellekte (RAM) bulunmalıdır. Eğer bu modüller bellek yetmiyor diye diske atılırsa

bir daha onları gerçek belleğe geri almak mümkün olmaz. Benzeri şekilde diskin nasıl kullanıldığını bilen modül, bellekte yer açmak için diske atılırsa, geri yüklenmesi olanaksız olacaktır.

İşletim sisteminin bazı modülleri de performans açısından her zaman bellekte yer almalıdır. Sistemde yüksek öncelikle çalışan ve hiçbir zaman sanal bellek işlemlerine tabi tutulmayan süreçler genellikle çekirdeğin bir parçası olacak şekilde geliştirilmiştir. Örneğin disk kotası denetimleri, IP paket filtreleme işleri, IP paket yönlendirme işleri olabildiğince yüksek performans gerektiren işler oldukları için çekirdekte yer alan programlarla yapılırlar.

Ancak, her işi çekirdekte halletmek de olası değildir. Başta LINUX olmak üzere, modern işletim sistemleri, tüm modülleriyle birlikte belleğe yerleştirilemeyecek kadar büyüktür. Modern LINUX sürümleri bu sorunu “Loadable Kernel Modules”; yani “yüklenbilir çekirdek modülleri” kavramıyla çözmüştür.

Kendi çekirdeğinizi oluşturmanız için şimdilik bir neden görememekle birlikte LINUX’un kaynak kodlarını yeniden derleyerek kendinize özel bir çekirdek oluşturmanızın mümkün olduğunu belirtmeden geçemeyeceğiz.

Dosya Sistemleri

Microsoft’un MS-DOS/Windows işletim sistemlerinde dosya ve dizinlerin yerini belirtirken A:, C: gibi sürücü isimleri kullanılması tercih edilmiştir. Bir başka deyişle, kullanıcılar kullandıkları bilgisayara bağlı disklerin sayısını ve isimlerini bilmek ve ilgilendikleri dizin ya da dosyaların bu fiziksel ortamlardan hangisinde bulunduğunu bilmek ve gerektiğinde belirtmek zorundadır. LINUX’ta durum biraz, hatta oldukça farklıdır.

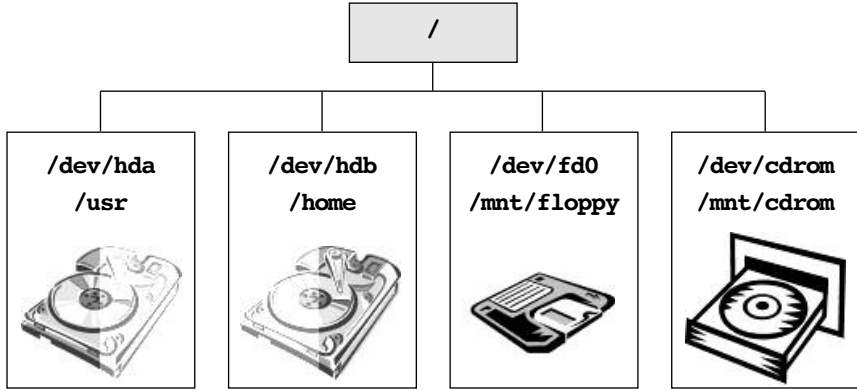
Şimdilik, LINUX bilgisayarınızın tek başına (bir bilgisayar ağına bağlı olmaksızın) çalışan bir bilgisayar olduğunu varsayalım. Bilgisayarınızın iki disk, bir disket, bir de CD-ROM sürücüsü olsun.

Disklerinizden ilki en az 2 parçaya (partition) bölünmüştür. (Bir bölüm LINUX işletim sistemi için, ikincisi de takas alanı için; hatırladınız mı?) İkinci diskiniz ise büyük olasılıkla tek parçadır.

Yukarıdaki varsayımlarımıza göre bilgisayarınızın beş diski varmış gibi düşünebilirsiniz. (İki parçaya ayrılmış birinci disk ve tek parça olan ikinci disk, disket sürücü ve CD-ROM sürücü.) MS-DOS ya da Windows kullanıyor olsaydınız, bu disklere A:, C:, D:, E: ve F: isimleriyle erişirdiniz.

Şimdi sıkı durun: LINUX kullanıcılarının, disklerin ne şekilde ayrılmış olduğundan, hatta bilgisayarda kaç disk sürücüsü bulunduğu haberi olması bile gerekmemektedir. LINUX'ta tüm diskler ve disk bölümleri (*partition*), disket sürücüler ve CD-ROM sürücüler, **/dev** dizinin altında birer **alt dizin** olarak yer alır.

Şematik olarak göstermek gerekirse:



Her disk ve disk parçası üzerinde diğerlerinden bağımsız bir dosya sistemi (*file system*) bulunmalıdır. Dosya sistemi, doğrudan erişimli bir veri saklama biriminde (disk, disket, CD-ROM gibi) dolu ve boş alanların yönetimini, dizin ve dosyaların yaratılmasını, silinmesini ve en önemlisi bunlara hızlı erişimi sağlayan bir veri yapısı bulundurur. Bu veri yapıları disklere formatlama (bir türlü “biçemleme” demeye alışamadık; zorla değil ya...) sırasında kaydedilir. Özet olarak, dosyaları oluşturan disk alanı bloklarının diskin fiziksel olarak nerelerinde yer aldığı, bu blokların kime ait olduklarının ve erişim haklarının saklandığı veri yapılarına “dosya sistemi” (*file system*) denir.

LINUX'ta disklerinizde kullanabileceğiniz birden fazla dosya sistemi seçeneği vardır. Biz, ext3 adı verilen sistemi kullanmanızı öneririz.

LINUX'ta dosya sistemleri, diskler (daha doğrusu disk bölümleri) üzerinde, formatlama işleminden sonra **mke2fs**, **mkreiserfs** gibi komutlarla yaratılır. (Merak etmeyin; bu işi sistem kurulumu sırasında farketmeden yaptınız bile.) Hatırlarsanız LINUX'unuzu kurarken "root" (/) dizininin hangi disk bölümüne bağlanacağını belirtmişsiniz. (/dev/hda6 gibi) İşte hiyerarşik LINUX dosya yapısının en tepe noktası bu "root" dizinidir. Sisteminizdeki tüm disk bölümleri, diskler, disket sürücüler, CD-ROM sürücüler, hatta başka bilgisayarlar üzerinde erişebileceğiniz disk/dizinler hep bu "root" dizinin altında "alt dizinler" olarak görünür.

Örneğin, /home dizini (kullanıcıların kişisel dizinlerinin yer aldığı dizin) bilgisayarınızın ikinci diski üzerinde olabileceği gibi / disk bölümünde yer alan gerçek bir alt dizin de olabilir. Özellikle merak edip bakmadıkça bir dizinin hangi diskte yer aldığını göremezsiniz. Yani **ls** komutu size bir dizinin hangi disk bölümünde ya da sürücüde yer aldığını söylemez.

Bilgisayarınızda disk bölümlerinin ve disket sürücü gibi çevre birimlerinin hangi dizinler altında erişilebilir olduğunu görmek için **mount** komutunu kullanabilirsiniz.

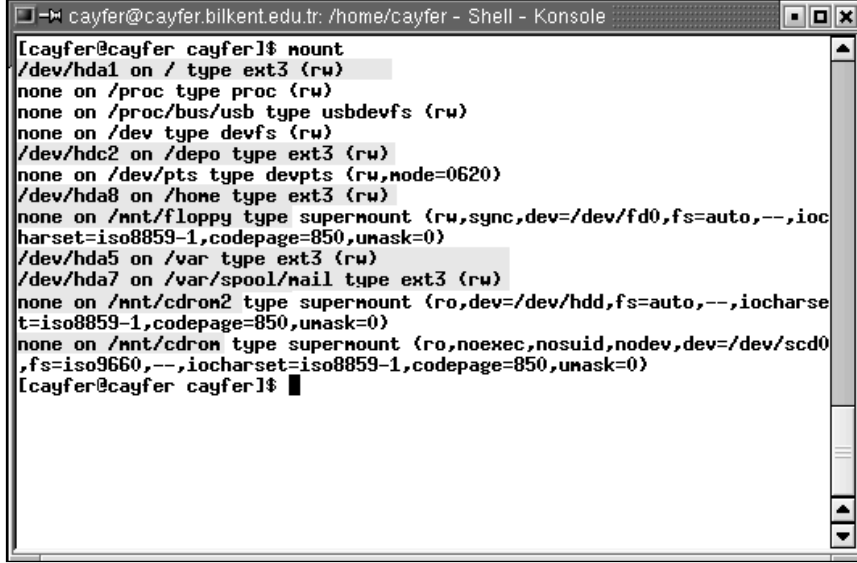
Örneğin, tek diskli ve bu diskte hem LINUX hem Windows barındıran bilgisayarda verilen **mount** komutu aşağıdaki gibi bir rapor üretir:

```

cayfer@notebook.lojman.bilkent.edu.tr: /home/cayfer - Shell - Konsol <4>
[cayfer@notebook cayfer]$ mount
/dev/hda6 on / type ext3 (rw,noatime)
none on /proc type proc (rw)
none on /proc/bus/usb type usbdevfs (rw)
none on /dev type devfs (rw)
none on /dev/pts type devpts (rw,node=0620)
none on /dev/shm type tmpfs (rw)
/nnt/cdrom on /nnt/cdrom type supernmount (ro,dev=/dev/hdc,fs=iso9660,--,iocharset=iso8859-1)
/nnt/floppy on /nnt/floppy type supernmount (rw,sync,dev=/dev/fd0,fs=vfat,--,iocharset=iso8859-1,unask=0,codepage=850)
/dev/hda1 on /nnt/windows type vfat (rw,iocharset=iso8859-1,unask=0,codepage=850)
[cayfer@notebook cayfer]$

```

Aynı komutu daha karmaşık disk yapısı olan, örneğin birden fazla diski olan ve bu diskleri de ayrıca **bölümlendirilmiş** olan bir bilgisayarda verirseniz alacağınız rapor aşağıdakine benzer şekilde olacaktır:



```
[cayfer@cayfer cayfer]$ mount
/dev/hda1 on / type ext3 (rw)
none on /proc type proc (rw)
none on /proc/bus/usb type usbdevfs (rw)
none on /dev type devfs (rw)
/dev/hdc2 on /depo type ext3 (rw)
none on /dev/pts type devpts (rw,node=0620)
/dev/hda8 on /home type ext3 (rw)
none on /mnt/floppy type supernmount (rw, sync, dev=/dev/fd0, fs=auto, --, iocharset=iso8859-1, codepage=850, unask=0)
/dev/hda5 on /var type ext3 (rw)
/dev/hda7 on /var/spool/nail type ext3 (rw)
none on /mnt/cdrom2 type supernmount (ro, dev=/dev/hdd, fs=auto, --, iocharset=iso8859-1, codepage=850, unask=0)
none on /mnt/cdrom type supernmount (ro, noexec, nosuid, nodev, dev=/dev/scd0, fs=iso9660, --, iocharset=iso8859-1, codepage=850, unask=0)
[cayfer@cayfer cayfer]$
```

Evet, biliyoruz! Feci bir görüntü ama zamanla alışılıyor. LINUX sistem yöneticisi olmak kolay değil. Şaka bir tarafa; bu işlerin çoğunu KDE veya GNOME altında çalıştırabileceğiniz yönetim araçlarıyla da yapabilirsiniz. Ancak LINUX'u konsol komutlarıyla kullanabilecek şekilde öğrenmek sizin için çok daha yararlı olacaktır. LINUX dışında UNIX makinelerin başına geçtiğinizde de iş yapabilmeniz için KDE'siz, GNOME'suz yaşamayı öğrenmelisiniz.

Yararlı bilgiler içeren satırları griye boyadık. Diğer satırlar yararsız değil elbette ama şimdilik o satırları anlamaya çalışmak için erken.

Bu listeyi dikkatlice incelediğinizde, bilgisayarda sadece iki fiziksel disk bulunduğunu (sadece **/dev/hda** ve **/dev/hdc** serisi disklerin adı geçiyor) ve bu disklerin birinci IDE kanalının ilk diski (Primary master, **hda**) ve ikinci IDE kanalının da gene ilk diski (Secondary Master, **hdc**) olduğunu göreceksiniz. Bu arada şunu da belirtmek gerekir: Bu bilgisayarda başka diskler de takılı olabilirdi; örneğin **/dev/hdd** diye bir disk de olabilirdi.

```
/dev/hda1 on / type ext3 (rw)
```

satır, bilgisayarın birinci IDE kanalının ilk diskinin (**hda**) bir numaralı bölümünün (**hda1**) bu sistemin root dizini olarak kullanıldığını gösteriyor. Ayrıca bu disk bölümünün okunabilir ve yazılabilir (**rw**) durumda olduğunu ve **ext3** dosya sistemi formatında yaratıldığını gösteriyor.

```
/dev/hdc2 on /depo type ext3 (rw)
```

satır, bilgisayarın ikinci IDE kanalının ilk diskinin (**hdc**) iki numaralı bölümünün (**hdc2**) bu sistemin “/” dizini altındaki **depo** isimli bir dizine bağlandığını gösteriyor. Yani bu diske bakmak isteyenler **/depo** dizinine bakmalılar. Ayrıca bu disk bölümü üzerindeki dosya sisteminin **ext3** olduğu ve şu anda okunabilir/yazılabilir durumda olduğunu gösteriyor. Tabii yetkisi olanlar için...

```
none on /mnt/cdrom type supermount (rw,dev=/dev/cdrom,fs=iso9660)
```

satır, bilgisayarın CD sürücüsünün (**/dev/cdrom**) sistemin **/mnt** dizinin altında **cdrom** isimli bir dizine (**/mnt/cdrom**) bağlandığını gösteriyor.

En baştaki “**none**”, CD sürücünde şu anda bir CD bulunmadığını belli ediyor. Ayrıca bu dizine **iso9660** dosya sistemine (standart CD dosya sistemidir) sahip CD’lerin takılabileceği ve bu CD’lerin oku/yaz kullanılabileceğini gösteriyor; yani **/mnt/cdrom**, yazılabilir CD’leri de destekliyor. “**supermount**” sözcüğü bu sürücüye bir CD takıldığında LINUX çekirdeğinin bu CD’yi belirtilen dizine otomatik olarak bağlayacağını gösteriyor. Sürücüye bir CD takıp içine bakmak isteyen birisi **/mnt/cdrom** dizinine bakmalıdır.

Benzer bir mantık disket sürücüler için de kullanılır.

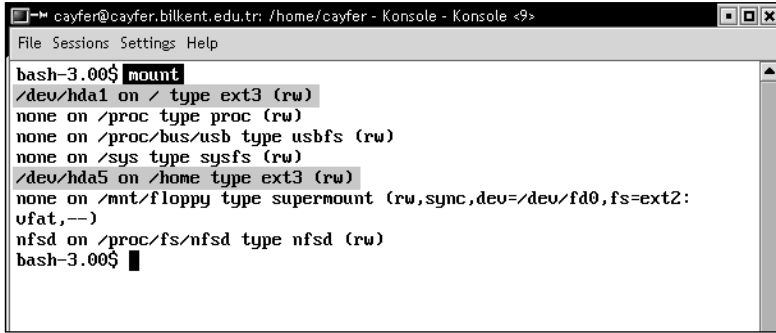
```
none on /mnt/floppy type supermount (rw,dev=/dev/floppy,fs=vfat)
```

satır, bilgisayarın disket sürücüsünün (**/dev/floppy**) sistemin **/mnt** dizini altındaki **floppy** isimli bir dizine (**/mnt/floppy**) bağlandığını gösteriyor. Yani disket takıp içine bakmak isteyen birisi **/mnt/floppy** dizinine bakmalıdır. Ayrıca bu dizine **vfat** dosya sistemine (standart MS-DOS disket dosya sistemidir) sahip disketlerin de takılabileceğini gösteriyor. “**supermount**” sözcüğü bu sürücüye bir disket takıldığında LINUX çekirdeğinin bu disketi belirtilen dizine otomatik olarak bağlamaya çalışacağını gösteriyor.

Gene yukardaki rapora göre **hda** diskinin sekiz numaralı bölümü (**hda8**) **/home** dizinine, beş numaralı bölümü **/var** dizinine; yedi numaralı bölümü de **/var/spool/mail** dizinine bağlanmıştır.

Üzerinde bir dosya sistemi olan bir disk birimine veya bölümüne okuma veya yazma amacıyla ulaşabilmeniz için, o dosya yapısının, “/” dosya yapınızda bir yerlerdeki bir alt dizine “mount edilmiş” (iliştirilmiş) olması gerekmektedir. (“/” dizini, bilgisayarın açılması sırasında otomatik olarak iliştirilmektedir. Eğer bu / dizini, bilgisayarın açılması aşamasında bağlanamazsa, o bilgisayar zaten açılmaz.)

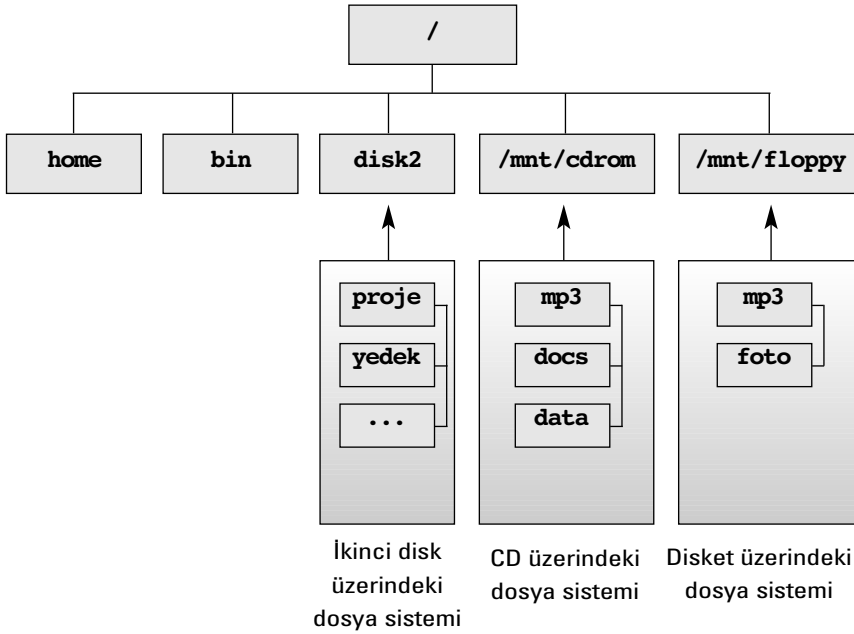
Bir bilgisayarda “mount edilmiş” disk ve disk bölümlerini daha kısa ve anlaşılır şekilde görebilmek için “df” komutunu da kullanabilirsiniz.



```
bash-3.00$ df
/dev/hda1 on / type ext3 (rw)
none on /proc type proc (rw)
none on /proc/bus/usb type usbfs (rw)
none on /sys type sysfs (rw)
/dev/hda5 on /home type ext3 (rw)
none on /mnt/floppy type supermount (rw, sync, dev=/dev/fd0, fs=ext2:
ufat, --)
nfsd on /proc/fs/nfsd type nfsd (rw)
bash-3.00$
```

Sistemin açılışı sırasında çeşitli dizinlere otomatik olarak iliştirilmesi istenen disk bölümleri **/etc/fstab** dosyasında belirtilir. Şimdilik bu dosyayla ilgili bir şey söylemek istemiyoruz; biraz erken. Sisteminize kurulum sırasında yerleştirilen **/etc/fstab** dosyası daha uzun süre işinizi görecektir. Bu dosyanın içeriğini merak ediyorsanız “**more /etc/fstab**” komutuyla görebilirsiniz.

LINUX'taki dosya yapılarını ters duran bir ağaca benzetirsek, dosya sistemlerini iliştirme işlemini, bir ağacı, bir başka ağacın dallarından birine iliştirmek (monte etmek) gibi düşünebilirsiniz.



LINUX işletim sisteminde **mount** komutu yalnızca bağlanmış diskleri listelemek için kullanılmaz. Sistem çalışırken disklerin bağlantısını çözmek, yeni disk ya da disk bölümlerini bağlamak için de kullanılır.

“mount” etmek derken disklerin bilgisayara fiziksel olarak takılıp çıkarılmasından söz etmiyoruz! O işi bilgisayarı kapatmadan yaparsanız başınız derde girer.



mount komutunu bu şekilde kullanabilmek için root kullanıcı yetkilerine sahip olmanız gerekecektir; yani eğer root kullanıcı değilseniz, **mount** komutunu yalnızca parametresiz olarak kullanmanıza izin verilecektir. **mount** komutu hakkında daha ayrıntılı bilgiyi sistem yönetimi ile ilgili bölümlerde bulabilirsiniz.



Disket sürücüler ve CD-ROM sürücülerini de küçük birer disk sürücü olarak düşünmelisiniz; bu nedenle, bu sürücüler kullanılabilmeleri için önce “/” dizini altında bir yereye iliştilmeniz gerekir. Disketler ve CD’ler, takılıp çıkarılabilir birimler olduklarından, bilgisayar açılırken otomatik olarak iliştilmezler (mount edilmezler). Ancak LINUX çekirdeği bu sürücülere bir disket ya da CD takıldığında ve yeni takılan birimi kullanan bir komut verdiğinizde bu birimi otomatik olarak iliştilir (supermount). Bu otomatik iliştilme işleminin başarılı olabilmesi için üzerinde anlamlı bir dosya yapısı olan disket ya da CD takmalısınız. Yani formatsız bir disket ya da boş bir CD takarsanız doğal olarak otomatik iliştilme işlemi başarılı olamayacaktır. Supermount kavramının bir önemi de root yetkileri olmayan kullanıcılarının da sisteme disket ve CD takıp çıkarabilmelerini sağlamaktır. Biliyorsunuz, iliştilme amacıyla mount komutunu yalnızca root kullanabilir.

İş biten disket ve CD’ler (bazen de diskler) **umount** edilmelidir; yani, bu birimlere takılı medyalar üzerindeki dosya sistemlerinin, “/” dosya sistemiyle bağlantısı kesilmelidir. Bir dosya yapısının bağlı bulunduğu dizinle ilişkisini kesmek için:

```
umount /cdrom  
umount /dev/hdc1  
umount /var/spool/mail
```

gibi komutlar kullanılır.

Dikkat ederseniz hangi dosya sisteminin “umount” edileceğini, isterseniz sürücü adıyla (**/dev/hdc1** gibi), isterseniz de bağlı bulunduğu dizin adıyla belirtebiliyorsunuz.

umount komutunu kullanabilmeniz için root kullanıcı olmanız gerekir. Normal kullanıcıların CD-ROM sürücü ve disket sürücülerini çözebilmeleri için **eject** komutunu kullanmaları gerekir.

```
eject cdrom  
eject floppy  
eject
```

Parametresiz kullanıldığında, varsa CDRÖM sürücüdeki CD unmount edilir ve kapağı açılır.

Siz hala sormadınız; bari konuyu biz açalım: **mount** komutunun verdiği listede sistemin kurulması sırasında ayırdığınız takas alanının (örneğin **/dev/hda5**) nereye iliştilildiğini göremiyorsunuz. Bu son derece normal çünkü takas alanı root dahil hiç kimsenin erişemeyeceği bir alandır. Takas alanının yönetimi tamamen LINUX çekirdeğine aittir. Sistem yöneticisinin takas alanıyla ilgili tek denetim şansı takas işlemlerini durdurmak ve başlatmaktır. (**swapon** ve **swapoff** komutları). Ancak bu işi yapmaya da pek gerek olmaz.



fsck

LINUX dosya sistemleri (ister ext2, ister reiserfs olsun) oldukça karmaşık veri yapılarıdır. Özellikle güç kaybı ya da reset düğmesine basılması nedeniyle kapanan sistemlerde bu dosya yapısının bozulma, daha doğrusu tutarlılığının kaybolması olasılığı vardır. Bir dosya sistemi içindeki tutarlılık bozulduğunda, sistem kullanılmaya devam edilirse sorun gittikçe büyür. Bu nedenle LINUX, sistemin düzgün kapatılıp kapatılmadığını izlemek için birtakım mekanizmalar kullanır. Düzgün olmayan bir kapatmadan sonraki ilk açılışın hemen başında dosya sistemleri üzerinde otomatik olarak tutarlılık testi başlatılır ve bu test bir şekilde başarıyla geçilmeden sistem tam olarak açılmaz.

Sorun çıkarma potansiyeli olan dosya sistemleri **fsck** programıyla test edilir. Test sırasında rastlanan sorunların çoğu **fsck** tarafından otomatik olarak düzeltilir. **fsck** yapacağı bir düzeltmenin tehlikeli olabileceğini, yani düzeltmenin başarılı olmama olasılığını hissederse yapacağı değişiklik için izin ister. Genellikle bu izinleri vermek zorundasınızdır.

Dosya sistemlerinin tutarlı olması çok önemli olduğu için LINUX her dosya sistemini belli bir sayıda; örneğin 20 mount edişte bir **fsck** ile kontrol eder. Bu nedenle sisteminizin bazı açılışları alıştığınız süreden uzun sürer. Bir dosya sisteminin **fsck** ile kontrol edilme süresi o diskin büyüklüğüne bağlıdır. 20 GByte ve oldukça dolu bir disk için bu süre 5 dakika kadar sürebilir. Sürenin uzun olması nedeniyle, birden fazla dosya sistemi kontrol edilecekse **fsck** kendi kendinin gereği kadar kopyasını paralel olarak başlatır.

Tipik LINUX bilgisayarların yılda, bilemediniz 3-5 kez kapatılması gerekebileceğini düşünürseniz, bu 20 seferde bir otomatik **fsck** başlatılmasının kullanıcıları ve sistem yöneticisini pek de rahatsız etmeyeceğini belirtmeliyiz.

Süreçler

LINUX işletim sisteminin çok kullanıcı ve çok işli bir işletim sistemi olduğunu şimdiye kadar birkaç kez vurgulamıştık. Burada bir daha açıklamak gerekirse; LINUX işletim sisteminin denetimindeki bir bilgisayar aynı anda birden fazla kullanıcıya hizmet edebilir. Her kullanıcı için birden fazla program çalıştırabilir; ve bu arada geri planda çeşitli servisler çalıştırılarak ağ üzerinden gelen isteklere de yanıt verilebilir. (Veritabanı sunucusu, web sunucusu gibi)

Bir LINUX bilgisayarı üzerinde çalışan işler:

- Kullanıcı programları,
- Servis (sunucu) programları ve
- Sistemin kendi gereksinimi için çalıştırılan programlardan oluşur.

İşletim sistemi, yöneticisinin ve kullanıcılarının belirttiği işler dışında kendi işlerini de bir sürü programı aynı anda çalıştırarak yürütür. Örneğin, **at** komutuyla belli bir tarih ve saatte başlatılması gereken işleri izleyen **atd** programı gibi; bilgisayar ağı üzerinden gelebilecek TCP/IP isteklerini değerlendiren **inetd** programı gibi, belirli aralıklarla disklere yapılan kayıt işlemlerinin fiziksel olarak disklere kaydedilmesi işini düzenleyen (*flushing disk buffers*) **update** programı gibi sistem yazılımları geri planda çalışır. Kimi sürekli çalışır, kimi de gerektiği zaman çalışır, işi bitince durur. Tipik bir LINUX bilgisayarında, kullanıcı programları dışında çok sayıda sistem programı sürekli çalışıyor durumdadır.

Bir LINUX bilgisayarı, belirli bir anda, merkezi işlem birimini (ya da birimlerini) ve belleği paylaşarak birlikte çalışan programlara genel anlamda **süreç** (*process*) adı verilir. Süreç kavramı program kavramından biraz daha değişiktir. Bir program birden fazla süreçten meydana gelebilir; hatta tek bir süreç olarak çalışmakta olan bir program gerek gördüğünde kendisinin bir kopyasını çıkarıp onu da yeni bir süreç olarak çalıştırmaya başlayabilir. Bunun en yaygın örneği web sunumu işini yapan **httpd** (*apache*) ve veritabanı

yönetim/sunum işini yapan **mysqld** yazılımlarıdır. Sistem yöneticisi, sistemin açılışı sırasında bu programlardan birer tane başlatır. Bu programlar, ağ üzerinden istek geldikçe kendi kendilerinin kopyalarını çıkarıp onları da çalıştırmaya başlarlar. (LINUX dilinde “spawn ve fork”)

Süreçlerin Merkezi İşlem Birimi (MİB) zamanını paylaşmaları işletim sisteminin çekirdeği tarafından koordine edilir.

MİB paylaşımına ilişkin önemli bir terim de “zaman dilimi” (*time slice*) kavramıdır. Her süreç, MİB’ni belirli ve kısa bir süre (tipik olarak 10-100 milisaniyelik zaman dilimleri) için kullanabilir. Zaman dilimini dolduran süreçler beklemeye alınıp, MİB, sırada bekleyen bir başka sürece tahsis edilir. Bu şekilde tüm süreçler aynı anda çalışıyormuş gibi bir etki elde edilir. Bu süreçlerin birden fazla kullanıcıya ait olmaları durumunda da, MİB kullanıcılar arasında paylaştırılmış olur. Eğer bilgisayarda birden fazla MİB varsa, LINUX çekirdeği işleri bu MİB’lerine otomatik olarak dağıtabilir. LINUX’un çok kullanıcı olma özelliğinin altında yatan temel mekanizma budur.

Herhangi bir anda, bilgisayarda çalışan süreçlerin neler olduğunu görmek isterseniz,

ps ax

komutunu kullanabilirsiniz.

```

cayfer@notebook.ijman.bilkent.edu.tr: /home/cayfer/public_html - Shell - Konsole
[cayfer@notebook public_html]$ ps ax
  PID TTY          STAT       TIME COMMAND
    1 ?           S            0:04 init
    2 ?           SW           0:00 [keventd]
    3 ?           SWM          0:00 [ksoftirqd_CPU0]
    4 ?           SW           0:04 [kswapd]
    5 ?           SW           0:00 [bdflush]
    6 ?           SW           0:00 [kupdated]
    7 ?           SW<          0:00 [ndrecoveryd]
   11 ?           SW           0:02 [kjournald]
   98 ?           S            0:00 devfsd /dev
  184 ?           SW           0:00 [khud]
  841 ?           SW           0:00 /sbin/cardmgr
  856 ?           SW           0:00 portmap
  870 ?           S            0:00 syslogd -n 0
  878 ?           SW           0:00 klogd -2
  938 ?           S            0:00 gpm -t ps/2 -n /dev/psaux
  955 ?           SW           0:00 yperv
 1055 ?           S            0:14 xfs -port -1 -daemon -droppriv -user xfs
 1111 ?           SW           0:00 rpc.yxfrd
 1126 ?           S            0:00 /usr/sbin/atd
 1147 ?           SW           0:00 saslauthd -a pam -T

```

ps ax komutuyla alınan çalışan süreçler listesi genellikle çok uzundur. (Tipik olarak 300 satır kadar.) Yalnızca içinde çalışmakta olduğunuz kabukla ilgili süreçleri görmek için **ps** komutunu parametresiz olarak kullanabilirsiniz. Kendinize ait süreçleri görmek isterseniz (farklı kabuklar içinde çalışıyor olabilirler) kullanmanız gereken komut:

ps u cayfer

gibi olmalıdır.

Her sürecin PID (Process ID) denilen kendine özgü bir numarası vardır. Bir süreçle bilgi alışverişinde bulunmak ya da o sürece mesaj göndermek isteyen diğer süreçler, bu numaraları kullanırlar. Örneğin 4261 numaralı süreci kesmek (LINUX terminolojisinde “öldürmek”) için bu sürece “kendini öldür” anlamında bir mesaj göndermek gerekir. LINUX'ta temel bazı haberleşme işleri için numarayla ve harf dizileriyle kodlanmış mesajlar vardır. Örneğin bir süreci öldürmek istiyorsanız o sürece “kendini öldür” anlamında “9” mesajını göndermeniz yeterli olacaktır. Tabii ki, bu sürecin sizin sözünüzü dinleyip intihar etmesi için sizin ya root olmanız ya da süreci başlatan kullanıcı, yani sürecin sahibi olmanız gerekir.

Çalışan süreçlerin listesini daha ayrıntılı bir şekilde görmek isterseniz

ps alx

komutunu kullanabilirsiniz.

F	UID	PID	PPID	PRI	NI	VSZ	RSS	MCHAN	STAT	TTY	TIME	COMMAND
100	0	1	0	8	0	1288	84	do_sel	S	?	0:04	init
040	0	2	1	9	0	0	0	contex	SW	?	0:00	[keventd]
040	0	3	1	19	19	0	0	ksofti	SWN	?	0:00	[ksoftirqd_CPU0]
040	0	4	1	9	0	0	0	kswapd	SW	?	0:04	[kswapd]
040	0	5	1	9	0	0	0	bdfld	SW	?	0:00	[bdfld]
040	0	6	1	9	0	0	0	kupdat	SW	?	0:00	[kupdate]
040	0	7	1	-1	-20	0	0	nd_thr	SW<	?	0:00	[ndrecoverd]
040	0	11	1	9	0	0	0	end	SW	?	0:02	[kjournald]
140	0	98	1	9	0	1720	192	devfsd	S	?	0:00	devfsd /dev
040	0	184	1	9	0	0	0	end	SW	?	0:00	[khubd]
040	0	841	1	9	0	1412	4	do_sel	S	?	0:00	/sbin/cardmgr
140	32	856	1	9	0	1416	4	do_pol	S	?	0:00	portmap
140	0	870	1	9	0	1360	176	do_sel	S	?	0:00	syslogd -n 0
140	0	878	1	9	0	1936	4	do_sys	S	?	0:00	klogd -2
140	0	938	1	9	0	1336	60	nanosl	S	?	0:00	gpm -t ps/2 -n /dev/psa
140	0	955	1	9	0	1412	4	do_pol	S	?	0:00	ypserv
140	414	1055	1	9	0	6972	2540	do_sel	S	?	0:14	xfs -port -1 -daemon -d
140	0	1111	1	9	0	1468	4	do_pol	S	?	0:00	rpc.yxfrd
040	2	1126	1	9	0	1312	108	nanosl	S	?	0:00	/usr/sbin/atd
040	0	1147	1	9	0	1500	4	wait_f	S	?	0:00	saslauthd -a pam -T
040	25	1179	1	9	0	9988	464	rt_sig	S	?	0:00	named -u named
040	25	1183	1179	9	0	9988	464	do_pol	S	?	0:00	named -u named
140	25	1188	1183	9	0	9988	464	rt_sig	S	?	0:00	named -u named
040	25	1189	1183	9	0	9988	464	nanosl	S	?	0:00	named -u named
040	25	1190	1183	9	0	9988	464	do_sel	S	?	0:00	named -u named
140	0	1203	1	9	0	2660	4	do_sel	S	?	0:00	/usr/sbin/sshd
140	0	1238	1	9	0	2068	4	do_sel	S	?	0:00	xinetd -stayalive -reus

Bu listedeki önemli bilgiler şunlardır:

ps Komutu Rapor Ayrıntıları	
Bilgi Alanı	Açıklama
UID	Sürecin sahibinin sayısal kullanıcı kodu.
PID	(Process ID) Süreç tanıtım numarası.
PPID	Süreci bir başka süreç başlattıysa o sürecin numarası. (Parent Process ID)
TTY	(Teletype : Çok eskilerden kalan bir alışkanlık.) Sürecin hangi terminalden başlatıldığını belirtir. ?: Herhangi bir teminalden değil, sistemin kendisinin başlattığı süreçleri gösterir. ptsN: (pts1, pts2 gibi) Ağ üzerinden bağlanmış kullanıcılar tarafından başlatılmış süreçleri gösterir. ttyN: (tty1, tty2 gibi) Seri arabirim üzerinden (modemle bağlantı gibi) bağlı kullanıcılar tarafından başlatılmış süreçleri gösterir.
STAT	(<i>Status</i>) Sürecin bulunduğu duruma ilişkin bir kod. R (<i>Runnable</i>): Çalışabilir durumda, sırasını bekliyor. S (<i>Sleeping</i>): Uyuyor. Bir şeylerin olmasını bekliyor Z (<i>Zombie</i>): Bu süreç ile bağlantılı tüm diğer süreçler bitmiş veya ölmüş; bunun da bitmiş olması gerekirdi ama bir nedenle ölememiş. ps listesinde hâlâ görünüyor olması zararsızdır.
TIME	Sürecin ne kadar zamandır çalıştığını gösterir.
COMMAND	Süreci başlatan komut satırıdır. (Varsa)

Şeytanlar (Daemons)

LINUX süreçleri arasında “**daemon**” (“deymın” diye okunur) sözcüğüyle tanımlanan bir özel süreç çeşidi vardır. Bunların normal süreçlerden tek farkı hiçbir şekilde konsola ya da ekrana mesaj göndermemeleridir. Daemon’lar geri planda sessizce çalışırlar. Eğer başlarına kayda değer bir iş gelirse bunu **/var/log/syslog** dosyasına ya da kendi log dosyalarına kaydederler. Genellikle sistemin açılışıyla birlikte başlatılıp sistem kapanıncaya kadar sürekli çalışırlar. Bir kural olmamakla birlikte daemon tipi süreçlere ilişkin programların isimleri genellikle “**d**” harfiyle biter. (**httpd**, **ftpd**, **named** gibi)



“Daemon”, İngilizce’de “şeytan”, “zebanî” anlamında kullanılan bir sözcüktür. Geri planda sessizce ama sürekli çalışan programlara daemon adı verilmesinin ilginç bir nedeni var:

Büyük fizikçi Maxwell, gazların dinamiğini kolay anlatmak için iki bölümlü hayali bir kutu tasarlamış. Kutunun iki bölümü arasında ancak bir gaz molekülü geçebilecek kadar bir delik olduğunu ve ancak yeteri kadar kinetik enerjiye sahip moleküllerin bu delikten diğer tarafa geçebileceğini söylemiş. Moleküllerin kinetik enerjileri de sıcaklıklarıyla doğru orantılı olduğu için bir süre sonra kutunun iki bölümünün ısısının eşitleneceğini böyle açıklamış. Kavramı dramatize etmek için de deliğin başında bir zebaninin oturduğunu ve delikten geçmeye çalışan tüm moleküllerin enerji düzeylerini kontrol edip ancak yeteri kadar hızlı olanların geçmesine izin verdiğini anlatmış. (Maxwell’in şeytanı hakkında daha ayrıntılı bilgiyi Bilim ve Teknik dergisinin Haziran 2003 sayısında bulabilirsiniz.)

MULTICS işletim sistemini geliştiren ekipte yer alan ve asıl mesleği fizik olan Fernande J. Corbato da arka planda sessizce çalışıp diğer süreçleri denetleyen programlara, Maxwell’in zebanisinden esinlenerek “daemon” adını vermiş.

Daemon’lara klasik örnekler olarak **httpd** ve **named** gösterilebilir. **httpd** web sunucu yazılımı, **named** ise “www.pusula.com” gibi alfabetik internet adreslerinin sayısal IP karşılıklarını bulan DNS yazılımıdır.

Bazen daemon tipi süreçler ölebilirler. Bunun nedeni genellikle programın bir hatası ya da içinden çıkılmaz bir sorun yüzünden sistem yöneticisi tara-

findan özellikle öldürülmeleridir. Böyle durumlarda daemon programın yeniden başlatılması çoğunlukla sorunu çözer.

Süreç Öldürmek

Diyelim ki başlattığınız bir iş kontrolden çıktı ve istediğiniz ya da beklediğiniz gibi davranmıyor ya da yanlış iş başlattığınızı farkettiniz. Doğal olarak bu işi hemen kesmek istiyorsunuz. İlk denemeniz gereken Ctrl-C tuşudur. Olmazsa Ctrl-D tuşu... (Fazladan basacağınız Ctrl-D kabuk programınızı sona erdirip terminal ekranınızın kapanmasına neden olabilir.) Ctrl-C ve Ctrl-D tuşlarıyla bir süreci öldürebilmeniz için, o sürecin çalıştırıldığı ekranın açık olması ve daha önemlisi o sürecin sahibi olmanız gerekir.

Öldürmek istediğiniz bir programı Ctrl-C veya Ctrl-D tuşlarıyla öldüremiyorsanız o programa ilişkin sürecin numarasını öğrenip, sürece “öl” mesajını göndermeniz gerekir.

Bunun için:

1. Süreç size ait değilse root kullanıcı olun. Bunun için herhangi bir telnet penceresinden,

su -

komutunu verip ardından sorulan şifreyi girin.

2. Uygun bir **ps** komutuyla (“**ps ax**”) çalışmakta olan süreçlerin bir listesini alın.
3. Bu listeye bakarak sorun çıkaran sürecin numarasını öğrenin. (Diyelim ki 5443)
4. “**kill 5443**” komutuyla 5443 numaralı sürece “*kendini öldür*” mesajını gönderin.
5. Tekrar aynı **ps** komutunu kullanarak sürecin listeden kaybolup kaybolmadığını kontrol edin.
6. Eğer sorun yaratan süreç hala direniyorsa,

kill -9 5443

komutuyla biraz daha sert bir emir olan “*kendini koşulsuz öldür (geber)*” mesajını gönderin.

Kim Korkar LINUX'tan?

Gene olmadı diyelim.

Süreci hâlâ öldüremiyorsanız, kabuk programınızı öldürmeyi deneyiniz. Hâlâ direniyorsa, başı bozuk süreci öylece bırakmayı da düşünebilirsiniz. Bu sürecin sisteme ne kadar yük getirdiğini **top** komutuyla görebilirsiniz. Eğer bu süreç performans açısından ya da bir başka şekilde sorun çıkarmıyorsa bırakın ortalıkta zombi gibi sürünsün. Yok eğer sorun çıkarıyorsa sisteminizi düzgün olarak kapatıp tekrar açın.



Sisteminizde denetimden çıkmış, öldüremediğiniz süreçler varsa ve bunlar diğer işleri bozuyorsa sisteminizi kapatıp açmaktan başka seçeneğiniz kalmamış demektir. Ancak böyle bir durumda bile bilgisayarınızı elektrik anahtarından kapatmayı veya reset düğmesine basmayı aklınızdan dahi geçirmemelisiniz! LINUX altında çalışırken böyle bir durumla karşılaşma olasılığınızın çok çok düşük olduğunu belirtmek isteriz.



Bazı programlar bellekte birden fazla kopya olarak yer alır ve çalışırlar. Bunun en çok rastlanan örneklerinden biri çok sayıda pencerede çalışan Mozilla ya da Netscape web tarayıcılarıdır. Bu yazılıma ait süreçlerin hepsini birden öldürmeniz gerekirse, teker teker süreç numaralarını bulup öldürmektense

killall -9 mozilla-bin

komutunu kullanabilirsiniz. Dikkat ederseniz **killall** komutunda süreç numarası değil süreç adı belirtiliyor. Web tarayıcınızın çalışırken başlatmış olduğu süreçlerin isimlerini bilmiyorsanız **ps ax** komutu işinize yarayacaktır.

Link Kavramı ve 1n Komutu

Şimdi biraz mistik bir konudan söz edeceğiz. LINUX işletim sistemi altında bazı dosyalar aslında bulundukları yerde olmayabilirler. Evet, yanlış okumadınız! Diskin üzerinde yer alan bazı dosyalar aslında orada olmayabilir; hatta bir dosyanın sistemde tek bir kopyası olmasına rağmen, bu dosya birden fazla dizinde, üstelik farklı isimlerle yer alabilir. Kavraması ve kullanması zor bir kavram fakat bir kez mecbur kalıp da kullandınız mı hoşunuza gideceğine emin olabilirsiniz.

Galiba en iyisi bir örnekle anlatmak:

Farzedin ki, bilgisayarınıza **matlab** isimli yeni bir uygulama programı yüklemeniz gerekiyor. Ancak, programın bir gereği olarak, program paketine ilişkin dosyaların **/usr/local/matlab** diye bir dizinin altına yer alması gerekiyor. Eh! Olabilir. Ancak bir sorun var! **/usr** diskinde, yeni programa ilişkin dosyalar için yeterli boş yer yok ve burada silebileceğiniz gereksiz dosyalar da yok!

İşte mistik **ln** kavramı, bu problemi LINUX'un şanına yaraşır bir yöntemle çözmenizi sağlar.

Yeteri kadar boş yeri olan disk bölümlerinden birinde, örneğin **/home** dizinin bulunduğu disk bölümünde, yeni yükleyeceğiniz program için bir dizin yarattınız: (**/home** dizini altında yeni dizin yaratabilmek için root kullanıcı olmanız gerekecektir.)

```
mkdir /home/matlab
```

Sonra, bu dizini, **/usr/local** altında yer alıyormuş gibi gösterebilmek için,

```
ln -s /home/matlab /usr/local/matlab
```

komutunu veriniz.

Böylece, gerçekte **/home** altında yer alan **matlab** dizini, aynı zamanda **/usr/local** altında da varmış gibi olacaktır. Bu dizini kullanırken isterseniz **/home/matlab**, isterseniz **/usr/local/matlab** dizin adreslerini kullanabilirsiniz. Böylece **matlab** dizinini **/usr/local** altında görmek isteyen **matlab** yazılımını kandırmış oldunuz.

Link kavramının çok işe yarayabileceği, bir öncekine benzeyen bir senaryo daha anlatabiliriz.

Diyelim ki, elinizde **mhsb2002** isimli bir dosya var ve muhasebe departmanının kullandığı muhasebe programı bu dosyayı mutlaka bu isimde görmek istiyor. Öte yandan yeni satın aldığınız bir mali analiz programı, aynı muhasebe verilerini **acct2002** adıyla görmek istiyor.

Söz konusu dosyanın adı **mhsb2002** olduğu zaman muhasebe departmanının sorunu yok ama siz mali analiz programını çalıştıramıyorsunuz. Analiz çalışmaları için dosyanın adını değiştirseniz, siz çalışabiliyorsunuz ama bu sefer muhasebe departmanındaki program kullanılamıyor. Dosyanın adını **mhsb2002** olarak tutup, kendi analiz çalışmalarınız için **acct2002** adlı bir kopyasını çıkarmak da düşünülebilir ama çok kullanıcıli ortamda siz analizler üzerinde çalışırken öte taraftan muhasebe personeli yeni kayıtlar girip sizin analizlerinizi eskimiş kayıtlar üzerinde yapmanıza neden oluyorlar. İşte böyle bir durumda link kavramı ve **ln** komutu gene sizi kurtaracaktır.

```
ln ./mhsb2002 ./acct2002
```

Bu komutla **mhsb2002** dosyasını **acct2002** isimli bir dosyaya bağladığınızda (aslında sadece tek bir asıl kopya var; o da **mhsb2002**. **acct2002** isimli bir dosya ise aslında yok, **acct2002** asıl dosyanın bir başka adı), bu sayede **mhsb2002** dosyasında yapılan her değişiklik **acct2002** diye tanınan dosyada da aynen gözlenebilecektir. İşin bir başka yararlı tarafı da, **acct2002** isimli dosyanın diskte hiç yer kaplamayacak olmasıdır.

Bu örnekler arasında, dikkatinizi çekmiş olduğunu umduğumuz önemli bir fark var. İlk örnekte, yani matlab örneğinde, **ln** komutunda **-s** diye bir parametre kullandık; oysa ikinci muhasebe örneğinde kullanmadık!

- Eğer **ln** komutuyla birbirlerine bağlanacak olan dosya sistemi elamanları birer dizinse; **-s** parametresini kullanmak zorundasınız.
- Eğer **ln** komutuyla birbirlerine bağlanacak olanlar birer dosyaysa ama farklı dosya sistemlerindeyse (örneğin, farklı disklerdeyse), gene **-s** parametresini kullanmak zorundasınız.
- **ln** komutuyla, bir dizini ve bir dosyayı birbirlerine bağlayamazsınız. Bağlanacak olanların ikisi de dizin, ya da ikisi de dosya olmalıdır.

Aynı dosya sisteminde yer alan ve **"-s"** kullanılmadan bağlanmış olan dosyalardan birini silmeniz diğerini etkilemez. Asıl dosyayı silseniz bile, LINUX bağlantıyı fark edip dosyayı diskten gerçekten silmeyecektir. LINUX her dosya için yapılmış bağlantıları sayar ve her silme işleminde bağlantı sayısını bir azaltır. Gerçek silme işi bu bağlantı sayısı sıfırlanınca yapılır.

Farklı dosya sistemlerinde yer alan bağlantılar için bu bağlantı sayma işine güvenmeyiniz. Farklı dosya sisteminde bağlantısı olan bir dosyayı silerseniz başınız derde girer. Asıl dosya silinir ve diğer sistemde gerçekte var olmayan bir dosyayı gösteren bir bağlantınız kalır.



Bir dosyanın gerçekten var olan bir dosya mı, yoksa sadece bir bağlantı mı (link) olduğunu anlamak için **ls** komutunu **-l** seçeneği ile kullanmanız gerekir. İçinde bağlantılı dosyalar bulunan bir dizinde **ls -l** komutunu vererek, alacağınız listede bağlantılı dosyaları ve hangi dosyaya bağlantılı olduklarını açıkça görebilirsiniz.

```

cayfer@notebook.lojman.bilkent.edu.tr /home/cayfer - Shell - Konsol <4>
[cayfer@notebook cayfer]$ ls -al /etc/rc*
lrwxrwxrwx 1 root root 7 Ara 29 17:52 /etc/rc -> rc.d/rc
lrwxrwxrwx 1 root root 10 Ara 29 17:52 /etc/rc0.d -> rc.d/rc0.d
lrwxrwxrwx 1 root root 10 Ara 29 17:52 /etc/rc1.d -> rc.d/rc1.d
lrwxrwxrwx 1 root root 10 Ara 29 17:52 /etc/rc2.d -> rc.d/rc2.d
lrwxrwxrwx 1 root root 10 Ara 29 17:52 /etc/rc3.d -> rc.d/rc3.d
lrwxrwxrwx 1 root root 10 Ara 29 17:52 /etc/rc4.d -> rc.d/rc4.d
lrwxrwxrwx 1 root root 10 Ara 29 17:52 /etc/rc5.d -> rc.d/rc5.d
lrwxrwxrwx 1 root root 10 Ara 29 17:52 /etc/rc6.d -> rc.d/rc6.d
lrwxrwxrwx 1 root root 13 Ara 29 17:52 /etc/rc.local -> rc.d/rc.local
-rwxr-xr-x 1 root news 4565 Eyl 13 17:41 /etc/rc.news
lrwxrwxrwx 1 root root 15 Ara 29 17:52 /etc/rc.sysinit -> rc.d/rc.sysinit

```

Bu örnek listeye göre, aslında **/etc/rc0.d** diye bir dosya bulunmadığı, bu isimde **/etc** dizini altındaki **rc.d** dizinin altındaki **rc0.d** dosyasına bir bağlantı yapılmış olduğu anlaşılmaktadır. (**/etc/rc0.d** → **rc.d/rc0.d**)

Dikkat ederseniz, **ls -l** komutunun verdiği listede, gerçek bir dosya (dizin) değil de, bağlantı olan dosyalara (dizinlere) ait satırların başında bir **l** harfi bulunmaktadır.

İpin ucunu kaçırmayacağınıza eminseniz, bağlantılara bağlantı yapabilirsiniz.

“Pipe” Kavramı

“Pipe” (boru) kavramı, daha önce açıklamış olduğumuz “Giriş/Çıkış Yönlendirme” kavramıyla kolayca karıştırılan, bu yüzden de dikkatle ele alınması gereken bir kavramdır. Kısaca bir tekrarlamak gerekirse; “çıkış yönlendirme (>)”, çalıştırılan bir programın, standart çıktı birimine yazacağı satırların bir dosyaya yönlendirilmesi işlemidir. Aynı mantıkla, verilerini standart giriş biriminden okuyan programlar için “giriş yönlendirme (<)”; verilerin bir dosyadan okunmasını sağlayan işlemidir.

“**Piping**” işlemiyse, gene bir çeşit yönlendirmedir; ancak, **bir programın standart çıktısı, bir başka programa standart girdi olarak** yönlendirilir.

“**Pipe**” kurmak için, aynı komut satırında en az iki program birden başlatmalı ve bu iki programa ilişkin komutların arasına “ ” karakterini yerleştirmeniz gerekir.

Şimdi **grep** ve **more** komutlarını birlikte kullanarak (biliyorsunuz; pipe kurmak için en az iki komut gerekiyor) pipe kavramının kullanımına birkaç örnek verelim: (**grep** programını şimdilik öylesine bir komut olarak düşünün lütfen. **grep** hakkında söyleyecek o kadar çok şey var ki, ona da dört-beş sayfa ayırdık.)

```
grep ayfer * | more
```

Bu komut satırında **grep** ve **more** programları aynı anda başlatılıyor. “**ayfer**” ve “*” **grep** programına parametre olarak gönderiliyor. **grep** programının çıktısı ise **more** programına girdi olarak yönlendiriliyor.

“**grep ayfer ***” komutu çalışma dizini içindeki dosyalarda “**ayfer**” karakter dizisini arar ve bulduğu dosyalarla bu dosyalar içinde “**ayfer**” sözcüğü geçen satırları standart çıktı birimine listeler. Eğer bu liste çok uzunsa, terminal pencerenizin içinde akar gider ve siz pek bir şey göremezsiniz.

grep programının çok uzun olabilecek çıktısını **more** programına girdi olarak yönlendirdiğinizde **more** programı standart girişten gelen satırları içinde bulunduğu terminal ekranına sayfa sayfa listeleyecektir. Ekran her dolduğunda sol alt köşede **-- more --** işareti görünecek ve listelemeye devam edilmesi için sizin bir tuşa basmanız beklenecektir.

Bir başka örnek:

```
ps ax | grep in.named
```

- **ps** ve **grep** programlarını birlikte başlatır,
- **ps** programının **ax** parametresiyle kullanıldığında oldukça uzun olabilecek çıktısını **grep** programına girdi olarak gönderir,
- **grep** kendisine gönderilen satırlar arasında, içinde “**in.named**” sözcüğü geçenleri bulur ve sadece bu satırları listeler.

Şimdi de sıkı bir pipe örneği...

```
echo Sistemde `who | wc -l` kullanıcı var
```

Bu komut satırında birkaç kademeli bir işlem yapılması istenmektedir. Ayrıca bu komutta kullanılan tırnak işaretinin karakter dizilerini sınırlandıran tırnaktan olmadığına, Q-Türkçe klavyede virgülle aynı tuşta yer alan “ters tırnak” işareti olduğuna dikkatinizi çekeriz.

who ve **wc** programları birlikte başlatılacak ve **who** programının çıktısı standart girişindeki satır, kelime ve karakterleri sayan **wc** programına gönderilecektir. (**-l** seçeneği yalnızca satırların sayılmasını sağlıyor.) **wc** programının çıktısıysa, (**who** komutunun listelediği satırların sayısı) tırnaklar arasına yerleştirilerek; örneğin üç kullanıcı varsa, “Sistemde 3 kullanıcı var” dizisine dönüşecektir. Bu dizi de **echo** programına girdi olarak transfer edilecektir. **echo** programı ise parametrelerini aynen ekrana gönderir. Böylece ekranda “**Sistemde 3 kullanıcı var**” dizisinin görünmesi sağlanır.

Bu örnekteki komutu, kişisel dizinizdeki **.bashrc** dosyasına eklerseniz, sisteme her bağlandığınızda, sistemde siz dahil, kaç kişinin çalıştığını öğrenmiş olursunuz.

Biraz Nefes Alalım

LINUX işletim sisteminde yüzlerce komut var! Bunların bir kısmını belki de hiç kullanmayacaksınız. Bir kısmını ise çok sık kullanacaksınız. Genellikle hangi komutlardan yararlanacağınız tamamen bilgisayar ne amaçla kullanıldığınıza bağlıdır.

Önceki bölümlerde biraz fazla teknik ayrıntıya kaçtığımızın farkındayız; bu yüzden bu bölümde biraz nefes almak amacıyla, çok önemli olmayan, fakat kullanımı da hoş olan birkaç komuttan söz etmek istiyoruz. Bu komutlardan söz ederken kullanacağımız genel form doğal olarak

```
komut [ -seçenekler ] [ parametre ] [ parametre ] ...
```

olacaktır. Tüm LINUX konsol komutları zaten bu formdadır.

Bu formdaki **[]** karakterleri, aralarında yer alan seçenek ve/veya parametrelerin isteğe bağlı olduğunu (*optional*) göstermektedir. Örneğin;

Kim Korkar LINUX'tan?

cp [-ri] dosya1 dosya2

formunda verilen bir komutta **-r** veya **-i** veya **-ri** isteğe bağlı parametreler; **dosya1** ve **dosya2** ise zorunlu parametreler olarak anlaşılmalıdır.

Kullanışlı LINUX Komutları

cal [ay] [yıl] (*calendar*)

Parametresiz kullanırsanız, içinde bulunduğunuz ay için bir takvim yaprağı listelenir.

```
cayfer@notebook.lojman.bilkent.edu.tr: /home/cayfer - Shell - Konsole <5>
[cayfer@notebook cayfer]$ cal
Şubat 2003
Pa Pz Sa Çr Pr Cu Ct
                1
 2  3  4  5  6  7  8
 9 10 11 12 13 14 15
16 17 18 19 20 21 22
23 24 25 26 27 28

[cayfer@notebook cayfer]$
```

Parametre olarak herhangi bir yıl girerseniz, o yıl için 12 aylık bir takvim listelenir.

```
cayfer@notebook.lojman.bilkent.edu.tr: /home/cayfer - Shell - Konsole <4>
[cayfer@notebook cayfer]$ cal 2003
2003

    Ocak                Şubat                Mart
Pa Pz Sa Çr Pr Cu Ct  Pa Pz Sa Çr Pr Cu Ct  Pa Pz Sa Çr Pr Cu Ct
    1  2  3  4          1  2  3  4  5  6  7  8  1  2  3  4  5  6  7  8
 5  6  7  8  9 10 11    9 10 11 12 13 14 15    9 10 11 12 13 14 15
12 13 14 15 16 17 18    16 17 18 19 20 21 22    16 17 18 19 20 21 22
19 20 21 22 23 24 25    23 24 25 26 27 28        23 24 25 26 27 28 29
26 27 28 29 30 31      30 31

    Nisan                Mayıs                Haziran
Pa Pz Sa Çr Pr Cu Ct  Pa Pz Sa Çr Pr Cu Ct  Pa Pz Sa Çr Pr Cu Ct
    1  2  3  4  5          1  2  3          1  2  3  4  5  6  7
 6  7  8  9 10 11 12    4  5  6  7  8  9 10    8  9 10 11 12 13 14
13 14 15 16 17 18 19    11 12 13 14 15 16 17    15 16 17 18 19 20 21
20 21 22 23 24 25 26    18 19 20 21 22 23 24    22 23 24 25 26 27 28
27 28 29 30            25 26 27 28 29 30 31    29 30

    Temmuz                Ağustos                Eylül
Pa Pz Sa Çr Pr Cu Ct  Pa Pz Sa Çr Pr Cu Ct  Pa Pz Sa Çr Pr Cu Ct
    1  2  3  4  5          1  2          1  2  3  4  5  6
```

sleep n

n parametresi olarak verilen saniye kadar bekler. Herhalde aklınıza ilk olarak böyle bir komutun ne işe yarayacağı sorusu gelmiştir. İlk bakışta pek işe yaramamış gibi görünen bu komut, kabuk programları yazmaya başladığınızda (*shell programming*) işinize yarayacaktır.

watch [-n saniye] komut

“komut” komutunu, **n** parametresi olarak verilen saniye aralıklarla sürekli olarak çalıştırır. Örneğin,

watch -n 5 who

who komutunu her 5 saniyede bir tekrarlar. Sisteme bağlanmasını beklediğiniz birisi varsa onu beklerken bu komut çok işinize yarayacaktır. **watch** programı Ctrl-C ile kesilinceye kadar çalışır. LINUX'ta programları durdurmanın tek yolu Ctrl-C değildir. Süreçlerle ilgili bölümde çalışmakta olan programları durdurmanın ve davranışlarını denetlemenin, bir başka deyişle çalışan süreçlere sinyal göndermenin yöntemini öğrenmiştiniz.

wc [-lwc] [dosya] (word count)

Parametresi olan dosyadaki satır, sözcük ve karakterleri sayar. Eğer parametre olarak bir dosya adı belirtilmezse, standart girişteki satırlar için bu sayım işini yapar. Sayım sonuçlarını standart çıkışa yazar.

- l** seçeneği verilirse, yalnızca satırları,
- w** seçeneği verilirse, yalnızca sözcükleri,
- c** seçeneği verilirse, yalnızca karakterleri sayar.

Hem satırları, hem de sözcükleri birlikte saydırmak isterseniz, **-lw** seçeneğini kullanabilirsiniz.


```
cayfer@notebook.ijman.bilkent.edu.tr: /home/cayfer - Shell - Konsol <5>
[cayfer@notebook cayfer]$ wc /etc/terncap
16432  77749 703515 /etc/terncap
[cayfer@notebook cayfer]$ 
[cayfer@notebook cayfer]$ wc -l /etc/terncap
16432 /etc/terncap
[cayfer@notebook cayfer]$ 
[cayfer@notebook cayfer]$ wc -lw /etc/terncap
16432  77749 /etc/terncap
[cayfer@notebook cayfer]$ 
[cayfer@notebook cayfer]$ wc -lw < /etc/terncap
16432  77749
[cayfer@notebook cayfer]$ █
```

tail [-n] [-f]

Bir dosyanın son **n** satırını görüntüler. Eğer **n** belirtilmezse son 10 satır görüntülenir. Kullanışlı değil mi? Hele bir de **-f** parametresini öğrenince bu komutu çok seveceksiniz...

-f parametresiyle birlikte kullanıldığında dosya sonuna geldiğinde **tail** programı işini bitirmez ve dosyaya yeni satırlar eklendikçe onları da göstermeye devam eder.

En iyisi durumu bir örnekle anlatmak:

Diyelim ki sisteminiz web servisi verecek şekilde kurulmuş. Eğer çıldırmadıysanız web servisini **apache** isimli programla veriyor olmalısınız. **apache** hem performans hem güvenlik hem de güvenilirlik açısından dünyanın en gelişmiş web sunucu yazılımıdır ve tüm LINUX dağıtımlarında standart olarak bulunur.



Bir anımızı anlatmadan geçemeyeceğiz... Birkaç yıl önce (2000 falan) Web sunucu güvenliği konularını tartışmak için bir liste açılmıştı. Listenin açıldığını duyuran şahıs “burada güvenlikle ilgili tartışmalar yapıp bilgi ve deneyim paylaşacağız” demişti. Birkaç saat sonra aynı listeye birisi “Microsoft IIS güvenliği de tartışılacak mı?” diye bir soru gönderdi. Yanıt çok ilginçti : “IIS’in güvenlikle ne ilgisi var ki?”

Neyse, bu kadar çamur atma yeter. (Şimdilik) Örneğimize dönelim... **apache** web sunucusu (aksi belirtilmedikçe) sunduğu web sayfalarını ziyaret edenlerin log dosyasını tutar. (log yerine uygun bir Türkçe sözcük bulamadık.) Bu dosyada hangi saatte hangi dosyaların hangi istemciler tarafından istendiğinin

kaydı tutulur. Bu bilgiler birçok web sitesi işleticisi için çok değerlidir. Log kayıtları üzerinde ayrıntılı istatistik çalışması yaparak en çok hangi sayfaların ziyaret edildiğini, ziyaretçilerin bu sayfalarda ne kadar zaman geçirdiğini, en çok hangi ülkeden ziyaretçi geldiğini falan öğrenebilirler. Apache log dosyalarının analizi için bir sürü hazır ve özgür program bulabilirsiniz. Bunların en popülerlerinden biri “**analog**” isimli pakettir. (*www.analog.cx*) Neyse...

Apache’nin log kayıtlarını dosyaya eklendikçe görmek isterseniz;

tail -f /var/log/httpd/access_log

komutunu kullanabilirsiniz. Bu komut **/var/log/httpd/access_log** dosyasının son 10 satırını gösterecek fakat **-f** parametresi verildiği için dosya bitince durmayacaktır. Dosyaya yeni satırlar eklendikçe onları da listelemeye devam edecektir. Böylece web sitenizin ziyaretçilerini gerçek zamanda izleyebileceksiniz. (Bu komutu deneyebilmeniz için sisteminizde bir web sunucusu kurulu ve çalışıyor olmalıdır; üstelik birilerinin de web sitenizi ziyaret ediyor olması gerekir ki log dosyasına yeni kayıt düşülsün.)

more

Uzun metin dosyalarını; örneğin log dosyalarını, ekrana sayfa sayfa listelemek için kullanılır. Dosyanın ekrana veya terminal penceresine sığıp sığmadığı görüntüledikten sonra sol alt köşede

--More-- (12%)

gibi bir satır görünür. Bu işaret dosyanın yüzde 12’sine geldiğinizi ve daha görüntülenecek satırlar olduğunu belirtir.

Boşluk tuşuna basarak dosyada sayfa sayfa ilerleyebilirsiniz. Satır satır ilerlemek için Enter tuşunu, geri dönmek içinse **b** tuşunu kullanabilirsiniz.

Bölme işaretiyle (/) aynı **vi**’da olduğu gibi arama da yapabilirsiniz. Örneğin **--More--** işaretinin karşısına **/Ayfer** yazarsanız dosya içinde ilk rastlanan “**Ayfer**” dizisine kadar ilerlersiniz. Son aramayı tekrarlamak için **n** tuşuna basmanız yeterlidir.

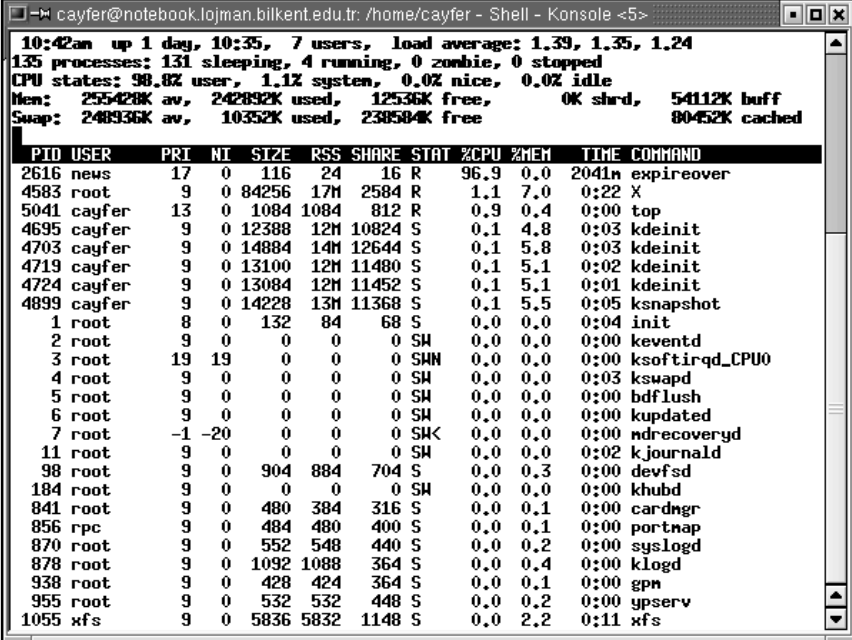
less

more komutunun neredeyse aynısıdır; ancak ekrana enine sığmayan satırlar

için sağa sola ok tuşlarıyla yatay kaydırma yapmanız da mümkündür. Aynı **more** komutunda olduğu gibi bölme işaretiyle dosyanın içinde istediğiniz karakter dizisini arayabilirsiniz.

top

Bilgisayarınız yavaşladığında, bilgisayarınızı en fazla meşgul eden işleri görmek için kullanabileceğiniz bir komuttur. Hem merkezi işlem biriminin kullanım oranlarını hem de takas alanı kullanım oranlarını görebilirsiniz.



```
10:42am up 1 day, 10:35, 7 users, load average: 1.39, 1.35, 1.24
135 processes: 131 sleeping, 4 running, 0 zombie, 0 stopped
CPU states: 98.8% user, 1.1% system, 0.0% nice, 0.0% idle
Mem: 255428K av, 242892K used, 12536K free, 0K shrd, 54112K buff
Swap: 248936K av, 10352K used, 238584K free, 0K swpd, 80452K cached
```

PID	USER	PRI	NI	SIZE	RSS	SHARE	STAT	%CPU	%MEM	TIME	COMMAND
2616	news	17	0	116	24	16	R	96.9	0.0	2041m	expireover
4583	root	9	0	84256	17M	2584	R	1.1	7.0	0:22	X
5041	cayfer	13	0	1084	1084	812	R	0.9	0.4	0:00	top
4695	cayfer	9	0	12388	12M	10824	S	0.1	4.8	0:03	kdeinit
4703	cayfer	9	0	14884	14M	12644	S	0.1	5.8	0:03	kdeinit
4719	cayfer	9	0	13100	12M	11480	S	0.1	5.1	0:02	kdeinit
4724	cayfer	9	0	13084	12M	11452	S	0.1	5.1	0:01	kdeinit
4899	cayfer	9	0	14228	13M	11368	S	0.1	5.5	0:05	ksnapshot
1	root	8	0	132	84	68	S	0.0	0.0	0:04	init
2	root	9	0	0	0	0	SW	0.0	0.0	0:00	keventd
3	root	19	19	0	0	0	SWN	0.0	0.0	0:00	ksoftirqd_CPU0
4	root	9	0	0	0	0	SW	0.0	0.0	0:03	kswapd
5	root	9	0	0	0	0	SW	0.0	0.0	0:00	bdfush
6	root	9	0	0	0	0	SW	0.0	0.0	0:00	kupdated
7	root	-1	-20	0	0	0	SW<	0.0	0.0	0:00	ndrecoveryd
11	root	9	0	0	0	0	SW	0.0	0.0	0:02	kjournald
98	root	9	0	904	884	704	S	0.0	0.3	0:00	devfsd
184	root	9	0	0	0	0	SW	0.0	0.0	0:00	khubb
841	root	9	0	480	384	316	S	0.0	0.1	0:00	cardmgr
856	rpc	9	0	484	480	400	S	0.0	0.1	0:00	portnap
870	root	9	0	552	548	440	S	0.0	0.2	0:00	syslogd
878	root	9	0	1092	1088	364	S	0.0	0.4	0:00	klogd
938	root	9	0	428	424	364	S	0.0	0.1	0:00	gpm
955	root	9	0	532	532	448	S	0.0	0.2	0:00	ypserv
1055	xfs	9	0	5836	5832	1148	S	0.0	2.2	0:11	xfs

En üst satırdaki

10:42am up 1 day, 10:35, 7 users, load average: 1.39, 1.35, 1.24

satırı sistemin 1 gün , 10 saat ve 35 dakikadır çalışmakta olduğunu göstermektedir.

Gene aynı satırda o anda 7 kullanıcının sistemi kullandığı (7 telnet bağlantısı yapmış tek kişi de olabilir) belirtiliyor. Bu kullanıcı sayısı sistemde kabuk programı çalıştıran kullanıcı sayısıdır; yani bu makinedeki web sitesini ziyaret edenler ya da bu makinedeki veritabanı sunucusundan yararlananların sayısını içermez.

Aynı satırda ortalama yük oranının son bir dakika içinde 1.39, son beş dakika içinde 1.35, son 15 dakika içindeyse 1.24 olduğunu görüyorsunuz. Bu ortalama yük oranları göreceli sayılardır ve tipik olarak birden küçüktür. Zaman zaman iki, üç, hatta 9'a kadar çıkması da normaldir. Yük oranlarınız genellikle 1 civarında değerlerde dolaşıyorsa sisteminizi tam ve ideal kapasiteyle kullanıyorsunuz demektir. Yük oranları genellikle iki veya üçten büyük değerlerde dolaşıyorsa makinenize biraz fazla yükleniyorsunuz demektir; yani artık merkezi işlem biriminizi daha hızlı biriyle değiştirmenin zamanı gelmiş demektir. Ancak üst modele geçmeye karar vermeden önce, **top** komutunun ürettiği rapordaki

```
Mem: 255428K av, 242040K used, 13388K free, 0K shrd, 54324 Kbuff
Swap: 248936K av, 10352K used, 238584K free 80480K cached
```

satırlarını da gözlemenizi öneririz. Eğer takas alanının kullanım ortalaması yüksekse (yukardaki örnekte görüntünün alındığı anda takas alanının pek kullanılmadığı görülmektedir) merkezi işlem birimini değiştirmektense öncelikle belleğinizi arttırmak daha yararlı olabilir.

top, ekranını her birkaç saniyede bir güncelleştirir. Gerek bu güncelleştirme sıklığını, gerekse rapor düzenini değiştirmek için **top** programına klavyeden bir tuşa basarak verebileceğiniz komutlardan bazıları aşağıdaki tabloda listelenmiştir:

top Komutları	
Tuş	Açıklama
q	top programını durdurur.
P (Büyük P)	İşleri merkezi işlem birimi kullanım oranlarına göre listeler.
M	İşleri bellek kullanım oranlarına göre listeler.
T	İşleri çalışmakta oldukları sürelerle göre listeler.
s	Ekranın güncelleşme sıklığını değiştirmek için kullanılır.
boşluk tuşu	Ekranı güncelleştirmek için kullanılır.

which

Kabuk programınızın komut satırından bir LINUX komutu yazdığınızda, bu komuta ilişkin bir program dosyasının diskten belleğe yüklenip çalışmaya başlayacağını biliyorsunuz. Peki bu program dosyasının hangi dizinde bulunduğunu bilebiliyor musunuz? Hatırlatmakta yarar var; sisteme verdiğiniz komutlara ilişkin program dosyaları **PATH** ortam değişkeninizde belirtilen dizinlerde ve belirtildiği sırayla aranacaktır. **PATH** ortam değişkeninizdeki dizinleri görmek için

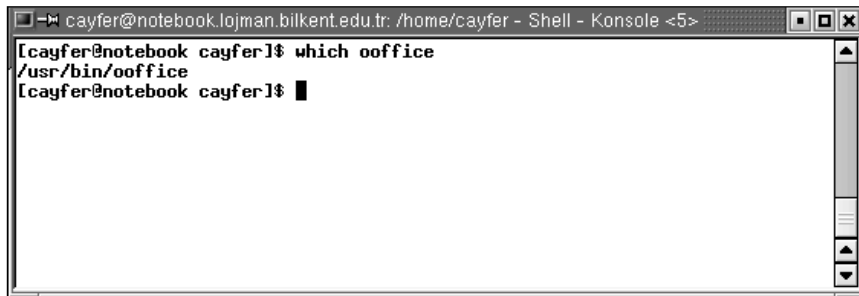
echo \$PATH

komutunu kullanabilirsiniz.



```
cayfer@notebook.lojman.bilkent.edu.tr: /home/cayfer - Shell - Konsol <5>
[cayfer@notebook cayfer]$ echo $PATH
/usr/local/bin:/bin:/usr/bin:/usr/X11R6/bin:/usr/ganes:/home/cayfer/bin
[cayfer@notebook cayfer]$
```

which komutuna, parametre olarak vereceğiniz bir komutun hangi dizindeki program dosyası kullanılarak çalıştırılacağını öğrenebilirsiniz.



```
cayfer@notebook.lojman.bilkent.edu.tr: /home/cayfer - Shell - Konsol <5>
[cayfer@notebook cayfer]$ which ooffice
/usr/bin/office
[cayfer@notebook cayfer]$
```

Yukarıdaki örnekte, telnet ekranınızda **ooffice** komutunu vermeniz durumunda, **/usr/bin** dizinindeki **ooffice** program dosyasının belleğe yüklenerek çalıştırılacağı görülüyor.

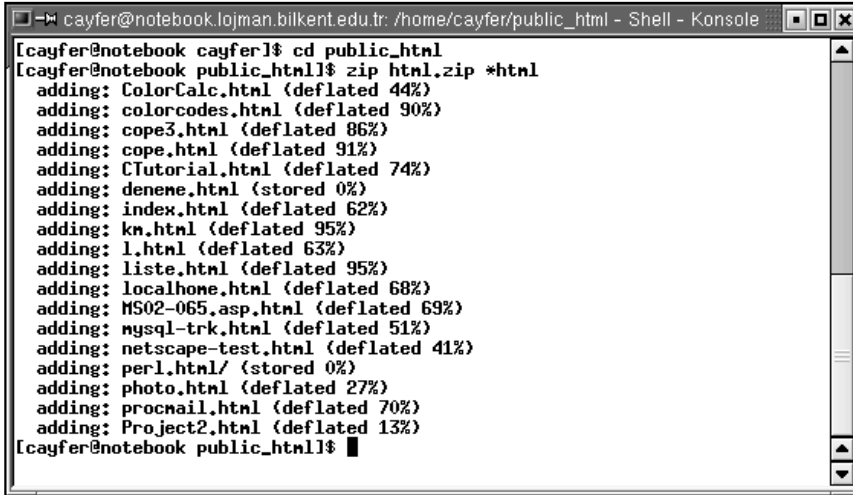
zip

PC dünyasının yakından tanıdığınız dosya sıkıştırma programıdır.

Genel formu:

zip [-r] zip_dosyası dosya1 dosya2 ... dosyaN

olan bu komutun ürettiği sıkıştırılmış dosyalar MS-DOS serisi işletim sistemi için Phil Katz tarafından geliştirilmiş olan PKZIP programının ürettiği dosyalarla uyumludur.



```

[cayfer@notebook cayfer]$ cd public_html
[cayfer@notebook public_html]$ zip html.zip *html
  adding: ColorCalc.html (deflated 44%)
  adding: colorcodes.html (deflated 90%)
  adding: cope3.html (deflated 86%)
  adding: cope.html (deflated 91%)
  adding: Ctutorial.html (deflated 74%)
  adding: deneme.html (stored 0%)
  adding: index.html (deflated 62%)
  adding: kn.html (deflated 95%)
  adding: l.html (deflated 63%)
  adding: liste.html (deflated 95%)
  adding: localhome.html (deflated 68%)
  adding: MS02-065.asp.html (deflated 69%)
  adding: mysql-trk.html (deflated 51%)
  adding: netscape-test.html (deflated 41%)
  adding: perl.html/ (stored 0%)
  adding: photo.html (deflated 27%)
  adding: procmail.html (deflated 70%)
  adding: Project2.html (deflated 13%)
[cayfer@notebook public_html]$
  
```

-r seçeneğiyle birlikte kullanıldığında sıkıştırılacak dosyalar arasında dizinlerin de bulunması durumunda, o dizinleri ve varsa, alt dizinleri de sıkıştırarak “**zip_dosyası**” içine yerleştirilecektir. Komutun tüm seçenekleri hakkında açıklayıcı bilgi için **zip** komutunu hiç parametre kullanmadan verebilirsiniz.

unzip

Adından da anlaşılacağı gibi **zip** komutuyla sıkıştırılmış dosyaları açan programdır.

gzip

Dosya sıkıştırma programları arasında LINUX dünyasında en popüler olanı, **gzip** programıdır. GNU organizasyonu tarafından genel kamu lisansı ile bilgisayar kullanıcılarına armağan edilmiştir.

Kim Korkar LINUX'tan?

gzip programının kullanma mantığı zip'e göre biraz farklıdır. **gzip** dosyaları teker teker ve kendi üzerlerine sıkıştırır. Örneğin:

```
gzip mail2002.log
```

diye bir komut verilirse, **mail2002.log** dosyası sıkıştırılarak **mail2002.log.gz** dosyasına dönüştürülür.

```
gzip *log
```

diye bir komut verildiğinde çalışma dizininde adı "**log**" ile biten beş tane dosya varsa, iş bittiğinde çalışma dizininde adı **.gz** ile biten beş tane sıkıştırılmış dosya oluşur.

Bir dizindeki dosyaları ve alt dizinlerini birlikte sıkıştırıp tek bir sıkıştırılmış dosya elde etmek istediğinizde kullanacağınız komut **gzip** değil, **tar** komutu olmalıdır. **tar** komutunu ileride ayrı bir bölümde anlatacağız.

gunzip

gzip komutuyla sıkıştırılmış dosyaları açmak için kullanılır. **gzip** komutunun dosyaları teker teker sıkıştırması gibi **gunzip** de sıkıştırılmış dosyaları teker teker açar.

Daha önce **gzip** ile sıkıştırılmış **mail2002.log.gz** dosyasını **gunzip** ile açmak istediğinizde;

```
gunzip mail2002.log.gz
```

komutunu vermeniz yeterlidir.

bzip2

bzip2 son yıllarda hızla yaygınlaşan bir dosya sıkıştırma programıdır. Sıkıştırma oranı konusunda GNU Lisansı ile dağıtılan **gzip**'den daha başarılı olduğu söylenir. **gzip** ve **gunzip** gibi kullanılır.

```
bzip2 büyük_dosya
```

```
bunzip2 büyük_dosya.bz2
```

BUNLARI BİLİYOR MUYDUNUZ?

Linux'un Desteklediği Donanım Platformları

LINUX işletim sistemi; Intel, AMD, Cyrix, Digital Alpha, SUN Sparc, Apple Macintosh, RS6000, Crusoe, PowerPC, Motorola 68K, Atari, NEC Alpha, IBM S/390, IBM S/370, VAX, MIPS, Playstation, XBox, Fujitsu AP1000+ serisi ve daha birçok merkezi işlem birimine uyarlanmış ve başarıyla kullanılmaktadır.

Tüm bu gerçeklere rağmen LINUX'un geleceği konusunda şüpheleri olanlara diyecek bir şey bulamıyoruz.

Kim Korkar LINUX'tan?

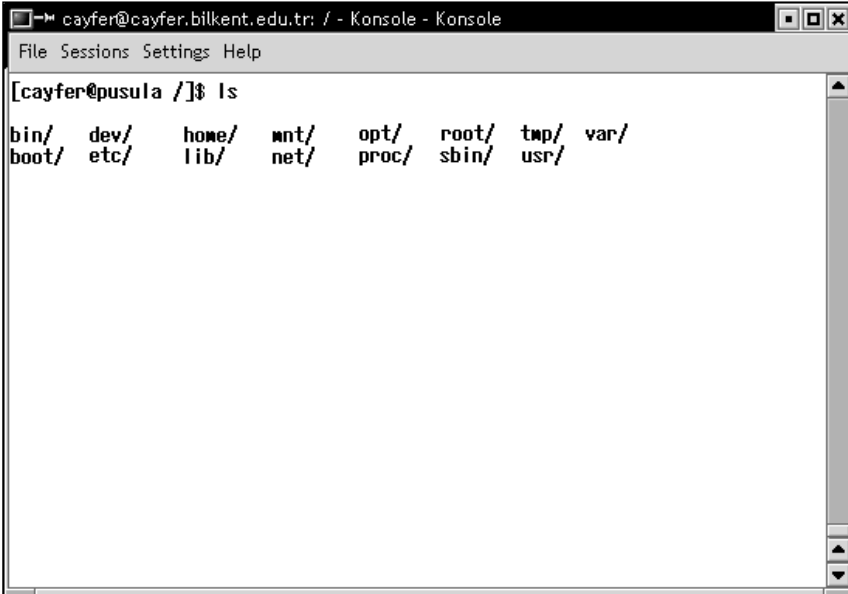
Çevreyi Tanıyalım

06

- /dev Dizini
- /etc Dizini
- lost+found Dizinleri

İyi bir bilgisayar kullanıcısı elinin altındaki kaynakları tanımalı, o kaynakların kuvvetli ve zayıf taraflarının yanı sıra kullanım alanlarını da iyi bilmelidir. LINUX için bu tanıma süreci, Windows serisi işletim sistemlerine göre daha uzun sürer. Sanıldığı gibi aksine, bu gecikme LINUX'un zorluğundan değil, kapsamlı olması ve esnekliğinden kaynaklanmaktadır. Eh tabii, kapsamlı olunca biraz da karmaşık oluyor ama gene de öğrenilemeyecek kadar değil.

Çevrenizi tanımaya başlamak için bilgisayar dünyasının “ne var ne yok?” sorusu olan **ls** komutunu “/” dizininde verdiğinizde



```
cayfer@cayfer.bilkent.edu.tr: / - Konsole - Konsole
File Sessions Settings Help
[cayfer@pusula ~]$ ls
bin/   dev/   home/  mnt/   opt/   root/  tmp/   var/
boot/  etc/   lib/   net/    proc/  sbin/  usr/
```

benzeri bir liste alırsınız. LINUX “/” dizininde pek fazla dosya ve dizin bulunmaz. Elbette siz isterseniz buraya dosya da koyabilirsiniz ama pek gerek-meyecektir; daha önemlisi geleneklere aykırı düşecektir.

/ Dizinindeki Dizinler	
Dizin	Kullanım Amacı
bin	Tipik LINUX kullanıcılarının kullanacağı komutlara ait program dosyalarının (<i>binaries</i>) saklandığı dizindir. “ <i>LINUX komutları arasında neler var acaba?</i> ” diye merak eden okuyucular bu dizindeki dosyaların listesini alıp sıradan man sayfalarına bir göz atıp komutları deneyebilirler. Bu komutları denerken root kimliğini kullanmamanızı hatırlatmak isteriz. root kimliğiyle verdiğiniz tüm emirler yerine getirilecektir; tüm dosya sistemini silmek buna dahildir. Normal kullanıcı kimliğiyle ise sisteme zarar veremezsiniz; olsa olsa kendi dosyalarınızı yok edersiniz ya da bozarsınız.
boot	Sistemin açılışı sırasında kullanılan dosyalar ve en önemlisi LINUX çekirdeği burada saklanır. Buradaki dosyaları pek kurcalamasanız iyi olur.
dev	Çok özel bir dizindir. Bu dizin altındaki dosyalar normal birer dosya değildir. /dev dizininin dosya listesini aldığınızda göreceğiniz dosyaya benzer şeyler aslında birer dosya değil, düğümdür (<i>node</i>). Bunun ne demek olduğunu biraz sonra anlatacağız. Şimdilik kısaca bu düğümlerin LINUX tarafından desteklenen donanım parçalarını tanımladığını belirtmekle yetinelim.
etc	Aman adı sizi yanıltmasın... Evet, “ etc ”, “ vesaire ” anlamındaki “ etc ” ama aslında sistemin en önemli dizinidir. Kullanıcı tanımlarının, LINUX’un ve sisteminizde çalışan önemli servislerin neredeyse hepsinin ayar dosyaları buradadır. /etc dizinini kaybeden bir sistem yönetici direksiyonunu kaybetmiş bir sürücüye benzer.
home	Kullanıcılarınızın kişisel dizinlerinin saklandığı dizindir. Genellikle burada her kullanıcı için kendi kullanıcı adıyla anılan bir dizin yer alır ve kullanıcılar buradaki kişisel dizinlerinde tam yetkiyle istediklerini yaparlar.
lib	Paylaşılan kütüphanelerin saklandığı dizinlerden biridir.
mnt	LINUX çekirdeğinin otomatik olarak iliştiirdiği (mount ettiği) birimler için dizinler içerir.

net	Genellikle boştur.
opt	AT&T (System V) UNIX ekolünden gelen sistem yöneticilerinin uygulama programlarını yüklemeleri için ayrılmıştır. Artık pek anlamı kalmadığı için genellikle boştur.
proc	Tamamen LINUX çekirdeğinin denetiminde bir dizindir. Aslında bu dizin diskte de değildir. Belleğin bir bölümü bu dizin için disk gibi kullanılmaktadır. (Sanal disk.)
root	Sistem yöneticisinin kişisel dizinidir. root kullanıcının kişisel dizininin /home altında olmaması strateji gereğidir. Genellikle sistem yöneticileri kullanıcı kişisel dizinleri için özel bir disk ya da disk bölümü ayırırlar. Sistemde meydana gelebilecek bir arıza nedeniyle bu disk ya da disk bölümü iliştilerilemezse root kendi dosyalarına gene de erişebilir diye böyle yapılır. Bu dizine de erişilemiyorsa zaten sistem yöneticisinin başı ciddi dertte demektir.
sbin	Genellikle sistem yönetimiyle ilgili komutların program dosyalarının saklandığı dizindir. Çoğunu çalıştırabilmek için root yetkisi gerekir. Bu nedenle bunları körü körüne denemek tehlikeli olabilir.
tmp	Herkese açık bir dizindir. Genellikle geçici dosyaların kaydedilmesi içindir. Sistem yöneticisi, yer gerektiği zaman bu dizindeki dosyaları hiç çekinmeden silebilir, hatta birçok sistem yöneticisi sistemlerinin açılışı sırasında bu dizinin otomatik olarak silinmesini tercih ederler.
usr	Çok işlevli bir dizindir. Adının çağrıştırdığı kullanıcılarla pek ilgisi yoktur. Genellikle uygulama programları ve bunlarla ilgili yazılım kütüphaneleri burada saklanır. Özellikle /usr/local dizini bir sistemde tüm kullanıcılara açık uygulama yazılımlarının yüklendiği dizindir. AT&T UNIX ekolünden gelen sistem yöneticileri için ayrılmış olan /opt diziniyle aynı amaçla kullanılır. Yani sisteme yeni bir uygulama programı (örneğin Acrobat PDF formatındaki dosyaları HTML'ye çeviren pdf2html gibi bir program) yükleyecekseniz, en mantıklısı program dosyalarını /usr/local/bin dizinine yerleştirmenizdir. /usr/lib dizini uygulama programlarına ait kütüphaneler için kullanılır.

var	<p>Önemli bir dizin!</p> <p>/var/mail veya /var/spool/mail: Sisteminizdeki kullanıcılara gelen e-posta mesajları, sahipleri okuyuncaya kadar bu dizin altında saklanır. Adına e-posta gelmiş olan her kullanıcı için burada bir dosya bulunur. Bir kişiye gelen tüm e-posta mesajları bu dosyada ardarda eklenmiş olarak saklanır. mail dizini aslında /var/spool altındadır. Tüm e-posta yazılımlarıyla uyumlu olması için bu dizinden /var/mail'e bir bağlantı atılmıştır. (ln komutunu hatırlıyor musunuz?)</p> <p>/var/log: Sistemin seyir defteri. Gerek LINUX'un kendisi, gerekse başından kayda değer bir şeyler geçen tüm programlar anılarını buraya kaydeder.</p> <p>/var/spool/lpd: Yazıcıya gönderilmeyi bekleyen işlerin saklandığı dizindir.</p> <p>/var/www: Eğer varsa bilgisayarınızda çalışan web sunucusuyla ve web sitenizle ilgili dosyaların saklandığı dizindir.</p>
------------	--

/dev Dizini

LINUX'ta tüm donanım parçaları birer dosya gibi kullanılır. İnanmayacaksınız ama fare bile aslında bir dosya gibi kullanılır. Sistem ve uygulama programları herhangi bir donanım parçasına erişmek istediğinde bu donanım parçasını **/dev** dizininde temsil eden düğüm'e (*node*) erişir. Bu düğümler **/dev** dizininde yer almakla birlikte, aslında gerçek birer dosya olmadıkları için diskte hiç yer harcamazlar.

Bu düğümlerin her birinin birer "**Major**" ve birer "**Minor**" numaraları vardır. Bir LINUX komutu ya da uygulama programı, **/dev** dizininde yer alan bir isim aracılığıyla bir donanım unsuruna ulaşmak istediğinde (örneğin, diskete kayıt yapmak için **/dev/floppy** düğümüne yazmak istediğinde), LINUX, bu major-minor numaralar aracılığı ile çekirdeğin (*kernel*) hangi modülünün harekete geçirileceğini anlar ve kontrolü, o donanım unsurunu tüm özellikleriyle tanıyıp denetleyebilen bir programa geçirir (*device driver*).

LINUX'un **/dev** dizinine yakından baktığınızda bu dizin altındaki düğümlerin çoğunun gene **/dev** dizini altında başka düğümlere bağlanmış olduklarını (link edilmiş olduklarını) göreceksiniz. Bu bağlantıları izlediğinizde en sonda düğümün kendisini bulursunuz.

```

cayfer@cayfer.bilkent.edu.tr: /home/cayfer - Konsol - Konsol <9>
File Sessions Settings Help

[cayfer@pusula cayfer]$ ls -al /dev/hda
lr-xr-xr-x  1 root    root      32 Jun  4 14:02 /dev/hda -> ide/host0/bus0/target0/lun0/disc

[cayfer@pusula cayfer]$ ls -al /dev/ide/host0/bus0/target0/lun0/disc
brw-----  1 root    root       3,  0 Jan  1 1970 /dev/ide/host0/bus0/target0/lun0/disc

```

Tipik bir LINUX bilgisayarının **/dev** dizininde yüzlerce düğüm yer alır. Burada yalnızca birkaç tanesinden söz etmek istiyoruz. Ne hepsini anlatmaya olanak var, ne de gereği var.

/dev Dizininde Yer Alan Bazı Önemli Düğümler	
Düğüm	İşlevi
/dev/console	Bilgisayarın karakter tabanlı ana ekranıdır. Sistemde ortaya çıkan donanım sorunları ve diğer önemli olaylara ilişkin mesajlar bu birime gönderilir. Eğer ekran o sırada X altında grafik kullanıcı arabirimiyle kullanılıyorsa ve xconsole yazılımının çalıştığı bir pencere yoksa /dev/console 'a gelen mesajlar görünmeyecektir.
/dev/mem	Sistemin ana belleğidir.
/dev/mouse	Fare donanımınızdır.
/dev/null	İşte bu ilginç bir düğümdür. Dipsiz kuyuya benzer. Buraya istediğiniz kadar veri kopyalayabilirsiniz. Ne disk doldu diye şikayet eder, ne de erişim haklarından dem vurur. Bir programın standart çıktıya gönderdiği mesajların çok uzun olduğunu ve bu çıktıların size gerekli olmadığını düşünün. Böyle bir durumda telnet ekranınıza yönelik gereksiz trafik yaratmamak ve hız kazanmak için programı başlatırken standart çıkısını /dev/null 'a yönlendirirsiniz olur biter. (/usr/local/bin/uzun_prog > /dev/null gibi.)

/dev/hda	IDE sabit disk. Birinci IDE arabirimdeki “ master ” disk.
/dev/hda1	Birinci IDE arabirimdeki “ master ” diskin birinci bölümü. (<i>partition</i>)
/dev/floppy	Disket sürücü.
/dev/cdrom	CD-ROM sürücü.
/dev/ttyA	Seri (RS-232) arabirim.
/dev/pty	Ağ üzerinden terminal bağlantısı için kullanılan sanal terminal arabirimi.
/dev/tape	Teyp sürücünüz.
/dev/eth0	Birinci Ethernet arabiriminiz.

/etc Dizini

/etc dizini de üzerinde biraz durmayı gerektiriyor.

Sistem yöneticisinin en çok dolaştığı dizindir. Bu dizinde birçok alt dizin ve dosya bulacaksınız. Hemen hemen hepsi sisteminizin davranışlarını etkileyen ayar dosyalarıdır. Bu nedenle yedeği alınması gereken ilk dizin **/etc** dizinidir. Bu dizindeki dosyalarda değişiklik yapmanız gerektiğinde kesinlikle ama kesinlikle önce o dosyanın bir kopyasını çıkarmanızı öneririz. Bu dizindeki bazı önemli dosyalara (aslında hepsi önemli ya, neyse) biraz değinmek istiyoruz.

/etc Dizinindeki Bazı Önemli Dosyalar	
Dosya / Dizin	İşlevi
/etc/passwd	Kullanıcı tanımlarınızın yer aldığı dosyadır. Sisteminize tanıttığınız her kullanıcı için ve sisteminizde süreç çalıştırması söz konusu olan tüm özel ve tüzel kişilikler için bu dosyada bir satır yer almalıdır. Bazı programlar güvenlik nedeniyle gerçek olmayan kullanıcı kimliğiyle çalıştırılır. Örneğin, Apache web sunucusu ya “ nobody ” ya da “ apache ” gibi bir kimlikle çalıştırılmalıdır. Bu durumlarda /etc/passwd dosyasında bu hayali kullanıcılar için de birer kayıt olmalıdır.

/etc/shadow	Kullanıcılarınızın şifrelerinin saklandığı dosyadır. Şifreler bu dosyada kriptolanmış olarak saklanır. Şifresini unutan birisinin şifresinin kriptolanmış halini burada görebilirsiniz ama bir işe yaramaz. LINUX'un kullandığı kriptolama algoritması tersine çevrilemez; yani kriptolanmış bir karakter dizisinin aslının ne olduğunu bulamazsınız. LINUX, sisteme bağlanmak isteyen bir kullanıcıdan şifresini aldıktan sonra bu karakter dizisini kriptolar ve kriptolanmış halini bu dosyadaki şifreyle karşılaştırarak doğru olup olmadığını kontrol eder.
/etc/group	Sisteminizde tanımlı kullanıcı gruplarını içerir.
/etc/rc.d	Sisteminiz açılırken çalıştırılan komut dizileri (<i>script</i>) dosyalarını içerir. Bu dizindeki dosyaların mantığını kitabın sistem yönetimi ile ilgili bölümlerinde bulacaksınız.
/etc/rc.local	Sisteminiz açılışı tamamlandığında, varsa yapılmasını istediğiniz işlere ilişkin komutları içeren dosyadır. Örneğin sisteminiz her açıldığında /tmp dizinin içindeki dosyaların ve dizinlerin silinmesini istiyorsanız bu dosyanın içinde bir yere "/bin/rm -r /tmp/*" gibi bir komut satırı ekleyebilirsiniz.
/etc/hosts.allow ve /etc/hosts.deny	Sisteminize TCP/IP üzerinden erişmesine izin vereceğiniz ya da izin vermeyeceğiniz IP adreslerinin tanıtıldığı dosyalardır. Bu dosyanın ayrıntılarını "Güvenlik" bölümünde bulacaksınız.
/etc/xinetd.d	Bilgisayarınızın internet servisleri ile ilgili ayarları. FTP sunucusu olarak hizmet verip vermeyeceğiniz; verecekseniz hangi port üzerinden vereceğiniz gibi ayarlar buradadır.
...	/etc dizinindeki ayar dosyaları bu kadar değil elbette. Burada yalnızca birkaç tanesinden örnek olarak söz ettik.

lost+found Dizinleri

Günün birinde bazı dizinlerde **lost+found** adında bir dizin belirdiğini farkedeceksiniz. İçine baktığınızda da adları sayılardan oluşan bir sürü alt dizin göreceksiniz.

Bir LINUX bilgisayar normal olmayan bir kapatmanın ardından açıldığında veya bir disk dosya sistemi 20 kez “mount” edildiğinde dosya sistemindeki dizin yapısı otomatik olarak kontrol edilecektir. Bir başka deyişle o dosya sistemi için **fsck** programı otomatik olarak çalıştırılacaktır.

fsck programı diskteki dosya yapısında bir gariplik, bozukluk veya tutarsızlık bulursa onarmak için elinden geleni yapacak, tehlikeli bulunduğu değişiklikler için izninizi isteyecektir. Açıkçası böyle durumlarda istenen izni vermekten başka bir çareniz de olmayacaktır.

fsck bulduğu bozuklukları giderebilirse ne ala; yok gideremezse sorun çıkaran disk bloklarındaki kayıtlı verileri o dosya sisteminin **lost+found** dizinine atıp işine devam edecektir.

fsck genellikle sorunlu dosya sistemlerini onarmayı başarır ama tabi bu sorunun derinliğine bağlıdır. **fsck** tarafından **lost+found** dizinine atılan dosya parçaları içinden hasarlı dosyaları ayıklamaya çalışmak samanlıkta iğne aramaktan farksızdır ama bazen insan bunu da yapmaya mecbur oluyor. Yeri gelmişken; bu gibi durumlara düşmemek için kendinize güvenilir bir yedekleme sistemi kurmalısınız. LINUX'unuzda bu iş için gereken her türlü yazılım zaten var.

bash Kabuğu ve Kabuk Programları

07

- Komut Satırının Yorumlanması ve Parametreler
- Kabuk Değişkenleri
(ya da Ortam Değişkenleri)
- Programları Arka Planda Çalıştırmak
- Ön Planda Çalışan Programları
Arka Plana Atmak
- Kabuk Programlama
 - İlk Kabuk Programı Örneği
 - İkinci Kabuk Programı Örneği

LINUX işletim sistemi, kullanıcıların verdikleri komutları çözümlemek ve bu komutları yerine getirecek programları başlatmak için **kabuk** (*shell*) programlarını kullanır. Bir başka deyişle, kabuk programları, kullanıcılarla bilgisayar arasındaki komut arabirimidir. Aslında, bu tip komut arabirimleri; yani “**komut yorumlayıcıları**” (*command interpreter*), tüm işletim sistemlerinde kullanılmaktadır; örneğin MS-DOS işletim sisteminde bu görevi COMMAND.COM üstlenmiş durumdadır.

LINUX işletim sisteminde, kullanıcıların birden fazla kabuk programı arasından seçim yapma ve beğendikleri komut yorumlayıcısını kullanma hakları vardır. Hatta aynı anda birden fazla kabuk programı bile kullanabilirler.

Bu kitapta LINUX işletim sisteminin en popüler kabuk programı olan **bash** kabuk programını anlatacağız. Ayrıntılarda önemli farklar olsa bile **bash** bilen birisi **csh**, **tcsh**, **sh**, **ksh** gibi diğer kabuk programlarını çok çabuk kavrayacaktır.

LINUX kullanımında deneyim kazanıp sistemin hakkını vermeye başladığınızda; özellikle sistem yöneticisi olma yolunda ilerledikçe, kabukların öne-

mi hızla ortaya çıkacaktır. Gerek sisteminize ileri düzey komutlar verirken, gerekse bir takım işleri otomatige bağlamaya başladığınızda kabukların yararlı özelliklerini kullanmaya ve daha önemlisi kabuk komutlarını kullanarak kendi komut dizilerinizi (*shell script*) yazmaya başlayacaksınız.

Bu aşamada, bu bölümün özellikle “**kabuk programlama**” ile ilgili kısımları size karışık gelebilir. Hele programcılık deneyiminiz yoksa bu bölümleri hiç üzülmenden atlayabilirsiniz. LINUX kullanmak için ille de programcı olmak gerekmez.

Komut Satırının Yorumlanması ve Parametreler

bash kabuk programı komut almaya hazır olduğunu ekranda

```
[cayfer@pusula cayfer] $
```

gibi bir karakter dizisiyle (**hazır işareti**: *prompt*) belirtir.

```
cp eski-dosya yeni-dosya
```

gibi bir komut verdiğinizde, kabuk programı, “**cp**” harflerini çalıştırmak istediğiniz programın adı olarak; “**eski-dosya**” ve “**yeni-dosya**” sözcüklerini ise bu **cp** programının iki parametresi olarak kabul edecektir.

Kabuğun yapacağı bir sonraki iş, çalıştırmak istediğiniz bu **cp** programının saklandığı disk dosyasını bulmak olacaktır. Bu arama işinin temelinde, sizin için tanımlanmış olan **PATH** kabuk ortam değişkeninin o andaki değeri yatmaktadır. Bu değişkenlerin değerleri tipik olarak

```
/usr/local/bin:/bin:/usr/bin:/usr/X11R6/bin:/home/cayfer/bin
```

benzeri bir karakter dizisi olacaktır.

Kendi **PATH** değişkeninizi görmek isterseniz “**echo \$PATH**” komutunu kullanabilirsiniz.

bash programı, **PATH** değişkeninin değeri içinde, “**:**” karakteriyle birbirlerinden ayrılmış olan dizinleri sırayla tarayıp bu dizinlerin içinde, sizin çalış-

tırma yetkiniz olan “**cp**” isimli bir dosya arayacaktır. Arama, dizin isimlerinin **PATH** değişkeninde veriliş sırasına göre yapılacaktır.

Örneğimize göre, **bash** programı, **cp** isimli dosyayı önce **/usr/local/bin** dizininde, orada bulamazsa **/bin** dizininde; orada da bulamazsa **/usr/bin**; olmazsa **/usr/X11R6/bin** dizininde, gene bulamazsa **/home/cayfer/bin** dizininde arayacaktır. Kabuk programınız söz konusu dosyayı bu dizinlerden hiçbirinde bulamazsa,

bash: cp: command not found

diye, komutu tanıyamadığına ilişkin bir hata mesajı vererek yeniden komut bekleme durumuna dönecektir.

Eğer **cp** program dosyası bu dizinlerden birinde bulunursa, bu dosyanın erişim yetkileri kontrol edilecektir. **cayfer**’in bu programı çalıştırmaya yetkisi varsa **cp** programı kabuk tarafından belleğe yüklenecek ve çalıştırılacaktır. Varsa, komut satırında verilen parametreler çözümlenip (çözümlemeden ne kastedtiğimizi biraz sonra açıklayacağız) **cp** programına aktarılacaktır.

Yüklenen program çalışmaya başladığında kontrol artık **cp** programına geçmiştir. Parametrelerin doğru sırada ve sayıda verilip verilmediğini her program kendisi kontrol eder ve gerekirse uygun hata veya uyarı mesajları üreterek, kullanıcıyı uyarır.

Bir komut verildiğinde o komuta ilişkin hangi program dosyasının belleğe yükleneceğini merak ettiğinizde

which komut

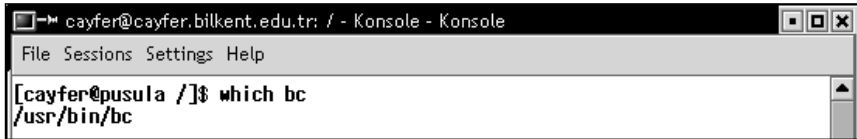
komutunu kullanabilirsiniz. “Böyle bir şeyi neden merak edeyim ki?” diyor-sanız bilgisayarınızda aynı isme sahip iki program dosyası olabileceğine dikkatinizi çekeriz.

Örneğin kendiniz C dili ile “**bc**” diye bir program yazabilir ve kişisel dizinize yerleştirebilirsiniz. Programınızı güzelce hatasız derledikten sonra çalıştırmak için “**bc**” komutunu verdiğinizde garip şeyler olduğunu gözleyebilirsiniz. İlk aklınıza gelen “*acaba doğru programı mı çalıştırıyorum?*” olmalıdır.

Bu soruya yanıt bulmak için:

which bc

komutunu verdiğinizde göreceksiniz ki sizin yazdığınız **bc** programı değil, **/usr/bin/bc** çalışıyor. Çünkü **bash** kabuğu çalıştıracağı komuta ilişkin program dosyasını **PATH** değişkenindeki dizinlerde ararken **/usr/bin** dizinindeki **bc** dosyası sizin kişisel dizininizdeki **bc** dosyasından daha önce bulunuyor.



Böyle bir durumda ya kendi programınızı yerini açıkça belirterek; yani komut olarak

/home/cayfer/bc veya
./bc

girerek (“**./bc**”, bu dizindeki **bc**” anlamında) çalıştırmalısınız; ya da **PATH** değişkeninizi kendi kişisel dizininiz daha önde olacak şekilde değiştirmelisiniz.

Kopyalama komutunu, çalışma dizinindeki tüm dosyaları

/disk2/home2/ayfer dizinine kopyalayacak şekilde

cp * /disk2/home2/ayfer

verdiğiniz varsayalım ve **bash** kabuğunun neler yapacağına bir göz atalım. Bu komutu gören **bash**, komut adı olan **cp** sözcüğünü bulduktan sonra, bu komutun parametrelerini bulup çıkarmaya çalışacaktır. Komut satırını tararken ***** karakterine rastlayınca, “tüm dosyalar” anlamına gelen bu işaret yerine, çalışma dizininde yer alan tüm dosya ve dizinlerin isimlerini yan yana gelecek şekilde yerleştirecektir.

Yani komut satırı,

cp abc dosya1 dosya2 xyz x123 /disk2/home2/ayfer

satırına benzer bir şekle dönüştürülecektir. (Çalışma dizininde sadece **abc**, **dosya1**, **dosya2**, **xyz** ve **x123** dosyalarının olduğunu varsayarak.) Bu dönü-

şümü ekranda gözleyemezsiniz; ancak bu tip dönüşümlerin olduğunu bilmeniz ve komutları verirken bu dönüşümleri dikkate almanız çok önemlidir.

Bazı durumlarda kabuk programlarının komutlarındaki parametreleri yorumlayıp açmaya çalışmasını istemezsiniz. Böyle bir gereksinim duyarsanız (ki **find** komutunu kullanırken duyacaksınız); kabuk programının kurcalamasını istemediğiniz parametreleri tırnak içinde yazmalısınız. Eğer kabuk programının irdelemeden komuta aktarmasını istediğiniz özel karakter tek bir karakterden oluşuyorsa, o karakteri tırnak içine almak yerine, önüne bir **"\"** (back slash) yerleştirebilirsiniz.



Bu arada, kabuk tarafından çalıştırılan programların sıfıncı parametrelerinin de bulunduğunu söylemeden geçemeyeceğiz. Bir program çalıştırıldığında, sıfıncı parametresi, programın kendi adıdır. Böylece, her program, hangi isimle kullanıldığını bilebilmektedir. Bu özelliğe tipik örnek **gzip** ve **gunzip** komutlarıdır. Bu iki komuta ait program dosyaları aslında tıpatıp aynı dosyadır. Aslında **gzip** isimli dosya gerçekten bu isimle diskte yer alırken, **gunzip** sadece bu dosyaya bir bağlantı da (*link*) olabilir. Yani **gzip** komutunu da verseniz, **gunzip** komutunu da verseniz aynı program dosyası belleğe yüklenip çalıştırılır. Bir dosyayı sıkıştırması gerektiğini mi yoksa sıkıştırılmış bir dosyayı açması mı gerektiğini anlamak için program sıfıncı parametreye, yani hangi isimle çalıştırıldığına bakarak karar verir.

Kabuk Değişkenleri (ya da Ortam Değişkenleri)

bash kabuk programları içinde çeşitli değişkenler tanımlamanız mümkündür. Hatta bazı standart değişkenler zaten öntanımlıdır. Öntanımlı kabuk değişkenleri arasında en önemlileri şunlardır:

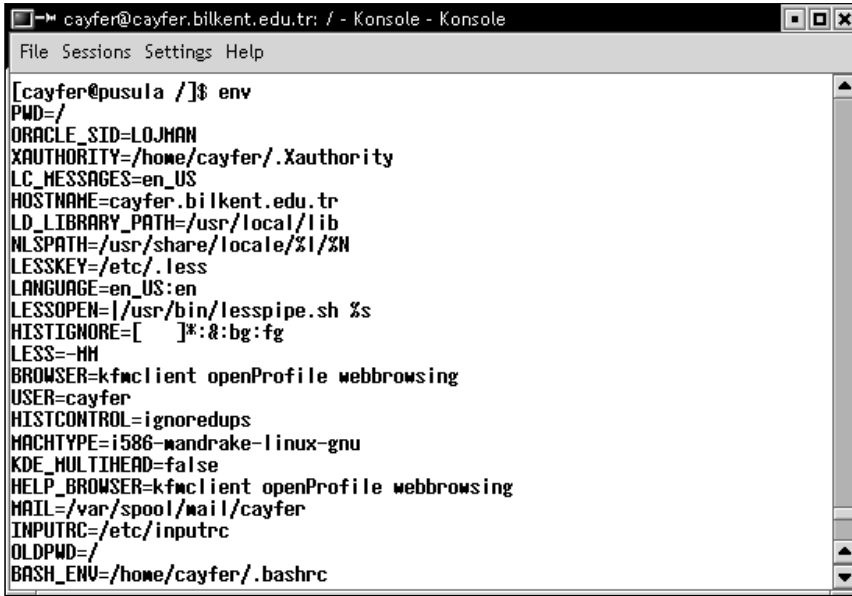
Bazı Önemli bash Kabuk Değişkenleri	
PATH	Bir komut verildiğinde, komut programını oluşturan dosyanın aranacağı dizinler listesini içeren değişkendir.
HOME	Kullanıcının kişisel dizininin adını içeren değişkendir. Sisteme bağlandığınızda kabuk programı tarafından otomatik olarak yaratılır.

TERM	Kullandığınız terminalin tipini belirleyen değişkendir. X altında çalışırken genellikle “ xterm ” değerini içerir. Windows bir makinadan telnet ile bağlanıldığında ise “ vt100 ” değeri verilmelidir; daha doğrusu önerilir.
DISPLAY	X altında çalışırken grafik ekran görüntülerinin gönderileceği X sunucusunun adresini ve ekran numarasını içermelidir. Kendi bilgisayarınızda çalışırken “ :0.0 ” değerini içerir. Bir başka bilgisayarda çalıştıracağınız X uygulamasının görüntüsünü kendi ekranınıza almak istediğinizde o diğer bilgisayara bağlantı kurmak için kullandığınız terminal ekranında çalışmakta olan kabuk programının DISPLAY değişkeninde “ 192.168.10.32:0 ” gibi bir değer olmalıdır. Buradaki 192.168.10.32 yerine önünde oturduğunuz grafik ekranın takılı olduğu bilgisayarın IP numarası veya adı gelmelidir.
HISTSIZE	Kabuk programınızın, en son verilen kaç komutu saklayacağını gösteren değeri içerir. Kabuk programları, her kullanıcının kullandığı son HISTSIZE komutu kişisel dizinlerdeki .bash_history isimli dosyalarda saklar.
SHELL	O anda çalışmakta olan kabuk programının yeri ve adıdır. (/bin/bash gibi.)
LD_LIBRARY_PATH	Programların kullandıkları, paylaşılan kütüphanelerin aranacağı dizinleri sırasıyla gösteren bir karakter dizisi içerir. Kullanım mantığı PATH değişkenine çok benzer; yalnızca aranan program değil, kütüphanedir.

Herhangi bir anda, kabuğunuzda tanımlı olan değişkenleri ve/veya değerlerini merak ederseniz,

env

komutunu kullanabilirsiniz.



```

cayfer@cayfer.bilkent.edu.tr: / - Konsole - Konsole
File Sessions Settings Help

[cayfer@pusula ~]$ env
PWD=/
ORACLE_SID=LOJMAN
XAUTHORITY=/home/cayfer/.Xauthority
LC_MESSAGES=en_US
HOSTNAME=cayfer.bilkent.edu.tr
LD_LIBRARY_PATH=/usr/local/lib
NLSPATH=/usr/share/locale/%l/%N
LESSKEY=/etc/.less
LANGUAGE=en_US:en
LESSOPEN=|/usr/bin/lesspipe.sh %s
HISTIGNORE=[ ]*:&:bg:fg
LESS=-MH
BROWSER=kfmclient openProfile webbrowsing
USER=cayfer
HISTCONTROL=ignoredups
MACHTYPE=i586-mandrake-linux-gnu
KDE_MULTIHEAD=false
HELP_BROWSER=kfmclient openProfile webbrowsing
MAIL=/var/spool/mail/cayfer
INPUTRC=/etc/inputrc
OLDPWD=/
BASH_ENV=/home/cayfer/.bashrc

```

Kabuk değişkenleri, standart isimli birtakım değişkenlerle sınırlı değildir. Kullandığınız uygulama programları, çalışma ortamını tanımlamak için özel değişkenlerin tanımlanmasını ve özel değer verilmesini gerektirebilir. Örneğin Oracle istemcileri **ORACLE_HOME** ve **ORACLE_SID** isimli iki kabuk değişkeni tanımlanmış olmasını ister.

Bir nedenle bir kabuk değişkeni tanımlamak ya da tanımlı bir değişkenin adını değiştirmek gerektiğinde; örneğin **DISPLAY** değişkenine “**192.168.10.33:0**” değerini vermeniz gerektiğinde

export DISPLAY=192.168.10.33:0

komutunu kullanmalısınız.

Komut satırından istediğiniz gibi kabuk değişkeni tanımlayabilir ya da değerlerini değiştirebilirsiniz; ancak bu değişiklik kalıcı olmaz. Kabuk programı kapatıldığında ya da öldürüldüğünde tanımladığınız değişken de kaybolur gider. Ayrıca bu şekilde yapılan bir tanım, yalnızca tanımın yapıldığı kabuk için geçerlidir. Yani, bir anda aynı bilgisayara bağlı olan üç telnet pencereniz varsa, bunlardan birinde vereceğiniz **export** komutu yalnızca o terminal penceresinde çalışan kabuk için geçerli olacaktır. Tüm pencereleriniz de geçerli ve kalıcı bir kabuk değişkeni tanımlı yapmak istiyorsanız, bu işi ya-

pan **export** komutunu kişisel dizininizdeki **.bashrc** dosyasına eklemelisiniz. Bu tanımın tüm kullanıcılar için geçerli olmasını istiyorsanız, aynı eklemeyi **/etc/bashrc** dosyasına da yapmalısınız (nokta yok).

Programları Arka Planda Çalıştırmak

Diyelim ki, çok büyük bir disk dosyasındaki (söz gelimi 200 Mbyte) müşteri kayıtlarını alfabetik sırada dizmek istiyorsunuz. Bu işin, kullandığınız bilgisayar sisteminde yarım saat süreceğini varsayalım. Eğer tek iş düzeninde çalışan bir işletim sistemi kullanıyor olsaydınız, **sort** sıralama komutunu verdikten sonra yemeğe çıkabilir veya köpeğinizi dolaştırmaya götürebilirdiniz; çünkü sıralama bitinceye kadar bilgisayarınızdan bir başka amaçla yararlanmanız söz konusu olamazdı. Oysa LINUX işletim sisteminde, sıralamayı arka planda bir iş olarak başlattıktan sonra, ön planda başka işler yapmanız mümkündür. Bunu yapabilmek için tek yapmanız gereken, arka planda yapılmasını istediğiniz işi başlatan komutun sonuna bir “&” işareti eklemektir.

```
sort musteri_dosyasi > sirali_dosya &
```

Diyeceksiniz ki “arka planda çalıştırmaya ne gerek var, yeni bir telnet penceresi açar, uzun işi orada başlatırım ve normal çalışmamda kullandığım pencereye geri dönerim!” Doğru! Yapabilirsiniz. Mis gibi de çalışır. Geri planda iş çalıştırmak için komut sonuna “&” işareti koyma fikri, yeni bir pencere açmaya üşenenler içindir...

Elbette ki her iş bu şekilde arka planda çalıştırılmaya uygun değildir. Örneğin, bir editör programı gibi, kullanıcının sürekli olarak klavyeden bilgi girmesini gerektiren programlar arka planda çalıştırılsa bile, sürekli ilgi istedikleri için bu tip bir çalışma anlamlı olmaz. Oysa yukarıdaki sıralama örneğimizde, sıralama süresince kullanıcıdan herhangi bir bilgi istenmeyecektir. Sıralama programı arka planda sessizce çalışıp işini bitirecektir.

Bazı programlar kullanıcıdan bir bilgi istememekle birlikte, sürekli olarak ekrana yaptıkları işin gelişmesini açıklayan bilgiler dökerler. Bu tip bir programı arka planda çalışmak üzere başlattığınızda, sürekli olarak ekrana gelen bilgiler yüzünden ön planda başka bir iş yapmanıza pek olanak kalmaz. Örneğin, genellikle bir dizindeki tüm dosya ve alt dizinleri tek bir dosya içinde toplamak (paketlemek) için kullanılan **tar** komutunu,

```
tar -cvf hepsi.tar /home/ayfer &
```

şeklinde verirseniz (bu komutla ilgili detaylı bilgiyi daha ileride vereceğiz; şimdilik komutun ne yaptığı ve parametrelerinin ne olduğu üzerinde durmayınız), program arka planda işini yapacaktır, ama bir yandan da paketlediği dosyaların isimlerini ekrana listeleyecektir. Bu şekilde her saniye yeni bir satır listelenen bir ekranda başka bir iş yapmak pek kolay olmayacaktır.

Ancak aynı komutu,

```
tar -cvf hepsi.tar /home/ayfer > /tmp/tarmesajlari &
```

şeklinde verirseniz, ekrana gelmesi gereken tüm mesajlar, **/tmp** dizinindeki **tarmesajlari** isimli bir dosyaya yönlendirilmiş olacaktır. İş bittikten sonra **tarmesajlari** dosyasına bakarak paketleme işinin başarıyla bitip bitmediğini ve kopyalanan dosyaların listesini görebilirsiniz.

Geri planda başlattığınız işlerin hata mesajı üretmeleri durumunda mesajlar **STDERR**'e gidecektir. Çalıştığınız ekrana zırt pırt hata mesajı gelmesini istemiyorsanız programı başlatırken hata mesajlarını; daha doğrusu **STDERR**'i bir dosyaya ya da dipsiz kuyuya yönlendirebilirsiniz.

```
cp -r /home/cayfer/tmp/* /yedeck 2> /tmp/hatalar  
cp -r /home/cayfer/tmp/* /yedeck 2> /dev/null
```

Dikkat ederseniz, **STDERR**'i yönlendirirken **"2>"** karakterleri kullanılıyor. Oysa **STDOUT** yönlendirilirken yalnızca **">"** kullanılıyordu.



Ön Planda Çalışan Programları Arka Plana Atmak

Bazı durumlarda, başlattığınız bir programın ne kadar süreyle çalışacağını önceden kestiremezsiniz. İşin uzun süreceğini ve sessiz çalışan bir iş olduğunu sonradan fark edersiniz ya da işin bu özelliklerini bilseniz bile, boş bulunup komut satırının sonuna **"&"** koymadan Enter tuşuna basıverirsiniz. Örneğin, bir dizindeki tüm dosyaları bir başka diske ya da dizine kopyalama komutunu,

Kim Korkar LINUX'tan?

```
cp -r /home/ayfer /disk2/home2
```

şeklinde verdiğiniz ve programı ön planda başlattığınızı varsayalım. Başlattıktan birkaç saniye (ya da birkaç dakika) sonra işin uzun süreceğini fark ettiniz ve “*Tüh! Keşke arka planda başlatsaydım!*” dediniz. Eğer kabuk programı olarak **bash** kullanıyorsanız sorun değil...

Klavyenizden,

^Z (Ctrl-Z)

tuşuna basarsanız (Control tuşu basılıyken Z tuşuna da basarsanız) ekranda,

Suspended.

mesajını görürsünüz. Bu mesaj, o sırada ön planda çalışan işinizin geçici olarak askıya alınarak durdurulduğunu göstermektedir. Ancak, buradaki “durduruldu” ifadesi, işinizin tamamlanmadan kesildiği anlamında değildir. Buradaki durdurma, müzik kaseti çalan teyplerdeki Pause düğmesinin görevine benzeyen bir durdurmadır. İşiniz çalışmaya ara vermiş ve devam edebilmek için sizden bir komut bekler durumdadır.

Bu noktada,

bg (Background)

komutu vererseniz, işiniz arka planda çalışmaya devam edecektir. Ancak, o anda ekranda bir de,

[1] cp ... &

mesajı görünecektir. Bu mesajın anlamı kısaca programınızın geri plana çekildiğini ve geri planda çalışan işleriniz arasındaki numarasının 1 olduğunu. Geri plandaki 1 numaralı işi tekrar ön plana almak isterseniz,

fg %1 (Foreground)

komutunu verebilirsiniz. (Eğer arka plana atılmış tek bir işiniz varsa, “%” işaretini ve ardındaki numarayı girmeniz gerekmez.)

Arka planda çalışmak üzere başlatılacak, ya da sonradan arka plana atılacak işlerin sayısı ile ilgili herhangi bir sınırlama yoktur. Ancak, arka plan ya da ön plan olsun, çalışan her işin bilgisayarın performansından bir pay alacağını unutmamalısınız.

Bazen, arka planda başlattığınız ya da sonradan arka plana attığınız işlerin hesabını şaşırtabilirsiniz. Böyle bir durumda,

jobs

komutunu vererseniz, arka plana atılmış işlerin bir listesini alırsınız.

Kabuk Programlama

Bu bölümde amacımız, okuyuculara kabuk programlamayı öğretmek değil, sadece bu kavramın nasıl bir şey olduğu konusunda fikir vermektir. Aslında oldukça karmaşık bir iş olan ve deneyim isteyen kabuk programlama, programcılık deneyimi olmayan LINUX kullanıcılarının pek ilgisini çekmemekle birlikte programcılık temeli olan okuyuculara oldukça ilginç gelebilir.

LINUX kabukları (**bash**, **csh**, **tsch** gibi) aslında oldukça gelişmiş birer programlama dilini çözümleyebilecek yeteneğe sahiptirler. Genel amaçlı işler için pek kullanışlı olmamakla birlikte ileri düzeydeki kullanıcıların ve sistem yöneticilerinin en değerli araçlarından biridir.

csh ve **bash** kabuk programlama dilleri birbirlerinden oldukça farklıdır. Bu kitapta vereceğimiz örnekler yalnızca **bash** kabuğu için olacaktır.

Her ne kadar LINUX kullanmak için **bash** ya da bir başka kabukla programlama yapmaya gerek olmasa da LINUX/UNIX sistem yöneticisi olmayı düşünenlerin bu işlerden biraz anlaması çok yararlı olacaktır. Kabuk programlama konusunda daha fazla ayrıntıya girmek isteyen okuyucular için internet'te "*bash programlama*", "*bash programming*" anahtar sözcükleriyle bir tarama yapmalarını öneririz.

Şimdi isterseniz birkaç örnek kabuk programına göz atalım:

İlk Kabuk Programı Örneği

İlk örnek çok da anlamlı bir iş yapmamakla birlikte kabuk programlarının neye benzediğini göstermek açısından tipik bir **bash** kodudur.

Öncelikle aşağıdaki LINUX komutlarını **vi** editörünü kullanarak “**ornek1.sh**” isimli bir dosyaya kaydediniz. (Aslında **vi** editörünü kullanmak zorunda değilsiniz elbette ama kabuk programlamayla ilgilendiğinize göre sistem yöneticisi olma yolunda ilerliyorsunuz demektir; bu durumda da **vi** bilmeden olmaz. Neyse, şimdilik istediğiniz editörü kullanabilirsiniz.)

```
#!/bin/bash
COUNT=1
while [ $COUNT -le 100 ]
do
ls | wc -l
sleep 2
COUNT=`expr $COUNT + 1`
done
```

Yukardaki **bash** kabuk programı iki saniyede bir toplam 100 kez “**ls**” komutunu çalıştırıp bu komutun ürettiği satırları sayacaktır.

Şimdi bu kısa bash programının satırlarına bir göz gezdirelim:

#!/bin/bash: **bash** kabuk programlarında açıklama satırları (*comment*) “**#**” ile başlar. Bu satır ilk bakışta bir açıklama satırı gibi görünse de aslında çok özel bir anlamı vardır. Bir kabuk programının ilk satırı olması; ayrıca ilk iki karakterinin “**#!**” olması, bu kabuk program dosyasındaki satırların **/bin/bash** programıyla yorumlanması gerektiğini belirtmektedir. Bir kabuk programının başında, programın hangi kabukla çalıştırılacağını belirtmesi oldukça önemlidir. Bu satır olmadığında, kabuk programı yalnızca **bash** kabuğu ile çalışan kullanıcılar tarafından başarıyla çalıştırılabilir. Oysa bu satır sayesinde, kullanıcının komutu verdiği sırada kullandığı kabuk ne olursa olsun, önce **/bin/bash** kabuğu başlatılacak, program da bu kabuk içinde çalıştırılacaktır. Program bittiğinde bu program için başlatılmış kabuk da bitecek ve kullanıcı kendi kabuğuna geri dönecektir.

COUNT=1: **COUNT** isimli bir değişkene 1 değeri atanıyor. Kabuk değişkenlerinin isimleri genellikle büyük harflerle yazılır. Bu kesin bir kural olmamak-

la birlikte kullanılacak değişkenlerin isimlerinin LINUX komutlarıyla karışmaması için yararlı bir gelenektir.

while [**\$COUNT -le 100**]: **\$COUNT** değişkeninin değeri 100'den küçük veya eşit olduğu sürece tekrarlanacak bir döngü başlıyor.

do: **while** kalıbının bir parçasıdır. Döngü içindeki komut satırları “**do-done**” deyimleri arasına yazılır.

ls | wc -l: “**ls**” komutu ve “**wc -l**” komutları birlikte çalıştırılıp **ls** komutunun çıktıları **wc** komutuna girdi olarak yönlendirilip satırlar sayılıyor.

sleep 2: İki saniye beklemek için **sleep** komutu.

COUNT=`expr \$COUNT + 1`: **\$COUNT** değişkeninin değerini bir arttıran komut.

done: **while** döngüsündeki deyim grubunun sonunu belirliyor.

ornek1.sh isimli dosyadaki bu **bash** komutlarını bir program gibi çalıştırmak için sisteme bu dosyanın “**çalıştırılabilir**” program dosyası olduğunu belirtmek gerekir. Bu komutu kimlerin kullanabileceğine bağlı olarak

chmod a+x ornek1.sh

veya

chmod u+x ornek1.sh

veya

chmod 755 ornek1.sh

gibi bir komutla dosyanın çalıştırma yetkilerini düzenleyebilirsiniz.

Son olarak da komutu çalıştırmak için:

./ornek1.sh

komutunu verebilirsiniz. Eğer **PATH** değişkeninizde “çalışma dizini” anlamında “.” yoksa, komutun başına “bu dizindeki” anlamına gelen “./” işaretini koymanız; yani komutu “**./ornek1.sh**” şeklinde yazmanız gerekecektir.

İkinci Kabuk Programı Örneği

Bu örneğimiz ileri düzey bir **bash** programı olacak. Amacımız **bash** programlamanın tüm detaylarını öğretmek olmadığı için bu programın ayrıntılarına girmeyeceğiz. Programcılık deneyimi olup bash programlamayı ayrıntılı olarak öğrenmek isteyen kullanıcıların başka kaynaklara yönelmesi gerekecektir.

```
#!/bin/bash
# lsx : Dizin yapısını hiyerarşik olarak görüntüler.

ara () {                                # Fonksiyon tanımı
  for DIZIN in `echo *`
  do
    if [ -d $DIZIN ] ; then
      W_DUZEY=0
      while [ $W_DUZEY != $DUZEY ]
      do
        echo -n "| "
        W_DUZEY=`expr $W_DUZEY + 1`
      done
      if [ -L $DIZIN ] ; then
        echo "+---$DIZIN" `ls -l $DIZIN | sed 's/^.*'$DIZIN' //'`
      else
        echo "+---$DIZIN"
        if cd $DIZIN ; then
          DUZEY=`expr $DUZEY + 1`
          ara
          DIR_SAYISI=`expr $DIR_SAYISI + 1`
        fi
      fi
    fi
  done
  cd ..
  if [ $DUZEY ] ; then
    DEVAM=1
  fi
  DUZEY=`expr $DUZEY - 1`
}
```

```

#
# Başlıyoruz... (Ana program)
#
CALISMA_DIZINI=`pwd`
if [ $# = 0 ] ; then
    cd `pwd`
else
    cd $1
fi

echo "Başlangıç dizini : `pwd`"
DEVAM=0
DUZEY=0
DIR_SAYISI=0
W_DUZEY=0
while [ $DEVAM != 1 ]
do
    ara
done
echo "Toplam $DIR_SAYISI dizin bulundu."
#
# Program başlatıldığındaki çalışma dizinine geri dön
cd $CALISMA_DIZINI

```

Yukardaki programı dikkatle incerseniz C, PASCAL gibi genel amaçlı programlama dillerindeki komut yapılarının eşdeğerlerini, hatta özyinelemeli (*recursive*) fonksiyonların dahi kullanılabildiğini göreceksiniz. Hatırlarsanız, **bash** kabuğunun aslında çok gelişmiş bir programlama aracı olduğunu daha önce de vurgulamıştık.

Şimdi bu **bash** programını bir UNIX komutu gibi kullanabilmek için yapmanız gerekenleri bir gözden geçirelim:

Öncelikle **bash** kodunu içeren dosyanın “çalıştırılabilir” bir dosya olması gerekir:

```

chmod 755  lsx           veya
chmod a+x  lsx

```

gibi komutlar bir dosyayı herkes tarafından çalıştırılabilir hale getirilebilir.

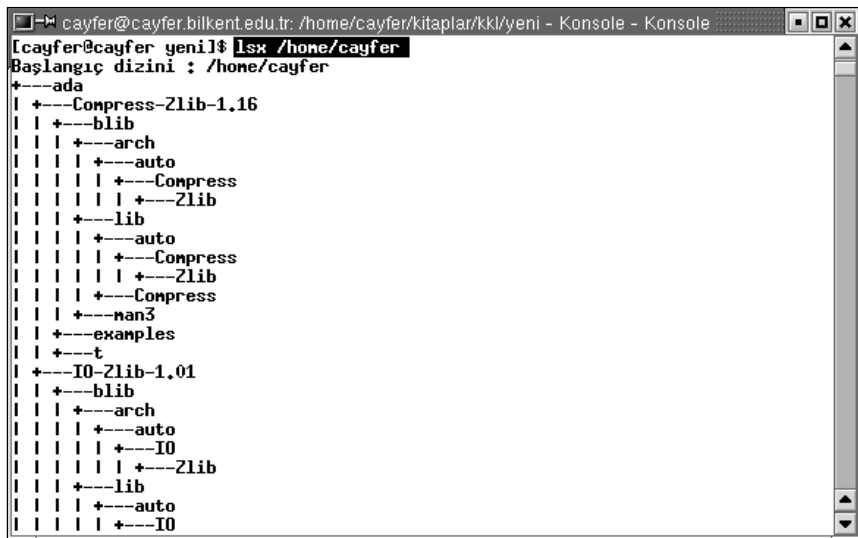
Kim Korkar LINUX'tan?

Dosyaya herkes kolayca erişebilsin diye dosya, kodu yazan programcının kişisel dizininden, **/usr/local/bin** dizinine kopyalanmalıdır. **/usr/local/bin** dizinine dosya kopyalayabilmek için root kullanıcı kimliğine bürünmek gerekecektir:

```
su -
```

```
cp /home/cayfer/lxx /usr/local/bin
```

Artık **PATH** değişkeninde **/usr/local/bin** olan tüm kullanıcılar rahatlıkla yeni **lxx** komutunuzu kullanabilecektir.



```
cayfer@cayfer.bilkent.edu.tr: /home/cayfer/kitaplar/kkl/yeni - Konsole - Konsole
[cayfer@cayfer yeni]$ lxx /home/cayfer
Başlangıç dizini : /home/cayfer
+---ada
| +---Compress-Zlib-1.16
| | +---blib
| | | +---arch
| | | | +---auto
| | | | | +---Compress
| | | | | | +---Zlib
| | | | | +---lib
| | | | | +---auto
| | | | | +---Compress
| | | | | | +---Zlib
| | | | | +---Compress
| | | | +---nan3
| | | +---examples
| | | +---t
| | +---IO-Zlib-1.01
| | +---blib
| | | +---arch
| | | | +---auto
| | | | | +---IO
| | | | | | +---Zlib
| | | | | +---lib
| | | | | +---auto
| | | | | +---IO
```

Diskler, Disketler ve mount Komutu

08

- **mount Edilmiş Diskleri Görmek**
- **Yeni Disk Takma**
- **Dosya Sistemi Tipleri**
- **fsck: Dosya Sistemi Kontrol Programı**
- **supermount**
- **CD Kullanımı**
- **Disket Kullanımı**
- **umount Komutu**
- **Başka UNIX Bilgisayarlardaki Diskleri Kullanmak**
- **mount İşlemlerini Otomatik Olarak Yapmak (fstab)**
 - **Başka Bir Windows Bilgisayarın Diskini Kullanmak**
- **Dosya Sistemi Yaratmanız Gerekirse...**

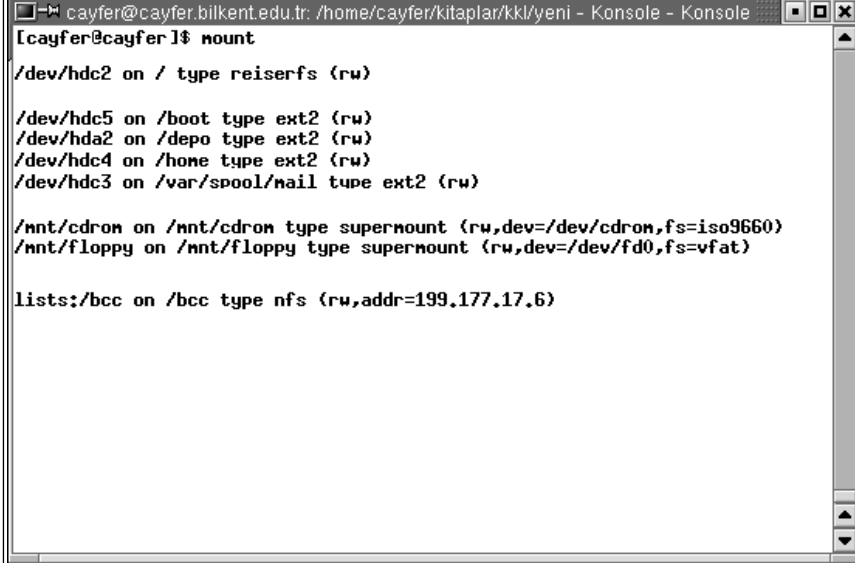
Şimdi biraz eski bilgileri tazeleyelim:

- Bir LINUX bilgisayarındaki tüm diskler tek bir dizin altında; yani “/” dizini altında toplanır.
- LINUX'ta, bir dosyanın yeri tarif edilirken hangi diskte bulunduğu değil, hangi dizinde yer aldığı belirtilir. Söz konusu dizinin gerçekte hangi fiziksel diskte yer aldığını, belirtilen dizine iliştilmiş (mount edilmiş) olan disk ya da disk bölümü belirler.
- Bir LINUX bilgisayarındaki her bir donanım unsuru (fare dahil) **/dev** dizini altında bir dosyaya; daha doğrusu bir “düğüm”e iliştilmiş olmalıdır. LINUX çekirdeği, bu “düğüm”ün özelliklerinden ilgili donanım parçasını hangi çekirdek modülünün denetleyeceğini anlar.
- LINUX'ta birinci IDE kanalının “master” diski **/dev/hda** olarak isimlendirilir. Birinci kanalın “slave” diski **/dev/hdb**; ikinci IDE kanalının “master” ve “slave” diskleri de sırasıyla **/dev/hdc** ve **/dev/hdd** olarak anılır.

- Varsa; SCSI diskler ise SCSI adreslerine göre **/dev/sda**, **/dev/sdb** gibi isimlerle anılır.
- Sistemin açılışı sırasında disk bölümlerinin hangi dizinlere iliştilirileceği, yani mount edileceği **/etc/fstab** dosyasında belirtilir.
- “/” dizini mount edilmeden sistem açılmaz.
- LINUX, çok çeşitli disk formatlarını, örneğin FAT16, FAT32 ve NTFS formatlı diskleri görebilir.
- Yalnızca üzerinde LINUX tarafından tanınan, geçerli bir “dosya sistemi” bulunan çevre birimleri “**mount**” edilebilir. Bunlar formatlı diskler ve disk bölümleri, formatlı disketler, dolu CD’ler, DVD’ler ve ZIP sürücüsü gibi özel bazı donanımlardır. Teyp kasetleri ve kartuşları **mount** edilemez çünkü veri içerse bile teyp kasetlerinin üzerinde bir dosya sistemi (*file system*) yoktur.

mount Edilmiş Diskleri Görmek

Bilgisayarınızda hangi disk bölümlerinin nerelere mount edilmiş olduğunu merak ettiğinizde **mount** komutunu parametresiz olarak kullanmalısınız.



```
[cayfer@cayfer]# mount
/dev/hdc2 on / type reiserfs (rw)

/dev/hdc5 on /boot type ext2 (rw)
/dev/hda2 on /depo type ext2 (rw)
/dev/hdc4 on /hone type ext2 (rw)
/dev/hdc3 on /var/spool/mail tupe ext2 (rw)

/nnt/cdrom on /nnt/cdrom type supermount (rw,dev=/dev/cdrom,fs=iso9660)
/nnt/floppy on /nnt/floppy type supermount (rw,dev=/dev/fd0,fs=vfat)

lists:/bcc on /bcc type nfs (rw,addr=199.177.17.6)
```

Bu “mount” komutu çıktısı, listenin kolay algılanması için biraz kısaltılmış ve yeniden düzenlenmiştir.

Yukarıdaki örnekte sistemin ikinci IDE kanalındaki master diskin ikinci bölümünün, yani **/dev/hdc2**'nin “/” dizini olarak **mount** edildiği ve dosya sistemi tipinin “reiserfs” olduğu görülmektedir.

Bu bilgisayar “**lists**” isimli bir başka bilgisayarın **/bcc** dizinini **nfs** üzerinden kendi **/bcc** dizinine bağlamış (mount etmiş) görünüyor.

Öte yandan birinci IDE kanalının ilk diskinin ikinci bölümünün de **/depo** dizinine iliştilendiği görülüyor. Bu **mount** işleminin yapılabilmesi için “/” dizini altında **/depo** isimli boş bir dizinin bulunması gerekir.

Biraz karışık oldu; değil mi? En iyisi bir senaryo üzerinde açıklamak galiba...

Yeni Disk Takma

Şimdi farzedin ki sisteminizin ikinci IDE kanalına “**slave**” olarak yeni bir disk taktınız. Bu diskte de tek bir disk bölümü var.

Bu durumda diskinizin adı **/dev/hdd**; üzerindeki yegane disk bölümünün adı da **/dev/hdd1** olacaktır.

Diski fiziksel olarak takıp bilgisayarınızı açmanız bu diske erişebilmeniz için yeterli değildir. Önce diskin bölümlendirilmesi ve ardından formatlanması ondan sonra da üzerine boş ve yeni bir dosya sistemi yaratılması gerekir.

Diskleri bölümleme ve formatlama işlemi LINUX'un **fdisk** komutuyla yapılır. Dosya sistemi yaratma işi de yaratmak istediğiniz dosya sistemine göre **mke2fs**, **mke3fs** veya **mkreiserfs** komutlarıyla yapılır. (Özel bir nedeni olmadıkça tüm disk bölümlerinizi “**reiserfs**” veya “**ext3**” dosya sistemleriyle donatmanızı öneririz.)

Bu aşamada bilgisayarınıza taktığınız yeni diskin üzerinde tek bir bölümün yer aldığını ve bu bölüm üzerinde de “**reiserfs**” formatında bir dosya sistemi yaratılmış olduğunu varsayıyoruz. Bölümleme ve dosya sistemi yaratma işlemlerini daha ilerideki bölümlerde açıklayacağız.

Önce bu yeni diski iliştmek için bir dizin seçmelisiniz (*mount point*). Eğer bu amaçla seçebileceğiniz bir dizin yoksa, yaratmalısınız. Yeni diskinizi, ör-

neğin yedekleme amacıyla kullanacaksınız, “**yedek**” isimli bir dizine ilişti-
mek anlamlı olabilir. Bu durumda bir kereye özgü “/” dizininiz altında “**ye-
dek**” isimli bir dizin yaratıp bundan sonra her açılışta uygun mount komu-
tunu vermeniz; daha doğrusu her açılışta **/dev/hdd1** disk bölümünün **/ye-
dek** dizinine iliştilmesini sağlamanız yeterli olacaktır:

```
su -  
mkdir /yedek  
mount -t reiserfs /dev/hdd1 /yedek
```

Bu işi yapabilmek için “**root**” kullanıcı kimliğine bürünmeniz gerektiğini
söylemeye gerek yok herhalde...

Eğer bir hata mesajı almazsanız **mount** işlemi başarıyla tamamlanmış demek-
tir. Tüm UNIX’lerde olduğu LINUX’ta da “*No news is good news!*”; yani
itiraz gelmezse işler yolunda demektir.

Şimdi artık yeni diskinizi kullanabilirsiniz; örneğin yedekleme amacıyla tak-
tığınız bu diske eski disklerinizden dosyaları kopyalamaya başlayabilirsiniz:

```
mkdir /yedek/home  
cp -r /home/cayfer /yedek/home
```

gibi...



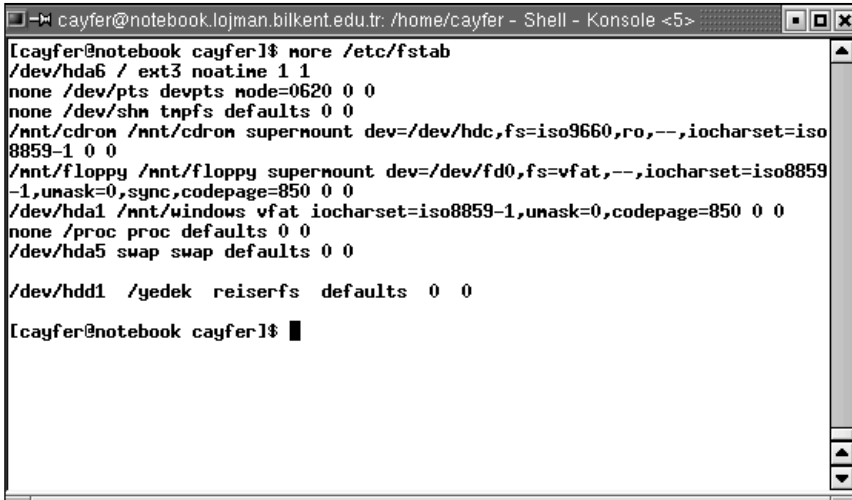
Yeri gelmişken: Yedekleme işlerini **cp** komutuyla yapmak pek iyi bir yöntem
değildir. İlk sorun dosya-dizin okuma yetkilerinde çıkar. Bir dosyayı kopya-
layabilmek için önce o dosyayı okuyabilmeniz gerekir. Doğal olarak root dı-
şında hiçbir kullanıcı diskteki tüm dosyaları okumaya yetkili değildir. Sorun
çıkmasın diye **cp** komutunu root kullanıcı olarak verirsiniz bu kez yeni kop-
yaların sahibi “root” olur. Kitabın sonraki bölümlerinde yedekleme işleri için
başka komutlar ve yöntemlerden söz edeceğiz. Dosya ve dizin kopyalarken
bu yeni kopyaların orjinallerini aynı kullanıcıya ait olmalarını sağlamak
gerekliyorsa **cp** komutunun **-p** parametresi sorunu çözecektir (**-p**: pre-
serve ownership).

mount komutuyla yeni diski **/yedek** dizinine iliştip kullanmayı başardınız
ama sistemi her açtığınızda bu **mount** komutunu tekrarlamanız gerekecek.
Sistemin her açılışında **/dev/hdd1** diskinin **/yedek** dizinine otomatik olarak

bağlanması için yapılması gereken iş **/etc/fstab** dosyasına uygun bir satır eklemek olacaktır. **root** kullanıcı olarak aşağıdaki komutla **/etc/fstab** dosyasına yeni bir satır ekleyebilirsiniz:

```
cat >> /etc/fstab
/dev/hdd1 /yedik reiserfs defaults 0 0
```

Aslında bu satır ekleme işini **vi** ile yapmak çok daha mantıklıdır elbette; ama **cat** komutunu öylesine bir hatırlatmak istedik.



```
cayfer@notebook.lojman.bilkent.edu.tr: /home/cayfer - Shell - Konsole <5>
[cayfer@notebook cayfer]$ more /etc/fstab
/dev/hda6 / ext3 noatime 1 1
none /dev/pts devpts mode=0620 0 0
none /dev/shm tmpfs defaults 0 0
/mnt/cdrom /mnt/cdrom supermount dev=/dev/hdc,fs=iso9660,ro,--,iocharset=iso
8859-1 0 0
/mnt/floppy /mnt/floppy supermount dev=/dev/fd0,fs=vfat,--,iocharset=iso8859
-1,unask=0,sync,codepage=850 0 0
/dev/hda1 /mnt/windows vfat iocharset=iso8859-1,unask=0,codepage=850 0 0
none /proc proc defaults 0 0
/dev/hda5 swap swap defaults 0 0

/dev/hdd1 /yedik reiserfs defaults 0 0
[cayfer@notebook cayfer]$
```

Artık sisteminiz her açıldığında yedekleme amacıyla kullandığınız disk **/yedik** dizinine otomatik olarak iliştilirilecektir.

Dosya Sistemi Tipleri

Bir dizine iliştiireceğiniz dosya sisteminin tipini mount komutunda **-t** parametresiyle belirtmelisiniz. LINUX'un tanıdığı bazı önemli dosya sistemi tipleri şunlardır:

Dosya Sistemi	-t Parametresi
Eski LINUX standart disk dosya sistemi.	ext2
Modern LINUX disk dosya sistemi.	ext3
Bir başka modern LINUX disk dosya sistemi.	reiserfs
MS-DOS dosya sistemleri.	msdos
NT Uyumlu (Win2K, XP) dosya sistemleri.	ntfs
CD dosya sistemleri.	iso9660
Windows-9x uyumlu dosya sistemleri.	vfat
Dağıtık (farklı bilgisayarlarda yer alan) dosya sistemi. (Network File System)	nfs

Yukarıdaki tabloya göre, MS-DOS formatlı bir disketi **/mnt/floppy** isimli bir dizine iliştiirmek için kullanmanız gereken

```
mount -t msdos /dev/fd0 /mnt/floppy
```

komutudur.

Bilgisayarınıza takacağınız bir NT diskini okuyabilmek için

```
mkdir /nt_diski
```

```
mount -t ntfs /dev/hdd1 /nt_diski
```

gibi komutlar işe yarayacaktır.

fsck: Dosya Sistemi Kontrol Programı

Disklerde kullandığınız dosya sistemi hangisi olursa olsun günün birinde bu dosya sistemindeki veri yapısının bozulma olasılığı vardır. Dosya sistemlerini oluşturan veri yapıları, disklerde kullanılan ve boş veri bloklarının yönetimini sağlar. Disk dosyalarının fiziksel olarak yerleştirildiği blokların hesabının tutulduğu, izinin sürüldüğü bu veri yapılarında, hızlı erişim için bir sürü karmaşık teknik kullanılmaktadır. Bu veri yapılarının güncelleştirilmesi sırasında kötü bir zamana denk gelebilecek bir enerji kesintisi ya da donanım arızası dosya sisteminin bütünlüğüne ve tutarlılığına zarar verebilir. Her ne kadar **ext3** ve **reiserfs** dosya sistemlerinde bu olasılık büyük ölçüde azaltılmış olsa da gene sıfırdan büyüktür.

LINUX, sistemin açılışı sırasında “mount edilen” tüm disk bölümlerine birer işaret koyar. Sistemin düzgün kapatılması sürecinde de başarıyla çözülen dosya sistemlerinden bu işaretleri kaldırır.

Açılış sırasında “mount edildi” işareti içeren dosya sistemlerine rastlanırsa LINUX, bilgisayarın normal yollarla kapatılmadığını, dolayısıyla bu dosya sisteminin veri yapılarında bir sorun olabileceği varsayımıyla otomatik olarak o dosya sistemi için “**fsck**” (*file system check*) programını çalıştırır.

fsck, dosya sisteminde bir bozukluk görürse düzeltmeye çalışır, genellikle de başarılı olur. Bu düzeltmeleri yapmadan, önce bir mesajla bulduğu sorunu gösterir ve sistem yöneticisinden düzeltme için izin ister:

```
cayfer@notebook.ijman.bilkent.edu.tr: /home/cayfer - Shell - Konsol <5>
[cayfer@notebook cayfer]# fsck /dev/hda6
Parallelizing fsck version 1.15 (18-Jul-1999)
e2fsck 1.15, 18-Jul-1999 for EXT2 FS 0.5b, 95/08/09
/dev/hda6 contains a file system with errors, check forced.
Pass 1: Checking inodes, blocks, and sizes
Pass 2: Checking directory structure
Pass 3: Checking directory connectivity
Pass 4: Checking reference counts
Pass 5: Checking group summary information
Block bitmap differences: -933729 -953798 -953799 -1021454 -1021
455 -1021456 -1023337 -1023338 -1023339 -1023340 -1023341 -102334
2 -1023343 -1023344
Fix<y>? yes

Free blocks count wrong for group #21 (0, counted=4).
Fix<y>? yes

/dev/hda6: ***** FILE SYSTEM WAS MODIFIED *****
/dev/hda6: 76493/525888 files (2.5% non-contiguous), 986179/10502
41 blocks
[cayfer@notebook cayfer]#
```

Genellikle bu izni vermemeniz için bir neden yoktur ve tüm sorulara “yes” diyerek devam etmeniz gerekir.

LINUX, dosya sistemlerinin kaç kez **mount** edildiklerinin de hesabını tutar ve her 20 ilişirmede bir, o dosya sistemi için sorun olmasa bile **fsck** çalıştırır.

Çok fazla gerekmemekle birlikte, dosya sistemlerinin bütünlüğünü kendiniz kontrol etmek istediğinizde

fsck /dev/hdc1

gibi komutlarla bu işi yapabilirsiniz. Doğal olarak bu iş için root kullanıcı yetkilerine sahip olmalısınız; ayrıca kontrol edilecek disk bölümünün sistemde herhangi bir dizine iliştirilmemiş olması gerekir. İliştirilmiş (*mounted*) dosya sistemleri üzerinde **fsck** çalıştırmak tehlikelidir. Eğer buna mecbur kalırsanız, **fsck** süresince, çalışan yazılımların söz konusu dosya sisteminde yer alan hiçbir dosya ya da dizinde değişiklik yapmayacağından emin olmanız gerekir.

supermount

Sistemin **/etc/fstab** dosyasına baktığınızda **/mnt/cdrom** ve **/mnt/floppy** dizinlerinin **/dev/cdrom** ve **/dev/fd0** çevre birimlerinin “supermount” seçeneği ile iliştirildiği dikkatinizi çekmiş olmalı. Bu seçenekler, söz konusu çevre birimlerine CD veya disket takıldığında, **mount** komutu vermeye gerek kalmaksızın takılan medyanın otomatik olarak **/mnt/floppy** veya **/mnt/cdrom** dizinlerine iliştirilmesini sağlamaktadır.

Aynı şekilde; söz konusu sürücülerdeki medyalar çıkarıldığında da ilgili dizinler otomatik olarak **unmount** edilir.

CD Kullanımı

Siz değiştirmediyse, CD sürücünüze ilişkin **/etc/fstab** satırında **supermount** parametresi belirtilmiş olmalıdır. Bu parametre sayesinde CD sürücünüze dolu bir CD taktığınızda, o CD “**mount**” komutunu vermenize gerek kalmadan **/mnt/cdrom** dizinine iliştirilir. Artık o CD'deki dosya ve dizinleri **/mnt/cdrom** dizini altında görebilirsiniz.

Eğer **/etc/fstab** dosyanızda **/dev/cdrom** ile ilgili bir satır yoksa, taktığınız CD'lerin içini görebilmek için

```
mount -r-t iso9660/dev/cdrom /mnt/cdrom
```

gibi bir komut vermeniz gerekecektir. (**-r** parametresini kullanmanız; yani iliştirilecek dosya sisteminin “**yalnız okunabilir: read-only**” olduğunu belirtmeniz şart değildir; ancak herşeyi kuralına göre yapmaya alışmak yararlıdır.)

Disket Kullanımı

Siz değiştirmediyse, disket sürücünüze ilişkin **/etc/fstab** satırında “**supermount**” parametresi belirtilmiş olmalıdır. Bu parametre sayesinde disket sürücünüze formatlı bir disket taktığınızda, o disket “**mount**” komutununa gerek kalmadan **/mnt/floppy** dizinine iliştilir. Artık o disketteki dosya ve dizinleri **/mnt/floppy** dizini altında görebilirsiniz.

Eğer **/etc/fstab** dosyanızda **/dev/fd0** ile ilgili bir satır yoksa, taktığınız disketi kullanabilmek için

```
mount -t msdos /dev/fd0 /mnt/floppy
```

gibi bir komut vermeniz gerekecektir.

umount Komutu

Sisteminizde bir dizine iliştilirilmiş olan bir dosya sistemini çözmek isterseniz “**umount**” komutunu kullanmalısınız. (Tabii ki bu işi gene yalnızca root kullanıcı yapabilir)

Dikkatli okuyucuların sorduğunu duyar gibiyiz: “Bu iş kaç yılda bir gerekecek ki; neden anlatıyorsunuz?” Evet, bazı kullanıcılar için bu işler hiç gerekemeyebilir; ama gerektiği zaman da bunları bilmek zorundasınız. Hele diskinizdeki bir dizine iliştmek isteyebileceğiniz bir diskin bir başka bilgisayarda takılı olabileceğini dikkate alırsanız, bu **mount-umount** komutlarına tahmin ettiğinizden daha çabuk gereksinim duyacağınıza emin olabilirsiniz.



Örneğin; **/yede**k dizinine iliştilirilmiş **/dev/hdd1** diskini çözmek istediğinizde

```
umount /yedek          veya
```

```
umount /dev/hdd1
```

komutlarından birini kullanabilirsiniz.

Başka UNIX Bilgisayarlardaki Diskleri Kullanmak

Disklerinin (daha doğrusu dizinlerinin) başka bilgisayarlar tarafından kullanılmasına izin veren UNIX bilgisayarların bu dizinlerini kendi bilgisayarınızdaki dizinlere iliştirebilirsiniz. Örneğin,

```
mount -t nfs sunucu.abc.com.tr:/yedek /yedek2
```

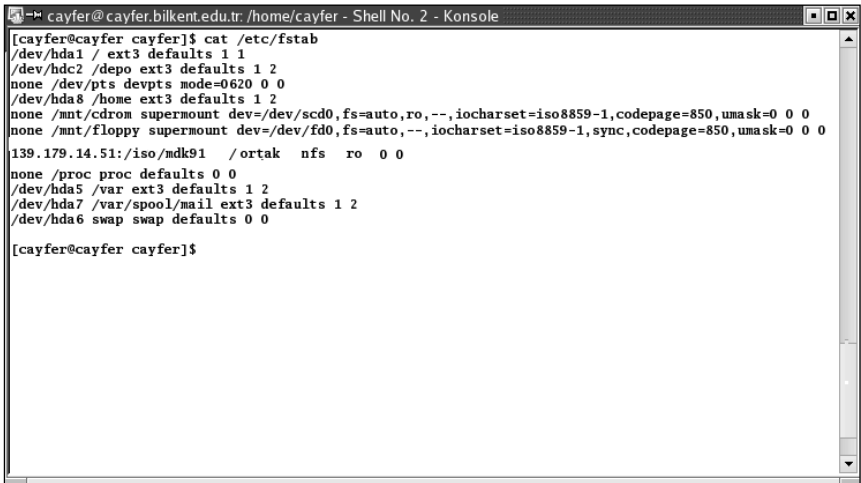
komutu, “**abc.com.tr**” ağındaki “**sunucu**” isimli bilgisayar tarafından **/yedek** adıyla paylaşıma açılmış dizini, kendi bilgisayarınızdaki **/yedek2** dizinine iliştirecektir.

Bir LINUX bilgisayardaki disk/dizinleri paylaşıma açmayı ve bu şekilde paylaşıma açılmış dizinlerin kullanımını düzenleyen yetki konularını daha ileride açıklayacağız.

mount İşlemlerini Otomatik Olarak Yapmak (fstab)

Bilgisayarınız açılırken otomatik olarak çeşitli dizinlere iliştirilecek disk bölümleri **/etc/fstab** dosyasında tanımlanır.

Aşağıdaki örnekte **/etc/fstab** dosyasını incelediğinizde:



```
[cayfer@cayfer cayfer]$ cat /etc/fstab
/dev/hda1 / ext3 defaults 1 1
/dev/hdc2 /depo ext3 defaults 1 2
none /dev/pts devpts mode=0620 0 0
/dev/hda8 /home ext3 defaults 1 2
none /mnt/cdrom supermount dev=/dev/scd0,fs=auto,ro,--,iocharset=iso8859-1,codepage=850,umask=0 0 0
none /mnt/floppy supermount dev=/dev/fd0,fs=auto,--,iocharset=iso8859-1,codepage=850,umask=0 0 0
139.179.14.51:/iso/mdk91 /ortak nfs ro 0 0
none /proc proc defaults 0 0
/dev/hda5 /var ext3 defaults 1 2
/dev/hda7 /var/spool/mail ext3 defaults 1 2
/dev/hda6 swap swap defaults 0 0
[cayfer@cayfer cayfer]$
```

Sistem açılırken hangi disk bölümünün hangi dizine iliştirileceğini göreceksiniz.

Bu örnekte **fstab** dosyasında dikkatinizi çekmek istediğimiz bir satır var:

```
139.179.14.51:/iso/mdk91 /ortak nfs ro 0 0
```

Bu satırda, sistem açılırken **139.179.14.51** IP numaralı bilgisayardaki **/iso/mdk91** dizininin bu makinedeki **/ortak** dizinine iliştiirileceğı belirtiliyor. Dosya sistemi tipi olarak verilmiş olan “**nfs**” parametresi, söz konusu dizinin bir başka UNIX bilgisayar tarafından paylaşımaya açıldığını göstermektedir. Bu örnek dosyada yer alan satırlardaki “**defaults**” sözcüğü **mount** işlemi sırasında kullanılacak seçenekleri; ardından gelen ve “**1**” veya “**0**” olan ilk rakam söz konusu dosya sisteminin **dump** komutu kullanıldığında yedeklenip yedeklenmeyeceğini; son rakam da **fsck** komutu kullanıldığında sistemdeki dosya sistemlerinin hangi sırayla kontrol edileceğini belirtmektedir. **fsck** sıra numarası aynı olan dosya sistemleri birlikte paralel olarak kontrol edilir. **fsck** sırası “**0**” ise o dosya sistemi kontrol edilmez. sunucu makinesindeki ortak dizininin **fsck** sırasının “**0**” olmasının nedeni, doğal olarak bu dosya sisteminin **fsck** kontrollerinden, dosya sistemi ya da dizinin gerçek sahibi olan bilgisayarın sorumlu olmasıdır.

Başka Bir Windows Bilgisayarın Diskini Kullanmak

Yerel ağınızda bulunan, Windows işletim sistemleriyle çalışan ve bazı dizinlerini paylaşımaya açmış bilgisayarlar varsa bu dizinleri siz de kullanabilirsiniz. Örneğin; **IP** adresi 123.123.12.3; adı da “pazarlama” olan bir WinXP bilgisayar kendi diskindeki bir dizini “teklifler” adı altında paylaşımaya açmış olsun. Bu dizine Linux bilgisayarınızdan erişebilmek için

```
mount -t smbfs //123.123.12.3/teklifler/mnt/windows
-o username=cua,password=sifre
```

komutunu veya

```
mount -t smbfs //pazarlama/teklifler /mnt/windows
-o username=cua,password=sifre
```

kullanabilirsiniz (elbette siz bu komutu tek satırda vermelisiniz). İkinci formdaki komutu; yani bilgisayarın **IP** adresi yerine ismini kullanabilmeniz için, söz konusu bilgisayarın bu isimle DNS kaydının yapılmış olması veya sizin bilgisayarınızda **/etc/hosts** dosyasında bir kaydı olması gerekir.

Yukarıdaki örnek komutlarda belirtilmiş olan “**cua**” kullanıcısı ve şifresi, Windows sisteminde tanımlanmış olması gereken bir kullanıcı kodu ve şifresidir. Sizin Linux bilgisayarınızda bu isimle bir kullanıcı kaydı gerekmez.

Dosya Sistemi Yaratmanız Gerekirse...

Disk ve disketlerinizi kendiniz formatlamanız gerektiğinde iki aşamalı bir iş yapmalısınız:

1. Önce medyayı formatlamalı,
2. sonra üzerine uygun bir dosya sistemi yaratmalısınız.

Örneğin MS-DOS formatlı boş bir disket hazırlamak istediğinizde,

```
/usr/bin/fdformat /dev/fd0  
/sbin/mkfs -t msdos /dev/fd0
```

komutlarıyla önce disketi formatlayıp sonra üzerine MS-DOS dosya sistemi oluşturmalsınız. Eğer **-t** parametresiyle dosya sistemi tipi belirtmezseniz **ext2** dosya sistemi kabul edilir ve ancak **LINUX/UNIX** bilgisayarlara mont edilerek kullanılabilen bir disket elde edersiniz.

Sabit diskleri formatlarken de benzeri bir yöntem izlenir. Önce,

```
/sbin/fdisk /dev/hdc
```

gibi bir komutla diskin üzerinde yer alacak bölümlleme tablosu (*partition table*) oluşturulur; daha sonra diskin üzerindeki bölüm sayısına göre:

```
/sbin/mkfs -t reiserfs /dev/hdc1  
/sbin/mkfs -t ext3 /dev/hdc2
```

gibi komutlarla her bölüm üzerinde yer alacak dosya sistemleri oluşturulur.

```
/sbin/mke2fs /dev/hdc2
```

komutu “**/sbin/mkfs -t ext2 /dev/hdc2**” ile eşdeğerdir. Dolu diskler üzerinde **fdisk** ve **mkfs** komutlarını denemeyiniz; tüm kayıtları silersiniz.

BUNLARI BİLİYOR MUYDUNUZ?

Linux'ta virüsler neden büyük bir tehdit değildir?

LINUX işletim sistemi altında erişim yetkileri çok iyi tanımlanmıştır. "root" kullanıcı dışında hiç bir kullanıcı sistem dizinlerine ve sistem dosyaları üzerine yazmaya yetkili değildir; e-posta ekinde gelen virüslü programları normal bir kullanıcının yetkileriyle çalıştırsanız bile sisteme zarar veremezsiniz. LINUX, baştan beri çok kullanıcılı bir işletim sistemi olarak düşünüldüğü ve geliştirildiği için, sistemin yetkisiz kullanıcılara ve hatalı yazılımlara karşı korunması iyice denenmiş, olgunlaşmıştır.

Kim Korkar LINUX'tan?

- **find**
- **grep**
- **tar Komutu ve tar Dosyaları**
- **tar Komutu ile Yedekleme**
- **rsh**
- **xargs**
- **at**
- **date, hwclock**
- **lynx**
- **cut**
- **tee**
- **script**
- **split**

find

Günümüzün tipik kişisel bilgisayarlarında disk kapasiteleri artık onlarca GigaByte ile ölçülmektedir. Bu kadar büyük disklerde de doğal olarak çok sayıda dizin ve onbinlerce dosya yer alabilmektedir. Zaman zaman adının yalnızca bir kısmını bildiğiniz ama bulunduğu dizini bir türlü hatırlayamadığınız dosyalar ve dizinler olacaktır. Tek tek bütün dizinlere girip **ls** komutuyla bu dosya ya da dosyaları aramaktansa **find** komutunu kullanmak hayatı kolaylaştıracaktır. Örneğin;

```
find /home/cayfer -name nerede.dat
```

komutu, **/home/cayfer** dizininden başlayarak, bu dizinde ve daha derinlerdeki dizinlerde adı “**nerede.dat**” olan dosyayı arar; bulursa yerini görüntüler.



```
cayfer@pusula.bilkent.edu.tr: /home/cayfer - Konsole - Konsole
[cayfer@pusula cayfer]$ find /home/cayfer -name nerede.dat
/home/cayfer/tmp/nerede.dat
/home/cayfer/fp/nerede.dat
```

Yukarıdaki örnekte, “**nerede.dat**” isimli iki dosya, **/home/cayfer** dizini altındaki **tmp** ve **fp** dizinleri altında bulunmuştur.

find komutuyla yalnızca adı ya da adının bir parçası bilinen dosyaları aramak için kullanılmaz. Dosyalar adları dışında da özelliklerine göre aranabilir.

Genel formu:

```
find başlama-dizini kriter[ler] [-exec komut “;”]
```

olan komutla:

- belirli bir tarihte değişikliğe uğramış dosyaları,
- belirli bir tarihten bu yana değişmiş dosyaları,
- belirli bir boydan daha büyük ya da küçük dosyaları,
- belirli erişim yetkilerine sahip dosyaları ve dizinleri,
- belirli bir kullanıcıya ait dosya ve dizinleri de arayıp bulabilirsiniz.

Üstelik, verdiğiniz arama kriterlerine uyan dosyalar ve dizinler üzerinde uygulamak isteyebileceğiniz LINUX komutlarını da **find** komutuna parametre olarak verebilirsiniz.

başlama-dizini: Arama işlemi, **find** komutunun bu ilk parametresinde belirtilen dizinden başlar ve varsa bu dizinin alt dizinleri de arama ağacına dahil edilir. Eğer arama işleminin, bilgisayarınıza bağlı ve mount edilmiş tüm dosya yapılarında (disk, CD ve başka bilgisayarların dosya sistemleri de olabilir) yapılmasını istiyorsanız, ilk parametre olarak “/” sembolünü kullanınız; yani aramayı tüm dosya sistemlerinizin en tepesinden başlatınız.

Bilgisayarınızın CD-ROM sürücüsü varsa, bu sürücüye bir CD takılıysa, bu CD mount edilmiş durumdaysa ve aramayı "/" dizininininden başlatırsanız, arama ağacı CD-ROM sürücüsünü de kapsayacaktır. CD'lerin kapasitele-
rinin büyüklüğü ve erişim hızlarının düşüklüğünden dolayı bu arama uzun
sürecektir. Aynı mantıkla, bilgisayar ağı üzerinden başka bilgisayarların
diskleri de sizin dosya sisteminize mount edilmiş durumdaysa, o diskler de
arama ağacına girecektir. Zaman kaybına yol açmamak için, gerekmedik-
çe aramayı "/" dizininininden başlatmamanızı öneririz. **-mount** parametresiy-
le aramanın başka dosya sistemlerine de atlamasını önleyerek bu dertten
de kurtulmak mümkündür elbette.



kriter[ler]: Aranana dosya veya dizinlerin ortak özelliklerini tanımlayan kriterlerdir. Birkaç örnek vermek gerekirse:

- name isim** Adı "**isim**" olan dosyalar.
(Farklı dizinlerde aynı isme sahip dosyalar olabilir.)
- name "abc*"** Adı "**abc**" ile başlayan dosyalar.
- name "[a-k]9"** Adı **a9**, **b9**, ..., **j9** veya **k9** olan dosyalar.

Dikkatinizi çektiyse, **-name** kriterinde dosya adı tam olarak yazıldığında tırnak (") kullanılmıyor; oysa * karakterini içeren bir kalıp kullanıldığında (wildcard) bu kalıbı tırnak (") içinde yazmak gerekiyor. Bunun nedeni şu-
dur: Bir komut verdiğinizde, bu komut önce kabuk programınız tarafından
irdelenir. Bu irdeleme sırasında rastlanan * karakterleri dosya adı kalıpla-
rının bir parçası olarak kabul edilip, * içeren parametre bu kalıba uyan
dosya isimleriyle değiştirilmeye çalışılır. Oysa, kalıplara uyan dosya isim-
lerinin kabuk programı tarafından değil, **find** programı tarafından bulun-
ması gerekmektedir. Kabuk programlarının irdeleme sırasında karşılaşa-
cakları * karakterlerine dokunmadan, parametreleri oldukları gibi çalıştı-
lacak programa aktarmaları için, kalıp tanımları tırnak içine alınır.



- user ayfer** Sahibinin adı "**ayfer**" olan dosyalar ve dizinler.
- group yönetim** Sahibi "**yönetim**" grubuna dahil olan dosyalar ve dizinler.

-perm 755	Erişim yetki düzeyi 755 olan dosyalar ve dizinler.
-newer dosya1	dosya1 isimli dosyadan daha sonraki bir saat ya da tarihte değişikliğe uğramış olan dosyalar ve dizinler.
-size 10	Diskte kapladığı alan 10 blok olan dosyalar. (1 blok = 512 Byte)
-size +100k	Diskte kapladığı alan 100 Kbyte'dan büyük olan dosyalar.
-size -100c	Diskte kapladığı alan 100 byte'dan az olan dosyalar.
-ctime 3	Tam 3 gün önce değişikliğe uğramış olan dosyalar ve dizinler.
-ctime +8	8 günden daha uzun bir süre önce değişikliğe uğramış olan dosyalar ve dizinler.
-ctime -8	8 günden daha kısa bir süre önce değişikliğe uğramış olan dosyalar ve dizinler.
-mtime 3	Tam 3 gün önce değişikliğe uğramış olan dosyalar ve dizinler.
-mtime +8	8 günden daha uzun bir süre önce değişikliğe uğramış olan dosyalar ve dizinler.
-mtime -8	8 günden daha kısa bir süre önce değişikliğe uğramış olan dosyalar ve dizinler.
-atime -3	3 günden daha kısa bir süre içinde bir şekilde erişilmiş olan dosyalar ve dizinler.
-amin -3	3 dakikadan daha kısa bir süre içinde bir şekilde erişilmiş olan dosyalar ve dizinler.
-mmin -3	3 dakikadan daha kısa bir süre içinde değişikliğe uğramış veya yaratılmış olan dosyalar ve dizinler.
-type f	Dosyalar.
-type d	Dizinler.

-ctime ve **-mtime** parametrelerinin her ikisi de dosyanın değişikliğe uğramasıyla ilgili süreleri kontrol eder; ancak aralarında küçük bir fark vardır:



-mtime dosyanın içeriğinde bir değişiklik yapıp yapılmadığını;
-ctime ise dosyanın içeriği yanısıra özelliklerinin de değişip değişmediğini kontrol eder.

Örneğin, sahibi değişen bir dosya **-mtime** tarafından farkedilmezken **-ctime** tarafından dikkate alınır.

Bu arama kriterlerini bir arada kullanabilirsiniz. Örneğin, sahibi “**hakman**” olan ve son 40 gündür kullanılmamış dosyaları bulmak isterseniz, kullanmanız gereken **find** komutu

```
find /home -user hakman -atime +40
```

olmalıdır.

Şimdi, sık kullanılan **find** formları için birkaç örnek verelim:

```
find /home/ayfer -name onemli.dosya
```

/home/ayfer dizininden başlayarak bu dizinde ve alt dizinlerinde “**onemli.dosya**” isimli dosyaları arar ve bulduklarının adını ve yerini standart çıktıya (ekrana) listeler.

```
find / -name core -exec /bin/rm {} “;”
```

“/” dizininden başlayarak tüm dizin yapısında “**core**” isimli dosyaları arar ve bulduklarını siler.

find komutu **-exec** parametresiyle birlikte kullanıldığında, bulunan dosya ve izin isimleri, “**{ }**” arasına parametre olarak yerleştirilecek ve **-exec**’den hemen sonra belirtilmiş olan program bu parametre ile çalıştırılacaktır. Yukardaki örnekte bulunan her bir “**core**” dosyası için “**/bin/rm core**” komutu çalıştırılmış olacaktır. En sondaki “**;**” parametresi çok önemlidir ve unutulmamalıdır. Sondaki bu “**;**” karakter dizisinin gerekliliği tamamen **find** programının yazılışından kaynaklanmaktadır.



find, sistem yönetiminden sorumlu olanların oldukça sık kullanacakları bir komuttur. LINUX, çeşitli programların kullanımı sırasında sistemin bütünlüğünü tehdit eden bir problemle karşılaştığında (ki bu genellikle hatalı yazılmış programlar yüzünden olur) “**core dumped**” mesajıyla birlikte, belleği **core** isimli bir dosyaya kopyalar. Bu **core** dosyaları, programcıların problemin nedenini bulmasına yardımcı olmak amacıyla yaratılır. Bu dosyaları irdeleyerek problemin nedenini bulmak pek kolay olmadığından, bu dosyaları içeriklerine bakmaksızın silebilirsiniz.

Bir başka önemli örnek:

```
find /home -user hasan -exec /bin/rm {} ";"
```

/home dizininden başlayarak **hasan** isimli kullanıcıya ait dosyaları arar ve bulduklarını siler. Sisteme erişim hakları iptal edilen bir kullanıcıya ait dosyaları tek harekette silmek için kullanılabilir.

```
find /home -name "*.mp3" -exec /bin/rm {} ";"
```

/home dizininden başlayarak adı ***.mp3** kalıbına uyan dosyaları arar ve bulduklarını siler.

```
find /home -type d -name [tmp, temp]
```

/home dizininden başlayarak adı **tmp** veya **temp** olan dizinleri bulur ve listeler.

find komutuyla birlikte kullanılabilen kriterleri çeşitli mantık operatörleriyle birleştirebilirsiniz.

Bunlar:

```
-a:  "ve"  
-o:  "veya"  
\!:  "değil"
```

operatörleridir.

Örneğin:

```
find . -name "*.tmp" -a -size +1000k
```

çalışma dizininde (".") ve varsa altındaki dizinlerde adı ***.tmp** kalıbına uyan

ve büyüklüğü 1000 KByte'den fazla olan dosyaları bulur.

```
find /home/cayfer \! -user cayfer
```

cayfer isimli kullanıcının kişisel dizininde yer alan ama **cayfer**'e ait olmayan dosyaları bulur.

Bu örnekteki “**değil**” anlamında kullanılan “\!” operatöründeki “\” işareti, ardından gelen “!” işaretinin özel bir anlamı olduğunu ve kabuk programı tarafından yorumlanmaya çalışılmaması gerektiğini belirtmek için kullanılmaktadır.

Hatırlarsanız, daha önce UNIX işletim sisteminde kendi komutlarınızı yaratabileceğinizden bahsetmiştik. İşte bu uygulamaya bir örnek vermek için uygun bir noktadayız.

find komutu oldukça yetenekli ve çok seçenekli bir komut olmakla birlikte klavyeden yazması da oldukça uzun bir komuttur. Dosyaları yalnızca adlarıyla arayan, **find** komutundan daha kısa bir LINUX komutu yaratmaya ne dersiniz?

Aşağıdaki **bash** kabuk programını herhangi bir editörle (tabii ki **vi** ile) kişisel dizininizde “**ff**” isimli bir dosyaya giriniz:

```
#!/bin/bash
case $# in
  1) find . -name "$1" ;;
  2) find "$1" -name "$2" ;;
  *) echo "Hata! Komutun kullanımı : ff [dizin] isim"
    echo "                                ff [dizin] \"xyz*\""
    echo "                                ff [dizin] \"*xyz\""
esac
```

Daha sonra,

```
chmod a+x ff
```

komutuyla, bu dosyanın erişim yetki kalıbını, tüm kullanıcılar tarafından çalıştırılabilen bir komut dosyası olacak şekilde değiştiriniz.

Bu kabuk programı, önce kendini çalıştıran komut satırında verilmiş olan parametrelerin sayısını kontrol ediyor. ("\$#") Eğer tek parametreyle başlatılmışsa "**find . -name parametre**" komutunu çalıştırıyor. Eğer iki parametreyle başlatılmışsa, birinci parametreyi aramanın başlatılacağı dizin kabul edip "**find param1 -name param2**" komutunu çalıştırıyor. Eğer parametre sayısı bir veya iki değilse üç satırdan oluşan bir hata mesajı veriyor.

Örnekler:

```
ff aranan.veri.dosyasi  
ff /home/ugur prog.c  
ff ~ file001.dat  
ff "*dat"
```

Yeni yarattığınız **ff** komutunu kullanmak istediğinizde komut programının bulunamadığına ilişkin bir mesaj alıyorsanız, çalışma dizininiz **PATH** değişkeninizde yer almadığı için olabilir. O zaman programı **./ff** komutuyla çalıştırabilirsiniz.

Yeni **ff** komutunuzu iyice denedikten sonra genel kullanıma sokmak için **ff** dosyasını **/usr/local/bin** altına kopyalayabilirsiniz. Ancak bu kopyalamayı yapabilmek için **root** kullanıcı yetkilerine gereksiniminiz olacaktır.



locate

Birçok modern Linux dağıtımında, arka planda çalışan "slocate" programı genellikle günde bir kez olacak şekilde disk(ler)inizdeki dosyaların isimlerini indeksler. Bu sayede "locate xyz" komutunu kullanarak normalde birkaç dakika sürebilecek **find -name "*xyz*"** benzeri komutun sonucunu saniyeler içinde görebilirsiniz. Tabi bu sonuç sisteminizin o andaki hali için değil, en son indekslemenin yapıldığı zaman için geçerli olacağından, sisteminizde ne kadar değişiklik olduğuna bağlı olarak kimi zaman doğru olmayabilir.

grep Komutu

Dosyaları adları ve sahipleri gibi özelliklerine göre arayıp bulma işini "**find**" komutu ile halledebilirsiniz. Ama bazen de dosyaları isimlerine göre değil, içeriklerine göre aramanız gerekecektir. Belli bir karakter dizisini içeren dosyaları ve/veya bir dosya grubu içinde belli bir karakter dizisi geçen satırları bulmak için;

grep [-irvnc] dizi dosya(lar)

komutunu kullanmalısınız.

Hemen birkaç örnek...

İçinde sisteminizin tüm tanımlı kullanıcıları için birer satır yer alan **/etc/passwd** dosyasında “**murat**” diye bir sözcük olup olmadığını kontrol etmek istediğinizde

grep murat /etc/passwd

komutunu kullanabilirsiniz. Bu dosyanın içinde “**murat**” sözcüğü geçen tüm satırlar standart çıktı birimine listelenecektir. Ancak komutu bu örnekteki gibi kullanırsanız içinde “**Murat**” geçen satırları yakalayamazsınız. Büyük-küçük harf ayırımı yapılmaksızın arama yapılmasını istiyorsanız komutu;

grep -i murat /etc/passwd

şeklinde vermeniz gerekir.

İçinde “**Murat**” ya da “**murat**” geçen satırların satır numaralarını da görmek isterseniz:

grep -ni murat /etc/passwd

formunu kullanabilirsiniz.

Diyelim ki bulunduğunuz dizinde, adı “**prog**” ile başlayan dosyalar arasında bir veya birkaç tanesinin içinde “**cayfer**” sözcüğünün bulunduğunu biliyorsunuz ama hangileri olduğunu hatırlayamıyorsunuz!

İşte çözüm:

grep cayfer prog*

Adı “**prog**”la başlayan dosyalarda “**cayfer**” sözcüğünün kaç defa geçtiğini öğrenmek isterseniz **grep** komutunu:

grep -c cayfer prog*

şeklinde **-c** parametresiyle kullanılabilir.

Kim Korkar LINUX'tan?

Eğer içinde belirli bir karakter dizisi geçen dosyaları bir dizin ağacında arayacaksanız **-r** parametresi çok işinize yarayacaktır.

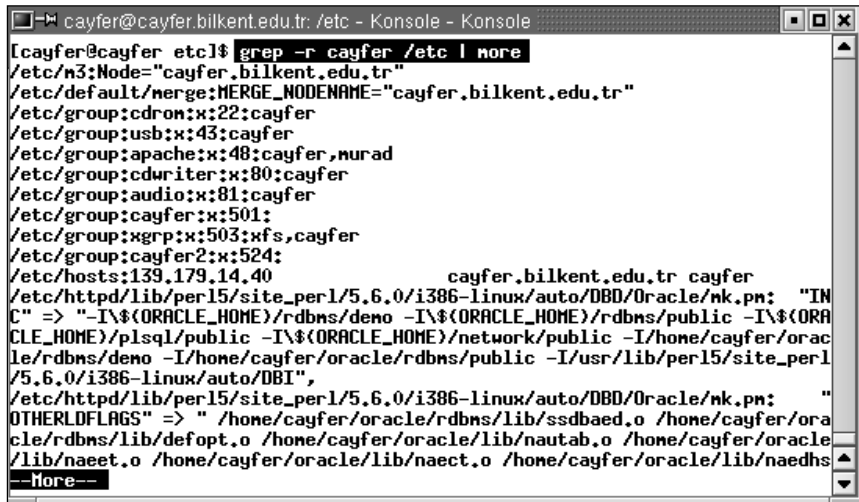
grep -r cayfer *

komutu “**cayfer**” karakter dizisini çalışma dizinindeki ve bu dizin altındaki dizinlerdeki dosyalarda arar ve içinde “**cayfer**” geçen dosyaların isimlerini listeler.

grep komutunun ürettiği listenin çok uzun olması durumunda, komutu

grep -r cayfer * | more

şeklinde kullanırsanız, **grep** komutunun ürettiği listeyi **more** komutuna yönlendirerek listenin ekrana sayfa sayfa görüntülenmesini sağlayabilirsiniz.



```
cayfer@cayfer.bilkent.edu.tr: /etc - Konsole - Konsole
[cayfer@cayfer etc]$ grep -r cayfer /etc | more
/etc/n3:Node="cayfer.bilkent.edu.tr"
/etc/default/merge:MERGE_NODENAME="cayfer.bilkent.edu.tr"
/etc/group:cdrom:x:22:cayfer
/etc/group:usb:x:43:cayfer
/etc/group:apache:x:48:cayfer,nurad
/etc/group:cdwriter:x:80:cayfer
/etc/group:audio:x:81:cayfer
/etc/group:cayfer:x:501:
/etc/group:xgrp:x:503:xfs,cayfer
/etc/group:cayfer2:x:524:
/etc/hosts:139.179.14.40 cayfer.bilkent.edu.tr cayfer
/etc/httpd/lib/perl5/site_perl/5.6.0/i386-linux/auto/DBD/Oracle/nk.pn: "IN
C" => "-I${ORACLE_HOME}/rdbms/deno -I${ORACLE_HOME}/rdbms/public -I${ORA
CLE_HOME}/plsql/public -I${ORACLE_HOME}/network/public -I/home/cayfer/orac
le/rdbms/deno -I/home/cayfer/oracle/rdbms/public -I/usr/lib/perl5/site_perl
/5.6.0/i386-linux/auto/DBI",
/etc/httpd/lib/perl5/site_perl/5.6.0/i386-linux/auto/DBD/Oracle/nk.pn: "
OTHERLDFLAGS" => " /home/cayfer/oracle/rdbms/lib/ssdbaed.o /home/cayfer/ora
cle/rdbms/lib/defopt.o /home/cayfer/oracle/lib/nautab.o /home/cayfer/oracle
/lib/naect.o /home/cayfer/oracle/lib/naect.o /home/cayfer/oracle/lib/naedhs
--more--
```

Bazı durumlarda size bir dosyada içinde “**cayfer**” geçen satırlar değil de, “**cayfer**” geçmeyen satırlar gerekir.

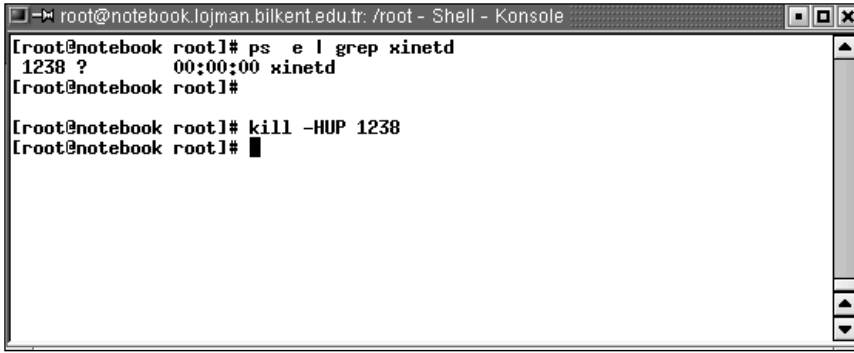
grep -v bash /etc/passwd

komutu **-v** parametresinden dolayı **/etc/passwd** dosyasındaki satırlar arasında, içinde “**bash**” geçmeyenleri listeleyecektir.

grep komutu ille de dosyalar içinde arama için kullanılmaz. Başka programların ürettiği çıktılar arasında da arama yapabilirsiniz. Örneğin, sisteminizin

internet servislerini ve bağlantılarını yöneten **xinetd** isimli sürecin konfigürasyonunda değişiklik yaptığınızda ve bu programı yeniden başlatmanız gerektiğinde **xinetd**'nin süreç numarasını bilmeniz gerekecektir. LINUX altında çalışan bilgisayarlarda, çok meşhur bazı işletim sistemlerinde olduğu gibi ayar değişikliklerinden sonra sistemi kapatıp açmak gerekmez. **xinetd** yazılımına kendisini yeniden başlatması için; daha doğrusu ayar dosyalarını yeniden yüklemesi için **-HUP** sinyalini göndermek yeterli olacaktır.

xinetd sürecinin numarasını, oldukça uzun olan “**ps e | grep xinetd**” listesinde gözle aramak yerine bu listeyi **grep** filtresinden geçirebilirsiniz.



```

root@notebook.lojman.bilkent.edu.tr: /root - Shell - Konsole
[root@notebook root]# ps e | grep xinetd
1238 ?        00:00:00 xinetd
[root@notebook root]#
[root@notebook root]# kill -HUP 1238
[root@notebook root]#

```

grep komutunun **gzip** veya **compress** ile sıkıştırılmış dosyalar içinde de arama yapabilen bir varyasyonu vardır: **zgrep**.

Sisteminizin **/var/log** dizinindeki dosyalara bakarsanız bazılarının isimlerinin “**1.gz**”, “**2.gz**” gibi karakterlerle bittiğini görürsünüz. Bunlar sistemin **logrotate** işleviyle belirli aralıklarla sıkıştırıp arşivlediği log dosyalarıdır. (**logrotate** ile ilgili ayrıntılı bilgiyi kitabın “Sistem Yönetimi” bölümünde bulabilirsiniz.) Bir dosya sıkıştırıldığında artık içinde okunabilir karakterler yer almayacaktır. Yani, sıkıştırılmış bir dosya içinde “**cayfer**” sözcüğünü **grep** ile aramak pek işe yaramaz.

Bu sıkıştırılmış dosyalarda bir şeyler arayacağınız zaman **grep** komutunu kullanabilmek için önce bu dosyaları **gunzip** gibi bir komutla açmanız ve ondan sonra **grep** komutuyla içlerinde arama yapmanız gerekir. Ama durun bir dakika... LINUX kullandığınızı unuttunuz herhalde... **grep** yerine

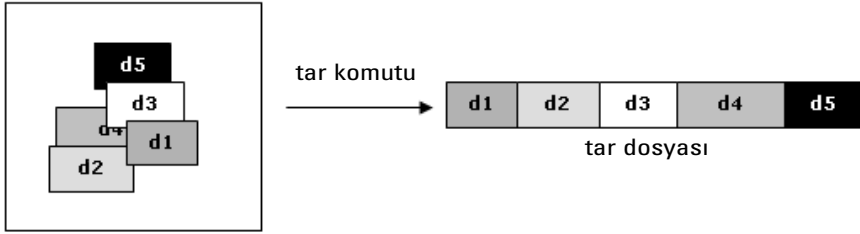
```
zgrep cayfer /var/log/mail/*
```

komutuyla **cayfer** sözcüğünü **/var/log/mail** altındaki tüm dosyalarda arayabilirsiniz. Üstelik **zgrep**, sıkıştırılmış dosyaları ve normal dosyaları ayırarak arama işlemini ona göre yapacak, aradığını bulursa da rapor edecektir.

tar Komutu ve tar Dosyaları

LINUX dünyasında çalışan birinin; hele sistem yöneticisiyse, çok sık karşılaştacağı bir kavramdır “**tar**”... “**Tape Archive**” sözcüklerinden türetilmiştir. UNIX işletim sisteminin yazılmaya başlandığı yıllarda teyp makaralarına ya da kasetlerine dosya kaydetmek ve bu dosyaları geri indirmek için geliştirilmiştir. Artık teyp kullanımı, yaygınlığını kaybetmiş olsa da (aslında büyük sistemlerde hala en yaygın yedekleme ve arşivleme ortamı teyptir, ancak küçük ofis ortamlarında artık teyp sürücülere pek rastlanmamaktadır) “**tar dosya**” mantığı aynen ve daha da yaygınlaşarak kullanılmaktadır.

tar dosyası hazırlamanın çok basit bir mantığı vardır: “**tar’lanmak**” istenen dosyaları peşpeşe ekleyip tek bir dosya elde etmek.



Yukarıdaki şemadaki gibi **d1**, ..., **d5** isimli dosyaları tek bir **d.tar** dosyasında birleştirmek için:

```
tar -cvf d.tar d1 d2 d3 d4 d5      veya
tar -cvf d.tar d?
```

komutlarından biri kullanılabilir.

tar dosyaları, ya da LINUX jargonuna uygun olarak söylemek gerekirse, “**tar yumakları**” (*tar balls*) son yıllarda program paketlerini taşımak için en çok kullanılan yöntemdir. Windows dünyasında da aynı bu şekilde çok kullanılan bir dosya paketleme yöntemi vardır: ZIP.

Diyeceksiniz ki “*ZIP dosyaları peşpeşe ekliyor eklemesine ama hem dosyaları sıkıştırarak toplam paketi küçültüyor hem de alt dizinleri de paketin içine yerleştirebiliyor.*”

Bir kere şu noktada anlaşılmı: Windows serisi işletim sistemleri kişisel kullanım için tasarlanmıştır ve bu kavram içinde de oldukça başarılıdır. Oysa UNIX, 30 yıldan fazla bir süredir geliştirilmektedir ve profesyonel kullanım için tasarlanmıştır. Bu nedenle bilişim dünyasında Windows'un yapıp da LINUX'un yapamadıkları değil, bunun tam tersi konuşulur. Sözün kısası

```
tar -cvzf web_cayfer.tgz ./public_html
```

komutuyla `/home/cayfer/public_html` dizini altındaki herşeyi `web_cayfer.tgz` dosyası içine sıkıştırarak paketleyebilirsiniz.

```
cayfer@notebook.lojman.bilkent.edu.tr: /home/cayfer - Shell - Konsol <4>
[cayfer@notebook cayfer]$ tar -cvzf web_cayfer.tgz ./public_html
./public_html/
./public_html/Animator.class
./public_html/C64-MinVICE1.7.zip
./public_html/CTutorial.html
./public_html/CTutorial_files/
./public_html/CTutorial_files/1.jpg
./public_html/CTutorial_files/10.jpg
./public_html/CTutorial_files/11.jpg
./public_html/CTutorial_files/11_002.jpg
./public_html/CTutorial_files/12.jpg
./public_html/CTutorial_files/14.jpg
./public_html/CTutorial_files/14_002.jpg
./public_html/CTutorial_files/15.jpg
./public_html/CTutorial_files/15_002.jpg
./public_html/CTutorial_files/15_003.jpg
./public_html/CTutorial_files/17.jpg
./public_html/CTutorial_files/17_002.jpg
./public_html/CTutorial_files/17_003.jpg
./public_html/CTutorial_files/18.jpg
./public_html/CTutorial_files/18_002.jpg
./public_html/CTutorial_files/1_002.jpg
```

Şimdi **tar** komutunda kullanılabilecek bazı önemli parametrelere ve **tar**'ın çok kullanıldığı işlere bir göz atalım:

tar Parametreleri	
c	Create: tar dosyası yaratılacağını belirtir.
x	Extract: Bir tar dosyasının açılacağını (çözüleceğini) belirtir.
t	Table of contents: Bir tar dosyasının içeriğinin listleneceğini belirtir.
v	Verbose: Bir tar dosyası yaratılırken ya da açılırken elden geçen dosyaların isimlerini ekrana listelemek için kullanılır.

z	tar dosyasının sıkıştırılmış bir dosya olarak kullanılacağını belirtir. Yani, dosya yaratılıyorsa sıkıştırılarak yaratılacaktır; dosya çözülüyorsa, önce gunzip ile açılması gerektiğini belirtir.
f	File: Yaratılacak, açılacak ya da içindekiler tablosu listelenecek tar dosyasının adının komut satırında verileceğini belirtir. tar dosyası yaratırken, yaratılacak dosya adının verileceğinin belirtilmesi biraz garip geldi, değil mi? Evet, haklısınız ama diskte gerçek bir dosya yaratmaksızın tar dosyası oluşturmak oldukça anlamlıdır. Bu şekilde kullanımın bir örneğini tar Komutu ile Yedekleme başlığı altında bulabilirsiniz.

Şimdi sıra örneklerde:

```
tar -cvf dat_dosyalar.tar *dat
```

Çalışma dizininde yer alan ve isimleri “**dat**” ile biten tüm dosyaları **dat_dosyalar.tar** adıyla birleştirir.

```
tar -cvzf dat_dosyalar.tgz *dat
```

Aynı işi dosyaları sıkıştırarak yapar. Sıkıştırma işi **gzip** programı kullanılarak yapılır.

```
tar -tf dat_dosyalar.tar
```

dat_dosyalar.tar isimli dosyanın içindeki dosya ve dizinlerin listesini döker.

```
tar -xvf dat_dosyalar.tar
```

dat_dosyalar.tar dosyasının içindeki tüm dosyaları çalışma dizinine açar.

```
tar -xvf dat_dosyalar.tar birinci.dat
```

dat_dosyalar.tar dosyasının içinden yalnızca **birinci.dat** isimli dosyayı çalışma dizinine açar. Eğer açmak istediğiniz dosya, **tar** dosyası oluşturulurken bir dizin altında yer alıyor idiyse, bu dosyayı açarken o dizini de belirtmelisiniz. (**tar -xvf d.tar /home/cayfer/mail** gibi.)

```
tar -xvf dat_dosyalar.tar -C /tmp/yeni_dizin
```

dat_dosyalar.tar dosyasının içindeki tüm dosyaları
/tmp/yeni_dizin dizinine açar.

```
tar -cvf /dev/rst0 /home
```

/home dizinindeki herşeyi birinci SCSI teyp birimindeki kasete kaydeder
(st0) ve işi bitirince kaseti başa sarar. (**r:** *rewind*)

```
tar -cvf /dev/nrst0 /home
```

/home dizinindeki herşeyi birinci SCSI teyp birimindeki kasete kaydeder
(st0) ve işi bitirince kaseti kaldığı yerde bırakır. (**nr:** *no rewind*)

```
tar -cvzf yedek.tgz /home -exclude /home/cayfer
```

/home/cayfer dizini hariç, **/home** altındaki tüm dosya ve dizinleri
yedek.tar dosyasında sıkıştırarak birleştirir.

tar programının yarattığı dosyaların uzantıları “**.tar**” ya da “**.tar.gz**” olmak zorunda değildir. Ancak tar dosyalarını bu şekilde isimlendirmek önemli bir alışkanlığınız olmalıdır. Aylar sonra karşınıza çıkan bir dosyanın tar dosyası olduğunu hatırlamayabilirsiniz; ancak isim verirken “**.tar**” veya “**.tar.gz**” uzantısı verdiğiniz dosyaları ne yapmanız gerektiğini her zaman için hatırlarsınız.



tar programının **c**, **v**, **f** gibi parametrelerinin başına “-” işareti koymak zorunda değilsiniz, yani “**tar cvf home.tar /home**” geçerli bir komuttur.

tar Komutu ile Yedekleme

Tüm akli başında bilgisayar kullanıcıları gibi dosya ve dizinlerinizi yedeklemelisiniz. Bilgisayarınızın diskinin arızalanmasına, bilgisayarınızın olduğu gibi çalınmasına, yanmasına ya da disklerin yanlışlıkla formatlanmasına hazırlıklı olmalısınız.

Kullanıcı olarak kendi dosyalarınızı, periyodik olarak (örneğin her akşam) bir başka disk üzerine kopyalayabilirsiniz. Bir başka bilgisayara kopyalamak

Kim Korkar LINUX'tan?

elbette daha güvenli olacaktır. Artık yedekleyeceğiniz dosyaların değerine göre kendiniz bir strateji geliştirirsiniz nasılsa...

Yedekleme amacıyla bu kopyalama işini sistem yöneticisi sıfatıyla, yani root kimliğiyle yapmanız gerektiğinde sizi küçük bir sorun bekliyor olacaktır:



Tipik bir UNIX bilgisayarda **/home** dizini altında kullanıcıların kişisel dizinleri yer alır. Normal koşullarda her kişisel dizinin ve altındaki dosyaların sahibi farklı kullanıcılar olacaktır. root kullanıcı olarak

```
cp -r /home /disk2
```

komutuyla **/home** dizinindeki herşeyi ikinci disk üzerine kopyalayabilirsiniz, ama **disk2** altına kopyalanan tüm dosyaların ve dizinlerin sahibi root olur!

Bu durumda yedekleri geri indirmeniz gerektiğinde indirilen tüm dosyaların sahiplerini yeniden düzenlemeniz gerekecektir. Zor iş! Oysa, tüm dosya ve dizinlerin baştan gerçek sahiplerinin kimliğiyle yedeklenmesi çok daha anlamlı olurdu. Bu nedenle **cp** komutunu yedekleme için kullanmanızı önermeyiz.

Aslında LINUX işletim sisteminin basit kopyalama komutu olan **cp** programı “**-p**” parametresiyle bu sorunu ortadan kaldıracılabilmektedir; yani, değişik kullanıcılara ait dosyaları

```
cp -rp /home /disk2
```

gibi bir komutla (“**-p**” parametresini kullanarak) kopyalarsanız, kopyalanan dosyalar yeni yerlerinde asıl sahipleri ve erişim özellikleriyle birlikte kopyalanır. Ancak bu özellik standart bir UNIX özelliği olmadığı için LINUX dışındaki UNIX türevlerinin hepsinde işe yaramaz. Sistem yöneticisi olarak yedek almanız gerektiğinde **tar** kullanmaya alışmanız daha yararlı olacaktır.

tar komutu yedekleme komutları arasında en kullanışlı olanıdır. Bir **tar** yumağını çözmek üzere gerekli **tar** komutunu “root” kullanıcı kimliğiyle verdiğinizde bile çözülen tüm dosya ve dizinler orijinal sahiplerinin kimliğine uygun olarak çözülür.

tar komutunun “**-f**” parametresini hatırlıyor musunuz? Hani yaratılacak tar dosyasının adını vereceğinizi belirten parametre... Bu parametreden söz

ederken “**tar**” komutunu bir **tar** dosyası yaratmadan da kullanmak olasıdır” demiştik. Evet; şimdi bu garip olayı açıklamanın zamanı geldi:

Diyelim ki bilgisayarınıza ikinci bir disk taktınız ve **/home** dizinini bu yeni diske taşımak istiyorsunuz. **/home** dizininizdeki dolu alan 24 GB olsun, yeni diskiniz de 40 GB olsun.

“**cp -r /home /disk2**” komutunun iş görmeyeceğini anlatmıştık. Peki, önce:

```
tar -cvf /disk2/eski_home.tar /home
```

komutuyla **/home** dizininin **/disk2** üzerinde bir tar yumağını oluştursak, sonra da

```
cd /disk2  
tar -xvf eski_home.tar
```

komutlarıyla bu yumağı **/disk2**’de açsak olur mu acaba?

Aslında olmaz! Çünkü ikinci diskinizde yeteri kadar boş yer yok! 40 GB’lık disk üzerine hem 24 GB’lık bir tar yumağını hem de bu yumağın açılmış halini sığdıramazsınız! **tar** komutunu **-z** parametresiyle birlikte kullanıp tar yumağını küçültmeyi düşünebilirsiniz ama bu taklalara hiç gerek yok! Çözüm UNIX’in zerafetinde yatıyor. Aynı anda iki tane **tar** programı başlatıp, birinin yarattığı tar yumağını arada hiç disk kaydı yapmadan, yani tar dosyası oluşturmadan, ikinci **tar** programına pompalayabilirsiniz:

```
tar -cv /home/cayfer | (cd /disk2; tar -x)
```

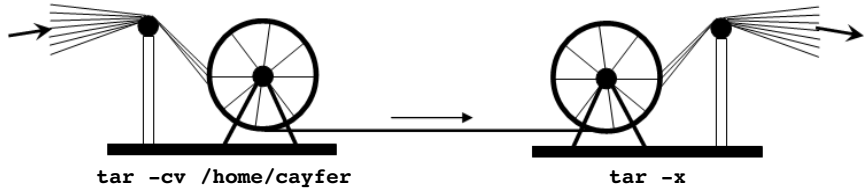
Dikkat ederseniz iki LINUX komutu birlikte başlatılıyor:

1. **tar -cv /home/cayfer**
2. **(cd /disk2; tar -x)**

Bunlardan birincisi (“**tar cv /home/cayfer**”) **-f** parametresi verilmeden kullanıldığı için, yani yaratılacak tar dosyası belirtilmediği için, oluşturacağı tar yumağını standart çıktıya gönderecektir.

İkinci komut bileşik bir komuttur. Noktalı virgülle ayrılmış iki komuttan oluşan ve parantezler arasına yazılmış olan bu bileşik komut da **tar** komutuyla birlikte başlatılacaktır.

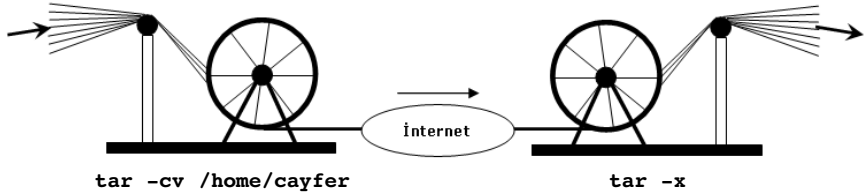
Bu bileşik komut, girdisini “**pipe**” işlemiyle bir önceki komutun çıktısından alacaktır. Bileşik komutun ilk parçası “**cd /disk2**” olduğu için önce çalışma dizini **/disk2** olarak değiştirilecek, sonra da **tar** komutu **x** parametresiyle çalıştırılacaktır. Bu ikinci **tar** komutunda da **-f** parametresi kullanılmadığı için çözülecek tar yumağı standart girişte aranacaktır. Birinci **tar** programı bir yandan yumağı oluştururken ikinci **tar** programı da bu yumağı yeni dizine çekecektir. “(**cd /disk2; tar -x**)” bileşik komutu yerine “**tar -x -C /disk2**” komutu da kullanılabilirdi elbette.



İşi biraz daha karıştıralım isterseniz:

Birinci **tar** programını bir bilgisayarda; ikinci **tar** programını da bir başka bilgisayarda (ikisinin de UNIX bilgisayarı olması kaydıyla elbette) çalıştırarak yedeklemeyi bir başka bilgisayar üzerine yapmaya ne dersiniz?

tar -cv /home/cayfer | rsh 192.168.1.2 (cd /yede; tar -x)



Evet, birinci **tar** komutu sizin makinenizde çalışırken ikinci **tar** komutu 192.168.1.2 IP numaralı bir başka makinedeki **/yede** dizini çalışma dizini iken çalışacak. Bu komutun çalışabilmesi için 192.168.1.2 IP numaralı bilgisayarda, sizin bilgisayarınız tarafından komut çalıştırılmasına izin verilmiş olması gerekir ki bu da başka bir bölümün konusudur.

tar Komutunu Kullanırken Dikkat Edilmesi Gereken Noktalar

tar komutunu kullanırken çok tekrarlanan bazı hatalara dikkatinizi çekmek istiyoruz.

- **tar** komutu, **tar** dosyası yaratırken dosya ve dizin ayırımı yapmaz. Parametre olarak verilen dosya kalıbına uyan her şey **tar** dosyasının içine paketlenir.

Dizinler ve alt dizinleri buna dahildir.

- “**tar -cvf yedek.tar ***” komutu (çalışma dizinindeki her şeyi **yedek.tar** dosyası olarak birleştir gibi okunan komut) aslında tam olarak istediğiniz işi yapmayacaktır. Komutun bu şekilde kullanılması durumunda adı “.” (nokta) ile başlayan dosyalar **tar** dosyasına dahil edilmeyecektir. Adları noktayla başlayan dosyaları da paketlemek istiyorsanız tüm dizini tar’lamak zorundasınız. Bir dizinin tümünü paketlerken istemediğiniz alt dizinleri **-exclude** parametresiyle paket dışında bırakabileceğinizi unutmayın.
- **tar** programı, dosya çözerken diskte aynı isimde bir dosya/dizin olsa bile uyardan üzerine yenisini indirecektir. Diskteki eski dosyaların üzerine kayıt yapılmaması için **-k** (*keep*) parametresini kullanabilirsiniz.
- **tar** programı, tar dosyası yaratırken bağlantılı dosyaları (linkleri) kopyalamaz ve bu bağlantıları izlemez. Örneğin **/var/spool/mail** dizini **/disk2/mail** dizinine linkli ise, **/var** dizini paketlenirken **/var/spool/mail** dizininde görünen ama aslında **/disk2/mail** dizininde yer alan dosyalar pakete alınmaz. Alınmasını istiyorsanız **-h** parametresini kullanmalısınız.

rsh Komutu

Bir önceki sayfada, bir dizinin, olduğu gibi bir bilgisayardan bir başkasına transfer edilmesine ilişkin **tar** örneğinde

```
rsh 192.168.1.2 (cd /yedek; tar -x)
```

şeklinde bir komut kullanmıştık.

Adı “Remote Shell” sözcüklerinin kısaltmasından gelen **rsh** komutu, bir bilgisayarın terminalinde çalışırken, başka bir bilgisayarda komut çalıştırıp, o komutun varsa ürettiği STDOUT’a gidecek görüntüyü çalıştığınız terminalle alabilmek için kullanılır.

Örneğin, 168.4.4.2 IP adresli bilgisayarda çalışırken 139.179.211.10 IP adresli bilgisayardaki kişisel dizininizdeki dosyaların listesini görmeniz gerekirse; ikinci bir terminal ekranı açıp, orada 139.179.211.10 bilgisayarına **telnet** veya **ssh** ile bağlanıp **ls** komutunu vermeniz gerekmez. Bu uzun yöntem yerine 168.4.4.2 IP adresli bilgisayarın terminal penceresinde

Kim Korkar LINUX'tan?

```
rsh 139.179.211.10 ls /home/cayfer
```

komutunu verirseniz “**ls /home/cayfer**” komutu, 139.179.211.10 IP adresli bilgisayarda çalıştırılacak, dosya ve dizin listesi, komutu verdiğiniz terminal penceresine listelenecektir.



ssh, telnet yerine kullanılabilir, daha doğrusu kullanılmasını hararetle önerdiğimiz güvenli bir terminal emülasyon programıdır.

Buraya kadar çok iyi; ama önlem alınmazsa, **rsh** komutu insanı dehşete düşürecek bir güvenlik riskini de yanında getirecektir. Düşünsenize; yerini bile bilmediğiniz bir bilgisayardan birileri sizin bilgisayarınıza yönelik olarak

```
rsh 168.4.4.2 /bin/rm -r /etc
```

komutunu verirse neler olur?

İşte bu yüzden **rsh** komutunun çalıştırılabilmesi için özel izin gerekir.

Bu özel izin iki şekilde verilebilir:

1. Her kullanıcı kendi izinlerini kendisi düzenler.
2. Sistem yöneticisi tüm sistem için geçerli izinleri düzenler.

Kullanıcılar kendi kimlikleriyle ilgili izinleri düzenlemek için kendi kişisel dizinlerinde adı **.rhosts** olan bir dosya (dosyanın adının başındaki noktaya dikkat!) hazırlarlar. Bu dosyada, **rsh** komutunun hangi bilgisayarlardan, hangi kimliklerle verilmesi durumunda komutun çalıştırılabileceği belirtilir. Örneğin **cayfer** isimli kullanıcı kendi kişisel dizininde (**/home/cayfer**), içinde

```
139.179.210.4
```

```
www.abc.com.tr  murat
```

```
www.xyz.edu.tr  omer
```

satırları olan bir **.rhosts** dosyası yaratırsa;

- 139.179.210.4 bilgisayarından kullanıcı adı “**cayfer**” olan kullanıcıya bu bilgisayara yönelik olarak “**cayfer**” kimliğiyle **rsh** komutu verme yetkisi verilmiş olur. Ayrıca,

- **www.abc.com.tr** bilgisayarından “**murat**” ve **www.xyz.edu.tr** bilgisayarı-
rından “**omer**” kimliği ile verilen **rsh** komutlarının “**cayfer**” kimliğiyle
çalıştırılmasına izin verilmiş olur.

Eğer bir başka bilgisayardaki tanımlı tüm kullanıcıların sizin bilgisayarınızda-
ki aynı isimle tanımlanmış kullanıcı kimlikleriyle **rsh** komutunu çalıştırmala-
rına izin vermek istiyorsanız her kullanıcı dizinine teker teker **.rhosts** dos-
yası yerleştirmektense bir **/etc/hosts.equiv** dosyası yaratıp içine diğer bil-
gisayarın ya da bilgisayarların IP adreslerini veya açık adlarını yazabilirsiniz.
Örneğin, bilgisayarınızda **/etc/hosts.equiv** dosyası varsa ve içinde

139.179.2.123

abc.xyz.edu.tr

satırları yer alıyorsa, 139.179.2.123 ve abc.xyz.edu.tr bilgisayarında kayıtlı
kullanıcılar, sizin bilgisayarınıza yönelik **rsh** komutu verebilirler. Sizin bilgi-
sayarınızda da aynı adla tanımlı kullanıcılar olması kaydıyla, **rsh** ile gönderi-
len bu komut(lar) sizin bilgisayarda çalıştırılacak, varsa komutun STDOUT’a
göndereceği mesajlar, rsh komutunu veren bilgisayara gönderilecektir.

Hatırlarsanız, daha önce LINUX’ta kullanıcı tanımlarının isimle değil, kul-
lanıcı numarasıyla tutulduğunu söylemiştik. Uzaktan çalıştırılan komutlarda
kullanıcı tanımlaması bunun bir istisnasıdır. Aynı insana ait hesap adının iki
değişik bilgisayarda aynı kullanıcı numarasıyla kaydedilmesini sağlamak ne-
redeyse olanaksız olduğu için, **rsh** izni düzenlemelerinde numara değil, isim
esas alınır.

rsh aslında UNIX’ta “**Remote Commands**” (Uzaktan komutlar) olarak anı-
lan bir komut ailesinin bir bireyidir. **rsh** ile aynı mantıkta çalıştırılan ve izin-
leri **.rhosts** ile **/etc/hosts.equiv** dosyalarıyla denetlenen

r**cp** (*remote copy*)
r**login** (*remote login*)

komutları da birer “**remote command**” olarak kullanılabilir.

İki ayrı bilgisayarda yer alan dosya sistemleri arasında dosya/dizin kopyala-
mak için **r****cp** komutunu kullanabilirsiniz. (Tabii ki **.rhosts** veya
/etc/hosts.equiv dosyalarıyla uygun izinlerin verilmiş olması kaydıyla.)

Örneğin:

```
rcp sunucu:/var/www/html/* /home/cayfer
rcp -r /home/cayfer/public_html/* 139.179.1.1:/var/www/html
```

gibi.

rlogin komutunu ise şifre vermeden bir başka makinedeki aynı isimli hesabınıza bağlanmak için kullanabilirsiniz. Ancak biz **rlogin** yerine **ssh** komutunu kullanmanızı ve şifre girmeye üşenmemenizi öneririz.

xargs Komutu

Kabuk programlarının dosya ismi kalıplarını işlerken karşılaşılabilecekleri bir sorun vardır. Ama bu sorunu bir örnekle açıklamak daha kolay olacak galiba...

Diyelim ki bir dizin içinde 5000 dosya var ve bunların yarısının isimleri ***.log**, gerisi de ***.bak** kalıbında. Siz adı ***.bak** kalıbına uyan dosyaları silmek istiyorsunuz ve **"rm *.bak"** komutunu veriyorsunuz, ardından da **"Argument list too long"** hata mesajını alıyorsunuz.

Aslında şöyle bir düşününce bu hata mesajını almanız son derece normaldir. Siz **"rm *.bak"** komutunu verdiğinizde, kabuk programınız çalışma dizinizde yer alan ve adı ***.bak** kalıbına uyan tüm dosya isimlerini bulup bunları aralarında birer boşlukla komutunuzun **rm** kısmının ardına dizmeye çalışacaktır. Bu kalıba uyan 2500 dosya olsa ve her birinin adı 10 karakter uzunluğunda olsa siz yaklaşık 25 Kbyte uzunluğunda bir komut satırı yazmış gibi olacaksınız. Eh, herşeyin bir sınırı olmalı değil mi? İşte sizin bu komut, **bash** programının komut satırı için ayırdığı tampon alanının dışına taşıdığı için **"too long"** mesajını alıyorsunuz.

UNIX'te her problemin bir çözümü olduğunu farketmişsinizdir artık herhalde. İşte bu sorunun çözümü de **xargs** komutudur.

```
ls *.bak | xargs /bin/rm
```

Yukardaki bileşik komutun ilk parçası çalışma dizininde yer alan adı ***.bak** kalıbına uygun dosyaların isim listesini üretecek (binlerce dosya isminden oluşan bir liste olabilir); ikinci bölümü de bu listedeki her bir dosya için **/bin/rm** komutunu çalıştıracaktır.

Benzeri bir örnek daha: Diyelim ki bir dizin ve altındaki tüm alt dizinlerdeki dosyaların arasından, içinde “**www.bilkent.edu.tr**” karakter dizisi yer alan dosyaları bulmak istiyorsunuz.

İlk akla gelen çözüm:

```
grep "www.bilkent.edu.tr" `find .`
```

komutunu yazmaktır.

Haydaaaa.. Bu da nerden çıktı diyebilirsiniz. Komutun analizi şöyle: Önce **find** komutu yalnızca “.” dan oluşan bir parametreyle başlatılıyor. **find** komutu bu haliyle çalışma dizini ve altındaki tüm dosyaların ve dizinlerin isimlerini listeliyor.

find komutu `` tırnakları arasında yer aldığı için önce bu komut çalıştırılacak ve komutun standart çıktıya gönderdikleri bu tırnaklar arasına yerleştirilecektir. Böylece **grep** komutunun sonuna uzunca bir dosya adı listesi eklenmiş olacaktır.

Buraya kadar her şey iyi; ancak dosya listesi çok uzunsa komut satırı da **bash** kabuğunun sınırlarını aşacaktır ve **grep** komutu çalıştırılamayacaktır.

Çözüm için gene **xargs** önereceğiz:

```
find . | xargs grep "www.bilkent.edu.tr"
```

Aslında yukardaki komutla yapılan iş

```
grep -r www.bilkent.edu.tr *
```

komutuyla da yapılabilirdi; ancak amacımız **xargs** komutuna bir örnek vermektir.

find komutunu anlatırken verdiğimiz “belirli bir kullanıcıya ait tüm dosyaları silme” işini hatırlıyor musunuz? İşte **xargs** ile benzeri bir iş yapan komut:

```
find / -user hasan | grep -v \.dat xargs /bin/rm
```

Yukardaki komut sahibi **hasan** olan dosyalar arasında adında “**.dat**” geçmeyen dosyaları silecektir.

find komutu sahibi **hasan** olan dosyaları listeleyecektir. Bu liste **grep** programına “**-v \.dat**” parametresiyle gönderilecek; **-v**'den dolayı içinde “**.dat**” geçen satırlar değil; geçmeyenler listelenecektir. Elde edilen bu dosya listesi de **xargs** aracılığıyla **/bin/rm** komutuna gönderilerek dosyalar silinecektir. Bu örnekte “**.dat**” yerine “**\.dat**” yazıldığı, yani noktanın işaretlenmiş olduğu dikkatinizi çekmiş olmalı. Bunun nedeni şudur: Eğer “**.dat**” yazsaydık, bu ifade **grep** tarafından “herhangi bir karakter ve ardından gelen dat” olarak değerlendirilirdi. Böylece adında “**sedat**” geçen dosyalar liste dışında kalırdı. Oysa basit nokta yerine “**\.**” yazmakla **grep**'e noktayı, “her karaktere uyan” bir joker karakter değil de, bildiğimiz “**.**” olarak değerlendirmesini istediğimizi belirtmiş olduk.

at Komutu

at [-m] saat

Vereceğiniz komutların belirli bir gün ve saatte başlatılmasını sağlar. Tarih belirtmezseniz “içinde bulunduğumuz gün” kabul edilir. Web sitenizin ana sayfasının 31 Aralık günü tam geceyarısı değişmesini istiyorsanız, önceden hazırlayacağınız **index_yeni.html** dosyasının saat tam 23:59'da **index.html** üzerine kopyalanması için:

```
at -m Dec 31 23:59
```

komutunu kullanarak

```
cd /var/http/html/  
/bin/cp index_yeni.html index.html
```

komutlarını geceyarısına bir dakika kala çalışacak şekilde programlayabilirsiniz.

```
Saat belirtirken    13:30  
                    now + 10 minutes  
                    4pm tomorrow  
                    midnight  
                    Jun 30 4am
```

gibi formlar kullanabilirsiniz.

Tarih vermek istediğinizde kullanmanız gereken form, “**aaa gg**” olmalıdır. Burada **aaa**, ay isimlerinin İngilizcelerinin üç harfli kısaltmalarından biri olmalıdır. (Jan, Feb, ... gibi) **gg** ise doğal olarak gündür... **-m** seçeneği ise, programın başarıyla çalıştırılması durumunda komutu veren kullanıcıya bir e-posta mesajı gönderilmesini sağlar.

at komutunu yukardaki formlardan birinde verdiğinizde belirttiğiniz saatte çalıştırılmasını istediğiniz komutları girmenizi isteyecektir. Bu komutları girmeniz tamamlandığında Ctrl-D tuşuna basarak standart girişten girilecek verilerin bittiğini belirtmelisiniz.

Aşağıdaki örnek **at** komutu, **/usr/local/bin/backup** komutuyla **/dev/hda1** diskinin yedeğini alma işini gece yarısına bir dakika kala başlatılacak şekilde kuyruğa atıyor. Ayrıca bu komutun çalışması bitince **/home/cayfer** dizinindeki “**mesaj**” dosyasını da “**mail ayfer@ieee.org**” komutuyla Ömer’e e-posta olarak gönderiyor.

```

cayfer@notebook.ijman.bilkent.edu.tr: /home/cayfer - Shell - Konsole <4>
[cayfer@notebook cayfer]$ at -m 23:59
warning: commands will be executed using (in order) a) $SHELL b) login shell c) /bin/sh
at> /usr/local/bin/backup /dev/hda1
at> mail ayfer@ieee.org < /home/cayfer/mesaj
at> <EOT>
job 3 at 2003-02-11 23:59
[cayfer@notebook cayfer]$

```

at komutuyla zamanının gelmesini beklemek üzere kuyruğa atılmış işleri **atq** komutuyla görebilirsiniz.

```

cayfer@notebook.ijman.bilkent.edu.tr: /home/cayfer - Shell - Konsole <4>
[cayfer@notebook cayfer]$ atq
1
    2003-02-11 23:59 a cayfer

[cayfer@notebook cayfer]$

```

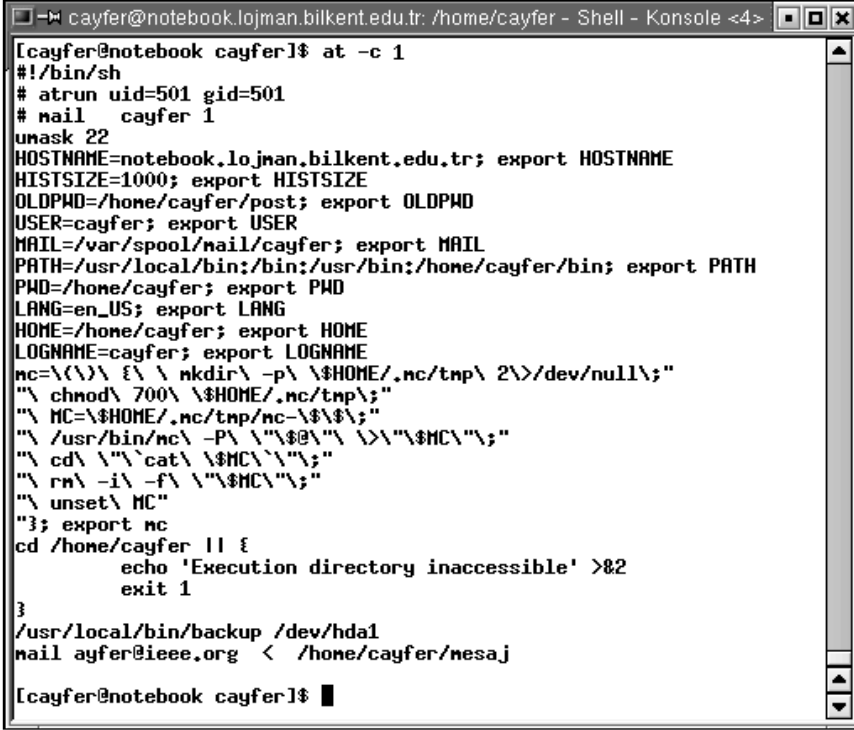
Bu listede kuyrukta bekleyen işlerin herbirinin bir numarası (7, 8, 9) ve beklediği kuyruğun adı (a) görünür.

Kuyrukta bekleyen işlerin ne olduğunu hatırlamak istediğinizde

at -c 1

Kim Korkar LINUX'tan?

gibi bir komutla, örneğin 1 numaralı işin ayrıntılı komut dizisini görebilirsiniz.



```
[cayfer@notebook cayfer]$ at -c 1
#!/bin/sh
# atrun uid=501 gid=501
# mail cayfer 1
unmask 22
HOSTNAME=notebook.lojman.bilkent.edu.tr; export HOSTNAME
HISTSIZE=1000; export HISTSIZE
OLDPWD=/home/cayfer/post; export OLDPWD
USER=cayfer; export USER
MAIL=/var/spool/mail/cayfer; export MAIL
PATH=/usr/local/bin:/bin:/usr/bin:/home/cayfer/bin; export PATH
PWD=/home/cayfer; export PWD
LANG=en_US; export LANG
HOME=/home/cayfer; export HOME
LOGNAME=cayfer; export LOGNAME
nc=\(\)\ \ \ nkdir\ -p\ \ $HOME/.nc/tnp\ 2\>/dev/null\;"
"\ chmod\ 700\ \ $HOME/.nc/tnp\;"
"\ MC=\$HOME/.nc/tnp/nc-\$\$\;"
"\ /usr/bin/nc\ -P\ \ "\$@"\ \ \ "\$MC\;"
"\ cd\ \ "\ cat\ \ $MC\ \ \;"
"\ rm\ -i\ -f\ \ "\$MC\ \;"
"\ unset\ MC"
"}; export nc
cd /home/cayfer || {
    echo 'Execution directory inaccessible' >&2
    exit 1
}
/usr/local/bin/backup /dev/hda1
mail ayfer@ieee.org < /home/cayfer/nesaj

[cayfer@notebook cayfer]$
```

Bu ayrıntılı döküm ilk bakışta sizi dehşete düşürebilir. Çünkü sizin yazdığınız komut satırlarının önüne bir çok kabuk komutu yerleştirilmiş olduğunu göreceksiniz. Sistem tarafından eklenen bu komutların espirisi şudur: Kuyruğa attığınız işin çalışma zamanı geldiğinde büyük olasılıkla siz sistemde olmayacaksınız. Dolayısıyla sizin için çalışmakta olan bir kabuk da olmayacaktır. Eh... peki sistem verdiğiniz komutu nasıl bir ortamda çalıştırsın? İşte bu baştaki komutlar, işinizin çalıştırılma zamanı geldiğinde sizin yerinize bir kabuk başlatıp sizin kişisel kabuk ortamınızı oluşturmak içindir. Bu laflar hoşunuza gitmediyse boşverebilirsiniz. Alt tarafı gece yarısı çalıştırmanız gereken bir iş olursa diğer işletim sistemi yöneticileri gibi kalkar iş yerine gider, işi adam gibi elle çalıştırıverirsiniz.

at komutuyla ileri bir saatte çalıştırılmak üzere programladığınız bir işi iptal etmek istediğinizde

atrm n

komutunu kullanabilirsiniz. Burada **n**, çalışmaya başlamak için zamanın gelmesini bekleyen işin **atq** komutu tarafından bildirilen sıra numarasıdır. Yani kuyruktaki bir işi iptal etmeden önce **atq** komutuyla o işin kuyruk sıra numarasını öğrenmeniz gerekir.

at komutunun çalışabilmesi için arka planda sürekli çalışan ve geçen zamanı kollayan bir yazılım olmalıdır. LINUX’da bu işi **atd** programı yapar. **at** komutunuzun çalışması için arkada **atd** çalışır durumda olmalıdır. (**at daemon**) Eğer **at** komutunu verdiğinizde **atd**’nin çalışmadığına ilişkin bir hata mesajı alırsanız **atd**’yi siz elinizle başlatabilirsiniz. **atd**’nin bilgisayarınızın her açılışında otomatik olarak başlatılması için birtakım işler yapmalısınız. Bu aşamada yalnızca **atd**’nin elle başlatılmasını göstermekle yetineceğiz:



```
/etc/rc.d/init.d/atd start
```

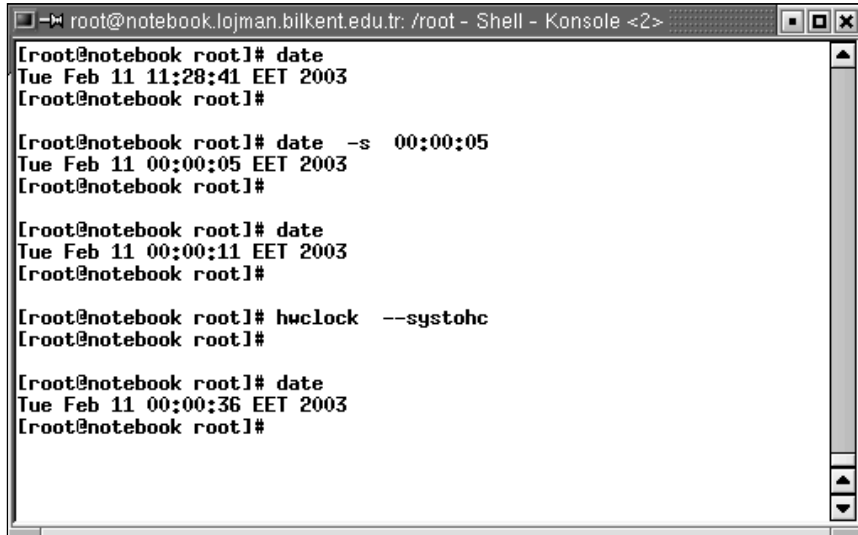
Açılış sırasında otomatik başlatma işini sistem yönetimiyle ilgili bölümlerde öğreneceksiniz.

at komutuyla belirli bir tarih ve saat için programlanan işler, zamanı geldiğinde yalnızca bir kez çalıştırılır. Periyodik olarak, örneğin, her gece yarısı tekrarlanmasını istediğiniz işler için **at** komutundan yararlanamazsınız.

Eğer belirli bir işin “her saat başı”, “her gece yarısı”, “her Pazartesi sabah 09:03’de” otomatik olarak başlatılmasını istiyorsanız, **cron** yazılımından yararlanabilirsiniz. **cron** yazılımı doğrudan bir komutla kullanılmaz. Periyodik olarak yapılmasını istediğiniz işleri **cron** yazılımının ayar dosyası üzerinde yapacağınız düzenlemelerle belirtirsiniz. Bu düzenlemeler için de “**crontab -e**” komutu kullanılır. **cron** hakkında daha ayrıntılı bilgiyi kitabın “Sistem Yönetimi” bölümünde bulabilirsiniz.

date, hwclock

Bazen sisteminizin saatinin yanlış olduğunu; biraz ileri gittiğini ya da geri kaldığını farkedersiniz. LINUX’un saatini **date** komutuyla değiştirebilirsiniz ama bu komut bilgisayarınızın saat devresinin ayarını değiştirmeyecektir. İşletim sisteminin tuttuğu saat ve tarihi bilgisayarın saat devresine kaydetmek için **hwclock** komutunu kullanmalısınız.



```
root@notebook.lojman.bilkent.edu.tr: /root - Shell - Konsole <2>
[root@notebook root]# date
Tue Feb 11 11:28:41 EET 2003
[root@notebook root]#

[root@notebook root]# date -s 00:00:05
Tue Feb 11 00:00:05 EET 2003
[root@notebook root]#

[root@notebook root]# date
Tue Feb 11 00:00:11 EET 2003
[root@notebook root]#

[root@notebook root]# hwclock --systohc
[root@notebook root]#

[root@notebook root]# date
Tue Feb 11 00:00:36 EET 2003
[root@notebook root]#
```

hwclock --systohc

komutu LINUX sistem saatini bilgisayarın saat devresine kaydeder.

hwclock --hctosys

komutu ise bilgisayarın saat / takvim devresindeki zamanı LINUX sistem saati olarak alır.

lynx

Bir senaryo: Yaptığınız bir çalışma için web adresini ezbere bildiğiniz bir yerden bir dosya indirmeniz gerekti. O sırada da Netscape, Konqueror veya Mozilla gibilerinden bir grafik tarayıcınız açık değil. Dosya indireceğiniz sitenin web sayfalarının grafik unsurları da sizi ilgilendirmiyor.

lynx böyle bir durumda çok işinize yarayacaktır. Karakter tabanlı terminal pencereleri için yazılmış küçük, hızlı ama yetenekli bir web tarayıcısıdır **lynx**. Belleğe çabuk yüklenmesi sayesinde indirmek istediğiniz dosyaya hızla erişebilirsiniz.

```

cayfer@cayfer.bilkent.edu.tr: /home/cayfer - Konsole - Konsole
[USEMAP:zenin1.jpg]
MAP: http://www.pusula.com/#FPMa0
1. http://www.pusula.com/hakkinda/index.html
2. http://www.pusula.com/kinkorkar/index.html
3. http://www.pusula.com/bilgisayarokulu/default.htm
4. http://www.pusula.com/multimedia/index.html
5. http://www.pusula.com/virgul/default.htm
6. http://www.pusula.com/other/index.html
7. mailto:pusula@pusula.com
8. http://www.pusula.com/english/index.html

Commands: Use arrow keys to move, '?' for help, 'q' to quit, '<-
Arrow keys: Up and Down to move. Right to follow a link; Left t
H)elp O)ptions P)rint G)o M)ain screen Q)uit /=search [delete]=h

```

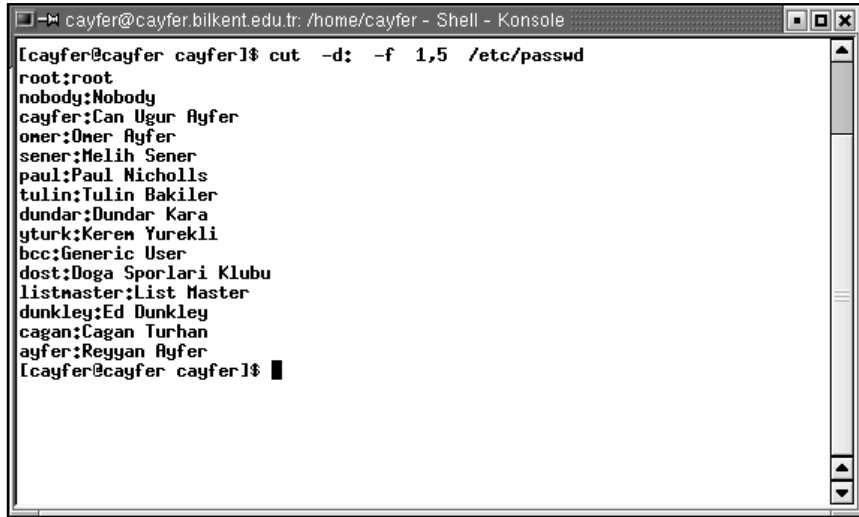
cut

cut, **more** ve **less** komutları basit metin dosyalarını ekrana görüntülemek için oldukça kullanışlı komutlar olmakla birlikte, bazen dosyaları olduğu gibi listelemek yerine bazı satırların bazı alanlarını listelemek de isteyebilirsiniz.

Örneğin **/etc/passwd** dosyasının yalnızca hesap ismi ve kullanıcı ismi alanlarından oluşan bir listeye gereksinim duyduğunuzda

```
cut -d: -f 1,5 /etc/passwd
```

komutuyla “:” ile ayrılmış olan alanlardan yalnızca birinci ve beşincileri listeyebilirsiniz.

A terminal window titled "cayfer@cayfer.bilkent.edu.tr: /home/cayfer - Shell - Konsol" displays the command `cut -d: -f 1,5 /etc/passwd` and its output. The output lists system and user accounts with their names in Turkish. The prompt `[cayfer@cayfer cayfer]$` is shown at the end.

```
[cayfer@cayfer cayfer]$ cut -d: -f 1,5 /etc/passwd
root:root
nobody:Nobody
cayfer:Can Ugur Ayfer
omer:Omer Ayfer
sener:Melih Sener
paul:Paul Nicholls
tulin:Tulin Bakiler
dundar:Dundar Kara
yturk:Keren Yurekli
bcc:Generic User
dost:Doga Sporlari Klubu
listmaster:List Master
dunkley:Ed Dunkley
cagan:Cagan Turhan
ayfer:Regyan Ayfer
[cayfer@cayfer cayfer]$
```

Kolay incelenebilmesi için gerçek örnek sadeleştirilmiştir.

cut komutuna **-d** parametresiyle satırlardaki alanları ayıran ayraç tanıtılır. Ayraç belirtmezseniz ayraç karakteri olarak <Tab> kabul edilir. İlk 5 alanı listelemek istediğinizde “**-f1, 2, 3, 4, 5**” parametresi yerine “**-f 1-5**” parametresini kullanabilirsiniz.

Elde ettiğiniz liste çok uzunsa, tüm LINUX komutlarında olduğu gibi **cut** programının standart çıktıya gönderdiklerini **more** veya **less** programına yönlendirebilirsiniz.

tee

Bazı durumlarda bir programın çıktısını iki değişik yere yönlendirmeye gereksinim duyabilirsiniz. Örneğin uzun bir dosyayı sıralayıp, sıralı dosyayı hem **less** ile sayfa sayfa görüntülemek hem de bu sıralı kayıtları bir dosyaya yönlendirmek isteyebilirsiniz.

```
sort adresler | tee adresler.sirali | less
```

komutu “**adresler**” dosyasını sıralayarak sıralı halini **tee** programına yönlendirilecek; **tee** programı standart girdisinden gelen verileri hem **adresler.sirali** dosyasına hem de kendi standart çıktısına yönlendirecek; son olarak da “**less**” komutu bu satırları sayfa sayfa listeleyecektir.

script

Diyelim ki çok sayıda komuttan oluşan bir dizi iş yapmak ve bu komutları ve aldığınız yanıtları sonradan incelemek üzere saklamak istiyorsunuz. **script** programı, sizi bu iş boyunca, hangi komutu verdiğiniz ve ne yanıt aldığınızı kağıda not etmekten kurtaracaktır.

```
script /tmp/log1
```

komutu, verildiği andan başlayarak Ctrl-D tuşuna basıncaya kadar verdiğiniz tüm komutları ve bu komutlara ilişkin STDOUT'a gönderilen herşeyi **/tmp/log1** dosyasında saklar. **script**'i Ctrl-D ile durdurduktan sonra bu **/tmp/log1** dosyasını bir anlamda "kaptanın seyir defteri" olarak kullanabilirsiniz.

script komutunu

```
script -a /tmp/log1
```

şeklinde kullanırsanız, **seyir** kayıtları **/tmp/log1** dosyasının olası eski içeriğine eklenir.

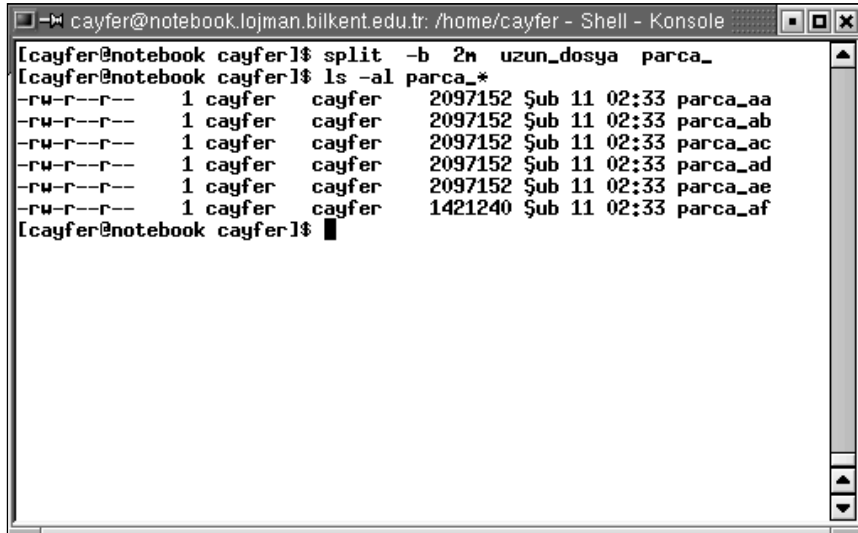
split

Bazen büyük dosyaları küçük parçalara ayırmak zorunda kalırsınız. Örneğin 50 Mbyte uzunluğunda bir dosyayı bir başkasına göndermek için e-posta'dan başka olanağınız yoksa, tek çıkar yol dosyayı e-posta sunucularının kabul edeceği büyüklükte (tipik olarak 10 Mbyte) parçalara bölüp öyle göndermektir.

split komutu, dosyaları isterseniz belli büyüklükte parçalara, isterseniz belli sayıda satır içeren parçalara bölebilir.

```
split -b 1m uzun_dosya parca_
```

uzun_dosya isimli dosyayı birer megabyte'lık parçalara ("**-b 1m**" parametresi) bölecektir. Parçaları oluşturan dosyaların isimleri de sırasıyla **parca_aa**, **parca_ab**, **parca_ac**, ..., **parca_az**, **parca_ba** gibi isimler olacaktır.



```
cayfer@notebook.lojman.bilkent.edu.tr: /home/cayfer - Shell - Konsol
[cayfer@notebook cayfer]$ split -b 2n uzun_dosya parca_
[cayfer@notebook cayfer]$ ls -al parca_*
-rw-r--r-- 1 cayfer cayfer 2097152 Şub 11 02:33 parca_aa
-rw-r--r-- 1 cayfer cayfer 2097152 Şub 11 02:33 parca_ab
-rw-r--r-- 1 cayfer cayfer 2097152 Şub 11 02:33 parca_ac
-rw-r--r-- 1 cayfer cayfer 2097152 Şub 11 02:33 parca_ad
-rw-r--r-- 1 cayfer cayfer 2097152 Şub 11 02:33 parca_ae
-rw-r--r-- 1 cayfer cayfer 1421240 Şub 11 02:33 parca_af
[cayfer@notebook cayfer]$
```

Öte yandan,

```
split -l 1000 uzun_dosya parca_
```

komutu, **uzun_dosya** isimli dosyayı 1000'er satırlık parçalara ("**-l 1000**") bölecek; parçaları **parca_aa**, **parca_ab**, **parca_ac**, ..., **parca_az**, **parca_ba** gibi isimlendirecektir. Satır uzunlukları eşit değilse, doğal olarak parça dosyaların büyüklükleri de byte ölçüsüyle aynı olmayabilecektir.

Daha sonra, bu parçaları tekrar birleştirmeniz gerektiğinde

```
cat parca_* > uzun_dosya
```

komutunu kullanabilirsiniz. Kabuk programınız "**parca_***" dosya isim kalıbını, bu kalıba uyan dosyaların isim listesine dönüştürürken alfabetik sıralama kullanacağı için parçalanmış dosyalar doğru sırada birleştirilecektir.

BUNLARI BİLİYOR MUYDUNUZ?

İşletim sistemi olmayan UNIX'ler

“UNIX” markasının kullanıldığı ilginç bazı ürünleri araştırırsanız:

- mikrodalga fırın uyumlu gıda saklama kapları
- yangın söndürücüsü
- ahşap kitap raf sistemi
- tükenmez kalem
- çamaşır askısı
- kimyasal tiner
- buğday gevreği
- televizyon anteni
- masaj aleti
- gözlük çerçevesi
- saç kurutma makinesi
- otomobil parçaları
- çocuk bezi
- kiralık oto hizmetleri
- bar, berber salonu
- mobilya

gibi ürün ve hizmetler bulabilirsiniz. (Ayrıntılar için bkz.
<http://cm.bell-labs.com/cm/cs/who/dmr/otherunix.html>)

Kim Korkar LINUX'tan?

- **Sistem Hakkında**
 - **Sistemin Açılması**
 - **Sistemin Kapatılması**
 - **Disklerin Yönetimi**
 - **Bellek Yönetimi**
- **Kullanıcı Yönetimi**
 - **Kullanıcı Hesapları**
 - **Kullanıcı Grupları**
- **Log Dosyalarının Yönetimi**
- **Ağ Yönetimi**
- **TCP/IP Ayarları**
- **Yazıcı Yönetimi**
- **Yedekleme**
- **Periyodik İşlerin Yönetimi**
- **LILO Yönetimi**
- **Webmin**
- **“root” Şifresini Unuttuğunuzda...**

UNIX sistem yöneticiliği bir meslektir, hem de önemli bir meslek! İnternet'in hızla yayılması nedeniyle de "Ağ Yönetimi" ve "Sistem Yönetimi" diye anılan iş kolları birbirlerine iyice yaklaşmış; hatta birleşmiştir.

Artık sistem yöneticileri ağ yönetiminden; ağ yöneticileri de sistem yönetiminden anlamak zorundadır.

Bir "Sistem Yöneticisi"nin öncelikli görevi sistemi ayakta tutmaktır. Bu sistem tek bir bilgisayardan oluşabileceği gibi dünyaya yayılmış birçok bilgisayardan da oluşabilir. LINUX işletim sisteminin çok güvenli olduğunu defalarca tekrarladık ama sonuçta insan yapısı bir bilgisayar üzerinde çalıştığını da unutmamak gerekir.

Bir bilgisayar sisteminin düzgün çalışmasında; iyi servis verilmesinde belki de en önemli eksen sistem yöneticisidir. Yeterince iyi bir sistem yöneticisi, işletim sistemi ve donanım özellikleri ne olursa olsun (Linux veya Windows; IBM veya toplama bilgisayar) başarılı ağ hizmetleri sunmayı başarabilir.

Donanım arızaları, kullanıcı hataları, sistem yöneticisi hataları, saldırılara yelik düşme, iletişim hataları, sahtekarlık, sabotaj gibi bir sürü nedenle bir sistem çökebilir. İşte sistem yöneticileri bu çöküşlerde sistemi hızla yeniden çalışır duruma getirmek, çöküş olasılığını azaltmak, çöküş olduğunda da zararı olabildiğince küçültebilmek için çalışmak, hem de çok çalışmak zorundadır.

Bir sistem yöneticisinin ikinci öncelikli görevi ise sistemin kullanıcılara sürekli ve güvenilir hizmet vermesini sağlamaktır. Bu görevin gerektirdiği bir sürü alt görev olmakla beraber en genel ve önemli olanları şunlardır:

- Sistemin kaynaklarının verimli bir şekilde kullanılmasını sağlamaya yönelik önlemleri almak.
- İşletim sistemini ve uygulama programlarını güncellemek.
- Sistemde yüklü program ve verilerin yedeklenmesi işlerini düzenlemek.
- Sistemin güvenliğini sürekli olarak denetlemek ve yeni saldırı teknikleriyle yeni ortaya çıkan güvenlik açıklarına karşı önlem almak.

Sistem Hakkında

Sistemin Açılması

Sistem yöneticilerinin en iyi bilmeleri gereken süreçlerden biri sistemin açılış sürecidir.

“boot” adı verilen sistemin açılış sürecini LINUX işletim sisteminde denetleyen yazılım genellikle LILO’dur. “**LILO**”; Linux Loader sözcüklerinden elde edilmiş bir kısaltmadır. Diğer bir popüler açılış yönetici yazılımı ise GRUB’dur, ama bu kitapta yalnızca LILO’dan söz edeceğiz.

Bilgisayara elektrik verilmesi ve kullanıcının bir login ekranıyla karşılaşması arasında olup bitenleri, fazla ayrıntısına girmeden de olsa, olabildiğince iyi anlamanız birçok sistem yönetimi konusunu kavramanıza yardımcı olacaktır. Şimdi bu açılış sürecini adım adım gözden geçirelim:

1. Bilgisayarınıza elektrik enerjisi verince anakartın üstündeki BIOS çipinde kayıtlı bulunan küçük bir program çalışmaya başlar.
2. Bu program, BIOS ayarlarınıza bağlı olarak sırasıyla sisteme bağlı disket, CD, disk sürücülerinin birinde işletim sistemini yükleyebilecek bir program arar. Bu program, “**boot sektörü**” olarak bilinen alanda, kullanılmakta olan sürücüden işletim sistemini yükleyebilecek bir program olmasıdır.
3. Bilgisayarınıza LINUX kurduğunuzda ilk diskin “**boot**” sektörüne (**MBR: Master Boot Record** da denir) LILO yazılımı yerleştirilir. (Elbette, GRUB ya da bir başka boot yöneticisi seçtiyseniz, boot sektörüne seçtiğiniz program kaydedilmiş olacaktır.)
4. LILO yazılımı BIOS tarafından belleğe yüklenip çalışmaya başladığında kullanıcıya çeşitli açılış seçenekleri sunabilir. Örneğin, üzerinde hem Windows XP, hem LINUX yüklü bir bilgisayarda kullanıcıya, sistemi istediği işletim sistemiyle açabilmesi için bir seçenek listesi sunulur:



LILO için gerekli ayarlar **/etc/lilo.conf** dosyasında yapılır. Ancak, LILO programı çalıştığı sırada ortada sistemi denetleyen bir işletim sistemi olmadığı için LILO, **/etc/lilo.conf** dosyasına erişemez. **lilo.conf** dosyasında değişiklik yaparsanız, **/sbin/lilo** komutuyla LILO'nun yeni ayarlara göre çalışacak şekilde yeniden hazırlanıp diskin boot sektörüne yazılmasını sağlamalısınız.

5. LILO'nun sunacağı seçenekler arasında "LINUX" seçtiğinizi varsayarak açılış sürecini incelemeye devam edelim... LILO, **/boot** dizininde **vmlinuz** dosyasında bulunan LINUX çekirdiğini (kernel) belleğe yükler ve çalıştırır.
6. Çekirdek programı, konsolun ekran kartına uygun bir görüntü ayarına geçer ve sisteme bağlı olan donanım unsurlarını tarayarak (bellek, merkezi işlem birimi, görüntü kartı, disk arabirimleri, ses kartları, paralel ve seri arabirimler, ağ bağlantı arabirimleri gibi) bunları tanımaya ve ilgili sürücü yazılımlarını (*device driver*) yüklemeye başlar.
7. Çekirdek, daha sonra yine LILO ayarlarında belirtildiği şekilde root dosya sistemini (yani **/** dizininin olduğu fiziksel diski) iliş­tirir (mount eder). Bu dosya sistemi ilk aşamada sadece okuma için (*read-only*) iliş­tirilir ve izin yapısı kontrol edildikten sonra, yani bu dosya sistemi için **fsck** yazılımı çalıştırıldıktan sonra **oku-yaz** (*read-write*) kullanımı için tekrar iliş­tirilir.
8. Bir sonraki adımda, sisteminiz açık kaldığı süre boyunca, "1" süreç numarasıyla sürekli çalışacak olan **init** programı başlatılır. Bu süreç, sistemi kullanıma hazır hale getirmek için **/etc/inittab** dosyasında belirtilen ayarlara göre bir dizi kabuk programı çalıştırarak sistemin "çalışma düzeyini" (*run level*) aşama aşama artırır.

Çalışma düzeyleri, işletim sisteminin hangi yeteneklerinin çalışmaya başladığını belirler. Tipik bir LINUX sisteminde yedi çalışma düzeyi vardır:



- 0: Sistemin kapanma işlemlerinin başlatıldığı düzey. (*Halt Level*)
- 1: Tek kullanıcı çalışma düzeyi. (*Single-user*)
- 2: Ağ desteği olmadan çok kullanıcı çalışma düzeyi.
- 3: Çok kullanıcı çalışma düzeyi. (*Multi-user*)
- 4: Bu çalışma düzeyi nedense kullanılmaz.
- 5: X Window'un çalışmaya başladığı düzey. (Grafik kullanıcı arabirimi)
- 6: Yeniden başlatma işlemlerinin başlama düzeyi. (*Reboot*)

Normal koşullarda; yani başarıyla açılmayı tamamlamış bir LINUX bilgisayar üçüncü (X Window kullanılmıyorsa) veya beşinci düzeyde çalışmasını sürdürür.

Sistemi kapatmak istediğinizde

```
init 0          veya
init 6
```

komutlarından biriyle sisteminizi sıfıncı düzeye indirerek kapatabilir (*halt-shutdown*) veya altıncı düzeye geçirerek yeniden başlatabilirsiniz. (*reboot*)

Yapacağınız bir yazılım veya veri bakım çalışması nedeniyle bilgisayarın başka kullanıcılara hizmet vermeksizin sadece konsoldaki kullanıcıya hizmet verecek şekilde çalışmasını isterseniz

```
init 1
```

komutuyla tek kullanıcı düzeye dönüp, işiniz bittiğinde

```
init 3
```

komutuyla tekrar çok kullanıcı düzeye geçebilirsiniz. Tabi bu arada kullanıcılarınız sizi dövüp sağlık düzeyinizi değiştirmediyse...

Sistemin açılışı sırasında her çalışma düzeyinde yapılacak işler **/etc/inittab** dosyasına tanımlanır. Bu dosyanın deseni ve içeriği konusunda daha fazla ayrıntıya girmek istemiyoruz. Meraklı okuyucular "man **inittab**" komutuyla oldukça ayrıntılı bilgi edinebilirler.

/etc/rc.d dizini altında değişik çalışma düzeylerine geçerken gerekli ve ilgili hizmetlerin başlatılıp durdurulması için kullanılan kabuk programları yer alır. Örneğin, **/etc/rc.d/rc5.d** dizinindeki dosyalar, sistemin beşinci çalışma düzeyine, yani X ortamına geçerken çalıştırılacak kabuk programlarıdır.

```
root@notebook.ijman.bilkent.edu.tr: /root - Shell - Konsole
[root@notebook root]# ls /etc/init.d/rc5.d
K55routed      S13partnon    S26ypxfrd     S66ypasswdd   S90webnbin
S03iptables    S14nfslock    S40atd        S75keytable    S95anacron
S05harddrake   S15gpm        S40saslauthd  S80postfix     S95innd
S08ipchains     S16ypserv     S45irda       S80prelude     S95kheader
S08ipvsadm      S17alsa       S55nand       S85httpd       S99devfsd
S10network     S18sound     S55sshd       S85proftpd     S99linuxconf
S11internet    S20random     S56rawdevices S90cron        S99local
S11pcmcia      S20xfs        S56xinetd     S90mysql       S99medusa
S11portnap     S25netfs      S60lpd        S90postgresql
S12syslog      S26apnd       S60nfs        S90squid
[root@notebook root]#
```

Sistem, beşinci düzeye geçtiğinde, **/etc/rc.d/rc5.d** dizinindeki kabuk programlarını, isimlerinin alfabetik sırasına göre (**ls** komutuyla listelendikleri sırada) çalıştırmaya başlar.

Bunun gibi **/etc/rc.d/rc3.d** dizinindeki dosyalar sistemi üçüncü düzeye çıkarmak için başlatılacak işleri düzenleyen komut ve programları içerir.

/etc/rc.d/rc.local dosyası, sistemin açılışı tamamlandıktan sonra varsa, başlatılacak işlere ilişkin komutları içerir. Örneğin sistemin açılışı tamamlandığında birilerine bunun otomatik olarak haber verilmesini istiyorsanız **/etc/rc.d/rc.local** dosyasının içine (sonuna olabilir)

```
echo `date` Sistem acildi | mail admin@bilkent.edu.tr
```

gibi bir satır ekleyerek her açılıştan sonra **admin**'e açılış tarihini ve saatini bildiren bir e-posta gönderilmesini sağlayabilirsiniz.

Sistemdeki çeşitli servisleri başlatıp durdurmak için kullanılabilecek kodları içeren dosyalar **/etc/rc.d/init.d** dizininde de yer alır. Bu dizindeki dosyalar, çeşitli sunucu yazılımları başlatıp durduracak şekilde yazılmış kabuk programlarıdır. Bu dosyaları devreye almak ve devreden çıkarmak için

chkconfig

komutunu kullanmalısınız.

Örneğin, açılırken artık sisteminizde web sunucusu yazılımı olan httpd'nin başlatılmasını istemiyorsanız

chkconfig --del httpd veya **chkconfig httpd off**

komutunu verebilirsiniz. Apache web sunucusunun sisteminiz açılırken tekrar otomatik olarak çalıştırılmasını istediğinizde

chkconfig --add httpd veya **chkconfig httpd on**

komutu iş görecektir. Bu kitabın düzeyi açısından **chkconfig** yazılımının daha fazla ayrıntısına girmeyeceğiz, ama meraklı okuyucuların komutun man sayfalarına bir göz atmasını öneririz.

/etc/rc.d/rcn.d dizinlerindeki program dosyaları aslında birer dosya değil; **/etc/rc.d/init.d** dizinindeki program dosyalarına bağlantıdır; (link) **/etc/rc.d/init.d** dizinindeki dosyalar sistem yöneticilerinin oldukça sık olarak kullandıkları araçlardır. Örneğin, bir yazılım konfigürasyon değişikliği nedeniyle sisteminizin web sunucusu yazılımını durdurup yeniden başlatmanız gerekirse (sistemi kapatıp açmayı aklınıza dahi getirmeyin; o yöntem eski işletim sisteminizde kullandığınız bir yöntemdi) vermeniz gereken komut



/etc/rc.d/init.d/httpd restart

olacaktır. Bu linklerin isimlendirilme sistemi, çalışma düzeyi değişirken her bir kabuk programının nasıl bir parametreyle ve hangi sırada çalıştırılacağını gösterir. Adı "S" ile başlayanlar "start" parametresiyle; "K" ile başlayanlar "stop" parametresiyle (kill) çalıştırılır.

Aslında birçok yazılım, konfigürasyon değişikliklerinden sonra, durdurulup tekrar başlatılmak yerine bu değişikliğin kendilerine bildirilmesiyle yetinir. Örneğin, DNS sunucunuzda yeni bir sembolik isim tanıtımı yaptığınızda **named** isimli sunucu yazılımı durdurup başlatmak yerine, **ps** komutuyla **named** programının süreç numarasını öğrenip, bu sürece **HUP** (*hang-up*) mesajını gönderebilirsiniz. **named** yazılımı, **HUP** mesajı aldığı anda konfigürasyon dosyalarını yeniden okuyacak şekilde programlanmıştır.

```
ps ax | grep named
kill -HUP 893
```

Konu dağıldı gene ya; neyse...

Sistemin Kapatılması

Tüm UNIX bilgisayarlarda olduğu gibi LINUX işletim sisteminin de adabına uygun bir şekilde kapatılması gerekir. Her ne kadar disklerinizi **ext3**, **reiserfs** gibi dosya sistemleriyle düzenleyerek enerji kesintilerine karşı önlem almış olmanız da, sistemleri düzgün kapatmak her zaman için iyi bir alışkanlıktır.

Bir LINUX bilgisayarı kapatmanın en kolay yolu, root kullanıcının herhangi bir terminalden

shutdown -h now

komutunu vermesidir.

init 0

komutu da sistemin çalışma düzeyini sıfır yapmak; yani kapatmak için kullanılabilir.

KDE'nin grafik ekranını kullanarak sistemi kapatmayı zaten şimdiye kadar çoktan keşfetmiş olmalısınız.

Acemilik dönemlerinizde bilgisayarı kapatıp açmaya (bir diğer deyişle “reboot etmeye”) gereksinim duyabilirsiniz. Bu gibi durumlarda

reboot

komutunu da kullanabilirsiniz.



Deneyimli sistem yöneticileri sistemlerini reboot etmekten pek hoşlanmazlar. Bir şeyler aksamaya başladığı zaman bu aksaklığa hangi sürecin neden olduğunu bulup o süreci durdurmaya çalışırlar. Bu yöneticiler için sistemlerinin uzun süre kesintisiz çalışması (sistemin ne kadar zamandır “up” olduğu) bir övünç kaynağıdır. LINUX dünyasında “up” süreleri genellikle aylarla ölçülür. Windows sistem yöneticilerine biraz garip gelebilir ama işletim sistemi sürüm güncellemeleri ve donanıma müdahale dışında LINUX sistemlerini kapatmaya gerçekten pek fazla gerek olmaz.

Pek kolay kolay olmaz ama, sisteminize komut veremediğiniz bir duruma düşerseniz tek çözüm bilgisayarınızı anahtarından kapatıp açmak olacaktır elbette. “**x**” altında bir kilitlenme sorunu yaşarsanız Ctrl-Alt-F1 tuşlarıyla grafik olmayan bir konsola geçip sistemi toparlamanız genellikle mümkün olabilmektedir. Ctrl-Alt-F1 ile elde edeceğiniz konsolda root olarak sisteme girip adında X geçen tüm süreçleri öldürmek işe yarayabilir. Eğer bu da işe yaramazsa bu konsol ekranından “reboot” komutunu vererek sistemin düzgünce kapatılmasını sağlayabilirsiniz. Yeri gelmişken; LINUX işletim sistemi altında bu duruma genellikle donanım arızalarından dolayı düşersiniz. Eğer sık sık bilgisayarınızı kapatıp açmak zorunda kalıyorsanız bellek modüllerini, CPU soğutma fanını ve güç kaynağını gözden geçirmenizi öneririz.

Disklerin Yönetimi

Tüm bilgisayarların belki de en önemli kaynağı diskleridir. Daha doğrusu en kolay tükenen ve en kolay arızalanan; bu nedenle de en çok sorun çıkaran kaynak genellikle disklerdir.

Disklerinizi ve disk bölümlerinizi başarıyla yönetebilmeniz için kitabın önceki bölümlerinde anlatılmış olan “**disk bölümleme**” ve “**mount**” kavramlarını iyi anlamış olmanız gerekir.

İyi bir sistem yöneticisi her sabah disklerinin dolu/boş oranlarını şöyle bir gözden geçirip, gerekirse sabah temizliği yapmalıdır. Daha da iyi sistem yöneticileri bu işleri otomatik yapacak kabuk programları yazıp, **cron** ile her sabah çalışmasını sağlarlar.

Disklerin dolu-boş oranlarını ve durumlarını gözlemek için en uygun komut

```
df -h          veya
df -k
```

komutudur.

```
cayfer@cayfer.bilkent.edu.tr: /home/cayfer - Shell - Konsole

[cayfer@cayfer cayfer]$ df -h
Filesystem      Size  Used Avail Use% Mounted on
/dev/hda1        4.0G  2.7G  1.1G   72% /
/dev/hdc2        19G   17G  1.2G   94% /depo
/dev/hda8        5.2G  4.1G  1.1G   80% /hone
/dev/hda5        6.5G  3.4G  2.8G   55% /var
/dev/hda7        2.1G   70M  1.9G    4% /var/spool/mail
[cayfer@cayfer cayfer]$

[cayfer@cayfer cayfer]$ df -k
Filesystem      1k-blocks    Used Available Use% Mounted on
/dev/hda1       4122352    2789184    1123760   72% /
/dev/hdc2       19370244   17186600   1199664   94% /depo
/dev/hda8       5350200    4265552    1084648   80% /hone
/dev/hda5       6734212    3506432    2885692   55% /var
/dev/hda7       2105808      70944    1927892    4% /var/spool/mail
[cayfer@cayfer cayfer]$
```

Göreceli olarak hızlı büyüyen sistem dosyaları genellikle **/var/log** altındaki log dosyalarıdır. **/tmp** dizini de, herkesin yazma hakkı olmasından dolayı disklerin kolayca dolmasına neden olan bir dizindir. Bu dizinleri zaman zaman kontrol edip, eski ve büyük dosyaları silmelisiniz.

/var/log dizinininde sistemde olup biten herşeyin kaydedildiği dosyalar yer alır; bu nedenle bu dizinin diskte kapladığı alan sürekli artar. Her ne kadar **logrotate** süreci bu dosyaları dönüşümlü olarak değiştirip, eskileri kaldırıp atsa da, bu dizin hiç değilse haftada bir gözden geçirilmelidir.

Bir dizinin diskte ne kadar yer harcadığını merak ettiğinizde

du -s /home/cayfer

gibi bir komutla yanıt alabilirsiniz.

```

cayfer@cayfer.bilkent.edu.tr: /home/cayfer - Shell - Konsol
[cayfer@cayfer cayfer]$ du -s /home/cayfer
3751652 /home/cayfer
[cayfer@cayfer cayfer]$ du /home/cayfer
4      /home/cayfer/tmp
16     /home/cayfer/bcc
7668  /home/cayfer/Desktop/Trash
16     /home/cayfer/Desktop/Removable media
7764  /home/cayfer/Desktop
76     /home/cayfer/GNUstep/Library/AfterStep/non-configurable
88     /home/cayfer/GNUstep/Library/AfterStep/desktop/thenes
92     /home/cayfer/GNUstep/Library/AfterStep/desktop
184    /home/cayfer/GNUstep/Library/AfterStep
4      /home/cayfer/GNUstep/Library/Icons
4      /home/cayfer/GNUstep/Library/WindowMaker/Styles
4      /home/cayfer/GNUstep/Library/WindowMaker/Themes
4      /home/cayfer/GNUstep/Library/WindowMaker/Backgrounds
4      /home/cayfer/GNUstep/Library/WindowMaker/IconSets
4      /home/cayfer/GNUstep/Library/WindowMaker/Pixmaps
4      /home/cayfer/GNUstep/Library/WindowMaker/Sounds
4      /home/cayfer/GNUstep/Library/WindowMaker/SoundSets
180    /home/cayfer/GNUstep/Library/WindowMaker

```

du komutu dizinde yer alan dosyaların toplam büyüklüğünü 1024 byte uzunluğunda “blok” cinsinden verir. Dikkat ederseniz **du** komutu, **-s** (summary) parametresiyle ilgilendiğiniz dizinin diskte işgal ettiği toplam kapasiteyi; **-s** parametresi olmadan kullanırsanız da alt dizinlerin toplam disk alanlarını rapor ediyor.

fdisk

Disklerin bölümlene tablolarıyla ilgili işlemler için kullanılır.

fdisk /dev/hda

gibi bir komutla birinci IDE kanalındaki ilk diskin bölümlene tablosu üzerinde çalışmaya başlayabilirsiniz. Bölümlene tablolarıyla oynamak tehlikelidir. Ne yaptığınızı bilmeden dolu diskler üzerinde bu komutu denememenizi öneririz.

fdisk programına verilebilecek komutları görmek için programı başlattıktan sonra “m” seçimini kullanabilirsiniz.

```
root@notebook.igjman.bilkent.edu.tr: /root - Shell - Konsole
[root@notebook root]# fdisk /dev/hda

The number of cylinders for this disk is set to 2432.
There is nothing wrong with that, but this is larger than 1024,
and could in certain setups cause problems with:
1) software that runs at boot time (e.g., old versions of LILO)
2) booting and partitioning software from other OSs
   (e.g., DOS FDISK, OS/2 FDISK)

Command (n for help): n
Command action
  a toggle a bootable flag
  b edit bsd disklabel
  c toggle the dos compatibility flag
  d delete a partition
  l list known partition types
  n print this menu
  n add a new partition
  o create a new empty DOS partition table
  p print the partition table
  q quit without saving changes
  s create a new empty Sun disklabel
  t change a partition's system id
  u change display/entry units
  v verify the partition table
  w write table to disk and exit
  x extra functionality (experts only)

Command (n for help): █
```

mkfs

fdisk ile bölümlenmesi yapılmış disklerde disk bölümleri üzerine dosya sistemi yaratmak için kullanılır. Örneğin:

```
mkfs -t reiserfs /dev/hda3
```

komutuyla ilk diskin üçüncü bölümü üzerine boş bir **reiserfs** dosya sistemi yaratılabilir.

Aynı iş:

```
mkfs.reiserfs /dev/hda3
```

komutuyla da yapılabilir. **mkfs** komutunun **-t** ile belirtilen parametreleri arasında **ext2**, **msdos** gibi seçenekler de bulunmaktadır. **-t** ile yaratılacak dosya sisteminin tipini belirtmezseniz **ext2** kabul edilir.

mount

mount komutuyla sisteminizde disk bölümlerindeki dosya sistemlerinin bağlanmış olduğu dizinleri görebilir; bunları çözebilir, bağlı olmayanları bağlayabilirsiniz. Daha önce ayrıntılı olarak açıkladığımız **mount** komutunu

burada yalnızca disk yönetimi ile yakından ilgili olması dolayısıyla tekrar an-
dık.

showmount -a

Bu komutla, bilgisayarınızda NFS üzerinden paylaşım açmış disklerin kim-
ler tarafından kullanılmakta olduğunu görebilirsiniz.

```

root@notebook.lojman.bilkent.edu.tr: /root - Shell - Konsole
[root@lists /root]# showmount -a
All mount points on lists:
cayfer:/bcc
cayfer:/home/cayfer
cayfer:/home/httpd
loj08031.lojman.bilkent.edu.tr:/disk2/mdk9.0
[root@lists /root]#

```

Yukardaki örnek listeye göre lists makinesi üzerinde paylaşım açılmış olan
“/bcc, /home/httpd, home/cayfer” dizinleri **cayfer** makinesi tarafından;
/disk2/mdk9.0 dizini de **loj08031** makinesi tarafından kullanılmaktadır.

NFS paylaşımı bir UNIX bilgisayar üzerindeki dizinlerin başka UNIX bilgi-
sayarlar tarafından “mount edilebilmesini” sağlayan servistir. Bu servisin de-
netimi, yani hangi dizinlerin hangi makineler için, bunlardaki hangi kullanıcı-
lar için paylaşım açılacağını denetimi **/etc/exports** dosyası ile yapılır.

```

root@notebook.lojman.bilkent.edu.tr: /root - Shell - Konsole
[root@lists /root]# cat /etc/exports
/projeler      cayfer(rw) 192.168.0.3(rw,no_root_squash)
/bcc           *.bcc.bilkent.edu.tr(rw)
/usr           *.bilkent.edu.tr(ro)
/pub           (ro,all_squash)
/pub/ozel      (noaccess)
[root@lists /root]#

```

Yukardaki örnek **exports** dosyasında,

- **/projeler** dizinine **cayfer** isimli makineden gelen bağlantı isteklerine (mount isteklerine) okuma yazma ve tam yetkiyle erişim hakkı verilmiş. **cayfer** isimli makinenin IP adresi öncelikle **/etc/hosts** dosyasındaki lis-
teden; orada yoksa DNS servisi üzerinden araştırılır.

/projeler dizinine, **cayfer** makinesinden başka bir de 192.168.0.3 IP adresli makineden oku-yaz olarak erişilmesine izin verilmiş ve bu erişimin root yetkisi ile yapabileceğini belirtilmiş. (**no_root_squash** parametresi.)

- **/bcc** dizinine **bcc.bilkent.edu.tr** ağındaki tüm makinelerden oku-yaz olarak erişilmesine izin verilmiş.
- **/usr** dizinine **bilkent.edu.tr** ağındaki tüm makinelerden yalnızca okumak için erişilmesine izin verilmiş. (*ro*)
- **/pub** dizinine her yerden yalnızca okumak için izin verilmiş, ancak erişen herkesin kullanıcı kodunun “**nobody**” kabul edilmesi ve paylaşılan **/pub** dizinindeki dosya erişim yetkilerinin bu kullanıcı koduna göre düzenlenmesi için gereken ayar yapılmış.
- **/pub** dizinine herkese erişim hakkı verilmiş olmasına rağmen bu dizinin altındaki **/pub/ozel** dizinine hiçbir şekilde dışardan erişilememesi sağlanmış.

/etc/exports dosyasında bir değişiklik yapıldığında; örneğin **/home/cayfer/public_html** gibi bir dizini paylaşım açmak için gereken satır eklendiğinde

```
exportfs /home/cayfer/public_html
```

komutuyla paylaşımı başlatabilirsiniz.

Paylaşılmakta olan bir dizini dışarıdan erişime kapatmak istediğinizde

```
exportfs -u /pub
```

gibi bir komut kullanabilirsiniz.

```
exportfs -ua
```

tüm paylaşımları kapatır.

```
exportfs -a
```

/etc/exports dosyasında adı geçen tüm paylaşımları açar.

/etc/rc.d/init.d/nfs kabuk programı NFS servislerini başlatıp durdurmak için kullanılabilir.

```
/etc/rc.d/init.d/nfs start
/etc/rc.d/init.d/nfs stop
```

gibi...

Eğer LINUX makinenizdeki bir dizini bir Windows makinenin erişimine açmak istiyorsanız NFS servisi işinizi göremeyeceği için; daha doğrusu Windows işletim sistemi NFS servislerinden nasıl yararlanılacağını bilmediği için; **samba** servisini kullanmalısınız. **samba** servisi bir LINUX bilgisayarın NT sınıfı bir sunucu gibi çalışmasını sağlar. Bir başka deyişle, Windows makineler yakınlarında gördükleri bir **samba** servisini NT sunucu zannederler. Çok ilginçtir ki, üzerinde **samba** çalışan bir LINUX makine, aynı donanım üzerinde çalışan bir NT sunucudan performans açısından daha başarılıdır. Gene politikaya girdik... Burada keselim, yoksa kalp kıracağız.



samba'nın kurulumu ve daha da önemlisi ayarlarının yapılması bu kitabın amaçları dışında kaldığı için ayrıntıya girmeyeceğiz. Aslında, Mandrake LINUX'unuzu kurarken siz aksini seçmediyseniz **samba** makinenizde kurulmuş ve çalışıyor olacaktır. **samba** kullanacaksanız, yapmanız gereken **/etc/samba/smb.conf** dosyasındaki parametreleri gereksinimlerinize göre değiştirip **samba**'yı



```
/etc/rc.d/init.d/smb restart
```

komutuyla yeni konfigürasyonla tekrar başlatmak olacaktır.

samba konfigürasyonu ile ilgili ayrıntıları

<http://www.belgeler.org>

adresindeki "Samba – Nasıl" dokümanında bulabilirsiniz.

ls of

Bilgisayarınızda bir şekilde kullanılmakta olan açık dosyaları ve soketleri listelemek için kullanılır. (Soket, TCP/IP programlamayla ilgili bir kavramdır. Ne anlama geldiğini bilmiyorsanız üzerinde durmayınız.) Çeşitli internet hizmetleri veren tipik bir LINUX bilgisayarında bu listede 10.000'den fazla dosya yer alabilir; dolayısıyla listeyi gözle taramak pek anlamlı değildir. **ls of** genellikle çıktısı **grep** ve **more** ile filtrelenerek kullanılarak belirli bir kulla-

nıcının açmış olduğu dosyaları ya da belirli bir programın kullandığı dosyaları gözlemek için kullanılır.

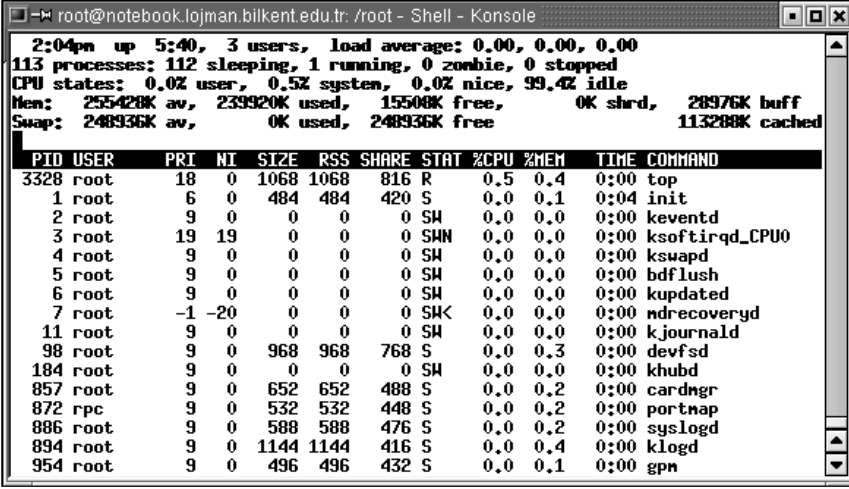
ls -l | grep cayfer | more

gibi.

```
root@notebook.lojman.bilkent.edu.tr: /root - Shell - Konsole
[root@notebook root]# ls -l | grep cayfer | more
bash      2112  cayfer  cwd    DIR      3,6    8192    309463 /home/cayfer
bash      2112  cayfer  rtd    DIR      3,6    4096     2 /
bash      2112  cayfer  txt    REG      3,6   675276  745307 /bin/bash
bash      2112  cayfer  nen    REG      3,6   539887  698241 /lib/ld-2.2.5
.so
bash      2112  cayfer  nen    REG      3,6    6876   923086 /usr/lib/gcon
w/ISO8859-9.so
bash      2112  cayfer  nen    REG      3,6   22523  226702 /usr/share/lo
cale/tr/LC_COLLATE
bash      2112  cayfer  nen    REG      3,6     59   745233 /usr/share/lo
cale/tr_TR/LC_NUMERIC
bash      2112  cayfer  nen    REG      3,6    371   601093 /usr/share/lo
cale/en_US/LC_IDENTIFICATION
bash      2112  cayfer  nen    REG      3,6     29   601092 /usr/share/lo
cale/en_US/LC_MEASUREMENT
bash      2112  cayfer  nen    REG      3,6     65   601088 /usr/share/lo
cale/en_US/LC_TELEPHONE
bash      2112  cayfer  nen    REG      3,6    161   601094 /usr/share/lo
cale/en_US/LC_ADDRESS
bash      2112  cayfer  nen    REG      3,6     83   601090 /usr/share/lo
cale/en_US/LC_NAME
--More--
```

Bellek Yönetimi

LINUX işletim sisteminde bellek yönetimiyle ilgili pek fazla işiniz olmaya-
caktır. Eksik bellekle çalışıyorsanız, elbette ki performans sorunlarınız ola-
caktır. LINUX genellikle bellek eksikliği hakkında doğrudan şikayet etmez.
Eğer **takas alanı** (*swap partition*) olarak ayırdığınız disk bölümü yetmiyorsa
bellek yetersizliği ile ilgili mesaj alabilirsiniz. Takas alanı, gerçek belleğin yet-
meme durumunda işletim sisteminin diskten yararlanması için kullanılır.
Çalışan bir sürece bellek tahsis etmek gerektiğinde, ana bellekte yer kalma-
dıysa, beklemede olan süreçler kaldıkları yeri ve durumu işaretleyen bilgiler-
le birlikte diske atılır, böylece kazanılan bellek gereksinim duyan sürece tah-
sis edilebilir. Bu yöntem doğal olarak programların çalışmasını çok ciddi şe-
kilde yavaşlatır. Takas alanı kullanmanın mantığı, bellek yetersizliği yüzün-
den programların kesilmesini önlemektir; yoksa kesin bir çözüm değildir.
Takas alanınızın kullanımını sık sık gözleyip, aşırı kullanılmaya başlanırsa
sisteminizin belleğini arttırmalısınız. Takas alanı kullanımıyla ve süreçlerin
bellek kullanımıyla ilgili bilgileri **top** komutuyla alabilirsiniz.



2:04pm up 5:40, 3 users, load average: 0.00, 0.00, 0.00
 113 processes: 112 sleeping, 1 running, 0 zombie, 0 stopped
 CPU states: 0.0% user, 0.5% system, 0.0% nice, 99.4% idle
 Mem: 255428K av, 239320K used, 15500K free, 0K shrd, 28976K buff
 Swap: 248936K av, 0K used, 248936K free, 113288K cached

PID	USER	PRI	NI	SIZE	RSS	SHARE	STAT	%CPU	%MEM	TIME	COMMAND
3328	root	18	0	1068	1068	816	R	0.5	0.4	0:00	top
1	root	6	0	484	484	420	S	0.0	0.1	0:04	init
2	root	9	0	0	0	0	SW	0.0	0.0	0:00	keventd
3	root	19	19	0	0	0	SWN	0.0	0.0	0:00	ksoftirqd_CPU0
4	root	9	0	0	0	0	SW	0.0	0.0	0:00	kswapd
5	root	9	0	0	0	0	SW	0.0	0.0	0:00	bdflush
6	root	9	0	0	0	0	SW	0.0	0.0	0:00	kupdated
7	root	-1	-20	0	0	0	SW<	0.0	0.0	0:00	ndrecoveryd
11	root	9	0	0	0	0	SW	0.0	0.0	0:00	kjournald
98	root	9	0	968	968	768	S	0.0	0.3	0:00	devfsd
184	root	9	0	0	0	0	SW	0.0	0.0	0:00	khud
857	root	9	0	652	652	488	S	0.0	0.2	0:00	cardmgr
872	rpc	9	0	532	532	448	S	0.0	0.2	0:00	portnap
886	root	9	0	588	588	476	S	0.0	0.2	0:00	syslogd
894	root	9	0	1144	1144	416	S	0.0	0.4	0:00	klogd
954	root	9	0	496	496	432	S	0.0	0.1	0:00	gpm

Eğer takas alanı hiç kullanılmıyorsa bilgisayarınızda gereksiniminizden daha fazla bellek var demektir. Zıyan etmemek için birazını o makineden söküp bir başkasına takabilirsiniz.

Takas alanı az geliyorsa ilk akla gelen diskin takas alanını büyütmektir ancak bu disk bölümlenmesinin değiştirilmesini gerektirir; bu da doğal olarak önce yedekleme, sonra diski yeniden düzenleyip (formatlayıp), dosya sistemlerini yeniden oluşturma ve yedekleri geri yükleme demektir ki, bu işlemler çalışan bir sistem için uzun süreli bir kesinti demektir.

Takas alanınız yetersiz kaldığında, takas alanı olarak ayrılan disk bölümünü yeniden oluşturmak yerine takas alanına ek yapmayı düşünmelisiniz. Bu yöntem, tek bir takas alanı kullanmak kadar yüksek performans sağlamasa da sisteminizi yeniden kurmayı göze alacağınız zamana kadar idare edecektir.

Bunun için önce disklerinizden birinde uygun boyda bir takas dosyası yaratın:

```
dd if=/dev/zero of=/tmp/ek_takas bs=1024 count=100000
```

komutu, /tmp altında yaklaşık 100 Mbyte uzunluğunda **ek_takas** isimli boş (daha doğrusu içi sıfırlarla dolu) bir dosya yaratacaktır. (Dosyanın blok uzunluğu 1024 byte olduğu için 100.000 blok aşağı yukarı 100 Mbyte eder.)



dd (device-to-device copy) komutu zaman zaman çok işe yarar. “**if=**” parametresi (input file) kopyalamanın nereden yapılacağını; “**of=**” parametresi de (output file) kopyalamanın nereye yapılacağını belirtir.

/dev/zero, aynı **/dev/null** gibi LINUX'un özel bir “çevre birimi” veya dosyasıdır. Yalnızca okunabilir. Boyu sonsuzdur; yani bu dosyadan sonsuza kadar veri okuyabilirsiniz. Ancak, okuduğunuz tüm veriler 0x00, yani ikil (*binary*) sıfırlardan oluşur.

Yukarıdaki takas dosyası yaratma komutu, **/dev/zero** dosyasından herbiri 1024 byte uzunluğunda 100000 blok okuyup bunları **/tmp/ek_takas** dosyasına kopyalıyor. Aslında, içi tamamen ikil sıfırlarla dolu 100 MByte'lık bir dosya yaratıyor.

Sonra bu dosyanın sahibini root kullanıcı yapın ve erişim yetkilerini, bu dosyaya root dışında kimsenin erişemeyeceği şekilde değiştirin:

```
chown root:root /tmp/ek_takas  
chmod 600 /tmp/ek_takas
```

Daha sonra **/etc/fstab** dosyası içine

```
/tmp/ek_takas swap    swap    defaults    0        0
```

satırını ekleyin.

Son olarak da yeni takas alanını devreye sokun:

```
swapon /tmp/ek_takas
```

Takas alanı değişikliklerinde bile sistemi “reboot” etmenize gerek olmadığı dikkatinizi çekti mi?

/etc/fstab'a eklediğiniz satır sayesinde sistemi her açışınızda bu takas alanı devreye girecektir.

Ek takas alanı olarak ayırdığınız disk alanına ihtiyacınız olursa

```
swapoff /tmp/ek_takas
```

gibi bir komutla, bu dosyanın takas olanı olarak kullanımına son verebilir ve dosyayı silip **/tmp** altında yer açabilirsiniz. **/etc/fstab** dosyasından ilgili satırı çıkarmayı da unutmayın ki bir dahaki sistem açılışında sorun çıkmasın.

Kullanıcı Yönetimi

Kullanıcı Hesapları

Kullanıcı hesaplarının açılıp kapatılmasının yanısıra, kullanıcıların kullanabilecekleri sistem kaynaklarını belirleme ve gerekirse sınırlama işleri de sistem yöneticisinin önemli görevlerindendir.

LINUX'ta kullanıcı hesabı açmanın pek çok yolu vardır; çünkü “hesap açma” temelde **/etc/passwd** ve **/etc/shadow** dosyalarına birer satır eklemekten oluşur.

```

root@cayfer.ilkent.edu.tr: /etc - Shell - Konsole

[root@caufer etc]# cat /etc/passwd
root:x:0:0:root:/root:/bin/bash
daemon:x:2:2:daemon:/sbin:/bin/sh
apache:x:72:72:system user for apache:/var/www:/bin/sh
mysql:x:78:78:MySQL server:/var/lib/mysql:/bin/bash
cayfer:x:501:501:Can Ugur Ayfer:/home/cayfer:/bin/bash
oner:x:502:502:Omer Ayfer:/home/oner:/bin/bash
sener:x:506:506:Metin Sener,php mysql:/home/sener:/bin/bash
paul:x:507:507:Paul Martin:/home/paul:/bin/bash
tulin:x:508:508:Tulin Tenizel:/home/tulin:/bin/bash
[root@cayfer etc]#

[root@cayfer etc]# cat /etc/shadow
root:$1$pXEMq6$3.z4q0.NjvDE.jjK89A:12055:0:99999:7:::
daemon:!:12055:0:99999:7:::
apache:!:12055:0:99999:7:::
mysql:!:12055:0:99999:7:::
cayfer:$1$9yQ9F/C.j$zpNbBBJMCALN1351:11875:0:99999:7:::1073862910
oner:$1$pk6yJ4UT$ps5CkuAbziF430/zT:11454:0:99999:7:::
sener:$1$r31z/vax$vykkB.BT6KttD.jn1hi:11972:0:99999:7:::1073862910
paul:$1$zMOugxSL$1B0uN439MeJ0YD0.isQ:11505:0:99999:7:::-1073744384
tulin:$1$6XMeJtQ/$RnDot0/nb7juvXfS13tZH0:11519:0:99999:7:::-1073744432
[root@cayfer etc]#

```

/etc/passwd dosyası, adının aksine şifrelerle ilgili hiçbir bilgi içermez. Eskiden, tüm UNIX'lerde **/etc/passwd** dosyası kullanıcıların şifre dahil her türlü bilgisinin tutulduğu dosya idi. Ancak, internet yaygınlaştıkça güvenlik sorunları da artmaya başladı. Kullanım mantığı gereği **/etc/passwd** dosyasının herkesin okuyabileceği bir dosya olması, kötü niyetli kişilerin kriptolanmış da olsa kullanıcı şifrelerini alıp başka bilgisayarlarda deneme yanılma yoluyla kırma çabalarına yol açtı. Bunun üzerine **/etc/passwd** dosyasının şifre hariç tüm özellikleri aynen korunacak şekilde şifrelerin başka dosyaya, yani **/etc/shadow** isimli bir dosyaya taşınması kararlaştırıldı ve root kulla-

nıcı dışında herkesin bu dosya üzerindeki tüm yetkileri kaldırıldı. Artık modern UNIX uyarlamalarının hemen hemen hepsi kullanıcı şifrelerini **/etc/shadow** dosyasında saklamaktadır. Yeri gelmişken; UNIX dünyasında kullanıcı hesaplarını ve şifrelerini saklamanın tek yolu **/etc/passwd** ve **/etc/shadow** dosyaları değildir. Bu iki dosya, en yaygın olarak kullanılan yöntemlerdir.

Bir hesap açmak ya da kapatmak için bu dosyaları **vi** ile düzenleyebileceğiniz gibi bu işi sizin yerinize yapacak özel araçlar da kullanabilirsiniz. KDE ortamında kullanıcı hesaplarının yönetimi için **userdrake** kullanabilirsiniz. Yeni kullanıcı tanımlarken terminal penceresinden komut vermeyi yeğlerseniz

```
adduser -c "Can Ugur Ayfer" -d /home/cayfer cayfer
```

komutundan yararlanabilirsiniz. Bu komutun aslında daha birçok parametresi vardır; ancak günlük kullanımda daha fazlası pek gerekmiyor. Komutun ayrıntılarını öğrenmek isterseniz komutu parametresiz olarak verin. Kendi kullanım kılavuzunu kendisi görüntüleyecektir.

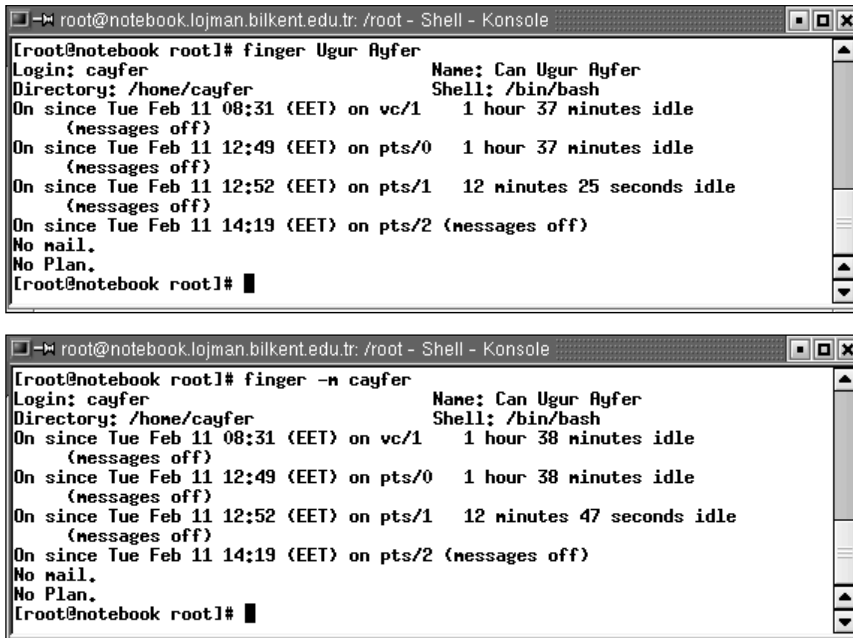
Hatırlarsanız kitabın başlarında UNIX'te kullanıcı kodlarının çok da anlamlı olmadığını, esas kimlik belirleyicisinin sayısal kullanıcı numarası olduğunu belirtmiştik. İşte bu konu şimdi gene gündeme geldi. **adduser** komutu, yeni tanıtılan kullanıcılara numara verme işini kendisi halleder; bunu yaparken de en son verilmiş numaranın bir fazlasını seçer. Bir nedenle kullanıcılarınıza kendiniz numara vererek hesap açmak isterseniz, **adduser** komutunun **-u** parametresinden yararlanabilirsiniz.

Bazen kodunu bildiğiniz ama açık adını hatırlayamadığınız; ya da tam tersine, açık adının bir kısmını da olsa bildiğiniz ama kodunu hatırlayamadığınız kullanıcılar olacaktır. Bu gibi durumlarda

finger

komutu çok işinize yarayacaktır.

Komutu "**finger ayfer**" şeklinde verirseniz, kullanıcı açık adında "**ayfer**" geçen kullanıcıların; "**finger -m ayfer**" şeklinde verirseniz de, kullanıcı kodu "**ayfer**" olan kullanıcının ayrıntılı bilgilerine erişirsiniz.



```

root@notebook.lojman.bilkent.edu.tr: /root - Shell - Konsol

[root@notebook root]# finger Ugur Ayfer
Login: cayfer                               Name: Can Ugur Ayfer
Directory: /home/cayfer                     Shell: /bin/bash
On since Tue Feb 11 08:31 (EET) on vc/1      1 hour 37 minutes idle
(messages off)
On since Tue Feb 11 12:49 (EET) on pts/0      1 hour 37 minutes idle
(messages off)
On since Tue Feb 11 12:52 (EET) on pts/1      12 minutes 25 seconds idle
(messages off)
On since Tue Feb 11 14:19 (EET) on pts/2 (messages off)
No mail.
No Plan.
[root@notebook root]#

root@notebook.lojman.bilkent.edu.tr: /root - Shell - Konsol

[root@notebook root]# finger -n cayfer
Login: cayfer                               Name: Can Ugur Ayfer
Directory: /home/cayfer                     Shell: /bin/bash
On since Tue Feb 11 08:31 (EET) on vc/1      1 hour 38 minutes idle
(messages off)
On since Tue Feb 11 12:49 (EET) on pts/0      1 hour 38 minutes idle
(messages off)
On since Tue Feb 11 12:52 (EET) on pts/1      12 minutes 47 seconds idle
(messages off)
On since Tue Feb 11 14:19 (EET) on pts/2 (messages off)
No mail.
No Plan.
[root@notebook root]#

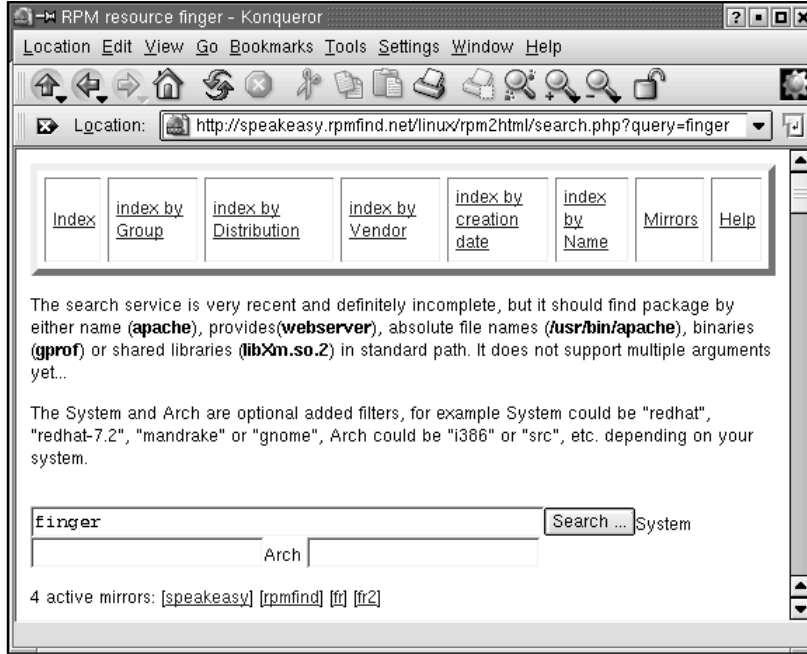
```

Kurulum sırasında **finger** programının kurulmasını özellikle istemediyse-
niz yukarıdaki **finger** komutu örneklerini denemek istediğinizde “**fin-
ger: command not found**” mesajını alacaksınız.

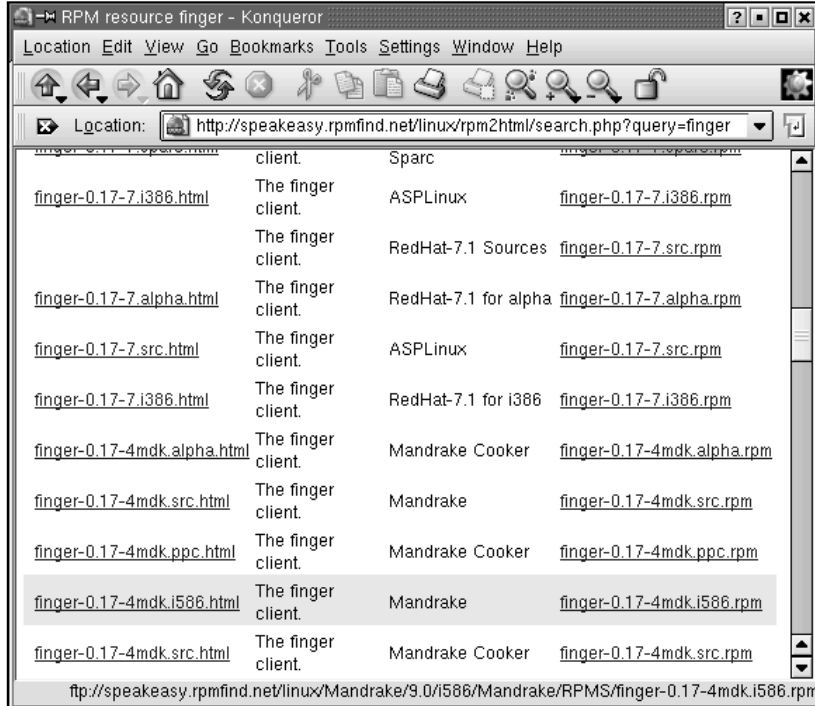
İsterseniz, yeri gelmişken **finger** programını sisteminize birlikte kuralım:

Bir LINUX programını kurmanın en kolay yolu o programın rpm paketini
(RedHat Packet Manager sözcüklerinin baş harfleri) bulup, paketi **rpm** ko-
mutuyla kurmaktır. Kurulum sırasında kurulmamış rpm paketlerini Mand-
rake dağıtım CD’lerinden birinde bulabilirsiniz; ama üç tane CD olduğunu
düşünürseniz bunlar arasında aramaktansa İnternet’te aramak daha kolay
olacaktır.

LINUX paketlerinin güncel sürümlerini indirmek için en uygun servisler-
den biri www.rpmfind.net sitesidir.



Anahtar sözcük kutusuna “**finger**” sözcüğünü yazıp aramayı başlatın.



Bulunan **rpm** paketleri arasında Mandrake dağıtımı için hazırlanmış olanlardan güncel bir sürüme ilişkin paketi seçin. Eğer varsa, adında i586, i686 geçen paketlerden birini seçin.

Adında “**src**” geçen paketler, programların kaynak kodlarını içeren paketlerdir. Kurulum için bu paketi kurduktan sonra modülleri derlemek gerekecektir. Boşyere derlemekle uğraşmamak için adında “**noarch**” (mimariden bağımsız) geçen hazır derlenmiş paketleri kurmak çok daha kolay olabilir.

Öte yandan, kendi sisteminize uyarlanması yerinde olan programlar (örneğin ekran kartınıza ve cpu cinsine göre daha iyi çalışacak şekilde derlenebilecek mplayer video oynatıcısı) kaynak kodlarından derlendiğinde daha başarılı olabilir.

finger komutunun **rpm** paketini, örneğin **/tmp** dizinine, indirdikten sonra

```
rpm -i /tmp/finger-0.17-4mdk.i586.rpm
```

komutuyla kurabilirsiniz.



Neyse, konumuza dönelim...

Bir kullanıcının hesabını kapatmak istediğinizde

```
userdel [-r] kullanıcı
```

komutunu kullanabilirsiniz. “**userdel kullanıcı**” komutu **/etc/passwd** ve **/etc/shadow** dosyalarından kullanıcı ile ilgili satırları siler. Kullanıcının (varsa) kişisel dizinini de silmek isterseniz **-r** parametresini kullanmalısınız.

Açılmış bir kullanıcı hesabıyla ilgili bilgileri değiştirmek istediğinizde

```
usermod parametre[ler] kullanıcı_adi
```

komutunu

Kim Korkar LINUX'tan?

```
[-c açık_adı]  
[-d kişisel_dizin]  
[-s kabuk]  
[-p şifre]  
[-u kullanıcı_sayısal_kodu]  
[-G grup1[,grup2[, ...]]]
```

gibi parametrelerle kullanabilirsiniz. Örneğin açık adı yanlışlıkla “Can Ugur Ayfer” olarak girilmiş olan **cayfer** kullanıcısının açık adını düzeltmek için

usermod -c “Can Ugur Ayfer” cayfer

komutunu verebilirsiniz. Bu arada tekrar hatırlatmadan geçemeyeceğiz: Aynı işi **/etc/passwd** dosyasını **vi** ya da bir başka editör ile düzenleyerek de yapabildiniz.

usermod komutunun oldukça kullanışlı üç parametresi daha vardır. Örneğin,

usermod -e 2002-12-31 cayfer

komutu, **cayfer** isimli kullanıcının şifresini 31 Aralık 2002 tarihinde geçersiz kılarak bu kullanıcının sisteme erişim hakkına son verecektir. Hesap kapanmayacak; ancak şifre bilinmeyen bir değere değiştirildiği için sistem yöneticisi yeni bir şifre verene kadar bu kullanıcı sisteme bağlanamayacaktır. Kullanıcılar şifrelerinin geçersiz kılınacağı tarih yaklaşıncaya kadar bu kullanıcıları uyarmaya başlar; ancak kullanıcıların bu uyarıyı görebilmeleri için sisteme “**login**” olmaları gerekir. Kabuk kullanmayan kullanıcılar (e-postalarını POP servisiyle uzaktan okuyanlar, yani sistemi doğrudan kullanmayanlar) doğal olarak bu uyarı mesajını göremeyecektir.

Bazen sistem yöneticileri bir kullanıcının hesabını geçici olarak erişilmez hale getirmek isterler. Örneğin bir hesabın bir başkası tarafından kullanıldığından şüphelendiğinizde ya da sisteminizin kullanım kurallarına uymayan kullanıcıların sisteme erişim yetkilerini geçici olarak kaldırmak istediğinizde

usermod -L cayfer

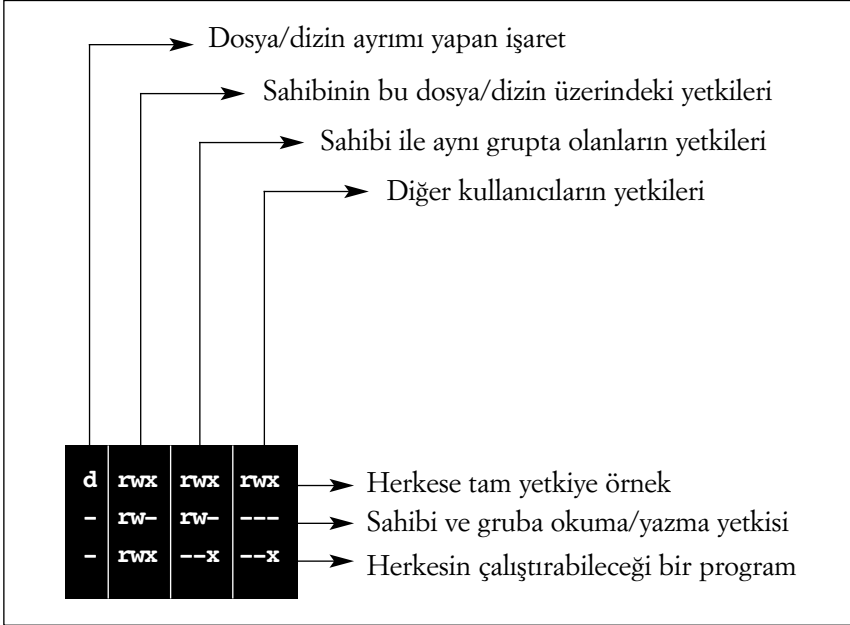
gibi bir komutla, **cayfer** isimli kullanıcının hesabını kilitleyebilirsiniz. Hesabı kilitlenmiş bir kullanıcının hesabını geri açmak için

usermod -U cayfer

komutu kullanılabilir.

Kullanıcı Grupları

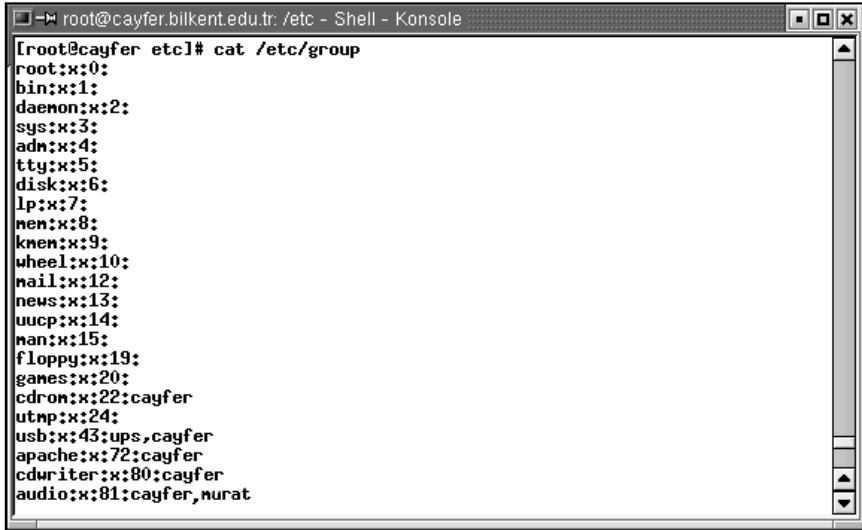
Dosya erişim yetkilerini şöyle bir gözünüzün önüne getirirseniz ortadaki üçlünün “dosyanın sahibiyle aynı gruptaki kullanıcıların” yetkileri olduğunu hatırlayacaksınız.



Kullanıcıları gruplayarak dosya ve dizinlere erişim yetkilerini düzenlemek oldukça kullanışlı bir yöntemdir. Örneğin, aynı yazılım projesi üzerinde çalışan tüm programcıları aynı gruba yerleştirip, bu projeyle ilgili izin ve dosyaların grup erişim haklarını istediğiniz gibi verebilirsiniz.

Bu arada herhangi bir kullanıcının birden fazla grupta yer alabileceğini de belirtmekte yarar var.

Sistemde tanımlı kullanıcı gruplarına ilişkin kayıtlar **/etc/group** dosyasında saklanır.



```
root@cayfer.bilkent.edu.tr: /etc - Shell - Konsole
[root@cayfer etc]# cat /etc/group
root:x:0:
bin:x:1:
daemon:x:2:
sys:x:3:
adm:x:4:
tty:x:5:
disk:x:6:
lp:x:7:
men:x:8:
kmem:x:9:
wheel:x:10:
mail:x:12:
news:x:13:
uucp:x:14:
nan:x:15:
floppy:x:19:
games:x:20:
cdrom:x:22:cayfer
utmp:x:24:
usb:x:43:ups,cayfer
apache:x:72:cayfer
cdwriter:x:80:cayfer
audio:x:81:cayfer,murat
```

Bu dosyadaki grup tanım kayıtlarının genel formunda önce grubun adı, sonra varsa grubun şifresi, sonra grubun sayısal numarası (kullanıcı hesaplarında olduğu gibi aslında önemli olan grupların isimleri değil, sayısal kodlarıdır), sonra gereği kadar virgülle ayrılmış olarak bu grubun üyelerinin isimleri yer alır.

Sisteminize yeni bir grup eklemeniz gerektiğinde, uygun bir grup tanım kaydını **/etc/group** dosyasına herhangi bir editörle ekleyebileceğiniz gibi

groupadd [-g grup_no] yeni_grup

komutunu da kullanabilirsiniz.

Benzeri şekilde grup iptal etmek istediğinizde

groupdel grup_adi

komutunu kullanabilir veya **/etc/group** dosyasını edit edebilirsiniz.

Bir kullanıcınızın hangi gruplara dahil olduğunu görmek isterseniz

groups kullanıcı

komutunu kullanabilirsiniz.

Gruplarınızdan birine yeni bir kullanıcı eklemek istediğinizde en kolay **/etc/group** dosyasını düzenleyerek ilgili grubun tanıtım kaydının sonuna bu kullanıcının adını eklemektir.

Örneğin **cayfer** isimli kullanıcıyı **apache** grubuna eklemek için **/etc/group** dosyasındaki **apache** grubuna ilişkin satırı

apache:x:48:webmaster,cayfer

olacak şekilde değiştirmeniz yeterlidir.

Kullanıcı gruplarıyla ilgili olarak sözünü etmek istediğimiz önemli bir komut daha var:

chgrp grup dosya

chgrp grup dizin

chgrp -R grup dizin

chgrp komutu bir dosyanın, bir dizinin veya altındaki herşeyle birlikte bir dizinin ait olduğu grubu değiştirmek için kullanılır. Çok büyük ölçüde **chown** komutuna benzer.

Bir kullanıcı sisteme bağlandığında kendisi için **/etc/passwd** dosyasında belirtilmiş olan grup kimliğine sahip olur. Oysa bir kullanıcının birden fazla grubun üyesi olabileceğini söylemiştik. Bir nedenle (ki bu genellikle erişim yetkileriyle ilgili bir neden olur) kullanıcı kimliğinizi değiştirmeden grup kimliğinizi değiştirmek isteyebilirsiniz. O zaman

newgrp grup_adı

komutunu kullanmalısınız. Eğer bu yeni grup şifre gerektiren bir grupsa, sizden grup şifresini girmeniz istenecektir.

Kullanıcıların bir grup kimliğine bürünmek için **newgrp** komutu kullandıklarında şifre sorulmasını istediğinizde

gpasswd grup

komutuyla o gruba bir şifre vermeniz gerekir.

Kullanıcı Disk Kotaları

Ne kadar hızla ucuzlasa da bilgisayar sisteminizin en değerli kaynağı disk kapasitesidir. Disk kapasiteleri 5 MegaByte iken de böyleydi, 100 TeraByte olduğunda da böyle olacak. Tipik kullanıcı davranışlarından birisi de hiçbir dosyayı silmemektir. Sahipleri “*Sakla samanı gelir zamanı*” diye düşünüyor olsalar gerek, MP3 ve DIVX dosyalar kişisel dizinlerinde döner dururlar.

Bu gibi durumlarda sistem yöneticilerinin en önemli silahlarından biri **disk kota** sistemidir.

Kullanıcıların kullanabilecekleri disk alanlarını kısıtlama işlemi her dosya sistemi için ayrı ayrı yapılır, ancak pratikte kullanıcı kişisel dizinlerinin yer aldığı **/home** dizininin yer aldığı dosya sisteminde kota uygulaması yapmak yeterlidir.

Kota uygulamaya başlamadan önce sisteminizin çekirdeğinde (*kernel*) “**kota desteği**” olduğuna emin olmalısınız. Bunun için

```
grep -i quota /var/log/dmesg
```

komutunu verdiğinizde

```
VFS: Diskquotas version dquot_6.5.0 initialized
```

gibi bir yanıt alırsanız, sisteminizin çekirdeğinde kota desteği var demektir. Eğer kota desteği olmayan bir LINUX çekirdeği kullanıyorsanız, kendinize disk kotası desteği olan bir çekirdek hazırlayıp, bu çekirdeği derleyip sistemin bu çekirdekle açılmasını sağlamalısınız. Çekirdek seçeneklerinde değişiklik yapıp yeniden derleme, bu kitabın kapsamı dışında kalan ileri düzey bir iştir. Çekirdek derlemeniz gerekirse başka kaynaklara başvurmanız gerekecektir.

Kota uygulamaya başlamak için **/etc/fstab** dosyasında, kota uygulanacak dosya sistemlerine ilişkin

```
/dev/hda3 /home ext2 defaults,usrquota 1 1
```

benzeri satırlar olacak şekilde bir düzenleme yapmalısınız. (**usrquota** parametresini eklemelisiniz.)

Bu değişikliğin ardından ilgili dosya sistemini yeniden mount etmeniz gerekir. Bunun en kolay yolu

mount -a

komutunu kullanmaktır. (Sisteme bağlı kullanıcılarınız varsa, büyük olasılıkla **/home** dizini altındaki kişisel dizinlerini kullanıyor olacaklarından bu komut **/home** dosya sistemini çözüp tekrar bağlamayacaktır. Bu nedenle **mount -a** komutundan önce sistemde sizden başka çalışan kimse olmamasına dikkat etmelisiniz.)

Daha sonra bu dosya sistemi için kota sistemini çalışır duruma getirmelisiniz:

quotaon /dev/hda3

Kota sistemini ilk kez çalışır duruma getirdiğinizde

quotacheck /dev/hda3

gibi bir komutla diskte kimin ne kadar yer harcadığının ve harcama hakkı olduğunun hesabının tutulduğu **aquota.user** dosyasının yaratılmasını sağlamalısınız. Eğer birden fazla disk bölümünde kota uygulayacaksanız, **quotacheck** komutunu her bir bölüm için ayrı ayrı vermelisiniz.

Şimdi sıra kotasını sınırlamak istediğiniz kullanıcılar için **/home** dizininde oluşturabilecekleri dosyaların toplam büyüklüğünü belirtmeye geldi. Bunun için **edquota** komutunu kullanmak gerekiyor:

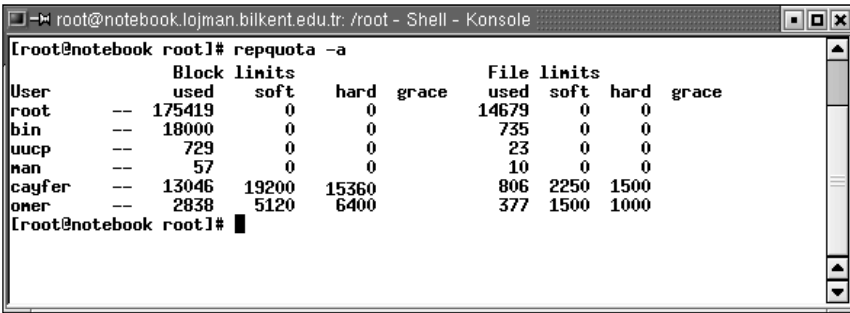
edquota cayfer

İşte **vi** kullanmanızı gerektiren bir noktaya geldiniz. Zamanında uyarmıştık! **vi** bilmeden olmaz diye...

/dev/hda6 diskinde o kullanıcıya ait dosyalarının kotası 200 MB olur; ancak geçici bir süre için (7 gün) toplamın 220 Mbyte'a kadar çıkmasına izin verilir. Bu sürenin dolmasından sonra, kullanıcının yeni dosya kaydetmesi ne izin verilmez.

Sistemde disk kotası verilmiş olan tüm kullanıcıları, kotalarını ve bu kotalarının ne kadarının kullanılmış olduğunu görmek için **repquota** komutunu kullanabilirsiniz.

repquota -a



User	Block limits				File limits			
	used	soft	hard	grace	used	soft	hard	grace
root	175419	0	0		14679	0	0	
bin	18000	0	0		735	0	0	
uucp	729	0	0		23	0	0	
man	57	0	0		10	0	0	
cayfer	13046	19200	15360		806	2250	1500	
oner	2838	5120	6400		377	1500	1000	

Log Dosyalarının Yönetimi

Log dosyalarının bir sistem yöneticisinin en önemli araçlarından olduğunu daha önce de belirtmiştik. Sisteme ne zaman kimin telnet ile bağlandığını, kimin FTP ile dosya çektiğini, kime kimden e-posta gittiğini, belirli bir saatte başlamak üzere programlanan işlerin başına neler geldiğini, sistemin açılması sırasında olup bitenleri, disk arızalarını, kimin hangi web sayfanıza baktığını ve bunun gibi birçok kayda değer olayı **/var/log** altındaki çeşitli dosyaları inceleyerek görebilirsiniz.

Log dosyaları genellikle ilgili oldukları işin adını içeren ya da anımsatan dosyalardır. Örneğin **/var/log/httpd/access.log** dosyası, web sitenizi ziyaret edenlerle ilgili bilgileri; **/var/log/messages**, sisteminizde oluşabilecek sorunlara; daha doğrusu çeşitli sistem yazılımlarının kayda değer bulduğu olaylara ilişkin kayıtları içerir.

Bazı log dosyalarını okuyabilmek için özel komutlar kullanmanız gerekebilir; örneğin sisteme telnet ile ya da konsoldan bağlanan (login eden) kullanıcıları görmek için **last** komutunu kullanmalısınız. Bu kayıtların tutulduğu

Kim Korkar LINUX'tan?

`/var/log/lastlog` dosyası basit bir metin dosyası olmadığı için **more** ya da **less** komutuyla bu log dosyasına göz atamazsınız.

```
cayfer@notebook.lojman.bilkent.edu.tr: /home/cayfer - Shell - Konsole
[cayfer@notebook cayfer]$ last | more
odavid      ftp      ppp109.bcc.bilke Sun Feb  9 12:16 - 12:21 (00:05)
murat       pts/5    oper1.bcc.bilken Sun Feb  9 12:07 - 12:08 (00:01)
alkin       pts/5    labb30706.bcc.bi Sun Feb  9 11:54 - 11:54 (00:00)
robin       ftp      ppp106.bcc.bilke Sun Feb  9 11:51 - 12:07 (00:15)
odavid      ftp      ppp109.bcc.bilke Sun Feb  9 11:37 - 11:51 (00:14)
karin       pts/4    dalyan.ee.bilken Sun Feb  9 11:10 - 12:20 (01:09)
akgul       pts/2    loj04051.lojman. Sun Feb  9 10:54 - 12:28 (01:34)
rabian      ftp      pc530b.cs.bilken Sun Feb  9 10:32 - 10:32 (00:00)
rabian      pts/2    pc530b.cs.bilken Sun Feb  9 10:30 - 10:32 (00:01)
alkin       pts/2    oper1.bcc.bilken Sun Feb  9 10:21 - 10:25 (00:03)
oner        pts/2    139.179.12.190   Sun Feb  9 09:57 - 09:57 (00:00)
stephen     ftp      sbz17-a.fen.bilk Sun Feb  9 09:52 - 09:53 (00:00)
ozdenir     pts/4    pclib31.lib.bilk Sun Feb  9 09:14 - 09:24 (00:09)
ozdenir     pts/2    pclib31.lib.bilk Sun Feb  9 09:11 - 09:21 (00:10)
menderes    pts/2    firat           Sun Feb  9 07:06 - 08:32 (01:25)
cahitakn    ftp      loj15601.lojman. Sun Feb  9 05:50 - 05:51 (00:00)
murata      ftp      d79011c.dorn.bil Sun Feb  9 00:14 - 00:14 (00:00)
menderes    pts/4    oper1.bcc.bilken Sat Feb  8 23:58 - 08:32 (08:33)
fast        ftp      ppp116.bcc.bilke Sat Feb  8 23:56 - 00:12 (00:15)
fae         ftp      ppp116.bcc.bilke Sat Feb  8 23:49 - 23:56 (00:07)
nur         pts/2    slip4.bcc.bilken Sat Feb  8 23:32 - 00:02 (00:29)
arslann     ftp      charon.bcc.bilke Sat Feb  8 23:25 - 23:25 (00:00)
kaliber     ftp      loj34131.lojman. Sat Feb  8 22:59 - 23:00 (00:00)
arslann     ftp      piranha.bcc.bilk Sat Feb  8 22:39 - 22:40 (00:00)
urucbey     pts/4    139.179.89.229   Sat Feb  8 22:26 - 22:26 (00:00)
kursadd     ftp      loj15671.lojman. Sat Feb  8 20:34 - 20:38 (00:03)
--More--
```

Bir sorunun kaynağını bulmak için hangi log dosyasına bakmanız gerektiğine karar vermek ve bu dosyaların içindeki kayıtları yorumlamak biraz deneyim gerektirmektedir. Merak etmeyin, kısa zamanda bu dosyaları yorumlamayı öğreneceksiniz.



`/var/log` dizinindeki dosyalar bir UNIX sistem yöneticisinin en önemli araçlarından biridir. Neredeyse sistemde olup biten herşeyin bu dosyalar da bir kaydını bulabilirsiniz. `/var/log/dmesg` dosyasında da sisteminizin en son açılışı sırasında olup bitenlerin kaydı tutulur. Her açılıшта bu dosya yeniden oluşturulur.

Açılış sırasında, modüller ve donanım sürücülerini yükledikçe bu dosyaya kayıt düşölür. Bu log dosyalarında hep olumsuz mesajlar yer almaz; başarıyla tamamlanan işler de buraya kaydedilir.

`/var/log/dmesg` dosyasının içeriğini görmek için

dmesg | less

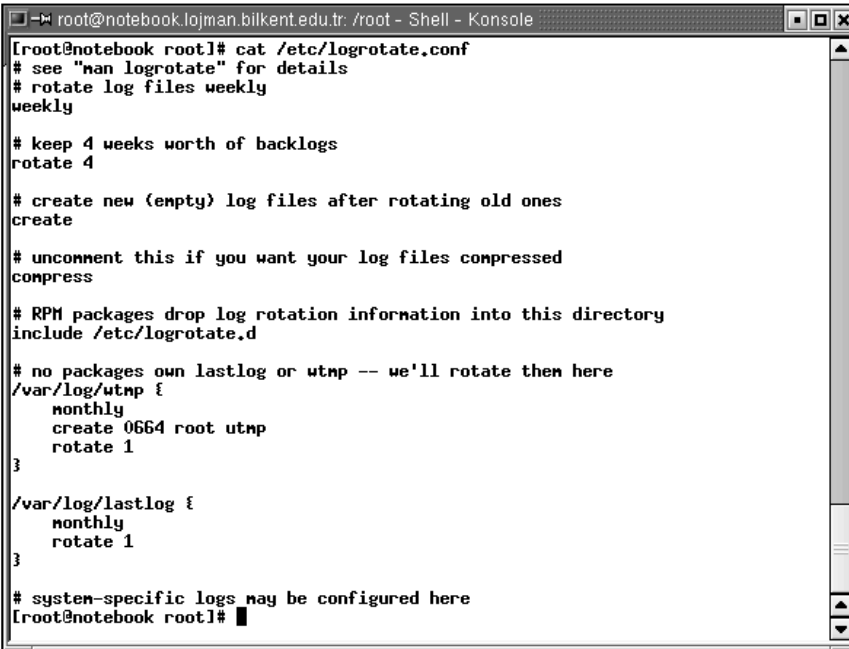
komutunu kullanabileceğiniz gibi

less /var/log/dmesg

komutundan da yararlanabilirsiniz.

Log dosyalarının esas sorunu içeriklerinin yorumlanmasından çok devamlı büyüyen dosyalar olmalarıdır. Diskiniz ne kadar büyük olursa olsun, log dosyaları günün birinde bu diski de dolduracaktır. Log dosyalarının aşırı büyümesini önlemek için LINUX sistemlerde **logrotate** isimli bir program çalışır.

logrotate programının görevi arada sırada log dosyalarını arşivlemek, eskiyen arşivleri de silip atarak **/var/log** dizininin aşırı büyüüp diski doldurmasını önlemektir. Örneğin, **logrotate** programı **messages** isimli log dosyasını haftada bir kez **messages.1.gz** isimli bir dosyaya dönüştürür ve yeni log kayıtlarını yeni bir **messages** dosyasında biriktirmeye başlar. Bu arada varsa eski **messages.1.gz** dosyası **messages.2.gz**'ye, varsa eski **messages.2.gz** **messages.3.gz**'ye dönüştürülür. Tipik olarak **messages.4.gz**, **messages.5.gz**'ye dönüştürülmeden önce, varsa **messages.5.gz** silinir. Böylece log dosyaları her hafta “döndürülmüş” olur. Log dosyaların döndürülmesinde kullanılacak mantık **/etc/logrotate.conf** dosyasında belirtilir. **logrotate.conf** dosyasına kurulum sırasında yerleştirilen değerler tipik bir LINUX sunucu için son derece uygun değerler olduğu için bunları değiştirmeye gerek duymayacağınızı varsayarak bu konuda daha fazla ayrıntıya girmek istemiyoruz.



```
[root@notebook root]# cat /etc/logrotate.conf
# see "man logrotate" for details
# rotate log files weekly
weekly

# keep 4 weeks worth of backlogs
rotate 4

# create new (empty) log files after rotating old ones
create

# uncomment this if you want your log files compressed
compress

# RPM packages drop log rotation information into this directory
include /etc/logrotate.d

# no packages own lastlog or utmp -- we'll rotate them here
/var/log/utmp {
    monthly
    create 0664 root utmp
    rotate 1
}

/var/log/lastlog {
    monthly
    rotate 1
}

# system-specific logs may be configured here
[root@notebook root]#
```

Ağ Yönetimi

Tüm UNIX sistemlerde olduğu gibi LINUX için de doğal ağ yazılımı TCP/IP üzerine kurulmuştur. Bu nedenle LINUX ağ yönetimi aslında TCP/IP ağ yönetimidir. TCP/IP ağ yönetimi başlıbaşına bir kitap konusu olduğu için bu kitapta tüm ayrıntılara girmemize olanak yok. Bir LINUX makinenin bir TCP/IP ağı nasıl bağlanacağını ve bu bağlantının nasıl denetleneceğini kısaca anlatmakla yetineceğiz.

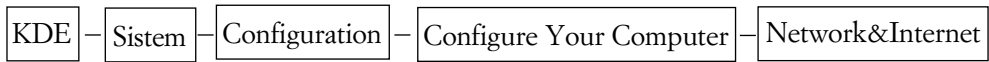
TCP/IP Ayarları

Bir bilgisayarın TCP/IP ağına bağlanabilmesi için o ağı bağlı en az bir ağ arabirimi olmalı ve bu arabirim için bir IP adresi, bir ağ geçidi (*gateway*), bir DNS sunucusu ve ağ maskesi (*netmask*) tanımlanmış olmalıdır.

Eğer bilgisayarınızı ağı Ethernet arabirimi kullanarak bağlayacaksanız bu ayarlar **/dev/eth0** adıyla erişilen ağ arabirimi için yapılmalıdır. Yok bağlantınızı bir modem aracılığıyla yapacaksanız bu ayarları **/dev/ttyS0** gibi bir isimle anılan seri arabirim veya **/dev/modem** adıyla anılan modem arabirimi için yapmalısınız. Kablo-Net veya DSL servisi üzerinden yapacağınız ağ bağlantıları için servis sağlayıcınıza danışmanızı öneririz.

Özellikle DSL bağlantılar için özel **PPPoE** (*Point to Point Protocol over Ethernet*) yazılımını kurmanız gerekebilir.

LINUX'ta TCP/IP ayarları için en kolay kullanılan araçlardan biri "**drakconf**" komutuyla da başlatılabilen "mandriva Control Venter" yazılımıdır. Bunun için:



seçimlerini yapabileceğiniz gibi root kullanıcı yetkisiyle çalışmakta olduğunuz bir terminal penceresinden

/usr/sbin/drakconf

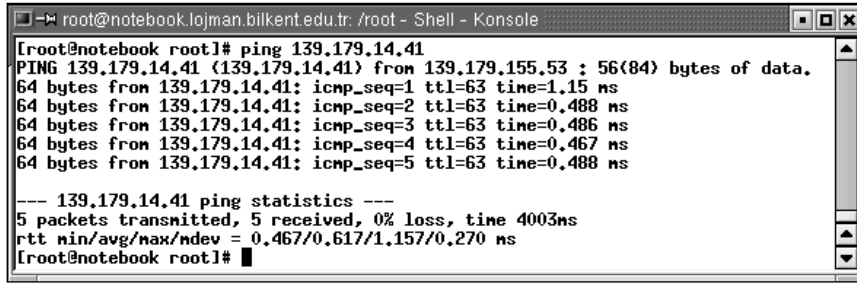
komutunu verebilirsiniz. Her iki yöntem de bilgisayarınızın TCP/IP ayarlarını yapmak için kullanışlıdır.



Ağ arabiriminiz ne olursa olsun, TCP/IP ayarlarınızı kontrol etmek için en kullanışlı komut **ping** komutudur:

ping 192.168.12.1

gibi bir komutla sizinle aynı ağda yer alan bir başka bilgisayara erişip erişemediğinizi kontrol edebilirsiniz. **ping** komutunun adı, masa tenisinden esinlenilmiştir. Temel olarak “*orda mısın?*” sorusu gönderip “*evet*” yanıtını bekleyen bir programdır. “*Orda mısın?*” sorusuna hiçbir zaman “*Hayır*” yanıtı gelmez.



```
root@notebook.lojman.bilkent.edu.tr: /root - Shell - Konsole
[root@notebook root]# ping 139.179.14.41
PING 139.179.14.41 (139.179.14.41) from 139.179.155.53 : 56(84) bytes of data.
64 bytes from 139.179.14.41: icmp_seq=1 ttl=63 time=1.15 ns
64 bytes from 139.179.14.41: icmp_seq=2 ttl=63 time=0.488 ns
64 bytes from 139.179.14.41: icmp_seq=3 ttl=63 time=0.486 ns
64 bytes from 139.179.14.41: icmp_seq=4 ttl=63 time=0.467 ns
64 bytes from 139.179.14.41: icmp_seq=5 ttl=63 time=0.488 ns

--- 139.179.14.41 ping statistics ---
5 packets transmitted, 5 received, 0% loss, time 4003ms
rtt min/avg/max/ndev = 0.467/0.617/1.157/0.270 ns
[root@notebook root]#
```

Hattınızın kalitesini; yani paket kaybı olup olmadığını kontrol etmek için 1500 byte uzunluğunda paketleri sürekli gönderebilirsiniz:

ping -s 15000 -c 200 192.168.12.1

Yerel ağda bir bilgisayara sorunsuz ulaştığınıza emin olduktan sonra

ping 128.12.3.66

gibi bir komutla sizin ağın dışında yer alan ve çalışır durumda olduğuna emin olduğunuz bir bilgisayarı ping'lemeyi deneyin. Başarısız olursanız kontrol edilmesi gereken ilk ayar ağ geçidi ayarınızdır.

Bir sonraki aşama DNS sunucu ayarlarınızı kontrol etmek amacıyla ping'lenecek bilgisayarın IP adresi yerine adını kullanmak olacaktır:

/bin/ping www.sunucu.com.tr

gibi bir komutla **www.sunucu.com.tr** makinesini de ping'leyebiliyorsanız bilgisayarınız sembolik internet adreslerinden sayısal IP adreslerini çözümleyebiliyor demektir. Üstelik ağ geçidiniz olan yönlendiricinin ayarlarını da doğrulamış olursunuz.

drakconf için “TCP/IP ayarlarını yapmak için kullanılan en kolay araçlardan biri” demiştik. TCP/IP ayarlarıyla ustaların yaptığı gibi konsoldan oynamak istediğinizde kullanmanız gereken komutlardan biri **ifconfig** komutudur.

```
/sbin/ifconfig eth0 192.168.10.3 netmask 255.255.255.0  
broadcast 192.168.10.255
```

(Aslında bu komut tek satır olmalı ama sayfaya sığmadı, ne yapalım.)

/dev/eth0 ağ arabiriminizin ne durumda olduğunu, hangi IP adresine ayarlı olduğunu, çalışmaya başladığından bu yana kaç paket gönderdiğini ve aldığı, bu transferler sırasında ne kadar iletişim hatası olduğunu

```
/sbin/ifconfig -a eth0
```

komutuyla görebilirsiniz.

```
root@notebook.lojman.bilkent.edu.tr: /root - Shell - Konsolle
[ root@notebook root ]# ifconfig -a eth0
eth0      Link encap:Ethernet HWaddr 00:20:E0:70:BC:38
          inet addr:139.179.155.53 Bcast:139.179.155.255 Mask:255.255.255.0
          UP BROADCAST RUNNING MULTICAST MTU:1500 Metric:1
          RX packets:410775 errors:0 dropped:0 overruns:0 frame:0
          TX packets:43792 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:100
          RX bytes:175503122 (167,3 Mb) TX bytes:8048600 (7,6 Mb)
          Interrupt:5 Base address:0x7000

[ root@notebook root ]#
```

traceroute komutuyla bir başka ağdaki bilgisayara gönderdiğiniz paketlerin izlediği yolu gözleyebilirsiniz. Bu komut, yerel ağınızın internet’e birden fazla çıkışı olduğu durumlarda çok işe yarayacaktır.

```
root@notebook.lojman.bilkent.edu.tr: /root - Shell - Konsolle
[ root@notebook root ]# traceroute www.mandrake.com
traceroute to www.mandrake.com (63.209.80.235), 30 hops max, 38 byte packets
 1  cisco7010-lib.bcc.bilkent.edu.tr (139.179.70.60)  0,635 ns  0,554 ns  0,520 ns
 2  egzos.bcc.bilkent.edu.tr (139.179.4.4)  0,772 ns  0,597 ns  0,603 ns
 3  139.179.7.2 (139.179.7.2)  0,838 ns  0,804 ns  0,758 ns
 4  routera.neteksan.net.tr (213.139.192.130)  85,392 ns  95,436 ns  1,202 ns
 5  195.175.17.1 (195.175.17.1)  1,699 ns  1,570 ns  1,593 ns
 6  ankH160-ankH20.ttnet.net.tr (195.175.10.1)  1,779 ns  1,893 ns  1,555 ns
 7  getepeH160-ankH160.ttnet.net.tr (195.175.7.2)  8,014 ns  7,999 ns  8,023 ns
 8  sl-gw18-nyc-4-1.sprintlink.net (144.232.171.217)  180,002 ns  180,093 ns  180,330 ns
 9  sl-bb21-nyc-12-1.sprintlink.net (144.232.13.162)  188,121 ns  181,430 ns  181,033 ns
10  sl-bb23-nyc-3-0.sprintlink.net (144.232.7.109)  181,178 ns  180,585 ns  181,440 ns
11  sl-bb20-nyc-8-0.sprintlink.net (144.232.7.13)  181,503 ns  182,701 ns  189,934 ns
12  pos6-2.core1.NewYork1.Level3.net (209.244.160.153)  189,091 ns  186,055 ns  191,247 ns
13  ae0-51.np1.NewYork1.Level3.net (64.159.17.1)  190,600 ns  186,148 ns  187,404 ns
14  so-3-0-0.np2.LosAngeles1.Level3.net (64.159.1.126)  250,934 ns  250,496 ns  256,411 ns
15  pos9-0.core2.LosAngeles1.Level3.net (209.247.10.206)  250,527 ns  252,072 ns  250,524 ns
16  gige6-0.ipcolo2.LosAngeles1.Level3.net (209.244.10.46)  250,375 ns  254,751 ns  253,892 ns
17  mandrake.com (63.209.80.235)  252,358 ns  249,866 ns  252,083 ns

[ root@notebook root ]#
```

Ağ bağlantınızın genel durumu ile ilgili ayrıntılı bilgi almak istediğinizde **netstat** komutu çok işinize yarayacaktır. Özellikle sizin bilgisayarınızla herhangi bir TCP/IP bağlantısı olan bilgisayarları görmek için çok kullanışlı bir komuttur.

“/bin/netstat -s” bilgisayarınızın TCP, UDP, ICMP trafik istatistiklerini listeler. Bunların ne anlama geldiğini bilmiyorsanız hiç dert etmeyin. O kadar da önemli değil.

“/bin/netstat -r” bilgisayarınız tarafından kullanılmakta olan yönlendirme tablosunu gösterir. Bilgisayarınızı bir yönlendirici (*router*) olarak kullanıyorsanız çok işinize yarayacaktır.



Yönlendiriciler, TCP/IP ağların belkemiğini oluşturur. Yönlendiriciler birden fazla bilgisayar ağını birbirine bağlamak için kullanılır. Örneğin, işyerinizdeki 50 bilgisayarını internet'e bağlamak istediğinizde şirketinizin ağını internet servis sağlayıcı kuruluşun bilgisayar ağına bağlayabilmek için bir yönlendirici kullanmanız gerekir. Bir modemle basit bir PPP bağlantısı bile kursanız aslında bu PPP bağlantıyı sağlayan bilgisayar bir yönlendirici olarak görev yapacaktır. Kullanılacak yönlendiricinin ağ arabirim sayısı ve çeşitleri bağlantı ortamına ve ağ gereksinimlerine göre büyük çeşitlilik gösterse de ilke olarak hepsi aynıdır: Birden fazla ağ üzerinde oturan ve gelip giden paketleri varış adreslerine göre uygun arabirime yönelten bilgisayarlara “yönlendirici” (*router*) denir.

Piyasada satılmakta olan yönlendiricilerin neredeyse tamamının salt yönlendirme işi yapacak şekilde, disksiz olarak UNIX işletim sistemiyle çalışabilen birer bilgisayar olduğunu düşünürseniz LINUX işletim sistemi ile kumsuz bir yönlendiriciyi çok ucuza kurabilirsiniz.

Çok sayıda meslektaşımız bu uygulamaya “arkasında ciddi bir firma olmadığı” gerekçesiyle karşı çıksa da, deneyimlerimiz 30.000 ABD doları tutarında yüksek performanslı bir yönlendirici yerine 2.000 ABD Doları karşılığında iki adet PC (bir tanesi yedek olarak hazır tutulmak üzere) ve LINUX işletim sistemiyle 3 internet bağlantısı olan, yaklaşık 5.000 bilgisayardan oluşan bir ağa mükemmel yönlendirme ve ateş duvarı (*firewall*) koruması sağlanabileceğini göstermiştir.

“/bin/netstat -a” bilgisayarınız üzerinde kurulu bulunan TCP/IP bağlantılarını, bu bağlantıların kullandığı port numaralarını ve bağlı bilgisayarın IP adreslerini veya açık adlarını listeler.

```

cayfer@notebook.lojman.bilkent.edu.tr: /home/cayfer - Shell - Konsole
[root@charon /root]# netstat -a
Active Internet connections (servers and established)
Proto Recv-Q Send-Q Local Address           Foreign Address         State
tcp        0      0 charon.bcc.bilkent.:www 195.87.142.4:1039       ESTABLISHED
tcp        0      0 charon.bcc.bilkent.:www 195.112.138.253:3280    ESTABLISHED
tcp        0    256 charon.bcc.bilkent.:ssh  firat.bcc.bilkent:48218 ESTABLISHED
tcp        0      0 charon.bcc.bilkent.:3552 139.179.11.24:inap2     TIME_WAIT
tcp        0      0 charon.bcc.bilkent.:www 195.112.138.253:3279    TIME_WAIT
tcp        0      0 charon.bcc.bilkent.:3551 139.179.11.24:snmp      TIME_WAIT
tcp        0      0 charon.bcc.bilkent.:www  vp180129,reshsg.uc:1679 ESTABLISHED
tcp        0      0 charon.bcc.bilkent.:www  vp180129,reshsg.uc:1679 FIN_WAIT2
tcp        0      0 charon.bcc.bilkent.:3550  firat.bcc.bilkent:inap2 TIME_WAIT
tcp        0      0 charon.bcc.bilkent.:www 195.112.138.253:3277    TIME_WAIT
tcp        0      0 charon.bcc.bilkent.:www 195.112.138.253:3276    TIME_WAIT
tcp        0      0 charon.bcc.bilkent.:www 195.112.138.253:3275    TIME_WAIT
tcp        0      0 charon.bcc.bilkent.:3549  firat.bcc.bilkent:inap2 TIME_WAIT
tcp        0      0 charon.bcc.bilkent.:3548  firat.bcc.bilkent:inap2 TIME_WAIT
tcp        0      0 charon.bcc.bilkent.:3547 139.179.11.24:inap2     TIME_WAIT
tcp        0      0 charon.bcc.bilkent.:www 217.131.5.151:36741     TIME_WAIT
tcp        0      0 charon.bcc.bilkent.:2721  firat.bcc.bilkent:inap2 ESTABLISHED
tcp        0      0 charon.bcc.bilkent.:www  adsl-64-171-7-189.:3287 CLOSE
tcp        0      0 charon.bcc.bilkent.:ssh  firat.bcc.bilkent:39285 ESTABLISHED

```

DNS çözümlemesi yapmak için kullanılan **host** komutunu

host www.mandriva.com

şeklinde kullanırsanız, bilgisayarınızın TCP/IP ayarları arasında belirtilmiş olan DNS sunucusundan **www.mandriva.com** bilgisayarının IP adresi sorulacaktır. Eğer birden fazla DNS sunucusu tanıtılmışsa ve birinciden yanıt gelmezse sorgu ikinci DNS sunucusuna yönlendirilecektir.

```

root@notebook.lojman.bilkent.edu.tr: /root - Shell - Konsole
[root@notebook root]# host cayfer.bilkent.edu.tr
cayfer.bilkent.edu.tr has address 139.179.14.40
[root@notebook root]#

[root@notebook root]# host 139.179.14.40
40.14.179.139.in-addr.arpa domain name pointer cayfer.bilkent.edu.tr.
[root@notebook root]#

```

Komutu

host www.cnn.com 128.12.34.1

şeklinde verirseniz **www.cnn.com** sembolik adresinin sayısal IP adresine iliş-

kin sorgu TCP/IP ayarlarındaki DNS sunucusuna değil, 128.12.34.1 IP adresli bilgisayara yönlendirilir.

Komutu

host 64.236.16.52

şeklinde bir IP adresiyle verirseniz, verdiğiniz IP adresinin hangi sembolik adrese karşılık geldiği sorgulanır.

Ağ yöneticileri zaman zaman *“falanca ağ / makine kime ait acaba?”* sorusuyla karşılaşılırlar. Örneğin **abc.com** diye bir ağdan bilgisayarınıza yönelik bir saldırı olduğunda bu ağın hangi kişi ya da kuruluşa kayıtlı olduğunu bulma gereğini hissedersiniz. Eğer ağın bir web sunucusu var ve burada da iletişim için e-posta adresi, telefon numarası varsa sorun olmaz. Ancak her zaman bu bilgileri bulamayabilirsiniz. Tüm internet “domain” isimlerinin kaydedilmesi gereğinden yola çıkarak sözkonusu ağın adının kimin üzerine kayıtlı olduğunu bulabilirsiniz. (Kötü niyetli kişiler gerçek isim ve telefonlarını vermezler ama olsun, bir ipucudur genede...)

Bu iş için **whois** programından yararlanabilirsiniz. (Eğer yazılım yönetimi bölümünde verdiğimiz örneği uygulamadıysanız **whois** programı bilgisayarınızda yüklü olmayacaktır.)

```
root@cayfer.bilkent.edu.tr: /root - Shell - Konsole
[root@notebook root]# whois google.com
Registrant:
  Google Inc.
  (DOM-258879)
  2400 E. Bayshore Pkwy Mountain View
  CA
  94043 US

Domain Name: google.com

Registrar Name: Alldomains.com
Registrar Whois: whois.alldomains.com
Registrar Homepage: http://www.alldomains.com

Administrative Contact:
  DNS Admin
  (NIC-1340142)
  Google Inc.
  2400 E. Bayshore Pkwy Mountain View
  CA
  94043 US
  dns-admin@google.com +1.6503300100 Fax- +1.6506181499
Technical Contact, Zone Contact:
  DNS Admin
  (NIC-1340144)
  Google Inc.
  2400 E. Bayshore Pkwy Mountain View
  CA
  94043 US
  dns-admin@google.com +1.6503300100 Fax- +1.6506181499
```

Yazıcı Yönetimi

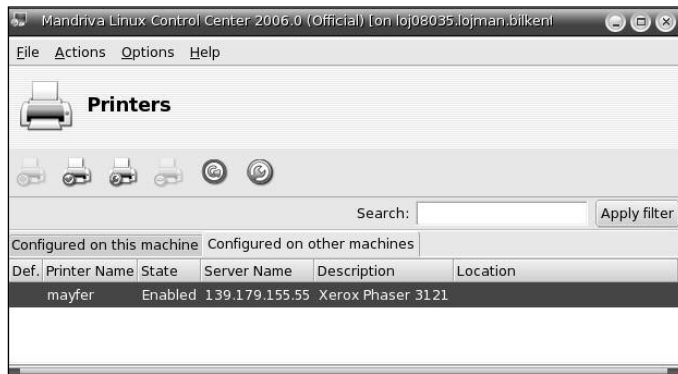
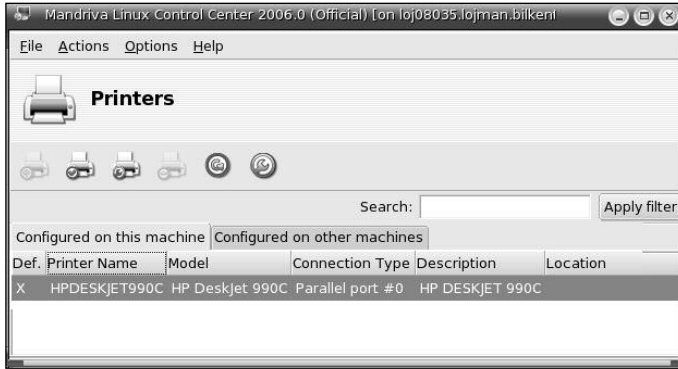
Genel olarak UNIX sisteminde yazıcı yönetimi sistem yöneticilerinin pek sevmediği bir konu olagelmıştır. Ancak, LINUX'un yaygınlaşmasıyla birlikte son derece başarılı ve kolay kullanılan yazıcı denetim sistemleri geliştirilmiştir. Bunların günümüzde en yaygın olarak kullanılanı **CUPS**'dir. (*Common UNIX Printing System*)

Doğrudan bilgisayarınıza da bağlı olsa, bir ağ yazıcısı da olsa, yazıcınızı CUPS ile yönetmenizi öneririz.

Doğal olarak ilk iş yazıcınızın sisteme tanıtılmasıdır. Bunun için Mandriva Control Center içinde **“Hardware”** ve **“Printers”** seçimlerini yaparak işe başlayabilirsiniz.



CUPS ile hem doğrudan bilgisayarınıza bağlı (local) yazıcıları; hem de ağ üzerinden erişebilen yazıcıları (network printer) kullanabilirsiniz.



CUPS sunucunuzu bir web tarayıcısıyla yönetebilirsiniz. Bunun için **cupsd** isimli daemon yazılımının arka planda çalışıyor olması gerekir. Bu son cümleyi eski konuları gözden geçirmek için bir fırsat olarak değerlendirmek istiyoruz



- **cupsd** yazılımının arka planda çalışıp çalışmadığı nasıl kontrol edilir?

"ps -ax | grep cupsd" komutuyla sisteminizde çalışmakta olan tüm süreçlerin listesinden, içinde **"cups"** sözcüğü geçen satırları listeleyebilirsiniz.

- **cupsd** yazılımı arka planda çalışmıyorsa nasıl çalıştırılır?

"/usr/sbin/cupsd &" komutuyla **cupsd** programını arka planda çalıştırabilirsiniz. ("**&**" karakterinin işlevini hatırladınız, değil mi?)

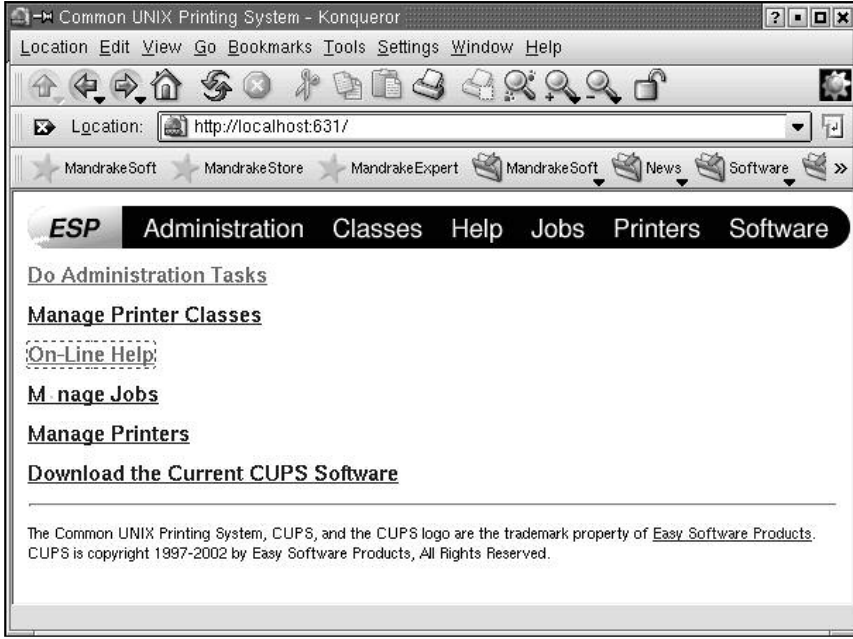
- **cupsd** yazılımı yüklü değilse ne yapmalı?

Yüklemeli... "Mandriva Control Center" sonra "Install Software"...

- **cupsd** yazılımının sistemin her açılışında otomatik olarak başlaması için ne yapmalı? (Her açılıştan sonra konsoldan root olarak **"/usr/sbin/cupsd &"** komutunu vermek zorunda kalmak hiç de hoş değildir doğrusu.)

"chkconfig --add cups" komutuyla **/etc/rc.d** dizinindeki düzenlemelerin yapılmasını sağlayabilirsiniz.

Evet, artık **cupsd** servisi çalıştığına göre CUPS denetimini web tarayıcıyla yapmak üzere beğendiğiniz web tarayıcısını çalıştırıp URL olarak **http://localhost:631** girebilirsiniz. Web tabanlı CUPS yöneticisi, standard http portu olan 80 numaralı port üzerinden değil, kendisine özel 631 numaralı port üzerinden servis verir.



Yedekleme

Sistem yöneticisinin en önemli görevi ya da önemli iki-üç görevinden biri “**yedekleme**”dir. Bilgisayarınızı hiç bozulmayacakmış gibi kullanmalı ama her dakika çökecekmiş gibi yedeklemelisiniz. Arızalanan bir diski değiştirmek kolay ve ucuzdur; ancak uygulama yazılımlarını tekrar baştan kurmak, sistem ayarlarını yeniden yapmak, özellikle de kayıtlı verileri yerine koymak çok zordur. Hatta bazen olanaksız olabilmektedir.

Sisteminizde yapılan işlerin önemine ve erişebildiğiniz donanım kaynaklarının özelliklerine göre kendinize özgü bir yedekleme stratejisi geliştirmeli ve bu stratejiye harfi harfine uymalısınız. Örneğin değerli dosyalarınızı ve dizinlerinizi periyodik olarak bir ikinci disk üzerine ya da daha iyisi, bir başka bilgisayara kopyalamalısınız.

Tipik bir LINUX sisteminde en değerli sistem dosyaları **/etc**, **/var** ve **/usr/local** dizinlerinde yer alır. **/home** da özenle yedeklenmesi gereken bir dizindir.

Daha önceki bölümlerde yedekleme için **tar** komutunu önermiştik. Aslında iki önemli yedekleme aracı daha vardır:

- **dump**
- **rsync**

dump büyük sistemlerde özellikle teyp kasetlerine yedekleme için kullanılır. Kullanması biraz deneyim gerektirir. Bu yüzden bu kitapta **dump** komutunun ayrıntılarına girmeyeceğiz. **dump** ile yedek alırken ayda bir “**tam yedek**” alıp, sonra da her gün “**artımlı yedekleme**” yapabildiğinizi belirtmek isteriz. (*incremental backup*, Türkiye Bilişim Derneği, Bilişim Terimleri Sözlüğü, <http://www.tbd.org.tr/sozluk.html>) “**Artımlı yedekleme**” de, en son tam yedeklemeden bu yana değişen dosyalar yedeklenir. Böylece yedekleme çabuk biter. Günümüzün tipik sunucu disk kapasitelerinin yüzlerce gigabyte olduğunu ve bu kapasiteleri yedeklemenin teyp veri transfer hızlarıyla saatlerce süreceğini düşünürseniz artımlı yedeklemenin ne denli önemli olduğunu kabul edersiniz herhalde.

rsync diskten diske ya da yerel bir ağ üzerinde makineden makineye yedekleme için kullanılabilen çok kullanışlı bir araçtır. Temel mantığı iki dizini senkronize etmektir.

Diyelim ki alçakgönüllü bir ağın yönetiminden sorumlusunuz ve ortalıkta bazı önemli dizinlerini yedeklemek istediğiniz on tane makine var. Her gece yarısı bu makinelerden **rsh** komutuyla yedeklemek istediğiniz dizin ve dosyaları yedekleme makinesine kopyalayabilirsiniz.

web sunucusunda

```
rsh -r /home depo:/web  
rsh -r /etc depo:/web
```

e-posta sunucusunda

```
rsh -r /var/spool/mail depo:/mail
```

ya da

depo makinesinde

```
rsh -r web:/home /web
```

```
rcp -r web:/etc /web
rcp -r mail:/var/spool/mail /mail
rcp -r muhasebe:/var/data /mhsb
```

gibi komutlar vererek bu şekilde bir yedekleme yapılabilir.

Ancak on makine için bu işler oldukça uzun sürecektir. Onun için **rcp** ile kopyalamak yerine **rsync** ile yalnızca değişen dosyaları kopyalamayı düşünebilirsiniz.

depo makinesinde vereceğiniz

```
/usr/bin/rsync -az muhasebe:/var/data/ /mhsb/var/data
```

komutuyla muhasebe makinesindeki **/var/data** dizinindeki dosya ve alt dizinlerle komutun verildiği makinedeki **/mhsb/var/data** dizinindeki dosyalar ve alt dizinlerin son değişiklik tarihleri karşılaştırılır. Eğer **muhasebe** makinesindeki dosya ve dizinlerin tarihi daha yeniyse, o dosyalar komutun verildiği makineye (**depo** makinesine) kopyalanır.

Önemli bir nokta da şu: Daha önce yedeklenmiş olan **mhsb:/var/data** dizinindeki **2002_bilancio** isimli bir dosya **muhasebe** makinesinden silinirse, **depo**'daki kopyası kalacaktır, çünkü son değişiklik tarih-saat karşılaştırması yalnızca gönderen makinedeki dosya ve dizinler için yapılır. Aslında bu özellik arada sırada işe de yarar. Yanlışlıkla silinmiş birçok dosyayı bu özellik sayesinde kurtarmak mümkün olabiliyor. Ancak bu “kalıntı” dosyalar büyük dosyalarsa, o zaman da disk kapasitesi sorununa yol açabiliyor. Eğer bu gibi durumlarda gönderici makinede yok olmuş dosyaların alıcı makineden de silinmesini isterseniz **rsync** komutunu **--delete** parametresiyle kullanabilirsiniz.

```
/usr/bin/rsync -az --delete muhasebe:/var/data/ /mhsb/var/data
```

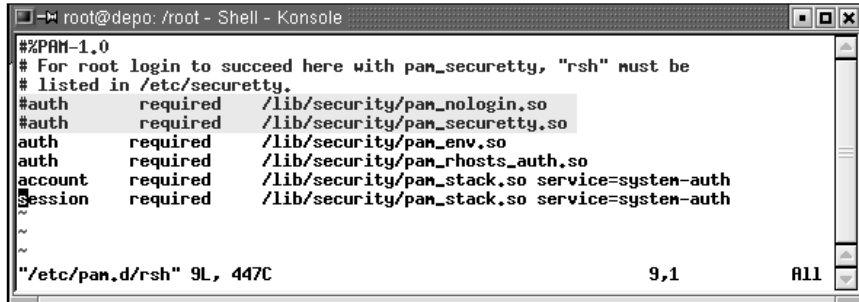
rsync programını kullanarak makineler arası yedekleme yapabilmeniz için hem gönderen hem alan makinelerde **rsync** ve **rsh** modülleri kurulu olmalıdır. Yedekleme işini genellikle root yetkisiyle yapmak isteyeceğiniz için güvenlik duvarlarını aşmak için yapmanız gereken iki iş daha olacaktır:

1. Gönderici makine üzerinde **/root** dizininde yani root kullanıcının kişisel dizininde adı **.rhosts** olan bir dosya yaratmalı ve bu dosyanın içine

192.168.0.12 root cayfer

gibi bir satır eklemelisiniz. Burada 192.168.0.12 yalnızca bir örnek olup, “diğer makine”nin IP numarası olarak değerlendirilmelidir. Bu satır, 192.168.0.12 IP adresli bilgisayardan gelen “**root**” ve “**cayfer**” kimlikli kullanıcıların her iki makinede de tanımlı olmak kaydıyla aynı kimliklerle kabul edilmelerini sağlamak içindir. **.rhosts** dosyasını düzenlerken liberal olmamanızı öneririz; önemli bir güvenlik gediği açabilirsiniz.

2. İkinci olarak da **/etc/pam.d/rsh** dosyasında **pam_nologin.so** ve **pam_securetty.so** satırlarının başına birer “#” yerleştirerek bu seçimleri geçersiz kılmalısınız. (Bunun neden yapıldığını ve pam’in ne olduğunu açıklamak biraz zor. Yeni başlayanlar için “**pam**” sözcüğünün “Password Authentication Module” isminin kısaltması olduğunu ve kullanıcıların özellikle ağ üzerinden yapabilecekleri işlerde yetki ve şifre değerlendirmelerinin nasıl yapılacağını belirleyen bir servis olduğunu açıklamak yeterli olacaktır. Güvenlik konusunda deneyim kazandıkça **pam** konusunda da bilginiz ve hakimiyetiniz artacaktır.)



```
root@depo: /root - Shell - Konsole
#%PAM-1.0
# For root login to succeed here with pam_securetty, "rsh" must be
# listed in /etc/securetty.
#auth      required    /lib/security/pam_nologin.so
#auth      required    /lib/security/pam_securetty.so
auth       required    /lib/security/pam_env.so
auth       required    /lib/security/pam_rhosts_auth.so
account    required    /lib/security/pam_stack.so service=system-auth
session    required    /lib/security/pam_stack.so service=system-auth

"/etc/pam.d/rsh" 9L, 447C          9,1          All
```

Periyodik İşlerin Yönetimi

Sisteminiz geliştikçe, daha doğrusu sisteminize bağımlılığınız arttıkça sistem yöneticisi olarak üstlendiğiniz sorumluluklar da artacaktır. Kullanıcılarınıza “*disk dolmuş da ondan falanca servis çalışmıyormuş*” dememelisiniz.

Bu sorumluluğun gereği olarak bazı işlerin periyodik olarak yapılması gerekecektir. Örneğin belirli dizinlerin her akşam yedeğinin alınması, her Pazar-tesi belirli geçici dizinlerin temizlenmesi, her gün 6 saate bir bazı raporların hazırlanıp müşteriye ya da patrona e-posta ile gönderilmesi gibi...

Bu şekilde belirli bir plana-programa bağlanmış işler LINUX altında **cron** servisinin desteğiyle yapılır.

cron servisi aslında üç unsurdan oluşur:

1. **crond:** cron daemon, geri planda pusuda yatan, yapılacak işlerin zamanının gelmesini bekleyen yazılım.
2. **cron dosyaları:** Her kullanıcı için hangi saatlerde, günlerde hangi işlerin yapılacağını saklandığı dosyalar. (**/var/spool/cron** altında, her kullanıcı için, kullanıcının adıyla anılan bir dosya yer alır.)
3. **crontab komutu:** Kullanıcının cron dosyalarını düzenlemek için kullanacağı komut.

/var/spool/cron altındaki dosyaları elle düzenlememelisiniz. Bu dosyalar da bir değişiklik yapmanız gerektiğinde

crontab -e

komutunu kullanmalısınız. **-e** parametresi dosyayı düzenlemek istediğinizi (*edit*) belirtmektedir.

crontab komutu, komutu veren kullanıcıya ait **cron** dosyasını **vi** editörüyle açar ve gerekli değişikliklerin yapılmasını bekler. **vi** editöründen çıktıldığında yeni **cron** dosyası devreye girmiş olur.



```

root@cayfer.bilkent.edu.tr: /etc - Shell - Konsole
0 4 * * * /usr/local/bin/yedekle
0,15,30,45 * * * * /hone/cayfer/ping.pl
30 0 * * * (cd /var/www/html/traffic/; perl gen_log.pl)
~
~
~
~
~
~
4,0-1 All
  
```

Yan sayfadaki örnekte **/usr/local/bin/yedekle** programı –ki aslında bir bash kabuğu için bir betik programıdır– (betik: script; Türkiye Bilişim Derneği, Bilişim Terimleri Sözlüğü – www.tbd.org.tr) hergün sabaha karşı saat

dörtte çalıştırılmaktadır. Bu cron dosyası “root” kullanıcısına ait olduğu için de bu yedekle betiği root yetkileriyle çalışacaktır.

Gene aynı örneğe göre **/home/cayfer/ping.pl** betiği 15 dakikada bir çalıştırılan bir Perl programdır. Görevi bazı önemli bilgisayarlara erişilebildiğini kontrol etmektedir. Söz konusu Perl programı ping'lenemeyen bilgisayarları bir e-postayla sistem yöneticisine bildirmektedir. (Bu arada Perl dilini öğrenmeyi düşünürseniz Pusula Yayıncılığın “Perl ve MySQL ile CGI Programlama” kitabını önerebiliriz.)

Son **crontab** satırı ise her geceyarısı 00:30'da **/var/www/html/traffic** dizinine geçilmesini ve ardından “**perl gen_log.pl**” komutunun çalıştırılmasını sağlar.

Şimdi gelelim **cron** dosyasının ayrıntılarına:

cron dosyalarında 6 bilgi alanı vardır. Bunlardan ilk beş tanesi işlerin başlatılacağı gün, saat ve dakikaları belirtmek için kullanılır, altıncısı da işleri başlatmak için kullanılacak komutları belirler.

Alan	Anlamı	Değer Aralığı	Açıklama
1	İşin seçilen saat başından kaç dakika sonra başlatılacağını belirler.	0 - 59	“30”, “saat başını otuz dakika geçe” demektir.
2	İşin hangi saatte başlatılacağını belirler.	0 - 23	Sayı yerine “*” girilirse “ her saat ” anlamına gelir.
3	İşin hangi günler başlatılacağını belirler.	1 – 31	Sayı yerine “*” girilirse “ hergün ” anlamına gelir. 7, 14, 21, 28 yazılırsa “her ayın 7’si, 14’ü, 21’i ve 28’i” anlamına gelir.
4	İşin hangi aylarda yapılacağını belirler.	1 – 12	Sayı yerine “*” girilirse “ her ay ” anlamına gelir. “1, 6” yazılırsa iş sadece Ocak ve Haziran aylarında çalıştırılır. “1-3” yazılırsa iş yalnızca Ocak, Şubat ve Mart aylarında çalıştırılır.
5	İşin haftanın hangi günlerinde çalıştırılacağını belirler.	0 - 7	Sayı yerine “*” girilirse “ ne gün olursa olsun ” anlamına gelir. 0. ve 7. günler Pazar kabul edilir. “1, 2” yazılırsa “Pazartesi ve Salı” anlamına gelir. “1-3” yazılırsa Pazartesi, Salı ve Çarşamba günleri anlamına gelir.



Sisteminizde **crond** çalışmıyorsa, **cron** dosyalarını düzenlemeniz bir işe yaramayacaktır. Sisteminizin her açılışında **crond**'nin de başlatılması için

```
chkconfig --add crond
chkconfig crond on
```

komutlarını vererek **/etc/rc.d** dosyalarında gerekli düzenlemelerin yapılmasını sağlayabilirsiniz.

İşte birkaç örnek:

15 * * * * prog1	prog1 15 dakikada bir çalışır.
0 17 * * * prog1	prog1 hergün saat 17'de çalışır.
0 10,12,14,16,18 * * * prog1	prog1 hergün saat 10 ile 18 arasında iki saatte bir çalışır.
59 23 * * 1-5 prog1	prog1 her iş günü geceyarısına bir kala çalışır.
* * * * * prog1	Sistemi mahvedebilecek bir cron satırı. Hergün, her saat ve her dakika yeni bir prog1 başlatılacak demektir. Artık neler olacağı prog1 programına bağlıdır.

LILO Yönetimi

LILO, yani **L**inux **L**Oader sisteminizin önemli bir yazılımıdır. Görevi, sistemin açılışı sırasında hangi disk bölümünden (*partition*) hangi işletim sisteminin hangi parametrelerle belleğe yükleneceğini belirlemek ve yönetmektir. Aynı amaca yönelik **GRUB** isimli bir yazılım da LINUX dünyasında oldukça yaygın olarak kullanılmaktadır, ama bu kitapta biz yalnızca **LIL**O'dan söz edeceğiz.

LILO sayesinde istediğiniz işletim sistemini istediğiniz diskin istediğiniz bölümünden yükleyebilirsiniz; üstelik bu yüklenen işletim sistemi LINUX olmak zorunda da değildir.

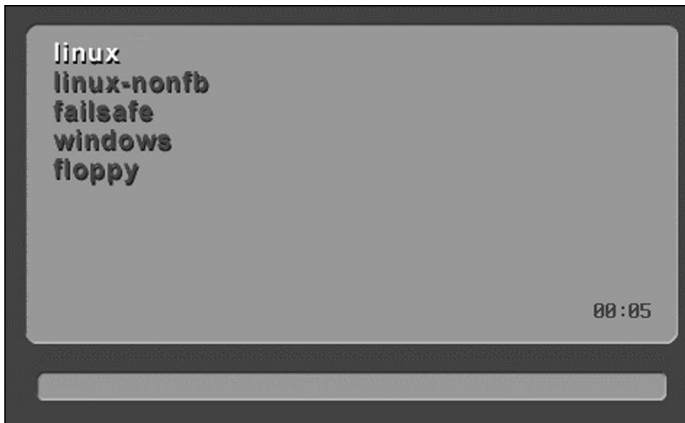
LILO'yu bilmek ve yönetebilmek, özellikle sistemini bazen Windows bazen de LINUX işletim sistemiyle açmak isteyen kullanıcılar için önemlidir. LINUX dünyasına yeni girenlerde gözlediğimiz ortak yaklaşım genellikle eski işletim sisteminin elinin altında olmasını istemeleridir. LINUX'u beğenip, işlerini bu platforma taşıyanlar bir süre sonra eski işletim sistemini içeren disk bölümüne bir **“mkfs atıp”**, o bölümü de LINUX disk alanı olarak kullanmaya başlıyorlar. Neyse...

Tekrarlama olacağını bile bile, burada PC türü bilgisayarların **“boot”** sürecini hatırlatmak istiyoruz.

Bilgisayarın BIOS yazılımı bellek, görüntü kartı gibi önemli bazı donanım unsurlarını kontrol ettikten sonra BIOS ayarları çerçevesinde disket sürücü, CD gibi birimleri kontrol eder ve genellikle bu sürücülerde bir ortam takılı olmadığı için diskten **“boot etmeye”** karar verir.

BIOS, diskten **“boot etmeye”** karar verirse birinci diskin **“boot sektörü”** diye adlandırılan bölgesinden bilgisayarın işletim sistemini yükleyecek yazılımı belleğe alır.

LILO işte bu boot sektöründen yüklenen yazılımdır. Sistem ilk açıldığında BIOS, boot sektöründen LILO'yu belleğe yükler. LILO da kendi ayarları doğrultusunda kullanıcıya işletim sistemi yüklemeyle ilgili seçenekleri sunar.



LILO yazılımı **/etc/lilo.conf** dosyasından yönetilir. Bu dosyada gereksinimleriniz doğrultusunda değişiklik yaptıktan sonra **/sbin/lilo** komutuyla yeni düzenlenen LILO konfigürasyonuna göre birinci diskin boot sektörlerine gerekli kayıtlar yerleştirilir.

Her ne kadar Mandriva Linux Control Center menüsündeki LinuxConf aracıyla LILO konfigürasyon dosyalarını düzenlemek mümkünse de bu işin temelini öğrenmenizde yarar vardır. Bu nedenle örneklerimizi **/etc/lilo.conf** üzerinde editörle değişiklik yapıp **/sbin/lilo** programını çalıştırma üzerine kuracağız.

Öncelikle tipik bir **lilo.conf** dosyasına göz atalım:

```
boot=/dev/hda
map=/boot/map
install=/boot/boot.b
vga=normal
default="linux"
keytable=/boot/tr_q-latin5.klt
prompt
nowarn
timeout=100
message=/boot/message
menu-scheme=wb:bw:wb:bw
#
image=/boot/vmlinuz
    label="linux"
    root=/dev/hda6
    initrd=/boot/initrd.img
    append="devfs=mount acpi=off"
    read-only
other=/dev/hdd1
    label="windows98"
    table=/dev/hdd
other=/dev/fd0
    label="floppy"
    unsafe
```

Dikkat ederseniz bu konfigürasyon dosyasının başında “**boot=**” diye başlayıp “**menu-scheme=**” diye biten global değişkenler ve ardından “**image**” ve “**other**” parametreleriyle bloklanmış gruplar yer alıyor.

Global parametreler arasında önemli olanlar:

boot=/dev/hda	LILLO’nun ilk diskin boot sektörüne yerleştirileceğini belirtiyor.
timeout=100	Sistem açıldığında kullanıcı LILLO menüsünden timeout/10 saniye içinde bir seçim yapmadığı takdirde varsayılan boot konfigürasyonunun kullanılacağını belirtiyor. Dikkat edin! timeout parametresi saniye cinsinden değil, saniyenin onda biri cinsinden belirtilir.
default="linux"	timeout/10 saniye sonunda hâlâ bir seçim yapılmadıysa etiketi “ linux ” olan konfigürasyonun kullanılacağını belirtiyor.

“**image=**” diye başlayan bloklar LINUX işletim sistemine ait tanımları içerir.

Örneğin,

```
image=/boot/vmlinuz
    label="linux"
    root=/dev/hda6
    initrd=/boot/initrd.img
    append="devfs=mount acpi=off"
    read-only
```

satırları “**linux**” isimli tanım için:

- Bu grubun “**linux**” ismiyle anılacağını,
- Bu grubun seçilmesi durumunda işletim sisteminin çekirdeğinin **/boot/vmlinuz** isimli dosyadan yükleneceğini,
- **root** dizin olarak ilk diskin altıncı bölümünün kullanılacağını (hda6),
- Yükleme sürecinde kullanılacak sanal disk görüntüsünün (*image*) **/boot/initrd.img** dosyasında yer aldığını (bunun ne demek olduğu, yalnızca çekirdek kodu geliştirenleri ilgilendirir),

Kim Korkar LINUX'tan?

- İşletim sisteminin çekirdeğine “**devfs=mount acpi=off**” parametrelerinin geçirileceğini,
- Dizinlerin salt oku mount edileceğini (LINUX gerekli kontrolleri yaptıktan sonra herşey yolundaysa, gereken tüm izinler oku-yaz olarak yeniden mount edilir) belirtiyor.

“**other=**” diye başlayan bloklar LINUX olmayan işletim sistemine ait tanımları içerir.

Örneğin,

```
other=/dev/hdd1  
    label="windows98"  
    table=/dev/hdd
```

satırları:

- Bu grubun “windows98” ismiyle anılacağını,
- Bu grubun seçilmesi durumunda yüklenen işletim sisteminin kullanacağı bölümlene tablosunun (*partition table*) ikinci IDE kanalındaki ikinci diskten (hdd) okunacağını belirtiyor.

Şimdi diyelim ki sisteminize üzerinde Windows-XP yüklenmiş yeni bir disk birinci IDE kanalının ikinci disk (*hdb, slave*) olarak taktınız ve sisteminizi bu diskten de açabilmek istiyorsunuz. Yapmanız gereken **/etc/lilo.conf** dosyasına

```
other=/dev/hdb1  
    label="windows-XP"  
    table=/dev/hdb
```

satırlarını ekleyip

```
/sbin/lilo
```

komutunu çalıştırmak olacaktır. Bir daha reboot ettiğinizde **LIL**O size Windows-XP yüklemeyi de bir seçim olarak sunacaktır. Bu işleri yapabilmemiz için sisteminizi önce LINUX ile açmanız gerektiğini belirtmeye gerek yok herhalde. Aslında sisteminizi neden XP ile açma gereksinimi duyduğunuzu da anlamadık ya, neyse...

Webmin

Bu bölümü okuyunca bize kızacaksınız... “Madem *webmin* vardı, ne diye bir *sürü komut ve ayar dosyası anlattınız?*” diyeceksiniz.

Gerçekten de “webmin”, LINUX işletim sistemini tek yazılımla yönetmek için geliştirilmiş web tabanlı bir uygulamadır.

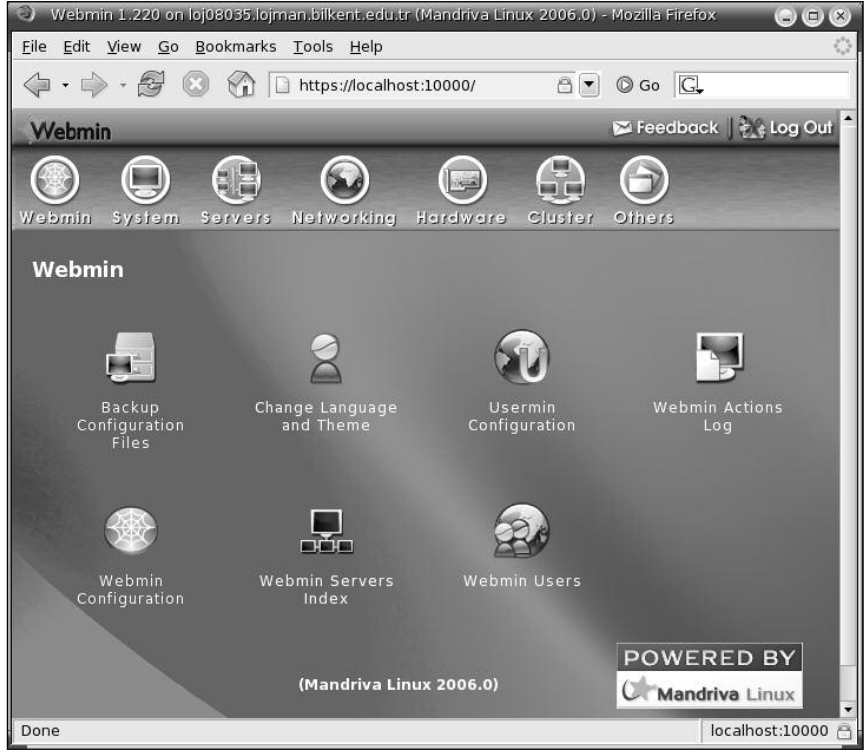
Webmin’i kullanmak için bilgisayarınızda herhangi bir web tarayıcı; örneğin Konqueror başlatıp URL olarak

https://localhost:10000

yazınız. Yani, 10000 numaralı porttan web servisi isteyiniz.



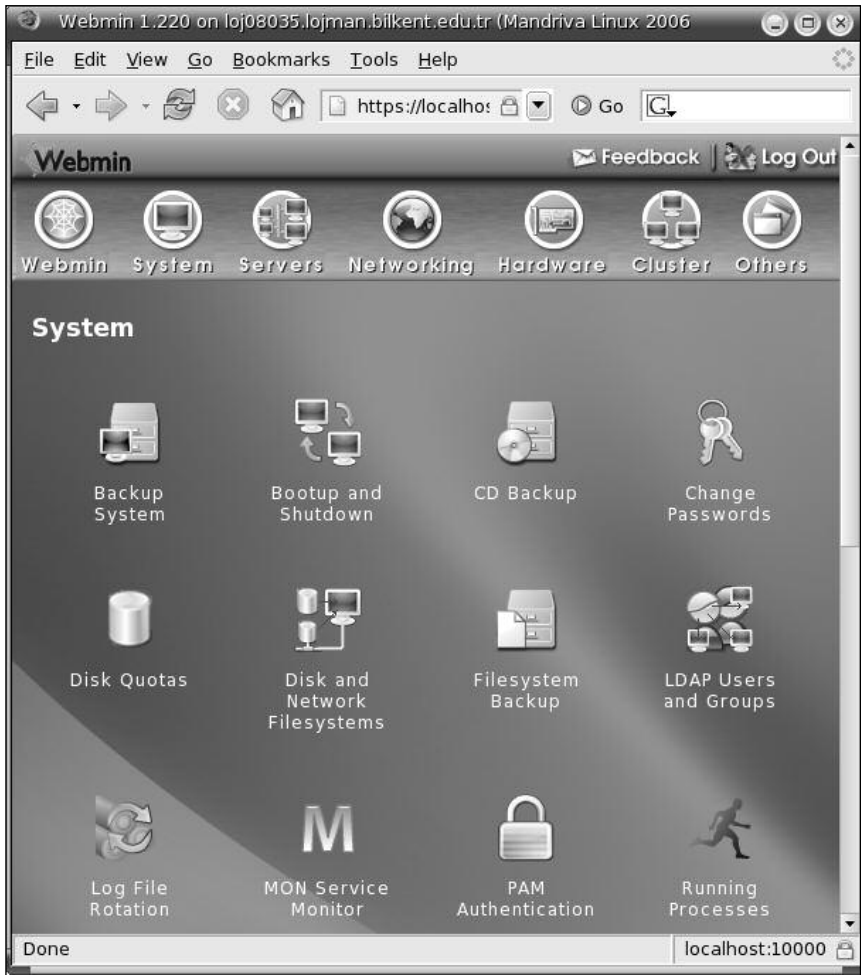
Kim Korkar LINUX'tan?



Doğal olarak root kullanıcı kimliğiyle servise bağlanmanız gerekecektir.

Artık webmin menüsünden yararlanarak, sisteminizin neredeyse tüm yönetim işlerini bu arabirim ile yapabilirsiniz.

Webmin'in işlevlerini bu kitapta anlatmayacağız. LINUX işletim sisteminin yönetim kavramlarının temellerini öğrenmiş olduğunuza göre gerisini siz kendiniz araştırabilir ve keşfedebilirsiniz.

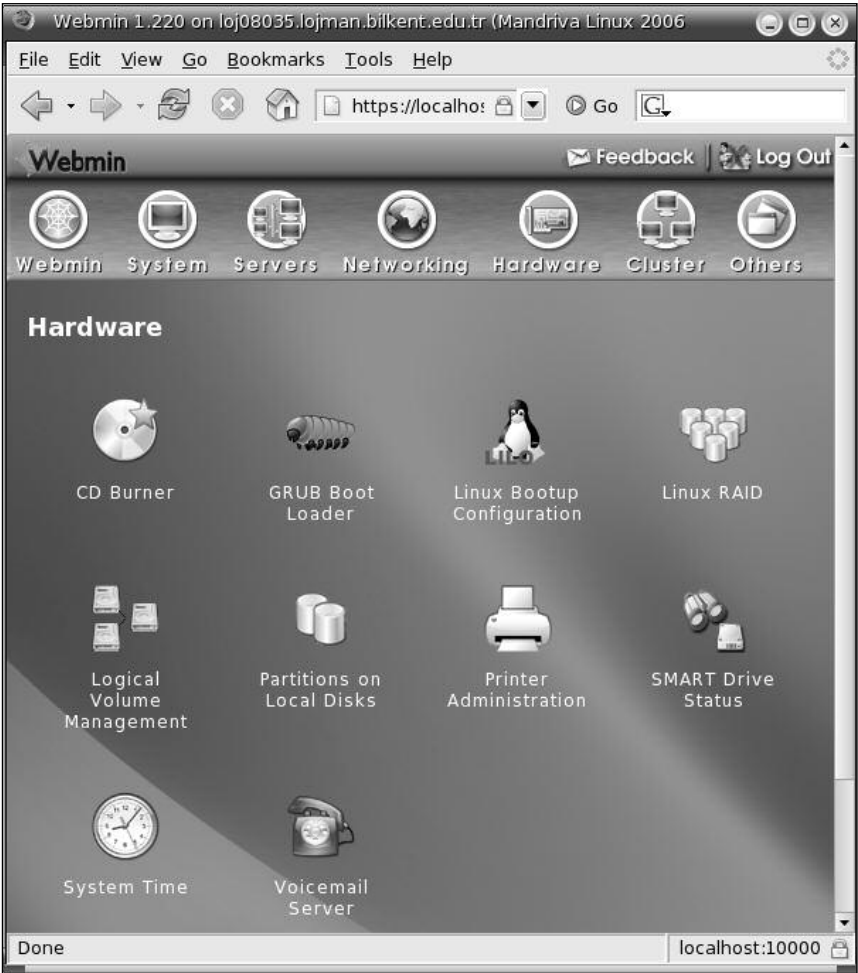




Apache web sunucusu, MySQL veritabanı yönetim sistemi, Postfix e-posta sunucusu, FTP sunucusu gibi LINUX'u LINUX yapan birçok sunucu servisinin ayarlarını bu menüden yapabilirsiniz.



Kim Korkar LINUX'tan?





“root” Şifresini Unuttuğunuzda...

Bir LINUX sistemin “root” şifresinin bilinmemesi, sanıldığı kadar az karşılaşılan bir durum değildir. Sistem yöneticisi şifreyi unutabilir; tatil deyken ya da işten ayrıldıktan sonra sisteme başkalarının “root” olarak bağlanması gerekebilir, ya da en kötüsü sisteminizi kıran birisi “root” şifresini değiştirebilir.

“root” şifresi bilinmeyen bir sistem üzerinde denetimi ele tekrar almanın tek yolu, “root” şifresini bilinen bir şifreyle değiştirmektir. Hatırlarsanız, daha önce kullanıcı şifrelerinden bahsederken, UNIX ve LINUX işletim sistemlerinde şifrelemenin tek yönlü olduğunu, şifrelenmiş hali bilinen bir şifrenin açık halinin ne olduğunun bulunmasının olanaksız olduğunu belirtmiştik. “root” şifresini değiştirmek için sisteminizi kapatın. Yeniden açılış sırasında LILO size yüklenebilecek işletim sistemlerini gösterdiği sırada bir kez Esc tuşuna basın.

Kim Korkar LINUX'tan?

Karşınıza

boot:

hazır işareti çıkacaktır. Bunun karşısına “**linux single**” yazıp Enter tuşuna basınız.

boot: linux single

LINUX çekirdeği belleğe yüklendiğinde “**single**” sözcüğü çekirdeğe parametre olarak geçirilecektir. LINUX çekirdeği de bu parametreyi görünce açılışı birinci çalışma düzeyi (run level 1) tamamlandığında kesecektir.

LINUX çalışır durumda olacak ancak yalnızca konsoldaki kullanıcıya hizmet edecektir. Yani, varsa web servisi, ftp servisi, e-posta servisi gibi hizmetleri yürüten yazılımlar başlatılmayacaktır.

Ayrıca, konsoldaki kullanıcının kendisini sisteme tanıtması da istenmeyecektir. “root” yetkileriyle bir **bash** kabuğu başlatılıp kullanıcıdan komut girmesi beklenecektir.

Bu konumda

passwd

komutunu vererek o anda geçerli olan kullanıcının; yani “root”un şifresini değiştirebilirsiniz. **passwd** komutunu veren kullanıcı “root” olduğu için de eski şifre sorulmayacaktır.

Şifreyi değiştirdiğinizde

exit

komutunu vererek çekirdeğin açılışa devam etmesini sağlayabilirsiniz.

Meraklısına...



Bir LINUX bilgisayarın açılışı sırasında LILO'ya "**linux single**" yazarak sistemin tek kullanıcı ve şifre sormadan "**root**" kullanıcı olarak açılmasını önleyebilirsiniz.

/etc/lilo.conf dosyasında ilgili ayar satırları arasına

```
password=eb!TKY-2
restricted
```

gibi satırlar ekleyip

lilo

komutunu vererseniz, artık sistem tek kullanıcı düzeyde açılırken bile "**eb!TKY-2**" şifresinin girilmesini isteyecektir. Dikkatinizi çekeriz... Bu şifre sistemin "**root**" şifresi değildir; tek kullanıcı düzeyde açma şifresidir.

Bu şifreyi pek fazla kullanmayacağınız için unutma olasılığınız çok daha yüksektir. Günün birinde, "**root**" şifresini unuttuğunuz için sisteminizi tek kullanıcı düzeyde açmanız gerekirse bu şifreyi hatırlamayacağınıza bahse gireriz. Bu nedenle, tek kullanıcı çalışma düzeyi koruması için **lilo.conf** dosyasına bu satırları koymanızı hiç önermiyoruz.

Eğer bir sistemin tek kullanıcı düzey şifresi ve root şifresi unutulursa yapılacak iş, diski sökmek ve bu diski şifreleri bilinen bir başka LINUX bilgisayara ikinci disk olarak takmak ve diskin üzerindeki **etc/shadow** dosyasında root kullanıcıya ait satırdaki kriptolanmış şifreyi bir editörle silmektir.

Kim Korkar LINUX'tan?

- “Paket” Kavramı
- Paket Bağımlılığı (Dependency)
- Sistem Güvenliği ile İlgili Haberleri İzleyin
- Paket Yöneticileri (Package Managers)
- Güvenlikle İlgili Yamaları İzleyin ve Uygulayın
- Yazılım Depoları (Media, Repositories)
- Yazılım Güncelleme
- Yazılım Yükleme
 - Mandriva rpm Paket Yöneticisi - rpmdrake ile Yükleme
 - urpmi - Gerçek Paket Yöneticisi
 - rpm Paketlerinden Yazılım Yükleme
- Kaynak rpm Paketlerinden (src-rpm) Yazılım Yükleme
- Kaynak Kodundan Derleyerek Kurulum
- İşletim Sisteminin ve Uygulama Programlarının Güncellenmesi
- Çekirdek Güncellemeleri (Kernel Updates)

“Paket” Kavramı

LINUX dünyasında, işletim sisteminin çekirdeği de dâhil olmak üzere, sistem ve uygulama yazılımları genellikle “paketler” halinde dağıtılır. Bir “paket” içinde, söz konusu yazılımı oluşturan dosyalar ve bu dosyaların nerele-re, nasıl kopyalanacağı bilgisi yer alır.

Bir bilgisayara “LINUX kurmak” demek; aslında o bilgisayara peş peşe binlerce paket kurmak demektir. Önce “kernel” paketi kurulur; sonra da işletim sistemi araçları ve uygulama programlarını içeren paketler yüklenir.

LINUX dünyasında; RedHat ve Debian dağıtımları için geliştirilmiş olan 2 önemli “paket” çeşidi vardır. RedHat'in kullandığı paket standardına “RPM” (RedHat Package Manager), Debian dağıtımının paketlerine “DEB” paketleri adı verilir.

Mandriva LINUX, paket standardı olarak RPM'i seçmiş ve kullanmaktadır.

“Paket Yönetici” leri de bu yazılım paketlerinin kolay kurulmasını, güncellenmesini, silinmesini sağlayan yazılımlardır. Örneğin “gimp” yazılımını içeren RPM paketinde 1260 kadar dosya yer almaktadır. “gimp yazılımını kurmak” demek bu 1260 dosyayı belki on değişik dizine dağıtmak demektir. Ayrıca, bir paketin kurulması bir takım ayar dosyalarında değişiklik yapılmasını da gerektirebilir. Bir de, en önemlisi, kurulum sırasında sistemde bulunan kütüphanelerin, bağlantılı başka yazılımların, yüklenmek istenen yazılıma uygun olup olmadığı sorusu vardır.

Paket Bağımlılığı (Dependency)

“gimp” örneği ile başladık, “gimp” ile devam edelim... Bir LINUX bilgisayara “gimp” paketinin başarıyla yüklenebilmesi için bilgisayarınızda önceden yüklü olması gereken 90 tane C kütüphanesi dosyası vardır. Bir başka deyişle “gimp” paketinin bağımlı olduğu 90 tane paket vardır. Bilgisayarınızda bu paketler, sürümlerine varıncaya kadar uygun şekilde kurulu değilse “gimp” paketini kuramazsınız.

Bu bağımlılık (dependency) konusu LINUX kullanıcılarının başını en çok ağrıtan konudur. “gimp” örneğindeki 90 paketin hangileri olduğunu bilmek, önce bunları bir yerlerden bulup yüklemek pek yapılası bir iş değildir. Ancak, doğal olarak bunun da çözümü var: “Paket Yöneticileri”...

Paket Yöneticileri (Package Managers)

RedHat'in “rpm” paket yöneticisi; Debian'in ise “dpkg” si var. Mandrake ve Mandriva da RedHat'in “rpm” paket yöneticisini kullanmayı tercih etmiş. Zamanla salt paket yöneticilerinin, özellikle bağımlılık sorunlarının çözülmesinde (dependency resolution) yetersiz kaldığı görülünce RedHat “yum”, Mandriva “urpmi”, Debian ise “apt” yazılımlarını geliştirdiler. Yöntemleri farklı da olsa, bu 3 yazılımın (urpmi, yum ve apt) temel işlevi paket

yüklemek; gerekiyorsa söz konusu paketin bağımlı olduğu diğer paketleri de bulup, onları da yüklemektir. Bir başka deyişle, kullanıcının “gimp” paketini yüklemek istediğini “urpmi” programına bildirmesi yeterlidir; “gimp”in bağımlılığı olduğu diğer 90 paketi bulup önce bunları yüklemek paket yöneticisi yazılımının görevidir.

Mandriva; RPM paketlerinin yönetimi için kullandığı “urpmi” için bir grafik önyüz olan Rpmdrake’yi geliştirmiştir. Bu program grubu 4 ana modülden oluşur:

1. rpmdrake
2. rpmdrake-remove
3. edit-urpm-sources.pl
4. MandrivaUpdate

“rpmdrake”, RPM paketlerinin indirip kurulma işlevini yerine getirir. Adından da tahmin edilebileceği gibi, “rpmdrake-remove”, yüklenmiş RPM paketlerinin kaldırılması ve silinmesi işini yapar. “edit-urpm-sources.pl” Perl programı ise “rpmdrake” ve “MandrivaUpdate” nin kullanacağı yazılım depolarına ilişkin ayarların yapılmasını sağlar. “MandrivaUpdate” ise işletim sistemi ve uygulama yazılımlarının güncellenmesinden sorumludur.

Yazılım Depoları (Media, Repositories)

Mandriva LINUX işletim sisteminizi kurmak için kullandığınız dağıtım CD’lerinin içinde Mandriva teknik personelinin sevdiği, seçtiği yazılımlar gelecek ve onlar kurulacaktır. Oysa LINUX dünyasında çok ama çok daha zengin yazılım kütüphaneleri vardır. Olabilecek tüm yazılımları CD’lerde dağıtmak pratik olarak mümkün olmadığı için büyük LINUX dağıtıcıları, kendi sunucularında “yazılım depoları” (repositories) barındırırlar. Standart dağıtım dışında bir yazılıma gerek duyarsanız, bu depolardan bilgisayaraınıza indirip yükleyebilirsiniz.

Size en yakın yazılım deposu kurulum CD’lerinizdir. İnternet bağlantınızı kullanarak erişebileceğiniz, kullandığınız dağıtıma ilişkin depo ya da depolar bulmanız da çok kolaydır. Örneğin, son bir yıldan beri çok beğenilen Ubuntu dağıtımı tek bir CD ile yapılmakta, tüm diğer yazılımlar İnternet üzerinden “Synaptic” isimli paket yöneticisi ile yüklenebilmektedir.

Yazılım Güncelleme

LINUX işletim sisteminin ve yazılımlarının güncellemesi iki nedenle yapılır: ya bir güvenlik açığı kapatılmıştır, ya da yazılım geliştirildiği için yeni sürümü çıkmıştır.

Evet! Güvenlik açığı kapatılmış olabilir; sonuçta LINUX kodlarını yazarlar da insanlar... Şu farkla ki - LINUX dünyasında açıkların kapatılma ve güncelleme yayınlanma süresi aylarla değil, saatlerle ölçülür.

Mandriva sisteminizi güncellemek için “rpm-drake” isimli programı çalıştırdığınızda önce sisteminize yüklü olan paketlerin yazılım depolarında yeni sürümleri olup olmadığına bakılacak; varsa yeni sürümler kurulacaktır.

Yazılım Yükleme

Diyelim, bir dergide işinize çok yarayacak bir LINUX yazılımından söz edildiğini okudunuz. Hemen bir terminal penceresinden programı çalıştıracığını umduğunuz komutu verdiniz, ancak “command not found” mesajını aldınız. Büyük olasılıkla o program sisteminizde kurulu değil. Şimdi bu yazılımı sisteminize yüklemek için kolaydan zora doğru çeşitli seçenekleriniz var:

1. yazılımı paket yöneticisi kullanarak yüklemek,
2. varsa, yazılımı kendi özel kurulum programıyla yüklemek,
3. yazılımın rpm paketini bulup indirmek ve yüklemek,
4. yazılımın kaynak kodunun rpm paketini indirip derlemek ve kurmak,
5. yazılımının kaynak kodlarını indirip derlemek ve kurmak.

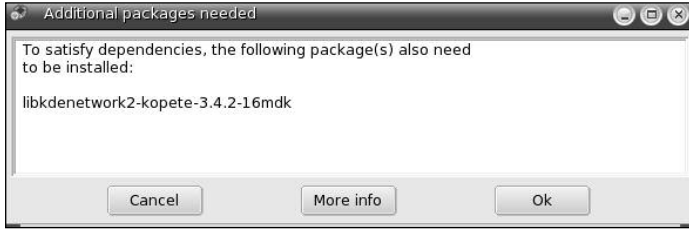
Kolayından başlayalım...

Mandriva rpm Paket Yöneticisi - rpm-drake ile Yükleme

KDE menüsünden System → Configuration → Packaging → Install Software seçin. Ardından arama kutusuna (Search) aradığınız programın adını veya adının bir kısmını yazıp (örneğin “kopete”) yazıp adında bu karakter dizisi geçen yazılım paketlerini aratın.



İlgilendiğiniz yazılım “KDENetwork-Kopete” olsun... Eğer 3.4.2 sürümünü yüklemek isterseniz, yazılımın adının yanındaki küçük kutuyu tıklayın. Kutucuğu tıkladığınızda karşınıza aşağıdakine benzer bir pencere çıkabilir:

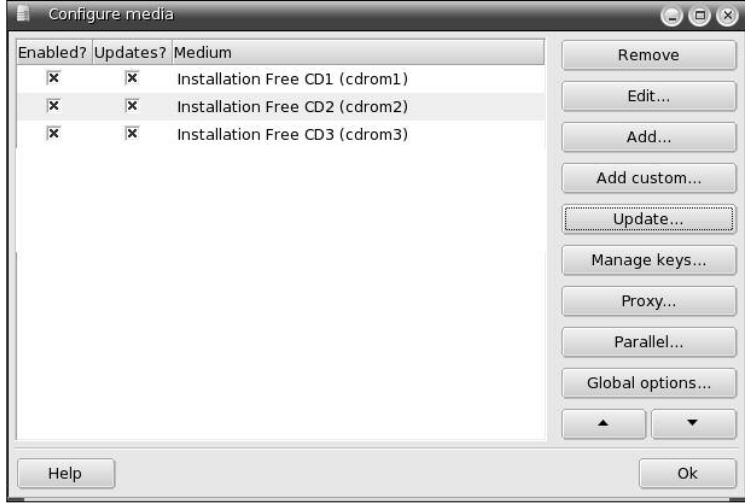


Bu örnekte, Kopete yazılımının yüklenebilmesi için “libkdenetwork2-kopete” kütüphanesinin de kurulması gerektiği belirtilmekte ve bunun için izniniz istenmektedir. Paket yöneticisi kullanmanın en önemli avantajı işte bu tip bağımlılıkların (dependency) otomatik olarak çözülmesidir.

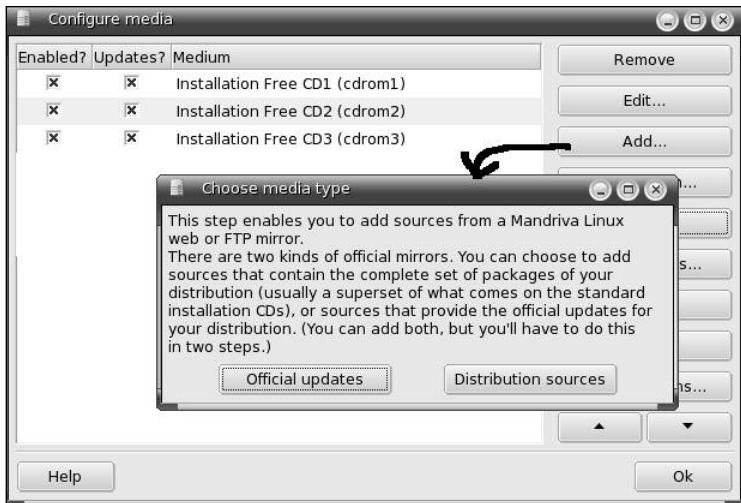
Rpmdrake paket yöneticisi, yeni yüklenecek, kaldırılacak ve en önemlisi güncellenecek paketleri, “media” adı verilen çeşitli kaynaklardan izler. “media” adı verilen bu kaynaklar, kurulum için kullandığınız CD’ler ve İnternet üzerinden erişilebilen Mandriva yazılım depolarıdır (İng.: Repositories).

Kim Korkar LINUX'tan?

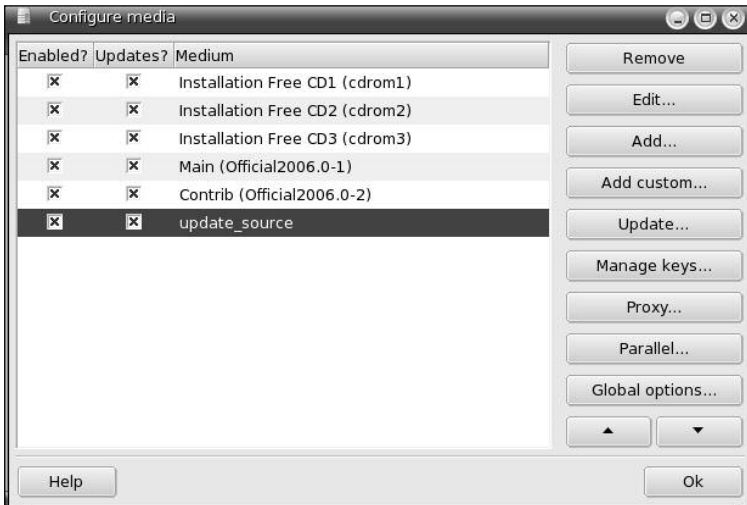
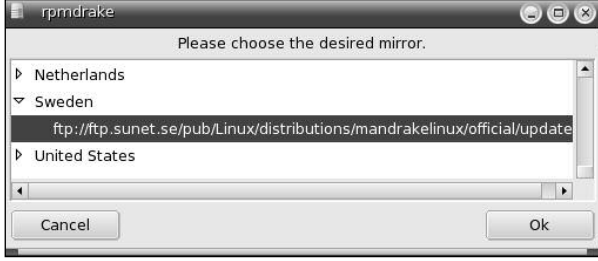
Yazılım ve güncelleme paketlerin indirilip yüklenebileceği kaynaklar rpmdrake yazılımının “Configure Media” işlevi ile belirtilir. CD’den yapılan standart bir kurulumda, bu yazılım kaynakları sadece 3 dağıtım CD’sinden oluşur.



Normal kullanım koşullarında yazılım deposu olarak bu 3 CD pek yeterli olmayacaktır; ne de olsa bu CD’lerden yazılım güncellemesi yapmak olası değildir. Bu yüzden paket yöneticimize güncellemeler ve CD’lerinizde yer almayabilecek yazılım paketleri için bir kaç yeni yazılım deposu; yani “media” eklemelisiniz.



rpmdrake için yeni yazılım deposu eklemek istediğinizde, “Official Updates” (resmi güncellemeler) ve “Distribution Sources” (dağıtım kaynakları) kategori seçimi yapmanız istenir. Her iki kategoriden de yazılım deposu seçmenizi öneririz. Örneğin;



rpmdrake için yazılım deposu tanımlarını bu şekilde tamamladıktan sonra artık yeni yazılımlar yüklemek; yüklü yazılımların sürüm ve güvenlik güncellemelerini yapmak çok kolaylaşacaktır.

urpmi - Gerçek Paket Yöneticisi

rpmdrake paket yöneticisi aslında “urpmi” isimli bir konsol uygulamasının grafik ön yüzüdür. Gerçek Mandriva paket yöneticisi “urpmi” yazılımıdır. LINUX yönetiminde deneyim kazandıkça birçok işi grafik arayüzlerle uğraşmadan, doğrudan terminal ekranından yapmayı tercih edeceksiniz. Terminal arabirimi ile kullanım, uzaktan yönettiğiniz sistemler söz konusu olduğunda daha da rahat çalışmanızı sağlayacaktır.

urpmi ile bir yazılım paketi kurmak istediğinizde; örneğin kapsamlı bir web sitesi geliştirme aracı olan “screem” kurmak isterseniz; bu işi rpmdrake kullanılarak grafik bir arayüz kullanarak yapabileceğiniz gibi root kimliğine sahip olduğunuz bir terminal penceresinden

urpmi screen

komutunu da verebilirsiniz. urpmi, screem programının kurulabilmesi için gerekli tüm paketleri tanımlanmış yazılım depolarından bulup indirecek ve doğru sırada kuracaktır.

```
root@0j08035.0jman.bilkent.edu.tr: /root - Shell - Konsole
Session Edit View Bookmarks Settings Help

# urpmi screem
To satisfy dependencies, the following 5 packages are going to be installed (9 MB):
dbus-x11-0.23.4-5mdk.i586
gtksourceview-1.2.1-1mdk.i586
libgnome-menu2-2.11.92-4mdk.i586
libgtksourceview-1.0_0-1.2.1-1mdk.i586
screem-0.14.1-5mdk.i586
Is this OK? (Y/n) y

installing /var/ftp/pub/Mandrivalinux/media/main/dbus-x11-0.23.4-5mdk.i586.rpm /var/ft
p/pub/Mandrivalinux/media/main2/gtksourceview-1.2.1-1mdk.i586.rpm /var/ftp/pub/Mandriv
alinux/media/main2/screem-0.14.1-5mdk.i586.rpm /var/ftp/pub/Mandrivalinux/media/main2/
libgtksourceview-1.0_0-1.2.1-1mdk.i586.rpm /var/ftp/pub/Mandrivalinux/media/main2/libg
nome-menu2-2.11.92-4mdk.i586.rpm
Preparing...
1/5: libgnome-menu2
2/5: gtksourceview
3/5: libgtksourceview-1.0_0
4/5: dbus-x11
5/5: screem
#
```

Özel Kurulum Programıyla Yükleme

Bazı LINUX yazılımları kendi özel kurulum programlarıyla birlikte dağıtılırlar. Bazıları da kurulum programına bile gerek olmadan, sadece program dosyasını bir dizine kopyalamakla kurulabilirler. Bu tip yazılımların nasıl kurulacağı, yazılımla birlikte dağıtılan

INSTALL.txt

README.TXT

benzeri isimli dosyalarda ya da yazılımın dağıtıldığı web sitesinde yer alabilecek kurulum bölümlerinde anlatılır. Bu kitapta özel kurulum yöntemlerini açıklamak pek olası değil çünkü bir yazılımın nasıl dağıtılacağı ve nasıl kurulacağı tamamen o yazılımı geliştiren kişi ya da ekibin tercihlerine bağlıdır.

Örneğin Mozilla-Firefox yazılımının 1.5.0.2 sürümünü *www.mozilla.com* adresinden indirdiğinizde, gene aynı web sitesinde “Releases” başlığı altında Firefox yazılımının nasıl kurulacağı şöyle anlatılmaktadır:

“Extract the tarball in the directory where you want to install Firefox:

```
tar -xzvf firefox-1.5.0.2.tar.gz
```

This will create a firefox subdirectory of that directory.”

Yani, firefox-1.5.0.2.tar.gz dosyasını /tmp dizinine indirdikten sonra

```
mkdir /usr/local/firefox  
cd /usr/local/firefox  
tar -xzvf /tmp/firefox-1.5.0.2.tar.gz
```

komutlarını peş peşe verdiğinizde kurulum tamamlanmış olacak ve artık istediğiniz zaman

```
/usr/local/firefox/firefox
```

komutuyla Firefox web tarayıcısını çalıştırabilecek duruma geleceksiniz.

Özel kurulum programı ile yüklenen yazılımlara bir örnek de Opera web ta-

Kim Korkar LINUX'tan?

rayıcısıdır. Bu yazılım sıkıştırılmış tar.gz dosyası ile dağıtılır. tar paketini açtıktan sonra paketin içinde yer alan **./install.sh** komutu ile Opera kurulur.

```
cd /tmp
tar -xvzf opera-8.54-20060330.5-shared-qt.i386-en.tar.gz
cd opera-8.54-20060330.5-shared-qt.i386-en/
./install.sh
. . .
opera
```

rpm Paketlerinden Yazılım Yükleme

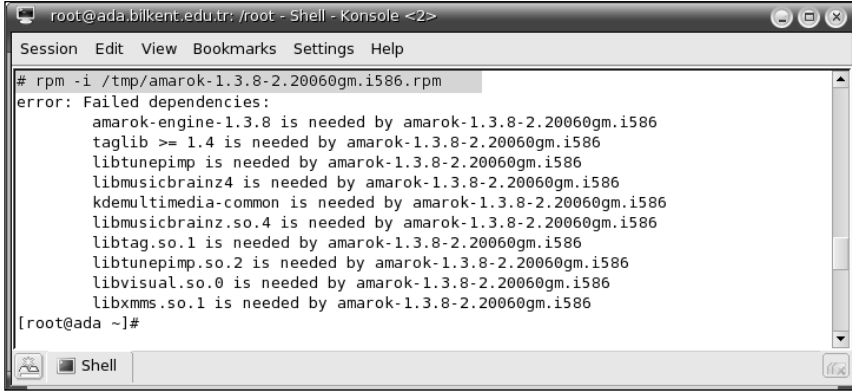
Bilgisayarınıza yüklemek isteyebileceğiniz bir yazılımı bazen yazılım depolarında (repository) bulamazsınız; bir başka deyişle; rpmdrake veya urpmi, yüklemek istediğiniz yazılımı, ayarlarında girilmiş olan yazılım depolarında bulamayabilirler. Böyle durumlarda, ilgilendiğiniz yazılımın rpm paketini başka kaynaklardan; örneğin yazılımı geliştirmiş olan kuruluş ya da şahsın web sitesinden, ya da *www.rpmfind.net* gibi rpm paketi arama servislerinden yararlanarak bulabilirsiniz.

İlgilendiğiniz yazılımın rpm paketini bilgisayarınızda geçici bir dizine (/tmp bu iş için çok uygundur) indirip

```
rpm -i /tmp/paket.rpm
```

gibi bir komutla yüklemeyi deneyebilirsiniz. “Denemek” diyoruz; çünkü bu paketin bağımlı olduğu başka paket(ler) varsa; rpm yazılımı yükleme işini yapmayıp; önce çözmeniz gereken bağımlılıklar olduğunu bildirecektir. Bazı yazılımlar rpm paketini indirip, **rpm -i paket.rpm** komutunu verdiğinizde tek seferde kurulabilir; bazı durumlarda da önce paketinizin bağımlı olduğu bir başka paketi; o başka paketi kurabilmek için ise daha da önce bir başka paketi kurmanız gerekebilir.

Örneğin *amarok.kde.org* sitesinden **amarok** yazılımının 1.3.8 sürümüne ait rpm dosyasını indirip rpm komutuyla kurmak istediğinizde



```

root@ada.bilkent.edu.tr: /root - Shell - Konsole <2>
Session Edit View Bookmarks Settings Help
# rpm -i /tmp/amarok-1.3.8-2.20060gm.i586.rpm
error: Failed dependencies:
  amarok-engine-1.3.8 is needed by amarok-1.3.8-2.20060gm.i586
  taglib >= 1.4 is needed by amarok-1.3.8-2.20060gm.i586
  libtunepimp is needed by amarok-1.3.8-2.20060gm.i586
  libmusicbrainz4 is needed by amarok-1.3.8-2.20060gm.i586
  kdemultimedia-common is needed by amarok-1.3.8-2.20060gm.i586
  libmusicbrainz.so.4 is needed by amarok-1.3.8-2.20060gm.i586
  libtag.so.1 is needed by amarok-1.3.8-2.20060gm.i586
  libtunepimp.so.2 is needed by amarok-1.3.8-2.20060gm.i586
  libvisual.so.0 is needed by amarok-1.3.8-2.20060gm.i586
  libxms.so.1 is needed by amarok-1.3.8-2.20060gm.i586
[root@ada ~]#

```

gibi amarok paketinin kurulabilmesi için önce 10 tane ek pakete gereksinim olduğuna ilişkin bir mesaj alabilirsiniz. Bu durumda yapılması gereken bu eksik paketlerin *rpm* dosyalarının İnternet’ten bulunup indirilmesi ve öncelikle bunların kurulmasıdır. Google bu konudaki en yakın yardımcınız olacaktır.

Açıkcısı, birçok kullanıcının LINUX işletim sistemini deneyip, kısa zamanda pes ederek terk etmesinin nedeni bu bağımlılık sorunlarıdır. Ancak, öte yandan, gereksinim duyabileceğiniz binlerce yazılım, urpmi ve/veya rpmdrake ile bu bağımlılık sorunlarını yaşamadan kurulabilecek şekilde yazılım depolarında bulunmaktadır. Yeni kullanıcıların genellikle yaptıkları hata “Software Media Manager” ile rpmdrake ve urpmi için yazılım depoları belirtmemeleri olmaktadır.

Sisteminize **rpm** ile yüklediğiniz programları gene **rpm** komutu ile silebilirsiniz. Örneğin **whois-4.5.29-1mdk.i586.rpm** isimli bir rpm dosyası kullanarak kurduğunuz **whois** programını silmek için

rpm -e whois

komutunu kullanabilirsiniz.

Sisteminize rpm programıyla yaptığınız bütün değişiklikler (LINUX kurulumu dâhil) bir veri tabanında tutulur. Böylece rpm kullanarak kurduğunuz programlar arasında koordinasyon sağlanabilir. Örneğin, bir program belli başka bir programın yeni bir versiyonu olmadan çalışmayacaksa rpm komutu kurulum sırasında sizi uyarır. Aynı şekilde sisteminizde zaten kurulu bu-

Kim Korkar LINUX'tan?

lunan bir programın daha yeni bir versiyonunu kurarken eskisinin silinmesi işi de temiz bi şekilde yürütülür.

rpm komutunun çok kullanılan bazı parametrelerine ve işlevlerine göz atmak isterseniz:

Sisteminizde kurulu olan tüm rpm paketlerini listelemek için:

rpm -qa

komutunu kullanmalısınız.

Bir rpm dosyasının paket adını öğrenmek için (aman dikkat! Bir rpm paketinin adı o paketin rpm dosyasının adı değildir) kullanabileceğiniz

rpm -qp opera-6.11-20021129.4-shared-qt.i386.rpm

gibi bir komut, dosya adı **opera-6.11-20021129.4-shared-qt.i386.rpm** olan rpm paketinin paket adının **opera-6.11-20021129.4** olduğunu gösterecektir.

Bir rpm paketinin içindeki dosyaları öğrenmek için:

rpm -qlp opera-6.11-20021129.4

Bir rpm paketinin ne işe yaradığını gösteren notları görmek için:

rpm -qip opera-6.11-20021129.4

rpm paketlerinin isimleri **whois-4.5.29-1mdk.src.rpm** gibi bir dendiğinde; yani içinde **.src.** dizisi geçiyorsa o rpm paket, programın/uygulamanın kaynak kodunu içeren rpm paketi demektir.



Kaynak kodundan oluşan programların kurulumu hazır derlenmiş paketlere göre göreceli olarak daha zordur. Söz konusu rpm paketleri açıldıktan sonra ilgili dizinlere yerleşen README, INSTALL gibi dosyaları güzelce okumak ve buralardaki işlemleri sırasıyla uygulamak gerekir. Sözün kısası: yüklemek istediğiniz paketin hazır derlenmiş sürümü varsa, onu kullanmayı tercih ediniz.

rpm dosya isimleri programın ismi ve sürümü dışında, aynı zamanda makine tipini de belirtir. Bir program indireceğiniz İnternet sayfasında aynı programın aynı sürümü için dosya isminin sonunda **i386**, **i586**, **noarch** gibi bazı uzantılar olan birkaç dosya görebilirsiniz. LINUX birçok platformda çalışan sürümlere sahip olduğu için ve rpm paketleri belli bir tip merkezi işlem birimi için önceden derlenmiş olduğundan kendi sisteminize uygun olan rpm'i indirmelisiniz. Alışılmış PC'ler için bu Intel firmasının i386 ve i586 platformudur. İsminde "noarch" geçen RPM paketleri rpm'in her donanım platforma uygun tek sürüm olarak dağıtıldığı anlamına gelir.

Kaynak rpm Paketlerinden (src-rpm) Yazılım Yükleme

Bilgisayarınıza yüklemek istediğiniz yazılımı sadece kaynak kodlarından oluşan rpm paketi olarak bulabiliyorsanız işiniz biraz zor demektir. Kaynak kodlarından yazılım yükleyebilmek için önce elinizdeki kaynak kod dosyalarının sorunsuz olarak derlenebilmesi gerekir. Bunun için bilgisayarınızda uygun derleyicinin, uygun kaynak kod kütüphanelerinin (standart ve özel *.h C dosyaları gibi) yüklü olması gerekir. İşin açıkçası, yeni başlayanlar için kaynak kod rpm paketlerini yüklemek bir kâbusa dönüşebilir.

Örneğin çeşitli GNU etkileşimli araçlar (GNU interactive tools) içeren "git" paketini kaynak rpm (source rpm) dosyasından kurmak için, önce "**git-4.3.20-9mdk.src.rpm**" dosyasını /tmp dizinine indiriniz ya da kopyalayınız.

Kim Korkar LINUX'tan?

Sonra

```
rpm --rebuild git-4.3.20-9mdk.src.rpm
```

Bu komut, “git” kaynak kod paketini açacak, gerekli tüm bağımlılık kontrollerini yapacak, herşey uygunsa kaynak kodları derleyip `/usr/src/RPM/RPMS/i586/` dizinine **git-4.3.20-9mdk.i586.rpm** paketini yaratacaktır. Artık

```
rpm -i /usr/src/RPM/RPMS/i586/git-4.3.20-9mdk.i586.rpm
```

komutuyla “git” paketini yükleyebilirsiniz.

Kaynak Kodundan Derleyerek Kurulum

LINUX yazılımları bazen derlenmiş, kurulumu hazır paketler yerine kaynak kodu paketleri olarak dağıtılır. Bu tip paketlerin kaynak kodlarını sizin derlemeniz ve kurmanız gerekir.

Kaynak kodu olarak dağıtılan paketlerin dosya isimleri genellikle. **src.tar.gz** veya **.src.tgz** diye biter ve bunlar birer “tar yumağı” dosyadır.

LINUX dünyasının, daha doğrusu Açık Kaynak Kodu (Open Source) dünyasının başarısının arkasındaki en önemli özelliklerden birinin bu yazılımların kaynak kodlarının serbestçe dağıtılması olduğunu söylemiştik. Bu nedenle, bir çok paket hem rpm paketi hem de kaynak koduyla dağıtılır.

Deneyimli kullanıcılar, yazılımları genellikle kaynak kodlarından kurmayı tercih ederler. Bunun en önemli nedeni sistemlerinin denetimini paket yöneticilerine bırakmamak; tüm güncellemeleri ve sürüm yükseltmelerini kendilerinin yapmak istemesidir.

“tarball” (tar yumağı) olarak da anılan bu dosyalar iki farklı program kullanılarak toparlanmıştır. Önce tar programı tarafından paketin içerdiği bütün dosyalar tek bir dosya haline getirilmiş, daha sonra gzip (GNU Zip) aracılığıyla sıkıştırılmıştır.



Açmak için siz de bu iki yöntemi tersine kullanmalısınız. Ancak tar ve gzip çok sık olarak birlikte kullanıldığından tar programına bir parametreyle (-z parametresi) gzip’lenmiş dosyayı açma yeteneği eklenmiştir.

```
tar -zxvf yeni_program.tar.gz
```

gibi bir komutla önce gzip’lenmiş açıp (z), sonra tar’ı açıp (x), bütün bunları yaparken açılan dosyaların listelenmesini sağlayabilirsiniz (v).

Genellikle basit bir programın bile **tar** yumağını açtığınızda karşınıza karmaşık, birçok izin ve dosyadan oluşan bir grup çıkacaktır. Bu kargaşanın içinden kolay çıkmanız için programı geliştiren(ler) size birkaç kolaylık da sağlamış olacaktırlar.

Kurulum için yapılması gerekenler genellikle **README** veya **INSTALL** gibi isimleri olan dosyalarda anlatılır. (Dosya isimlerinin büyük harf olmasının nedeni, **ls** tarafından üretilcek dosya listelerinde bu dosyaların isimlerinin üst sırada görünmesini sağlamaktır.)

Genellikle ilk adım olarak programın kaynak kodlarının sizin sisteminize uygun bir şekilde derlenmesini sağlamak için, paketle birlikte dağıtılan **configure** isimli bir komut kullanılır. Kurmakta olduğunuz programın karakteristik özellikleriyle ilgili (örneğin sistemde nereye kurulacağı veya hangi tip işlemci için optimize edileceği) bu komuta parametre olarak verilir. Olası parametreler ve değerleri için kurmakta olduğunuz yazılımın **INSTALL** ve/veya **README** dosyalarına bakmalısınız. Genellikle

```
cd /tmp/kaynak  
./configure
```

gibi bir komut yeterlidir. (Baştaki “./”, kullandığınız kabuk programına bulunduğunuz dizindeki **configure** isimli programı çalıştırmak istediğinizi belirtmek içindir.)

configure bilgisayarınızın özelliklerini tek tek kontrol eder, aynı zamanda başka programlara veya kütüphanelere ihtiyacınız olup olmadığını size bildirir. Bu aşama başarıyla sonuçlandığında kaynak kodunun bulunduğu dizinde (ve eğer varsa diğer alt dizinlerde) **Makefile** isimli bir dosya yaratır.

İkinci adım olarak vereceğiniz **make** komutu bu **Makefile** dosyasında listelenen parametrelere göre bütün programın ve modüllerinin derlenmesi için gerekli işlemleri sırayla yapacaktır. Bazı programları derlemek ve kurmak için birden fazla **make** komutunu peş peşe kullanmanız gerekebilir. (Böyle durumlarda vermeniz gereken **make** komutlarının neler olduğu **README** veya **INSTALL** gibi açıklama dosyalarında belirtilmiş olacaktır.)

Örneğin,

```
./configure  
make  
make test  
make install
```

oldukça sık rastlanan bir kurulum sürecidir.

İşletim Sisteminin ve Uygulama Programlarının Güncellenmesi

Tüm yaşayan yazılımlar gibi işletim sistemleri de sürekli gelişmektedir. Bu gelişme, yazılımlara yeni yetenekler eklenmesi ve hataların düzeltilmesi **ve en önemlisi, bulunan güvenlik gediklerini kapatan yeni sürümlerin yayınlanması** şeklinde olmaktadır. LINUX güvenlidir deyip durduk, genel kanı da zaten bu doğrultuda... Ancak hiçbir yazılım kusursuz değildir. LINUX'un "güvenliği ve güvenilirliği yüksek" yüksek bir işletim sistemi olarak tanınmasının en önemli nedeni bir modülde güvenlik gediği ya da hata bulunduğu anda düzeltilmiş sürümün (şaka değil) birkaç saat içinde yayınlanması ve dağıtımına başlanmasıdır.

Mandriva, hazırlayıp dağıttığı LINUX paketlerinin güncellenmesi için gereken hizmetleri verme görevini de üstlenmiştir. Aslında yalnızca Mandriva değil, genellikle tüm LINUX dağıtımcıları eşdeğeri hizmetleri ücretsiz olarak vermektedir.

İşletim sisteminizi ve onunla birlikte dağıtılan uygulama programlarını güncellemek istediğinizde, rpmdrake programını başlatmak için KDE menüsünden

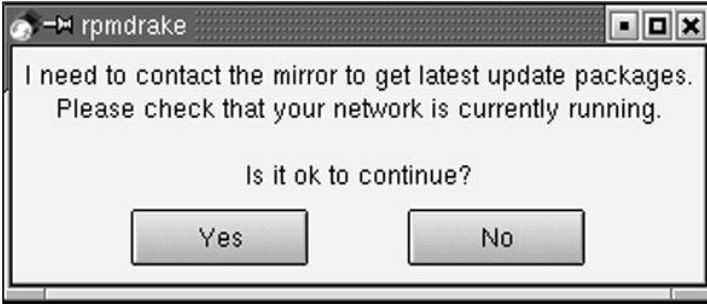
KDE SystemConfiguration → Packaging → Mandriva Update

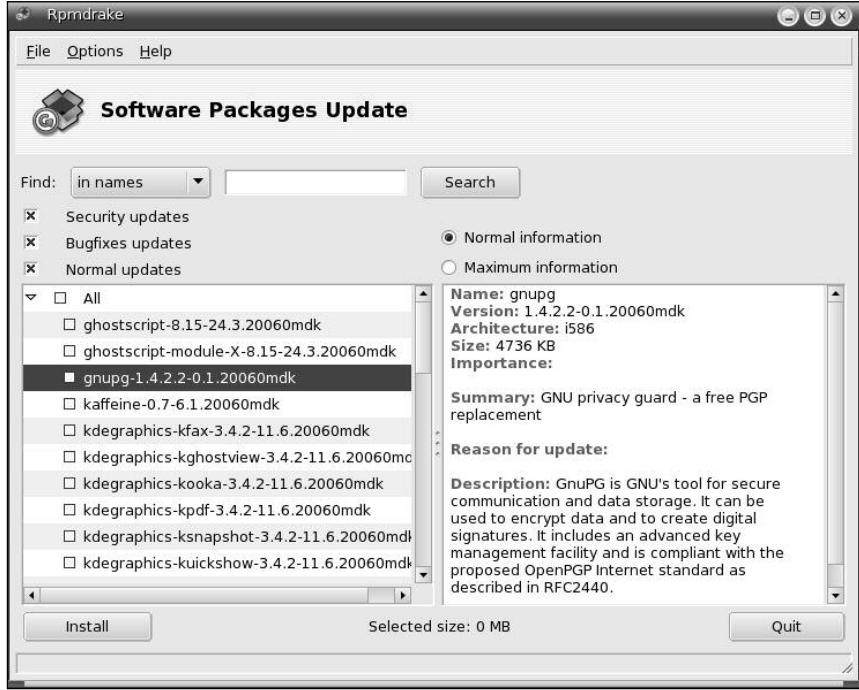
veya Mandriva Linux **Control Center** yazılımını başlatmak için

KDE → System → Configuration → Configure Your Computer

seçimini yaparak **Mandriva Update** programını başlatmalısınız. Bu programı başlatırken bilgisayarınızın İnternet bağlantısı çalışır durumda olmalıdır.

“Mandriva Update”, başlatıldığında önceden belirlenmiş Mandriva ve yansılarları olan sitelere (mirror) bağlanıp, sizin makinanızda yüklü olan yazılımlarla bunların güncel sürümlerini karşılaştıracak ve sürüm güncelleştirmeye aday yazılımların bir listesini üretecektir.





“Mandriva Update” güncelleştirmeleri üç grup altında toplar:

1. Güvenlik güncellemeleri (Security updates),
2. Hata düzeltme güncellemeleri (Bugfixes updates),
3. Normal güncellemeler.

İlgilendiğiniz gruptaki güncellemeleri seçmek ve uygulamak için “Mandriva Update” ekranındaki butonları kullanabilirsiniz. Uygulanacak güncelleme dosyasının seçilmesi için sağ tarafındaki küçük kutu içinde çarpı işareti görünecek şekilde tıklanmalıdır. Eğer bilgisayarınız sürekli internete bağlı olarak çalışacaksa ve önemli servisleri üstlenecekse, güvenlikle ilgili yazılım güncellemelerini en az haftada bir kez kontrol etmenizi ve yeni güncellemeler varsa derhal uygulamanızı öneririz.

Çekirdek Güncellemeleri (Kernel Updates)

Mandriva zaman zaman çekirdek güncellemeleri yayınlamaktadır. Bu güncellemeler genellikle güvenlik gediklerini kapatmaya yöneliktir. Bir de, çekirdek sürüm yükseltmeleri söz konusudur; ne de olsa LINUX yaşıyor, büyüyor ve gelişiyor.

Örneğin, Mandriva 2006'nın şu sıralardaki (Mayıs 2006) çekirdek sürümü 2.6.12'dir. Nerden mi bildik?

```
# uname -sr
Linux 2.6.12-12mdksmp
```

Siz de kendi çekirdek sürümünüzü öğrenmek için

```
uname -sr
```

komutunu kullanabilirsiniz.

Bu çekirdek adındaki “mdk” kodu çekirdeğin Mandrake (Mandriva'nın eski adı) tarafından hazırlandığını ve derlendiğini; “smp” kodu ise çekirdeğin “Symmetric Multi Processor”; yani çoklu işlemci için hazırlanmış bir çekirdek olduğunu belirtiyor. Aslında yukarıdaki çekirdek sürümünü rapor eden bilgisayar tek işlemcili, ancak bu işlemcinin “Hyper Threading” özelliğinden ötürü işletim sistemine iki işlemcili gibi görünüyor.

Mandriva'nın yayınladığı yeni bir çekirdek sürümü olup olmadığını öğrenmek için

```
# urpmi kernel-
```

komutunu kullanabilirsiniz.

Mandriva Update yazılımı ile işletim sisteminizi ve uygulama yazılımlarını güncellediğinizde, varsa, çekirdek güncellemeleri (kernel updates) uygulanmaz. Çekirdek güncellemeleri için sizin özellikle çekirdek güncellemelerini kontrol etmeniz ve varsa uygulamanız gerekir.



Kim Korkar LINUX'tan?

```
root@loj08035.lojman.bilkent.edu.tr: /root - Shell - Konsole
Session Edit View Bookmarks Settings Help
# urpmi kernel-
no package named kernel-
The following packages contain kernel-:
gnbd-kernel-
kernel-2.4.28.0.rc1.5mdk
kernel-2.6.12.12mdk
kernel-BOOT-2.6.12.12mdk
kernel-doc
kernel-enterprise-2.4.28.0.rc1.5mdk
kernel-i586-up-1GB-2.4.28.0.rc1.5mdk
kernel-i586-up-1GB-2.6.12.12mdk
kernel-i686-up-4GB-2.6.12.12mdk
kernel-multimedia-2.6.12-12.mm.1mdk
kernel-multimedia-enterprise-2.6.12-12.mm.1mdk
kernel-multimedia-i686-up-4GB-2.6.12-12.mm.1mdk
kernel-multimedia-source
kernel-multimedia-win4lin-2.6.12-12.mm.1mdk
kernel-multimedia-win4lin-enterprise-2.6.12-12.mm.1mdk
kernel-multimedia-win4lin-i686-up-4GB-2.6.12-12.mm.1mdk
kernel-smp-2.4.28.0.rc1.5mdk
kernel-smp-2.6.12.12mdk
kernel-source-2.6
kernel-source-stripped-2.6
kernel-uml
kernel-uml-2.4-2.4.22-2mdk
kernel-uml-2.6.10-2mdk
kernel-win4lin-2.6.12.12mdk
kernel-win4lin-i586-up-1GB-2.6.12.12mdk
kernel-win4lin-i686-up-4GB-2.6.12.12mdk
kernel-win4lin-smp-2.6.12.12mdk
kernel-xbox-2.6.12.12mdk
kernel-xen0-2.6.12.12mdk
kernel-xenU-2.6.12.12mdk
[root@loj08035 ~]#
```

Uzunca bir çekirdek listesi... Bu örnek listede özel amaçlı olmayan çekirdeklerden en son sürüm numarasına sahip olanı “kernel-2.6.12.12mdk”. “**uname -sr**” komutu da aynı sürüm numarasını rapor ettiğine göre çekirdek güncellemeye gerek yok. Ama, “urpmi kernel-” komutu, örneğin, kernel-2.6.23.3mdk” gibi bir çekirdek bulunduğunu rapor etseydi,

urpmi kernel-2.6.23.3mdk

gibi bir komutla çekirdeğinizi güncellemeliydiniz.

Çekirdek güncellemesinin aktif olabilmesi için bilgisayarınızı kapatıp açmanız; daha doğrusu işletim sistemini yeniden yüklemeniz gerekir.

LINUX’a alıştıkça, sisteminizi kapatıp açmak size zor gelecek. Başlangıçta eski alışkanlıklarla, her sorun çıktığında içinizden bilgisayarınızı kapatıp aç-

mak gelecek. Ancak bir süre sonra bunun normal olmadığını, donanımında sorun olmayan bir bilgisayarın sadece çekirdek güncellemelerinden sonra veya disk/bellek eklemek için kapatılması gerektiğine alışacaksınız. LINUX ve UNIX kullanıcıları, bilgisayarın kendileri için çalışması gereken bir makine olduğunu; kullanıcıların makine için çalışmasının normal olmadığını bilirler. Yıllardır bu işlerin içindeyiz; USB bellekteki fotoğrafları gösteren bir yazılım yükledikten sonra neden bir işletim sisteminin yeniden yüklenmesi gerektiğini anlayabilmiş değiliz.

Kim Korkar LINUX'tan?

- **Sistem Güvenliği**
 - **Kolay Tahmin Edilebilecek Şifre Kullanmayın ve Kullandırmayın**
 - **Sistem Güvenliği ile İlgili Haberleri İzleyin**
 - **Olabildiğinizde Güvenli İletişim Yapan Yazılımlar Kullanın**
 - **Güvenlikle İlgili Yamaları İzleyin ve Uygulayın**
 - **Gereksiz Yazılım Yükleme**
 - **Gereksiz Servisler Çalışmasın**
 - **Gereksiz Portlar Açık Olmasın**
 - **Bilgisayarınızdaki Tüm Servisler Dünyaya Açık Olmasın**
 - **Log Dosyalarınıza Bakın**
 - **Sisteminizi Yedekleyin**

Sistem Güvenliği

İnternet'in yaygınlaşmasıyla "bilgisayar sistem güvenliği" de önemli sorunlar listesinde önce üst sıralara; sonunda da en üste tırmandı. LINUX, sistem güvenliği açısından en şanslı işletim sistemidir, çünkü kodu herkese açıktır. Bu iddia çok kişi tarafından ciddiye alınmamakla birlikte son yıllarda yaşanan deneyimler iddiayı doğrulamıştır.

Kapalı kodlu bir işletim sisteminde bir güvenlik açığı bulunduğunda, düzeltme yamaları ya da yeni sürümün geliştirilmesi, duyurulması ve yayınlanması zaman almaktadır. Bu sürenin ayları bulabildiği görülmüştür. Oysa, benzeri bir durum LINUX işletim sisteminde ortaya çıktığında, yamalar veya yeni sürümler birkaç saat içinde dünyaya yayılmaya başlamaktadır.

Sisteminizin güvenliği için salt işletim sistemine güvenmek çok büyük hatadır. "LINUX işletim sistemi altında virüs olmaz", "LINUX güvenlidir, kim-

se kıramaz” gibi inanışlar tamamen yersizdir. Bal gibi virüs de bulaşır; sisteminiz de kırılır...

Sistem yöneticilerinin en önemli görevlerinden birisi de sistemin güvenliği ile ilgili çalışmaları disiplinli bir şekilde yapmaktır.

Kolay Tahmin Edilebilecek Şifre Kullanmayın ve Kullandırmayın

Adı üstünde; “şifre”... Eğer yapışkanlı sarı bir kağıda yazıp ekranın köşesine yapıştırırsanız ya da şifre diye adınızı veya “abc123”, “qwerty” gibi bir dizi kullanırsanız hiç şifre kullanmayın daha iyi.

Şifreleri kesinlikle e-postayla göndermeyin. Şifreleriniz kolay hatırlayabileceğiniz kadar anlamlı ama tahmin edilemeyecek veya sözlüklerde bulunamayacak kadar anlamsız olsun. En iyisi atasözü, şarkı adı gibi cümlelerin baş harflerinden ve noktalama işaretlerinden oluşan şifrelerdir. “**Hehmi!**” gibi... (Hayatta en hakiki mürşit ilimdir!).

Sistem Güvenliği ile İlgili Haberleri İzleyin

En başta, kendinize en az bir LINUX güvenlik sitesi bulmalısınız. Bu site(ler)deki haberlere hiç değilse iki-üç günde bir göz atmalısınız. Yeni ortaya çıkan bir güvenlik açığı varsa hemen mümkün olduğunca ayrıntılarını öğrenip, sisteminizde önerilen kontrolleri yapıp gerekli güncellemeleri uygulamalısınız. Google’da “linux, güvenlik, security” gibi sözcüklerle yapacağınız bir arama sizi istemediğiniz kadar çok kaynağa yönlerecektir. <http://security.metu.edu.tr/belge.php> ve <http://www.linuxsecurity.com> genellikle her türlü gereksiniminizi karşılayacak kaynaklar içermektedir.

Olabildiğince Güvenli İletişim Yapan Yazılımlar Kullanın

Ethernet ağlarını dinlemek (sniff etmek de denir) çok kolay olduğu için, ne kadar iyi şifre seçip kullansanız da güvende olamazsınız. Bu hat dinleme sorununa bir çözüm olarak TCP/IP protokolüne **SSL** (*Secure Socket Layer*) özelliği eklenmiştir. SSL kullanan protokoller, her bağlantıda sunucu ile istemciyi bir kriptolama sistemi üzerinde anlaştırıp, ağ arabirime bastıkları tüm paketlerin bu sisteme göre kriptolanmasını sağlarlar. Genellikle RSA

(Rivest, Shamir, Adleman) adı verilen teknikle kriptolanan bu paketleri çözmek olanaksız değilse de pratik zaman sınırları içinde çözülemezler.

Bilgisayarınıza “telnet” kullanarak erişmeyin; onun yerine “ssh” kullanın. Ssh, tüm terminal haberleşmesinin kriptolu yapılanmasını sağlayacaktır.

Güvenlikle İlgili Yamaları İzleyin ve Uygulayın

Tüm LINUX dağıtım kuruluşları gibi Mandrake de zaman zaman güvenlik yamaları yayınlamaktadır. Her hafta en az bir kere “Mandrake Control Center” altındaki “Software Manager” ile güvenlik yamalarını sorgulayıp, gerekli gördüklerinizi; hatta hepsini uygulayın. Her olasılığa karşı bu işleri sisteminizin yoğun kullanılmadığı zamanlarda; örneğin Cuma günleri akşam saatlerinde yapın. *“Cuma akşamları yama aramakla uğraşamam; arkadaşlarla eğlenmeye gideceğiz”* diyenlerdenseniz siz sistem yönetici olamazsınız; en azından “iyi” bir sistem yöneticisi olamazsınız. Sistem yöneticileri gezmezler, eğlenmezler (daha doğrusu sistemlerinin başında daha mutludurlar); herkes gibi geceleri uyumazlar; normal yemek yemezler; organizmaları kahveden protein, karbonhidrat gibi yaşamsal maddeleri soğurma yeteneğini geliştirmiştir.

Gereksiz Yazılım Yüklemeyin

Kesinlikle gerekli olmayan yazılımları sisteminize yüklemeyin; hele sisteminiz önemli servis(ler) veriyorsa... Kaynağı belli olmayan yazılımlardan uzak durun.

Gereksiz Servisler Çalışmasını

Sisteminizde kullanmayacağınız servisleri kapatın. Örneğin sisteminizden bir web sunucusu olarak yararlanmayacaksanız, **httpd** çalışmasın; hem boş yere bellek harcar hem de gereksiz risk almış olursunuz. Sisteminizin açılışı sırasında başlatılacak servisleri “Mandrake Control Center - System” altındaki “DrakXServices” yazılımıyla görsel olarak seçebilirsiniz.

Kim Korkar LINUX'tan?

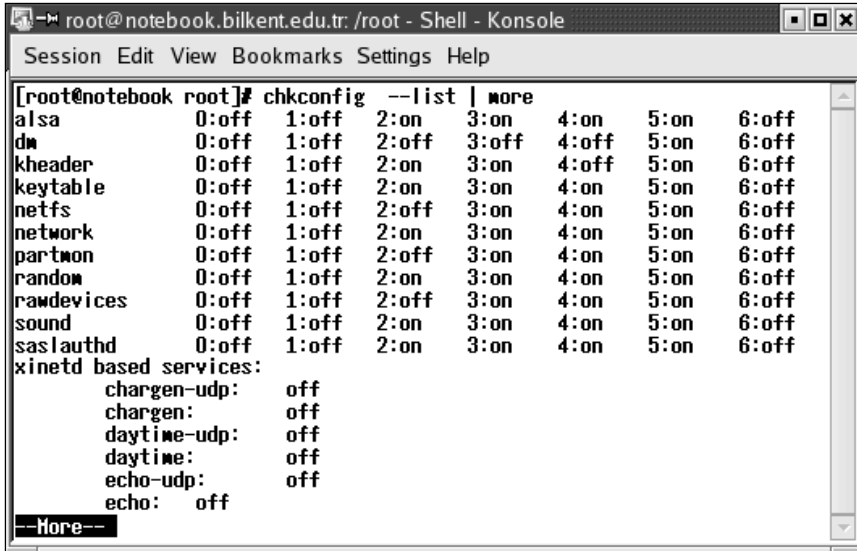
Bu işleri konsoldan yapmayı yeğlerseniz

chkconfig

komutunu kullanabilirsiniz.

chkconfig --list

komutuyla, **chkconfig** size sisteminizde çalışan servisleri, her çalışma düzeyi için (run level) ayrı ayrı belirtecektir.



```
[root@notebook root]# chkconfig --list | more
alsa          0:off  1:off  2:on   3:on   4:on   5:on   6:off
dm            0:off  1:off  2:off  3:off  4:off  5:on   6:off
kheader       0:off  1:off  2:on   3:on   4:off  5:on   6:off
keytable      0:off  1:off  2:on   3:on   4:on   5:on   6:off
netfs         0:off  1:off  2:off  3:on   4:on   5:on   6:off
network       0:off  1:off  2:on   3:on   4:on   5:on   6:off
partwon       0:off  1:off  2:off  3:on   4:on   5:on   6:off
random        0:off  1:off  2:on   3:on   4:on   5:on   6:off
rawdevices    0:off  1:off  2:off  3:on   4:on   5:on   6:off
sound         0:off  1:off  2:on   3:on   4:on   5:on   6:off
saslauthd     0:off  1:off  2:on   3:on   4:on   5:on   6:off
xinetd based services:
  chargin-udp: off
  chargin:     off
  daytime-udp: off
  daytime:     off
  echo-udp:    off
  echo:        off
--More--
```

Örneğin

mysql **0:off 1:off 2:on 3:on 4:on 5:on 6:off**

satırı **mysql** veritabanı sunucusunun 0,1 ve 6 çalışma düzeylerinde çalışmayacağını, ama 2, 3, 4 ve 5. düzeylerde çalışır durumda olacağını (olduğunu) gösterir.

Meraklısına...



Daha önce dördüncü LINUX çalışma düzeyinin kullanılmadığını söylemiş-tik. Hangi çalışma düzeyinde hangi yazılımların çalıştırılacağı **/etc/init-tab** dosyasından denetlenir. Eğer kendi gereksinimleriniz için özel bir dü-zey tanımlamak ve bu düzeyde birtakım yazılımları çalıştırmak isterseniz **/etc/inittab** dosyasında gerekli değişiklikleri yaparak kendi sisteminize özgü dördüncü düzey tanımınızı yapabilirsiniz.

Çalışma düzeylerini şöyle bir hatırlatmak gerekirse...

0	Kapanış düzeyi. Bu düzeyde pek çalışan program bulamazsınız...
1	Tek kullanıcı düzeyi. Sistem, gerek ağ üzerinden gerekse konsolundan birden fazla kullanıcıya hizmet vermez.
2	Çok kullanıcı düzeyi ama ağ üzerinden izin/dosya paylaşımına izin verilmez (NFS yoktur).
3	Sistemin tam kapasite ile çalıştığı çok kullanıcı düzeyidir
4	Pek kullanılmaz
5	XFree86 pencere sisteminin çalıştığı düzeydir
6	"Reboot" düzeyidir. Yani, sistemi kapatan yazılımların çalıştığı düzeyidir.

Sisteminizde **mysql** servisine gereksiniminiz yoksa

chkconfig mysql off

komutuyla servisin bir dahaki açılışta ve sonrasında başlatılmasını önleyebi-lirsiniz.

mysql veritabanı sunucusunu kullanmaya başladığınızda, açılışlarda kendi-liğinden başlaması için

chkconfig mysql on

komutunu kullanabilirsiniz.



chkconfig komutuyla birtakım servislerin otomatik olarak başlatılmasını sağladığınızda (--add seçeneği) ya da engellediğinizde (--del seçeneği), komutunuzun etkisi ancak sistemin bir dahaki açılışında görülür.

Yaptığınız değişikliğin etkisini hemen görmek istediğinizde; örneğin mysql'in hem hemen durdurulmasını hem de bir dahaki açılışta çalışmasını sağlamak için

```
/etc/rc.d/init.d/mysql stop  
chkconfig mysql off
```

komutlarını kullanmalısınız.

chkconfig komutunun aslında tek yaptığı iş, **/etc/rc.d/init.d** dizinin-deki açılış denetim komut dizelerini (*script*) düzenlemektir.

Gereksiz Portlar Açık Olmasın

Bildiğiniz gibi (daha doğrusu bilmeniz gerektiği gibi) TCP/IP iletişiminin temelinde “port” kavramı yatar. İnternet hatları üzerinden bilgisayarınıza ulaşan veri paketlerinin hangi yazılım tarafından karşılanacağını paketin içindeki port bilgisi belirler. Örneğin, 23 numaralı port genellikle “telnet” servisi ile ilgilidir. Yani, sisteminize varış port numarası (destination port) 23 olan bir veri paketi ulaştığında, bu paket “telnet” sunucu yazılımına, yani **in.telnetd** yazılımına iletilir.

Hangi port numarasını hangi yazılımın karşılayacağına ilişkin tanımları **/etc/services** dosyasında bulabilirsiniz.

```

root@notebook.bilkent.edu.tr: /root - Shell - Konsole
Session Edit View Bookmarks Settings Help

[root@notebook root]# more /etc/services
# /etc/services:
# service-name port/protocol [aliases ...] [# comment]

tcpmux      1/tcp                # TCP port service multi
plexer
tcpmux      1/udp                # TCP port service multi
plexer
rje         5/tcp                # Remote Job Entry
rje         5/udp                # Remote Job Entry
ftp         21/tcp               fsp fspd
ftp         21/udp
ssh         22/tcp                # SSH Remote Login Proto
col         22/udp                # SSH Remote Login Proto
col
telnet      23/tcp
telnet      23/udp
# 24 - private mail system
smtp        25/tcp                mail
smtp        25/udp                mail
time        37/tcp                timeserver
time        37/udp                timeserver

--More--(6%)

```

TCP ve UDP portlarının ne demek olduğunu bilmiyorsanız en kısa zamanda TCP/IP temellerini öğrenmeniz gerekir. Açıkçası, TCP/IP bilmeden sistem yöneticisi olma şansınız pek yok! TCP/IP öğrenmek için Murat Yıldırımoğlu'nun Pusula Yayıncılık tarafından yayınlanan *TCP/IP* isimli kitabından (ISBN: 975-7092-25-8) yararlanabilirsiniz.

Bilgisayarınızdaki Tüm Servisler Dünyaya Açık Olmasın

İnternet üzerinde yer alan bilgisayarlar üzerindeki TCP/IP trafiğini denetim altına almak için kullanılan yazılımlara “Ateş duvarı, Firewall” denir. Bu yazılımların temel görevi sisteme giren ve sistemden çıkan tüm ağ paketlerini inceleyip, paketleri sistem yöneticisinin direktifleri doğrultusunda işlemektir. Örneğin, bir ateş duvarı yazılımı, **192.168.13.*** gibi adreslerden gelen telnet paketlerine izin verip bunun dışındaki IP bloklarından gelen tüm telnet paketlerini reddedebilir.

Sisteminizi dışarıya karşı korumak için LINUX’da pek çok araç vardır. Bu araçlardan en yaygın olanı “**iptables**” adıyla bilinen filtre yazılımıdır.

TCP Wrapper, temel olarak bilgisayarınıza hangi bilgisayarların, hangi servislerle erişebileceğini ya da erişemeyeceğini belirlemenizi sağlar. TCP Wrapper, yalnızca sisteminize giren paketleri denetlemek için işe yarar; sisteminizden çıkan paketler bu yazılımla denetlenemez.

TCP Wrapper, **/etc/hosts.allow** (sisteme erişmesine izin verilecek bilgisayarlar) ve **/etc/hosts.deny** (sisteme erişmesine izin verilmeyecek bilgisayarlar) dosyalarıyla denetlenir. Bu dosyalar **vi** ile düzenlenebilecek basit dosyalardır. Dosyalarda yapacağınız değişiklikler hemen etkili olur; yani, herhangi bir yazılımı; hele hele işletim sistemini yeniden başlatmanız gerekmez.

TCP Wrapper'ın denetim mantığı şöyledir:

1. Sisteminize bir TCP/IP paketi geldiğinde, port numarasına bakılarak hangi yazılıma iletileceğine karar verilir.
2. Paket eğer **/etc/services** dosyasında tanımlı bir servise gönderilecekse, önce **/etc/hosts.allow** dosyası taranarak paketin özelliklerine uygun bir tanım olup olmadığına bakılır. Örneğin gelen bir telnet paketiye ve **/etc/hosts.allow** dosyasında "**in.telnetd: 139.179.14.: ALLOW**" gibi bir satır varsa ("139.179.14. ile başlayan bir IP adresinden gelmek kaydıyla, tüm telnet paketlerini kabul et" anlamında), paket kabul edilir.
3. **/etc/hosts.allow** dosyasında, gelen pakete uygun bir satır bulunamazsa, benzeri bir tarama **/etc/hosts.deny** dosyasında tekrarlanır. Bu dosyada, sisteminize erişmesini istemediğiniz bilgisayarlar tanımlıdır.
4. Eğer bu dosyada da gelen pakete uygun bir kural kalıbı bulunamazsa, paketin içeri girmesine izin verilir.

hosts.allow ve **hosts.deny** dosyalarında yer alabilecek kalıp tanım satırlarının genel formatı:

servis: IP alan tanımı

şeklinde dir. Örneğin **/etc/hosts.allow** dosyasında yer alan

in.telnet.d: 139.179. ALLOW

gibi bir satır, IP adresi "139.179." ile başlayan bilgisayarlardan gelen telnet paketlerini kabul edecektir.

/etc/hosts.deny dosyasında yer alabilecek

in.telnet.d: 192.168.100.12 DENY

gibi bir satır, IP adresi 192.168.100.12 olan bilgisayardan gelecek telnet servisi isteklerine olumsuz yanıt verilmesini sağlayacaktır.

Genellikle, sağlamcı bir politika izlemek amacıyla **/etc/hosts.deny** dosyasında tek satır olur:

ALL: ALL:

Yani, “nereden gelirse gelsin, hiçbir **xinetd** servisi isteğini kabul etme!” Sisteme erişmesine izin verilecek bilgisayarlar da **/etc/hosts.allow** dosyasında belirtilir.

Her iki dosyada da “#” ile başlayan satırlar açıklama satırlarıdır.

hosts.deny ve **hosts.allow** dosyalarına tipik birer örnek vermek gerekirse:

/etc/hosts.deny	/etc/hosts.allow
ALL: ALL:	in.telnetd: 139.179.14. ALLOW ALL: 139.179.14.41 ALLOW ALL: 139.179.23. EXCEPT 139.179.23.45



TCP Wrapper ile yalnızca **xinetd** üzerinden verilen servisleri denetleyebilirsiniz. O yüzden **hosts.allow** ve **hosts.deny** dosyalarıyla sisteminizin trafiği üzerinde tam denetim sağlayamazsınız. Örneğin bu araçla bilgisayara-
rınıza yönelik web trafiğini (http, 80. port üzerinden gelen trafik) denetleye-
mezsiniz çünkü http, xinetd'nin denetlediği bir servis değildir. Benzeri şekil-
de ftp sunucusu olarak pro-ftpd çalıştırıyorsanız (-ki, standart kurulumda
böyle olacaktır), gene erişim denetimini TCP Wrapper ile yapamazsınız.

Sisteminize yönelik ve sisteminizden kaynaklanan trafiği tam olarak dene-
tim altına almak için "**iptables**" veya "**ipchains**" ateş duvarı yazılımını
kullanmalısınız. Son yıllarda "**iptables**" daha popüler bir yazılım olarak
öne çıkmaktadır.

Eğer sisteminizde esnek bir trafik denetim sistemi kurmak istiyorsanız
"**shorewall**" paketini kurmanızı öneririz. **shorewall**, **iptables** üzerine ku-
rulmuş bir filtreleme sisteminin denetim yazılımıdır. "**shorewall**"u kurmak
için Mandrake Control Center yazılımından "Security" seçimini yapıp
"DrakFirewall" programını başlatınız. "shorewall" yazılımının ayarları en
kolay, web tabanlı bir sistem yönetim aracı olan "webmin" ile yapılır.

Log Dosyalarınıza Bakın

LINUX işletim sisteminin seyir defteri olan log dosyalarına sık sık göz atma-
nız önemlidir. Özellikle de **/var/log/security** dosyası...

Log dosyaları genellikle çok sayıda ve karmaşık satırlardan oluşur.
"**more xyz.log**" gibi komutlarla listelenip gözle kontrol edilmeleri zor-
dur. Webmin servisinin "System – System Logs" seçimiyle daha kolay iz-
lenebilecek log listeleri alabilirsiniz.

Sisteminizi Yedekleyin

Başkalarına ait bilgisayarlara girmek nedense çok sayıda hasta ruhlu insan
için bir tutkudur. Bu tip insanlar, bir bilgisayarın güvenliğini kırmayı başar-
dıklarında bu zaferlerini kutlamak isterler; bu yüzden de sistemleri tamamen
çökertmek yerine "başarı"larını belgeleyen bir işaret bırakmayı yeğlerler.
Sisteminize girildiğini hissettiğinizde yapabileceğiniz en akıllıca şey diskleri
formatlayıp işletim sistemini baştan yüklemek olacaktır. Bu işin kolay bölü-

müdür; öte yandan iyimser bir bakış açısıyla da sürüm güncellemek için iyi bir fırsattır. Ancak; iş daha önce yapılmış ayarları, yüklenmiş uygulama programlarını, tanımlanmış kullanıcıları, onların kişisel dosyalarını yerine koymaya gelince işiniz zor olacaktır. Disiplinli ve dikkatli bir şekilde yedeklenmiş bir sistemde bu dosya/dizinleri yerine koymak zaman alsa bile kolayca yapılabilir. Napolyon sistem yöneticisi olsaydı, eminiz ki “*para, para, para*” yerine “*yedek, yedek, yedek*” derdi.

BUNLARI BİLİYOR MUYDUNUZ?

Kaç İşlemci?

UNIX ve türevi işletim sistemlerini oluşturan yazılımların en önemli özellikleri, her birinin “kendi işini, ama yalnızca kendi işini çok iyi ve hızlı yapan”, esnek ama gereksiz işlevleri olmayan, “küçük” programcıklar olmalarıdır. Bu yazılımlar; anlamlı olabildiği her durumda birer “filtre” olarak yazılmıştır. “Filtre” olarak kullanılabilen programlar girdilerini STDIN'den alan, bu girdileri işleyip çıktılarını da “STDOUT” a gönderen programlardır. Bu sayede, bu programları “pipe” (|) ve “yönlendirme” (< , >) işlemleri ile peşpeşe ya da birlikte çalıştırılarak gereksinimlere göre anlamlı işler yapabilen zincirler oluşturulabilir.

UNIX'in çok önemli bir başka tasarım ilkesi de; olabildiğince, tüm verilerin, ayar/seçenek değerlerinin basit metin dosyalarında saklanması ve işlenmesidir.

Yararlı LINUX Sunucu ve Servis Yazılımları

13

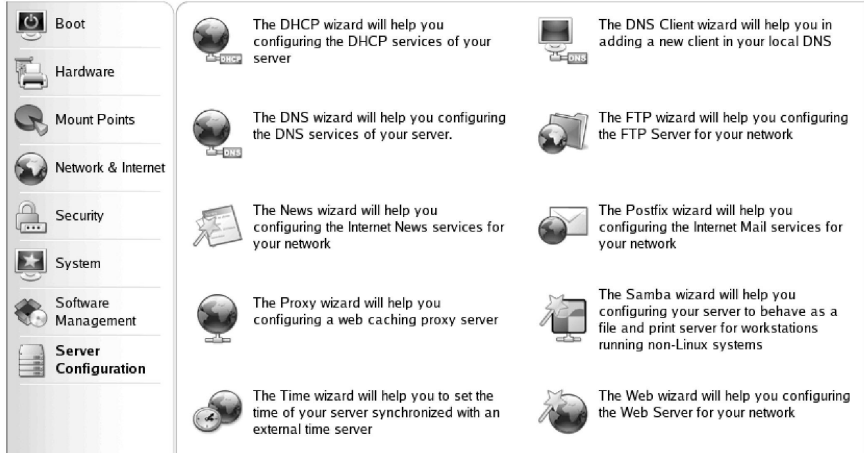
- **Yararlı LINUX Sunucu Yazılımları**
 - **ftp Sunucusu**
 - **apache Web Wunucusu**
 - **postfix e-Posta Sunucusu**
 - **procmail**
 - **samba Sunucusu**
 - **named (DNS) Sunucusu**
 - **ssh Sunucusu (Secure Shell)**
 - **NIS Sunucusu (Network Information Services)**
 - **iptables Ateş Duvarı**
 - **DHCP Sunucusu**
 - **MySQL ve PostgreSQL Veritabanı Sunucuları**
 - **squid Proxy Sunucusu**
 - **ppp Çevirmeli Ağ Sunucusu**

Mandrake LINUX'un, çeşitli sunucu yazılımlarını denetlemeyi kolaylaştıran “**Server Configuration**” isimli bir GUI (Graphical User Interface : Grafik Kullanıcı Arabirimi) aracı vardır; ancak bu araç siz özellikle kurmadıkça “Mandrake Control Center” menüsünde yer almaz. Her ne kadar bu bölümde “Server Configuration” yazılımının kullanımını anlatmayacaksak da, şimdi bu yazılımı yüklemenizde yarar var.

“**Server Configuration**”, Mandrake dağıtım CD'lerinizin ilkinde yer almaktadır. Yazılımı yüklemek için birinci CD'yi takıp

```
rpm -ivh /mnt/cdrom/RPMS/drakwizard-1.8-1mdk.noarch.rpm
```

komutunu verin. “Server Configuration” yazılımı kurulacak ve bundan sonra Mandrake Control Center yazılımını çalıştırdığınızda seçenekler arasında görünecektir:



Artık bu yeni araçla oynayıp yeteneklerini keşfetmek size düşüyor. Biz, önemli sunucu yazılımlarına ve bunlarla neler yapabileceğinize dönelim.

Yararlı LINUX Sunucu Yazılımları

Mandrake ve diğer tüm LINUX dağıtımları içinde son derece güçlü, yetenekli bir çok sunucu yazılımı yer almaktadır. Bunların neredeyse herbiri birer kitap yazılmasını gerektirecek kadar kapsamlı yazılımlar olduğu için burada yalnızca işlevleri hakkında kaba bilgi vermekle yetineceğiz. LINUX'unuzla yapmak istediğiniz işler için yararlı olacak yazılımları ayrıca çalışmanız ve öğrenmeniz gerekecektir. Standart kurulumlarda, burada sözünü ettiğimiz sunucu yazılımların hepsi kurulmaz; bazılarını ayrıca özel olarak kurmanız gerekecektir. Bu sunucu yazılımların çoğunun ayarları **webmin** ile yapılabilmektedir.

ftp Sunucusu

"File Transfer Protocol", Internet protokolleri arasında en önemlilerinden; daha doğrusu en çok kullanılanlarından birisidir. TCP/IP ağlarda (bir başka deyişle: İnternet'te) bilgisayarlar arası dosya transferinde kullanılır. Uygun FTP istemci programlarıyla (**ncftp**, **gftp** gibi) bir FTP sunucusu ile yetkileriniz doğrultusunda iki yönlü dosya transferi yapabilirsiniz.

Bilgisayarınızda **proftpd** sunucu yazılımı çalışıyorsa FTP istemcileri size dosya gönderebilir, sizden dosya çekebilir. Sizin bilgisayarınızda hesabı olan

kullanıcılar bir FTP istemci yazılımıyla sisteminize bağlandıklarında, kendi kullanıcı yetkileri çerçevesinde dosya çekip gönderebilirler. Eğer bilgisayarınızda hesabı olmayan kullanıcıların da bilgisayarınızdan dosya çekip göndermelerine izin vermek istiyorsanız “**anonymous**” (kimliği belirsiz) kullanıcıların erişimine izin vermeniz gerekir.

FTP sunucunuzu bir “anon ftp sunucusu” olarak kurmak istiyorsanız **/etc/proftpd.conf** konfigürasyon dosyasını aşağıdaki gibi düzenleyip “**/etc/rc.d/init.d/proftpd restart**” komutuyla FTP sunucu yazılımını durdurup tekrar başlatmalısınız.

```

ServerName                "Benim Sunucum"
ServerType                 standalone
DefaultServer             on
Port                      21
Umask                     022
MaxInstances              30
User                      nobody
Group                     nogroup
<Directory /*>
  AllowOverwrite           on
</Directory>
<Anonymous ~ftp>
  User                    ftp
  Group                   ftp
  UserAlias               anonymous ftp
  MaxClients              10
  DisplayLogin            welcome.msg
  DisplayFirstChdir       .message
  <Limit WRITE>
    DenyAll
  </Limit>
</Anonymous>

```

Bu arada bir hatırlatma yapmadan geçemeyeceğiz: Eğer FTP sunucu yazılımınız olan **proftpd** daemon’u sisteminiz açıldığında otomatik olarak çalıştırılmıyorsa

“**chkconfig proftpd on**” komutuyla bu sorunu halledebilirsiniz.

Bundan sonra yapmanız gereken küçük bir iş daha var: **/etc/passwd** dosyasına bakarsanız burada “**ftp**” diye bir kullanıcının tanımlı olduğunu fakat bu kullanıcının kabuk programının “**/bin/false**” olduğunu göreceksiniz. “**/bin/false**” aslında bir kabuk değildir; başlatıldığında hemen duran bir programdır. Genellikle kullanıcıların sisteme telnet ve ssh istemcileriyle bağlanmalarını önlemek için kullanılır.

“ftp” isimli kullanıcının sisteminize bağlanmasını istemeyeceğiniz; ama ftp ile dosya alıp vermesine izin vermek isteyeceğiniz için **/etc/passwd** dosyasındaki kabuk tanımını değiştirmeden **/etc/shells** dosyasında **/bin/false**’un kabul edilebilir bir kabuk olduğunu belirtmelisiniz. Bu işi vi ile **/etc/shells** dosyasına, içinde “**/bin/false**” olan bir satır ekleyerek yapabilirsiniz. (Bakın! Sonra uyarmadılar demeyin... vi öğrenmeden olmaz!)

apache Web Sunucusu

Dünyanın en iyi, en gelişmiş, en güvenli ve en yetenekli web sunucusu Apache’dir. Bu yalnızca bizim fikrimiz değil. *survey.netcraft.com* adresine bir göz atarsanız dünyadaki web sunucularının yüzde yetmiş yakın bir bölümünün Apache ile servis verdiğini göreceksiniz. MS-IIS’de fena değil aslında ama bir de güvenlik ve performans sorunları olmasa... IIS, yalnızca web uygulamalarını ASP ile yazmakta ısrar edenler için anlamlı. Ehh.. kendileri bilir. (Gene sataşmadan duramadık.)

Apache web sunucusu, kendinden önceki web sunucu yazılımlarına saygıdan olsa gerek “**httpd**” (hyper text transfer protocol daemon) adıyla kaydedilmiş bir program dosyasıyla çalışır. Bu nedenle sisteminizde çalışmakta olan süreçlerin listesini aldığınızda, içinde “**apache**” geçen bir süreç göremezsiniz. Apache sunucunuz çalışırken

```
ps ax | grep httpd
```

komutunu verirsiniz birden fazla **httpd** süreci çalıştığını göreceksiniz. Bu normaldir. Apache sunucusu, başlatıldığında, gelebilecek web isteklerini ayrı ayrı süreçlerle karşılayabilmek için kendisinin bir kaç kopyasını birden başlatır. Web istekleri artarsa, gerektiği kadar kendi kopyasını başlatır (UNIX dünyasında bu kavrama “spawning” denir). Süreç numarası en küçük olan **httpd** sürecini öldürürseniz, tüm **httpd** süreçleri ölür ve bilgisayarınız artık web servisi vermez.

Bilgisayarınızdan web servisini başlatmak için sabırsızlanıyorsanız **/var/www/html** dizinine uygun bir **index.html** dosyası yerleştirerek hemen yayına başlayabilirsiniz. Web sitenizle ilgili tüm html dosyaları bu dizinde yer almalıdır. **CGI** (Common Gateway Interface), yani web uygulama yazılımlarınız varsa onları da **/var/www/cgi-bin** dizinine yerleştirebilirsiniz.

Apache, hakkında bunun gibi bir kitap daha yazılması gereken bir yazılımdır. Ayarları oldukça karmaşık olabilmektedir. Ancak, basit bir web servisi için hiçbir ayar değiştirmenize gerek olmayacaktır. Merak ediyorsanız, Apache'nin ayar dosyalarını **/etc/httpd/conf** altında bulabilirsiniz.

Apache web sunucusunu durdurma, yeniden başlatma gibi işlemleri **/usr/sbin/apachectl** komutuyla yapmak daha doğrudur. Örneğin web sunucunuzun ayarlarında bir değişiklik yaptığınızda

/usr/sbin/apachectl restart

komutuyla programın yeni ayarlarla tekrar başlatılmasını sağlayabilirsiniz. Durdurmak için

/usr/sbin/apachectl stop

Güvenli iletişim yetenekleriyle (Secure Socket Layer) başlatmak istediğinizde ise

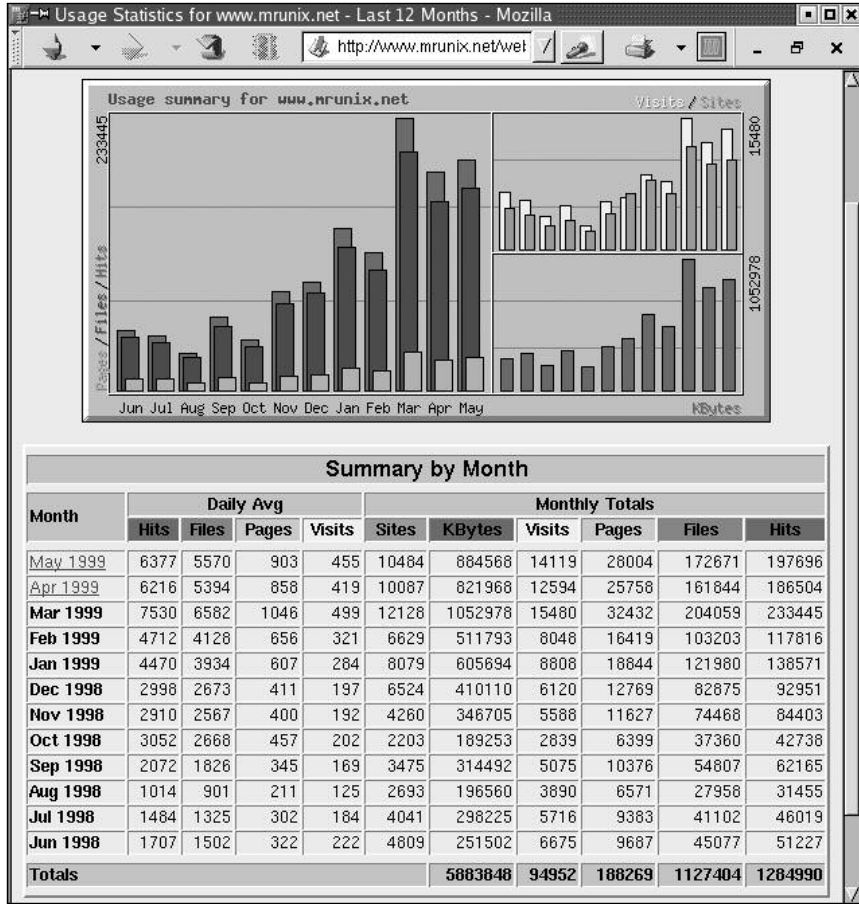
/usr/sbin/apachectl startssl

komutlarını kullanabilirsiniz.

Kullanıcılarınız, kendi kişisel web sitelerini yayınlamak isterlerse, kendi kişisel dizinlerinde "**public_html**" dizini yaratıp, altına kendi web sitelerine ilişkin dosyaları yerleştirirlerse, Apache sunucunuz bu sayfaları "**http://www.abc.com.tr/~kullanici**" adresinden yayınlayacaktır.

Apache'nin yetenekleri arasında "sanal web sunucusu" hizmeti de vardır. Bir bilgisayarla birçok web sitesinin yayını yapabilirsiniz. **/etc/httpd/conf/vhosts/Vhosts.conf** ayar dosyasında yapacağınız değişikliklerle bilgisayarınıza **http://www.abc.com.tr** adresiyle gelen isteklere ayrı; **http://www.xyz.org.tr** adresiyle gelenlere ayrı web siteleri sunabilirsiniz.

Web sitelerinin başarısı ziyaretçi sayısı ile ölçülür. Apache web sunucusu, sunduğu sayfaları kime ve ne zaman sunduğunu kaydeder. Apache log dosyalarını **/var/log/httpd/access.log** isimli dosyada bulabilirsiniz. Gözle izlenmesi olanaksız olan bu log dosyasının analizi ve görsel raporlar elde etmek için **webalizer** yazılımını kullanmanızı öneririz.



Web tabanlı uygulama programlarında çok yaygın olarak kullanılan PHP programlama dili desteği Apache ile birlikte kurulmaktadır. Geliştireceğiniz web uygulamalarının her türlü bilgisayar donanım ve işletim sistemi platformunda çalışabilmesini istiyorsanız; hele yüksek performans istiyorsanız; hele hele web ziyaretçilerinizin her gelişlerinde sitenizden yararlanabilmelerini istiyorsanız PHP öğrenmenizi ve kullanmanızı öneririz. Tamam, tamam biliyoruz... Web uygulamaları ASP ile de geliştirilmekte, hem de çok yaygın

olarak; ama biz profesyonel iş yapacak olan yazılımcılara sesleniyoruz.

Bu arada web uygulamalarında tartışmasız üstünlüğü olan bir betik dili olan Perl programlama dilinin de LINUX dağıtımınızla birlikte standart olarak kurulduğunu belirtmeliyiz. Uygulamalarınızda ister PHP, ister Perl, ister C, ister Python, ister bir başka dil/araç ya da karışımını kullanın; artık o sizin bileceğiniz iş.

postfix e-Posta Sunucusu

e-Posta servisi olmayan İnternet sunucusu olur mu? Olmaz tabii. Aslında olur! Windows tabanlı İnternet servislerinde genellikle web servisi için ayrı; e-posta sunum hizmetleri için ayrı bilgisayar kullanılmak zorunda kalındığı için e-posta servisi olmayan İnternet sunucusu olur diye kabul etmek zorundayız. LINUX dünyasındaysa, birkaç yüz Mega Hertz'lik alçakgönüllü bir bilgisayar hem web servisinizi, hem ftp servisinizi, hem de e-posta servisini verebilir.

SMTP (Simple Mail Transfer Protocol) günümüz e-posta sunucuları arasında e-posta mesajı iletiminde kullanılan standart protokoldür. UNIX dünyasında SMTP servisi için yaygın olarak kullanılan üç yazılım bulunmaktadır: **sendmail**, **postfix** ve **qmail**. Bunlar arasında kurulması en kolay olanı; üstelik virüs ve spam kontrolünün en kolay yapıları **postfix** yazılımıdır.

postfix sunucusunun temel ayarları **/etc/postfix** dizinindeki **main.cf** dosyasında yapılır. Bu ayar dosyası içindeki açıklayıcı notlar, ayarları kolayca yapmanızı sağlayacaktır. **postfix** ile virüs ve spam filtreleri kurulmasına ilişkin Türkçe "Nasıl" belgelerini **www.belgeler.org** adresinde bulabilirsiniz.

postfix ile e-posta alışverişi yanında yapılabilecek yararlı işlere birkaç örnek vermek gerekirse:

- Virüs ve spam filtreleme yapabilirsiniz (amavis, spamassassin gibi ek ve tabii ki özgür yazılımların desteği ile),
- e-posta dağıtım listeleri; yani jenerik e-posta adresleri yaratabilirsiniz. Örneğin web sayfanızda "**bilgi@abc.com.tr**" olarak ilan ettiğiniz bir e-posta adresine gelen mesajları 5 değişik gerçek e-posta adresine dağıtabilirsiniz (**/etc/aliases** dosyası),

- kullanıcılarınız kendilerine gelen tüm e-posta mesajlarını başka bir adrese yönlendirebilirler (`~/ .forward` dosyaları).

e-Posta iletim mekanizmasını kısaca anlatmak için çok uygun bir noktaya geldik galiba:

İşyerinizin alan adının **abc.com.tr**; e-posta sunucusu olarak seçtiğiniz bilgisayarın adının da **sunucu.abc.com.tr** olduğunu varsayalım. Böylece kullanıcılarınızın e-posta adresleri **ali@abc.com.tr** veya **ali@sunucu.abc.com.tr** olacaktır. Dünyanın herhangi bir yerinden gönderilecek bir e-posta mesajının e-posta sunucunuzu bulabilmesi için **abc.com.tr** alanı için e-posta sunucusunu belirten bir “MX kaydının” dünyaya ilan edilmesi gerekir.

MX kayıtları (Mail Exchange), ait oldukları alanların DNS kayıtlarını tutan sunucular tarafından tutulur ve yayınlanır. Örneğin **abc.com.tr** şirketi DNS kayıtlarını kendisi tutuyorsa, şirketin DNS sunucusuna **sunucu.abc.com.tr** için bir MX kaydı girmelidir. Eğer şirket DNS kayıtlarını kendi tutmuyorsa büyük olasılıkla İnternet servis sağlayıcısı tutuyordur; bu durumda MX kaydı İSS'na yaptırılmalıdır.

Şimdi artık Patagonya'dan **ugur@abc.com.tr** adresine bir e-posta göndermek isteyen bir bilgisayar kendi DNS sunucusuna “*abc.com.tr'nin e-Posta sunucusu kim?*” diye sorduğunda yanıt olarak **sunucu.abc.com.tr** olacaktır. Ardından, “sunucu.abc.com.tr kim?” diye sorgulayıp IP adresini (örneğin 195.194.12.32) öğrenecektir. e-Posta sunucunuzun IP adresini öğrenen Patagonya'daki bilgisayar, 195.194.12.32 IP adresli bilgisayarınızla SMTP protokolünde bir görüşme başlatacaktır. Eğer 195.194.12.32, SMTP konuşabilen bir bilgisayar ise (örneğin **postfix**, **qmail** veya **sendmail** gibi bir e-posta sunum programı çalışıyorsa) aralarında aşağıdakine benzer bir iletişim gerçekleşecektir:

- ben Patagonya'dan falanca, sizin kullanıcılardan birine e-posta iletilmesini isteyecektim...

- *Kullanıcının adı ne?*

- **cayfer**

- *İyi... Gönder...*

- Mesajın ayrıntıları şöyle:

From: ...
To: ...
BCC: ...
Subject: ...
Body: ...
Ekleri: ...
bitti.

- *Tamam.. Aldım.. Ben iletirim...*

- Kapatıyorum... Görüşürüz...

Sizin sunucunuz mesajın tamamını aldıktan sonra, alıcının posta kutusuna yerleştirilmek üzere mesajı olduğu gibi **procmail** yazılımına iletacaktır. **procmail** de varsa kullanıcının filtrelerini (spam filtresi, virus filtresi gibi) uygulayacak; eğer mesaj kabul edilecekse, **/var/spool/mail** dizininde kullanıcın adıyla anılan posta kutusu dosyasının sonuna ekleyecektir.

Kullanıcınız, posta kutusunda kendisini bekleyen e-postaları görmek istediğinde birkaç şey yapabilir:

1. LINUX sisteminize login olur, **pine** veya **mail** konsol komutuyla posta kutusunda kendisini bekleyen mesajları görebilir, yanıtlayabilir, silebilir;
2. Bir **POP3** (Post Office Protocol 3) istemcisi kullanarak (KMail, Eudora, Mozilla Thunderbird, Evolution, hatta tehlikeyi seven birisi ise Outlook) mesajlarını görebilir, yanıtlayabilir, silebilir;
3. Bir **IMAP** (Internet Message Access Protocol) istemcisi kullanarak (KMail, Eudora, Mozilla Thunderbird, Evolution, hatta tehlikeyi seven birisi ise Outlook) mesajlarını görebilir, yanıtlayabilir, silebilir.

e-posta mesajının sizin bilgisayarınızdan gönderilmesi durumunda da aynı senaryo tekrarlanacaktır. Mesajınızı gönderen yazılım (örneğin KMail), siz gönder butonunu tıkladığınızda mesajı, kendi ayarlarında **SMTP** (Simple Mail Transfer Protocol) sunucusu olarak gösterilmiş olan bilgisayara iletir. O bilgisayardaki **postfix**, **sendmail** veya **qmail** gibi bir servis sizin mesajınızdaki alıcının adresinde görünen alanın (domain) MX kaydını DNS kana-

lıyla sorgular. MX kaydı bulunursa, kayıta belirtilen sunucu ile bir SMTP görüşmesi açar ve mesajı gönderir. Mesajın alıcının posta kutusuna yerleştirilmesi artık karşıdaki e-posta sunucusunun görevidir.

procmail

procmail sunuculukla pek ilgisi olmayan bir yazılımdır. Tüm kullanıcılarınızın farkında olmadan çok sık kullanacakları; bu nedenle hem varlığından, hem de neler yaptığından haberinizi olması gereken bir program olduğu için söz etmeden geçemedik.

procmail, kullanıcılarınıza iletmek üzere bilgisayarınıza ulaşan e-posta mesajlarını karşılayan ve kullanıcıların posta kutularına yerleştiren yazılımdır. Eğer kullanıcıların kişisel dizinlerinde kendileri için hazırladıkları **.procmailrc** diye bir dosya varsa, bu dosyada yer alan satırlar procmail'e komut (makrosu) olarak yorumlanıp gelen e-posta mesajı bu komutlar doğrultusunda değerlendirilir. Gelen mesaj bu kontrol sonunda özel bir posta kutusuna yerleştirilebilir, çöpe atılabilir, bir başkasına yönlendirilebilir ya da normal posta kutusuna yerleştirilebilir. Aşağıdaki örnek **.procmailrc** dosyası,

- a. cyberspam.com adresinden gelen mesajların doğrudan çöpe atılmasını (**/dev/null** dipsiz kuyusuna yönlendirilmesini),
- b. omer@ayfer.net'den gelen mesajların da "omer" isimli bir dosyaya aktarılmasını sağlamaktadır.

```
# Örnek .procmailrc dosyası
SHELL=/usr/bin/sh
MAILDIR=${HOME}/Mail
LOGFILE=${MAILDIR}/procmail.log

:0
* ^From: *@cyberspam\.com
/dev/null
# Omerden gelen mesajları ayrı posta kutusunda sakla
:0:
* $ ^From:.*omer@ayfer\.net
/home/cayfer/Mail/omer
# Diğer mesajlari kabul et
:0:
```

\$ {DEFAULT}

samba Sunucusu

samba, “Server Message Block (SMB)” adıyla anılan protokolün LINUX işletim sistemine bir uyarlamasıdır. “NETBIOS”, “LanManager” ve “Common Internet File System (CIFS)” isimleriyle de anılan bu protokol, Windows tabanlı bilgisayarların dosya ve yazıcı kaynaklarının ağ üzerinden paylaşmasını sağlayan protokoldür.

Üzerinde **samba** sunucusu çalışan bir LINUX bilgisayar, bulunduğu ağ üzerinde bir NT sunucusu gibi davranır. Gerek yazıcı ve dosya paylaşımı, gerekse “Domain Controller” işlevlerinde son derece başarılıdır. Samba ile LINUX bilgisayarınızı bir iş grubuna (Workgroup) yerleştirip seçeceğiniz dizin ve yazıcıları Windows tabanlı bilgisayarlar kullansın diye paylaşım açabilirsiniz.

Samba sunucunun ayarları **/etc/samba/smb.conf** dosyasından yapılır. Oldukça uzun olan bu ayar dosyasının ayrıntılarını gerek dosyanın açıklama satırlarında, gerekse internette “Samba Nasıl” sözcükleriyle yapacağınız arama sonucunda karşınıza gelecek dokümanlarda bulabilirsiniz.

Bir Windows bilgisayarın paylaşım açtığı kaynaklara LINUX bilgisayarınızdan erişmeniz gerektiğinde ise, samba paketinin **smbclient** veya **smb-mount** isimli yazılımın yararlanabilirsiniz. LINUX/UNIX ve Windows tabanlı işletim sistemleriyle kullanılan bilgisayarların birlikte kullanıldığı ağlarda iki işletim sistemi arasındaki paylaşımları hep LINUX/UNIX işletim sistemi üzerinden yapmanızı öneririz.

named (DNS) Sunucusu

TCP/IP protokolüyle çalışan her bilgisayarın bir **DNS** (Domain Name System) sunucusuna gereksinimi vardır. DNS sunucuları, sembolik İnternet adreslerinin sayısal IP adreslerine çevrilmesini sağlayan; bir bakıma İnternet’in “bilinmeyen numaralar” servisleridir. Bu servisin yazılımı, ilk olarak Berkeley Üniversitesi’nde geliştirilmiş olan BIND (Berkeley İnternet Name Domain) paketidir. Paketin adının BIND olmasına rağmen, çalışan programın adı “**named**” dir.

İnternet üzerinde bir bilgisayara erişebilmeniz için ağa basacağınız paketlerin alıcı adresi bölümünde karşıdaki bilgisayarın IP adresi bulunmalıdır. Sayısal IP adreslerini ezberlemek zor olduğu için İnternet bilgisayarlarına sistematik bir şekilde düzenlenmiş, alan adları içeren sembolik isimler verilir.

Örneğin

www.bilkent.edu.tr

sembolik adresi, “**tr**” alanının, “**edu**” alt alanında yer alan “**bilkent**” ağının “**www**” isimli bilgisayarı demektir. TCP/IP bir ağ üzerinde yer alan bu bilgisayarın bir de IP adresi olmalı ve birileri bu IP adresinin hangi sembolik isme karşılık geldiğini bilmeli ve soran olduğunda da bunu bildirmelidir.

DNS mekanizması ana hatlarıyla şöyle çalışır:

Patagonya’da web tarayıcısının başında oturan bir kullanıcı, URL olarak “**http://www.bilkent.edu.tr**” girdiğinde, o tarayıcı yazılımın, **www.bilkent.edu.tr** isimli bilgisayarın IP adresini öğrenmesi ve http protokolünün web sayfası isteme kurallarına uygun bir paket hazırlayıp, paketin alıcı adresi bölümüne bu IP adresini yerleştirmesi gerekir.

Bunun için, Patagonya’daki kullanıcının kullandığı işletim sistemi, kendi TCP/IP ayarlarında belirtilmiş olan DNS sunucusuna “*www.bilkent.edu.tr de kim ola ki?*” sorusunu; yani DNS sorgusunu gönderir.

Diyelim ki, Patagonya’daki bilgisayar açısından Bilkent’e yönelik daha önce hiç bir sorgu yapılmamış olsun... Bu durumda Patagonya’daki DNS sunucusu bu sorguya ne yanıt vereceğini bilemeyecektir. Bu kez, Patagonya’daki DNS sunucusu, sorguyu kendi TCP/IP ayarlarında belirtilmiş olan bir üst düzey DNS sunucusuna aktaracaktır.

Sorgu bu şekilde yukarı doğru çıkarken, yol üzerinde biryerlerde bir DNS sunucusu “*www.bilkent.edu.tr’yi bilemem ama ‘edu.tr’ adreslerini kimin bildiğini biliyorum!*” yanıtını verecektir. Bu örneğimiz için edu.tr adreslerini bilen DNS sunucusu, ODTÜ’deki bir DNS sunucusu olacaktır.

Bu yanıt, Patagonya’daki bilgisayara geri iletilince, bu bilgisayar sorgusunu biraz değiştirerek ODTÜ’deki sunucuya “*bilkent.edu.tr adreslerini kim bilir?*” olarak yöneltir. ODTÜ bu soruya “*139.179.10.13 IP adresli bilgisayara tüm bilkent.edu.tr adreslerini sorabilirsiniz!*” yanıtını verir.

Son adımda da Patagonya'daki bilgisayar Bilkent Üniversitesi'nin 139.179.10.13 IP adresli DNS sunucusuna "*www.bilkent.edu.tr'nin IP adresi nedir?*" diye sorar, yanıtını alır ve http istek paketini bu adrese gönderir. Adresi çözülen kayıtlar, DNS sunucuları tarafından "*belki birazdan birileri gene sorar*" mantığıyla "DNS kaşesi" adı verilen tampon bellekte bir süre saklanır.

Bu senaryo, sorguların aktarıldığı hiyerarşideki DNS sunucularının ayarlarına bağlı olarak değişebilirse de, ana hatlarıyla mekanizma bu şekilde işler. Eğer hiç kimsenin tanımadığı bir adres sorgulandıysa, sorgu internetin en üst düzey DNS sunucularına kadar çıkabilir. En üst düzeydeki bu DNS sunucular (*root level servers, top level servers*) fazla ayrıntıya girmeden "**.com**", "**.edu.tr**" gibi ağ alanlarına DNS servisi veren bilgisayarların listesini tutarlar. Yani, bu en üst düzey DNS sunucular, örneğin "**bilkent.edu.tr**" ağının kayıtlarını tutan bilgisayar ya da bilgisayarları bilmeyebilir, ama "**edu.tr**" alanının kayıtlarını tutan bilgisayar ya da bilgisayarları bilirler. "**edu.tr**" kayıtlarını tutan DNS sunucusu, "**bilkent.edu.tr**" için kayıtları kimin tuttuğunu bilir. Bilkent Üniversitesi'nin DNS sunucusu da üniversitedeki tüm kayıtlı bilgisayarları bilir.

Çok sık yanlış anlaşılan bir kavrama burada açıklık getirmek istiyoruz: bir bilgisayarın sembolik adresleri çözebilmesi için o bilgisayarda DNS sunucusu yazılımı çalışması gerekmez. Aslında tek gereksinim olan, söz konusu bilgisayara makul bir sürede yanıt verebilecek bir DNS sunucusunun yakınlarında biryerlerde bulunmasıdır. Küçük ağlarda (yaklaşık 250 bilgisayara kadar) ve daha önemlisi genellikle "sorgulayan" istemcilerden oluşan ağlarda, DNS hizmeti genellikle internet servisini sağlayan kuruluştan alınır. Eğer ağınızda çok bilgisayar varsa ve/veya ağınızdaki bilgisayarların IP adresleri çok sorgulanıyorsa kendi DNS sunucunuzu kurmanız genel ağ performansını arttıracaktır.

"*Kendi DNS sunucunuzu kurmanız yararlı olur*" dediğimize bakıp da bu iş için yeni bir bilgisayar satın almanız gerektiği sonucunu çıkarmayın sakın. Ağ üzerindeki herhangi bir LINUX bilgisayar bu işi rahat rahat yapacaktır. DNS sunucunuzu Windows tabanlı bir işletim sistemi üzerinde çalıştıracaksanız; yani bir WINS sunucu kuracaksanız o zaman başka... Salt DNS işleri için oldukça güçlü bir bilgisayar ve işletim sistemi lisansı satın almanız ve bu bilgisayarda başka iş çalıştırmamanız gerekecektir.

named ayarları **/etc/named.boot** dosyası ve **/var/named** dizini altındaki dosyalarda yapılır. DNS sunucu kurmak kolay değildir. DNS mantığını iyice öğrenmek ya da kurulmuş bir sistemi inceleyip, ona bakarak çalışmak gerekir.

ssh Sunucusu (Secure Shell)

Yerel ağ hatlarını dinlemek, gelip geçen tüm Ethernet paketlerini seyretmek hatta kaydetmek mümkündür. Bilişim terminolojisinde “Sniff etmek, yani koklamak” olarak anılan bu işi yapmak için pek çok yazılım bulmak olasıdır. Bu yazılımlar bilgi çalmanın yanısıra ağ yöneticileri tarafından ağ sorunlarını bulmak için de yoğun olarak kullanılmaktadır.

Ethernet ağlarda “Hub” yerine “Switch” kullanarak hattın dinlenmesini önleyebileceğinizi sanıyorsanız yanılıyorsunuz. Nasıl yapıldığını bu kitapta elbette anlatmayacağız ama bunun mümkün olduğunu bilmenizde yarar var.

Hattın dinlenmesini önleyemeyeceğinize göre bari dinleyenlerin neler olup bittiğini izleyemeyecekleri bir düzen kurmalısınız. Hırsıza kilit dayanmayacağı gibi çözülemeyecek bir şifre sistemi kurmak ta olası değil ama hiçbir şey yapmadan da olmaz. Secure Shell kavramı işte bu iş için geliştirilmiştir. İki bilgisayar haberleşmeye başlamadan önce karşılıklı bir şifre/parola sistemi üzerinde anlaşır ve haberleşme seansı boyunca bu sistemi kullanırlar. Yeni bir seans başladığında yepyeni bir şifre/parola sistemi kullanılır. Bu haberleşmeleri dinleyip de kırmak isteyenler epeyce uğraşacaktır.

ssh, bilgisayar bilimlerinde “Symmetric Key Encryption” olarak sınıflandırılan “RSA”, “3DES” ve “Bluefish” sistemlerini kullanarak hat üzerinde gelip giden tüm paketleri kriptolar. LINUX bilgisayarınıza klasik “telnet” yerine “**ssh**” istemcisini kullanarak bağlanırsanız hatlarınızı dinleyenler neler olup bittiğini anlayamayacaktır. **ssh** ile terminal bağlantısı kurduğunuzda görsel olarak hiçbir şey değişmeyecek, size herşey telnet ekranı gibi bir ekranda görünecektir. Bilgisayarlar arası dosya kopyalarken “**rcp**” yerine “**scp**” kullanabilirsiniz. ssh kullanan ftp istemcileri de bulabilirsiniz (Mandrake LINUX dağıtımında **sftp** isimli bir güvenli ftp istemcisi, “secure ftp” yer almaktadır.)

ssh sunucunuzun ayarlarını **/etc/ssh/** dizinindeki dosyalarda yapabilirsiniz. Eğer paranoyak değilseniz, varsayılan ayarlar işinizi görecektir. Ancak; unutmayın; internete bağlı hiçbir bilgisayar yüzde yüz güvenli iletişim yapamaz!

NIS Sunucusu (Network Information Services)

NIS'in ne işe yarayacağını en kolay bir örnekle açıklayabiliriz. Diyelim ki, işyerinizde 10 tane UNIX/LINUX bilgisayar ve 30 da kullanıcınız var. Bu durumda bilgisayarlar paylaşıyor demektir; yani, kimin ne zaman hangi bilgisayarın önüne oturacağı ya da uzaktan login edeceği belli olmayacaktır. Bir kullanıcının rastgele seçtiği bir bilgisayara login olabilmesi için, seçtiği bilgisayarda tanımlı bir kullanıcı hesabına sahip olması gerekir. Bir başka deyişle sistem yöneticisinin, örneğin, **cayfer** kullanıcısı için 10 bilgisayarda da birer hesap açması gerekecektir. Daha da kötüsü, bu kullanıcı şifresini değiştirmek istediğinde 10 bilgisayarda da bu değişikliği yapması gerekecektir. Olacak iş değil! İşte buna benzer durumlarda NIS kullanmanız gerekecektir. Bir ağda yer alan bilgisayarlar arasında paylaşılabacak **/etc/passwd**, **/etc/shadow**, **/etc/hosts** gibi dosyaları NIS ile bilgisayarlar arasında paylaştırabilirsiniz. Makinelerinizden birini ya da bir kaçını NIS sunucu olarak belirleyip, örneğin **/etc/passwd** ve **/etc/shadow** dosyalarınızı bu bilgisayara yerleştirip, kullanıcı hesaplarını tek noktadan yönetebilirsiniz.

NIS servisinin adı eskiden “Yellow Pages” idi. Ancak, “Yellow Pages” sözcüklerinin isim hakkı AT&T telefon şirketine ait olduğu için bu isim terkedildi ve yerine NIS kullanılmaya başlandı. Daha önce de belirttiğimiz gibi, UNIX dünyası “tutucu” denebilecek kadar geleneklere bağlı olduğu için NIS ayar dosyaları ve ilgili süreç/program isimleri “**yp**” ile başlar şekilde kaldı. NIS ayarları da bu nedenle **/etc/ypserv.conf** isimli dosyadan yapılmaktadır.

iptables Ateş Duvarı

Bilgisayarınıza gelen ve bilgisayarınızdan çıkan TCP/IP paketleri üzerinde tam bir denetim kurmak istiyorsanız, bir “*firewall*” yazılımı, yani bir “ateş duvarı” yazılımı kullanmanız gerekir. Aslında **iptables**, sunucu kategorisine giren bir yazılım değildir. Sisteme, ağ ara birimleri üzerinden gelen ve giden paketleri filtrelemek, değiştirmek ve gerektiğinde yönlendirmek için kullanılır.

LINUX altında bu aracınız da hazır: son derece güçlü, yetenekli ve hızlı bir ateş duvarı yazılımı olan **iptables** ile **/etc/hosts.deny** ve **/etc/hosts.allow** dosyalarıyla denetleyemeyeceğiniz trafiği de kontrol altına alabilirsiniz. Örneğin, 140.1.13.23 adresinden ağınızdaki 134.43.23.1 adresli makineye gelen SMTP paketlerini kesebilirsiniz. Ağınızdaki 134.43.23.1 adresli bilgisayara gelen http paketlerini yük dağıtımını yapmak üzere 134.32.23.250 adresli bilgisayara yönlendirebilirsiniz.

iptables ile port düzeyinde filtreler kurabilirsiniz. Örneğin, günün birinde MS-SQL sunuculara musallat olan SQL Slammer gibi bir virüsün 1434 numaralı porttan saldırdığını öğrendiğinizde

```
/sbin/iptables -I FORWARD -p udp --dport 1434 -j DROP
```

komutunu vererek 1434 numaralı porttan gelen tüm UDP paketleri durdurabilirsiniz.

iptables ayarları yeni başlayanlara biraz karışık görünecektir; bu nedenle **shorewall** paketi, **iptables** ayarlarını yapmanızı kolaylaştıracak bir paket olarak geliştirilmiş ve tabii ki özgür bir yazılım olarak hizmetinize sunulmuştur.

DHCP Sunucusu

Bir TCP/IP ağda yer alacak her bilgisayar için, IP numarası, ağ maskesi (*net-mask*), ağ geçidi (*gateway*), DNS sunucusu ya da sunucularının belirtildiği ayarların bir şekilde yapılması gerekir. Bu ayarlar, statik olarak elle yapılabilir gibi, otomatik olarak da yapılabilir. TCP/IP ayarların otomatik yapılmasını sağlayan servis **DHCP** (**D**ynamic **H**ost **C**onfiguration **P**rotocol) servisi. Bu servisi kullanan ağlarda, bilgisayarların TCP/IP modüllerine, ayarların DHCP ile yapılması gerektiği belirtilirse, TCP/IP modülleri yüklendiğinde ağ arabirimine

“MAC adresim 03:04:de:3a:29:1f; bana IP ayarlarımı verecek bir DHCP servisi var mı?”

anlamına gelen bir paket gönderirler. Eğer ağda DHCP sunucusu varsa, bu sunucu yazılımı, gelen isteği, değişik politikalara göre değerlendirip istekte bulunan bilgisayara

“IP adresin 192.168.23.45,
ağ geçidin 192.168.2.1,
ağ masken 255.255.255.0,
DNS sunucun da 192.168.2.240”

gibi bir yanıt gönderir.

DHCP politikaları arasında, verilen bir IP adresinin ne kadar süreyle bir başkasına verilmeyeceği, belirli MAC adresine sahip istemcilere önceden kararlaştırılmış IP adreslerinin verilmesi gibi kavramlar kullanılabilir.

DHCP kullanıp kullanmamak bir ağ yönetim stratejisidir. Kimi ağ yöneticileri, kullanıcılarının trafiklerini izlemek ve denetlemek istedikleri için statik IP adresleri tahsis etmeyi tercih ederler. Örneğin, **iptables** veya benzeri bir ateş duvarı yazılımıyla ağınızdaki bilgisayarların üzerinden geçen trafik üzerinde sınırlamalarınız varsa, DHCP ayağınıza bağ olacaktır. Düşünsenize; siz 192.168.3.2 IP adresli bilgisayara ftp trafiği gelmesini istemediğiniz için iptables ayar dosyasına bir satır eklemiştiniz ama bir sonraki gün, söz konusu bilgisayar DHCP servisinden bir başka IP adresi alarak açılmış. Linux’ta çare tükenmez... Siz de uygun IP adresleriyle durduramıyorsanız, bir iptables filtresi kullanarak MAC adresiyle durdurursunuz.

Öte yandan, yeni yeni yaygınlaşan telsiz Ethernet teknolojisi (Wireless Ethernet, 802.11b, 802.11a, 802.11g standartları, Wi-Fi) sizi DHCP kullanmaya zorlayacaktır. Wi-Fi hizmeti verdiğiniz bir bölgeye kucağında bilgisayarıyla gelen bir kullanıcının önce ağ yöneticisini bulup, uygun bir IP numarası istemek zorunda olması hiç de mantıklı olmaz.

DHCP kullanıp kullanmamak sizin kararınıza kalmıştır. Bu kararı alırken, kullanıcılarınızı DHCP kullanmaya zorlayamayacağınızı bilmelisiniz. Siz DHCP servisi verseniz de vermeseniz de, ağınızdaki bir kullanıcı bilgisayarına istediği IP adresini girebilecektir. Tahmin edebileceğiniz gibi DHCP sunucu yazılımı **dhcpcd**’nin ayarları **/etc/dhcpcd.conf** dosyasından yapılabilir.

Dikkat ettiyseniz neredeyse tüm sunucu yazılımların ayar dosyaları **/etc** dizininin altında yer alıyor. Hatırlarsanız, daha önce yedeklemeden söz ederken **/etc** dizininin önemini vurgulamıştık. Haklıymışız; değil mi?



MySQL ve PostgreSQL Veritabanı Sunucuları

Her ikisi de özgür yazılım dünyasının çok beğenilen, güvenilen birer “ilişkisel veritabanı yönetim” (**RDBMS: Relational Database Management System**) sistemleridir. İsimlerinden de anlaşılacağı üzere; her ikisi de SQL veritabanı sorgulama dilini desteklemektedir. Hangisinin daha iyi olduğu konusunda birşey söylemek oldukça güçtür.

PostgreSQL'in, işleyebildiği SQL komut sayısı ve komut varyasyonları daha zengindir. Eğer bir veritabanı uzmanıysanız, MySQL, PostgreSQL, Oracle, Sybase, MS-SQL, IBM DB2, Informix gibi veritabanı yönetim sistemlerinin çok ayrıntılı bir karşılaştırmasını

<http://www.mysql.com/information/features.html>

adresinde incelemeden hangi veritabanı sistemini kullanacağınıza karar vermeyin.

Sıradan bir veritabanı yönetim sistemine gereksinim duyuyorsanız, ya da bu işlere yeni başlayacaksanız, MySQL kullanmanızı öneririz. Aslında MySQL ya da PostgreSQL sunucularından birini seçmek zorunda değilsiniz; ikisini birden kurup çalıştırabilirsiniz. Unutmayın, LINUX dünyasındasınız; alışmanız belki biraz zaman alacak ama özgürsünüz.

Eğer MySQL seçerseniz, gerek veritabanı sunucusunu, gerekse veritabanlarınızı yönetmek için web tabanlı bir uygulama olan **phpMyAdmin** yazılımını kurmanızı öneririz (www.phpmyadmin.net).

İlişkisel veritabanlarını pek tanımayan okuyucularımız için kısaca özetlemek gerekirse; veritabanı sunucu yazılımları, desen olarak birbirine benzeyen ve çok sayıda veri kaydı arasından belirli bir koşulu sağlayan kayıtları çok hızlı bir şekilde bulup çıkarmak için geliştirilmiş yazılımlardır. Doğal olarak bu sunucu yazılımları, veritabanlarına kayıt ekleme, güncelleme, kayıt çıkarma işlevlerini; ve en önemlisi çok kullanıcı ortamında erişimi de desteklemektedir. Uygulama programları, ilişkisel veritabanı yönetim sistemi denetimindeki dosyalara doğrudan erişemezler. Veritabanına bir kayıt eklemek gerektiğinde, uygulama programı veritabanı sunucusuna

```
INSERT musteriler values ("ABC Ltd", "Falanca Mah",  
"602.Sokak", "Bayi");
```

benzeri SQL komutları gönderir ve komutun yerine getirilmesini bekler. Kayıtlı veriler arasından seçim yapmak içinse

```
SELECT firma_adi, adresi FROM musteriler  
WHERE tip = "Bayi";
```

benzeri bir SQL komutu gönderip, tipi "Bayi" olan müşterilerin kayıtları istenebilir. Verilen kriterleri sağlayan kayıtlar bir küme olarak uygulama programına geri gönderilir.

Perl dili ile MySQL kullanımı hakkında ayrıntılı bilgiye gereksinim duyarsanız, PUSULA Yayıncılık tarafından yayınlanmış olan *Perl ve MySQL ile CGI Programlama* (ISBN:975-7092-89-4) kitabını önerebiliriz.

squid Proxy Sunucusu

LINUX dünyasının en çok kullanılan "proxy" sunucusudur. "Proxy server" teriminin Türkçe karşılığı olarak "vekil sunucu" kullanılmaktaysa da "vekil" sözcüğü "proxy" sözcüğünün anlamını tam olarak vermediği için; biz "proxy" demeye devam edeceğiz.

Proxy sunucuları, bir bilgisayar ağının İnternet çıkışında devreye "seri" olarak girmiş ve proxy yazılımı çalışan bilgisayarlardır. Genellikle, proxy sunucudan yararlanmak isteyenler TCP/IP istemci yazılımlarının (örneğin web tarayıcılarının) ayarlarında bunu belirtirler. Böylece söz konusu uygulama TCP/IP paketlerini proxy sunucuya yönlendirirler. Örneğin, "http proxy" görevini yapan bir proxy sunucu, içerden kendisine gelen tüm http isteklerini yakalar; gerekiyorsa istemci olarak kendisini göstererek paketi yeniden düzenler ve alıcısına öyle gönderir. Gelen yanıtları da önce kendi diskine kaydeder, sonra da dosyayı istemiş olan, içerdeki bilgisayara iletir. Bir süre sonra ağın içinden birileri gene bu dosyayı İnternet'ten indirmek isterse, bu isteği İnternet hattına aktarmak yerine kendi disklerine kaydettiği dosyayı geri gönderir. Aslında mekanizma bu kadar basit değildir; örneğin, istenen bir dosyayı kendi disklerinden sunmadan önce dosyanın istendiği yerden o dosyanın en son değişikliğe uğradığı saat ve tarihi öğrenir; eğer kendi diskindeki kopya daha eski değilse, dosyayı İnternet'ten indirmek yerine kendi diskinden gönderir.

Proxy sunucuları internet hatlarından tasarruf sağlar; ayrıca sık sık ziyaret edilen web sitelerinin kullanıcılara çok daha hızlı sunulmasını sağlar.

Proxy sunucuları sadece web iletişimde kullanılmaz, FTP proxy sunucuları da anlamlıdır.

Proxy sunucusu olarak kullanılacak bir bilgisayarın işe yarar bir disk kapasitesine sahip olması gerekir. Arkasındaki ağın büyüklüğüne; daha doğrusu arkadaki kullanıcı sayısına ve bunların web davranışlarının özelliklerine bağlı olarak birkaç Gigabyte'dan birkaç yüz GigaByte'a kadar disk gerekecektir. Bir üniversite için, 60 GByte civarında bir proxy sunucu disk kapasitesi iş görecektir; oysa bir servis sağlayıcı için daha fazlası gerekecektir.

Proxy sunucuları kullanıcı tarafında genellikle isteğe bağlı olarak kullanılır. Yani istemeyen kullanıcılar, web tarayıcılarının proxy ayarlarını yapmayarak proxy sunucunuzdan yararlanmamayı seçebilir. Servis verdiğiniz ağın özelliklerine bağlı olarak kullanıcılarınızı proxy sunucusunu kullanmaya zorlamak isteyebilirsiniz. Böyle bir durumda squid yazılımını "saydam proxy sunucusu" (transparent proxy server) olarak kurmalısınız.

Kurduğunuz bir proxy sunucusunun yararını ölçmek istediğinizde, sunucunun raporlarından ve log kayıtlarından yararlanmalısınız. "Kaşe isabet oranı" (*Cache hit ratio*), sunucunuzun başarı düzeyini en iyi gösterecek olan değerdir.

squid, sorumlu olduğu ağdaki trafiği çok sıkı denetlemek isteyen yöneticiler için önemli bir araçtır. Yerel ağda kopyası bulunan bir dosyayı almak için internet hattının boşuna işgal edilmesini engellediği gibi, ağ içinde kimin hangi adrese gittiğinin de ayrıntılı bir şekilde izlenmesine olanak sağlar. İşyerinizdeki kullanıcıların bütün gün chat yaptıklarından, fal ve oyun sayfalarında dolaştıklarından şüpheleniyorsanız, squid kurup istatistik raporlarını inceleyebilirsiniz. "tor" ve "privoxy" gibi yazılımlarla squid'in yetenekleri daha da genişletilebilir. Örneğin "privoxy" ile spam filtrelemeye benzer bir yöntemle web sitelerindeki reklamlar filtrenebilir.

ppp Çevirmeli Ağ Sunucusu

İşyerinizde kurduğunuz LINUX bilgisayara evden modemle bağlanmak istediğinizde iki seçeneğiniz vardır:

1. Basit telnet bağlantısı kurmak
2. ppp bağlantısı kurup, hem LINUX makinenize, hem de onun üzerinden internete de bağlanmak.

ppp (Point-to-Point protocol), seri arabirimler üzerinden TCP/IP bağlantısı kurmak için geliştirilmiş bir protokoldur. İki bilgisayar arasında ppp protokolu çalışmaya başladığında her iki tarafta da seri arabirimlere birer IP numarası verilir. Böylece aranan bilgisayarın iki ağ arabirimi olur: biri Ethernet arabirimi; diğeri de modemin bağlı olduğu seri arabirim.

“Aranan bilgisayar” artık iki bilgisayar ağı üzerinde aynı anda yer alan bir yönlendiricidir (router).

Aralarında ppp bağlantısı olan bilgisayarların arasında istemci-sunucu gibi bir ilişki yoktur; her ikisi de eşit düzeyde noktadan-noktaya bağlı bilgisayarlardır; bir başka deyişle “ppp sunucusu” aslında pek doğru bir deyim değildir.

Bir bilgisayarın, modem üzerinden gelen seri bağlantıları kabul edebilmesi için söz konusu seri arabirimin (**/dev/ttySn**) basit terminal bağlantısı yapabilir şekilde ayarlanmış olması gerekir. Bu ayarı yapabilmek için, örneğin ilk seri arabirim **/dev/ttyS0** için **getty** programı çalışabilir durumda olmalıdır.

getty servisiyle normal terminal bağlantısı yapabilen bir kullanıcı, kendi bağlantısı için bir **pppd** süreci başlatıp kendi bilgisayarıyla aradığı bilgisayar arasında bir ppp bağlantısı kurabilir.

Evet! Yukarıdaki satırlarda biraz garip laflar ettiğimiz farkındayız ama ne yapalım, bu işler böyle. Bu kitapta, bir bilgisayarın ppp servisini verecek şekilde kurulması için yapılması gereken herşeyi anlatmaya olanak yok; amacımız yol göstermek. Eğer bu konuda ayrıntılı bilgiye gereksinim duyuyorsanız, internette çok sayıda kopyasını kolayca bulabileceğiniz “Serial Howto” ve “ppp Howto” dokümanlarını okumalısınız.

Kim Korkar LINUX'tan?

LINUX işletim sistemi ile birlikte gelen veya sonradan yüklenebilen önemli ağ servisleriyle ilgili daha ayrıntılı bilgiye gereksinim duyduğunuzda PUSULA Yayıncılığın *LINUX Ağ Servisleri* isimli kitabından yararlanabilirsiniz (ISBN: 975-6477-13-x).

BUNLARI BİLİYOR MUYDUNUZ?

Linux Dağıtımları

Mayıs 2006 tarihinde, dağıtılan, satılan ve desteklenen 357 değişik LINUX dağıtımı olduğunu biliyor muydunuz?

Evet, tam 357 değişik LINUX dağıtımından'dan söz ediyoruz... Bu kitabın birinci baskısının yayınlandığı Temmuz 2003'de dağıtım sayısı 135 idi.

LINUX'un arkasında profesyonel destek olmadığını, sorunların çözülmesinde kullanıcıların yalnız olduğunu iddia edenlere ithaf olunur.

LINUX dağıtımlarını sürekli izleyen, inceleyen, irdeleyip eleştiren www.distrowatch.com sitesinde ayrıntılı bilgi bulabilirsiniz.

GNU Genel Kamu Lisansı

GNU Public License (GPL)

EK

Genel Kamu Lisansı

This is an **unofficial** translation of the GNU General Public License into Turkish. It was not published by the Free Software Foundation, and does not legally state the distribution terms for software that uses the GNU GPL-only the original English text of the GNU GPL does that. However, we hope that this translation will help Turkish speakers understand the GNU GPL better.

Bu, GNU Genel Kamu Lisansının (GPL) Türkçe'ye **gayriresmi** çevirisidir. Bu çeviri Free Software Foundation tarafından yayınlanmamış olup GNU GPL kullanan yazılımların dağıtım şartlarını belirleme açısından hukuki bağlayıcılığı yoktur. Hukuki açıdan yalnızca GNU GPL'in İngilizce metni bağlayıcıdır. Bu çeviri, Türkçe kullanıcılarının GNU GPL'i daha iyi anlayabilmeleri için hazırlanmıştır.

GNU Genel Kamu Lisansı (GPL)

Sürüm 2, Haziran 1991

Telif Hakkı © 1989, 1991 Free Software Foundation, Inc. 59 Temple Place
- Suite 330, Boston, MA 02111-1307, USA

Bu lisans dökümanının birebir kopyalarını yapma ve dağıtma izni herkese verilmiştir, fakat metinde değişiklik yapma izni yoktur.

Giriş

Yazılım lisanslarının çoğu sizin yazılımı paylaşma ve değiştirme hakkınızın elinizden alınması için hazırlanmıştır. Buna karşılık, GNU Genel Kamu Lisansı sizin serbest yazılımları değiştirme ve paylaşma hakkınızın mahfuz tutulması ve yazılımın bütün kullanıcıları için serbest olması amacı ile yazıl-

mıştır. Bu Genel Kamu Lisansı, Free Software Foundation'un çoğu yazılımı ve bu lisansı kullanmayı düstur edinen diğer yazılımcıların yazılımları için kullanılmaktadır. (Free Software Foundation'un bazı yazılımları GNU Kitaplık Genel Kamu Lisansı -- GNU LGPL -- altında dağıtılmaktadır.) Siz de bu lisansı yazılımlarınıza uygulayabilirsiniz.

Serbest yazılımdan bahsettiğimiz zaman fiyattan değil, özgürlükten bahsediyoruz. Bizim Genel Kamu Lisanslarımız, sizin serbest yazılımların kopyalarını dağıtma özgürlüğünüzü (ve isterseniz bu hizmet için para almanızı), yazılım kaynak kodlarının size dağıtım esnasında veya eğer isterseniz verilmesini, yazılımı değiştirebilmenizi, yazılımın parçalarını yeni yazılımlar içerisinde kullanabilmenizi ve bunları yapabileceğinizi bilmenizi sağlamaktadır.

Haklarınızı koruyabilmemiz için sizin haklarınızı kısıtlama veya sizin bu haklarınızdan feragat etmenizi isteme yollarını yasaklayıcı bazı kısıtlamalar getirmemiz gerekmektedir. Bu kısıtlamalar eğer serbest yazılım dağıtıyor veya değiştiriyorsanız size bazı yükümlülükler getirmektedir.

Örneğin böyle bir programın kopyalarını, bedava veya ücret karşılığı dağıtıyorsanız alıcılara sizin sahip olduğunuz bütün hakları sağlamalısınız. Onların da kaynak kodlarına sahip olmalarını veya ulaşabilmelerini sağlamalısınız. Onlara da haklarını bilebilmeleri için bu şartları göstermelisiniz.

Haklarınızı koruma iki aşamada gerçekleşmektedir:

1. Yazılıma telif hakkı alınmaktadır.
2. Yazılım lisansı olarak size, hukuki olarak, yazılımı kopyalama, dağıtma ve/veya değiştirme hakkı tanıyan bu lisans sunulmaktadır.

Ayrıca, yazarların ve bizim korunmamız için bu serbest yazılımın herhangi bir garantisi olmadığını herkesin anlamasını istiyoruz. Eğer yazılım başkası tarafından değiştirilmiş ve değiştirilmiş hali ile tarafınıza ulaştırılmış ise alıcıların, ellerinde olan yazılımın orijinal olmadığını, dolayısıyla başkaları tarafından eklenen problemlerin ilk yazarların şöhretlerine olumsuz etkide bulunmaması gerektiğini bilmelerini istiyoruz.

Son olarak, bütün serbest yazılımlar yazılım patentleri tarafından sürekli tehdit altında bulunmaktadır. Serbest bir yazılımın dağıtıcılarının bireysel

olarak patent lisansı almalarını ve bu yol ile yazılımı müseccel hale getirmelerine imkan vermemek istiyoruz. Bunu engellemek için, yazılım için alınacak her patentin herkesin serbest kullanımına izin vermesi veya patentlenmemesi gerektiğini açık olarak ortaya koyuyoruz.

Kopyalama, dağıtım ve değiştirme ile ilgili kesin şart ve kayıtlar aşağıda yer almaktadır.

Kopyalama, Dağıtım ve Değiştirme ile İlgili Şart ve Kayıtlar

Bu Lisans, telif hakkı sahibi tarafından içerisine bu Genel Kamu Lisansı altında dağıtıldığına dair ibare konmuş olan herhangi bir yazılım veya başka eseri kapsamaktadır. Aşağıda “Yazılım”, bu kapsamdaki herhangi bir yazılım veya eser, “Yazılımı baz alan ürün”, ise Yazılım veya telif kanunu altında Yazılım’dan iştikak etmiş, yani Yazılım’ın tamamını veya bir parçasını, değiştirmeden veya değişiklikler ile, veya başka bir dile tercüme edilmiş hali ile içeren herhangi bir ürün, manasında kullanılmaktadır. (Bundan sonra tercüme “değiştirme” kapsamında sınırsız olarak içerilecektir.) Her ruhsat sahibine “siz” olarak hitap edilmektedir.

Kopyalama, dağıtım ve değiştirme haricinde kalan faaliyetler bu Lisans’ın kapsamı dışındadırlar. Yazılım’ı çalıştırma eylemi sınırlandırılmamıştır ve Yazılım’ın çıktısı yalnızca çıktının içeriği (Yazılım’ı çalıştırmak yolu ile elde edilmesinden bağımsız olarak) Yazılım’ı baz alan ürün kapsamına girer ise bu Lisans kapsamındadır. Bu koşulun sağlanıp sağlanmadığı Yazılım’ın ne yaptığı ile ilgilidir.

1. Yazılım’ın kaynak kodlarını birebir, aldığınız şekilde, herhangi bir ortamda ve vasıta ile, uygun ve görünür bir şekilde telif hakkı bildirimi ve garantisiz olduğuna dair bildirim koymak, bu Lisans’dan bahseden herhangi bir bildirim aynen muhafaza etmek ve bütün diğer alıcılara Yazılım ile birlikte bu Lisans’ın bir kopyasını vermek şartı ile kopyalayabilir ve dağıtabilirsiniz.

Kopyalamak fiili işlemi için bir ücret talep edebilir ve sizin seçiminize bağlı olarak ücret karşılığı garanti verebilirsiniz.

2. Yazılım’ın kopyasını veya kopyalarını veya herhangi bir parçasını değiştirerek Yazılım’ı baz alan ürün elde edebilir, bu değişiklikleri veya ürünün

kendisini yukarıda 1. bölümdeki şartlar dahilinde ve aşağıda sıralanan şartların yerine getirilmesi koşulu ile kopyalayabilir ve dağıtabilirsiniz.

- a. Değiştirilen dosyaların görünür bir şekilde dosyaların sizin tarafınızdan değiştirildiğine dair, tarihli bir bildirim içermesini sağlamalısınız.
- b. Yazılım'dan veya Yazılım'ın bir parçasından tamamen veya kısmen iştirak etmiş ve sizin tarafınızdan dağıtılan veya yayınlanan herhangi bir ürünün bütün üçüncü şahıslara bu Lisans şartları altında ücretsiz olarak ruhsatlanmasını sağlamalısınız.
- c. Eğer değiştirilen yazılım olağan kullanım altında komutları interaktif olarak alıyor ise, yazılım, en olağan kullanım için interaktif olarak çalıştırıldığı zaman uygun bir telif hakkı bildirimi, garantisi olmadığına (veya sizin tarafınızdan garanti verildiğine), kullanıcıların bu yazılımı bu şartlar altında tekrar dağıtabileceklerine ve kullanıcının bu Lisansın bir kopyasını nasıl görebileceğine dair bir bildirim yazdırmalı veya göstermelidir. (İstisna: Eğer Yazılım'ın kendisi interaktif ise fakat böyle bir bildirim olağan kullanım esnasında yazdırmıyor ise, sizin Yazılım'ı baz alan ürününüz böyle bir bildirimde bulunmak zorunda değildir.)

Bu şartlar değiştirilmiş eserin tamamını kapsamaktadır. Eğer eserin tespit edilebilir kısımları Yazılım'dan iştirak etmemiş ise ve makul surette kendi başlarına bağımsız ve ayrı eserler olarak kabul edilebilir ise, o zaman bu Lisans ve şartları, bu parçaları ayrı eser olarak dağıttığınız zaman bağlayıcı değildir. Fakat, aynı parçaları Yazılım'ı baz alan bir ürün bütününe bir parçası olarak dağıttığınız zaman bütününe dağıtımı, diğer ruhsat sahiplerine verilen izinlerin bütüne ait olduğu ve parçalarına, yazarının kim olduğuna bakılmaksızın bütün parçalarına tek tek ve müşterek olarak uygulandığı bu Lisans şartlarına uygun olmalıdır.

Bu bölümün hedefi tamamen sizin tarafınızdan yazılan bir eser üzerinde hak iddia etmek veya sizin böyle bir eser üzerindeki haklarınıza muhalefet etmek değil, Yazılım'ı baz alan, Yazılım'dan iştirak etmiş veya müşterek olarak ortaya çıkarılmış eserlerin dağıtımını kontrol etme haklarını düzenlemektir.

Buna ek olarak, Yazılım'ı baz almayan herhangi bir ürünün Yazılım ile (veya Yazılım'ı baz alan bir ürün ile) bir bilgi saklama ortamında veya bir dağıtım ortamında beraber tutulması diğer eseri bu Lisans kapsamına sokmaz.

3. Yazılım'ı (veya 2. bölümde tanımlandığı hali ile onu baz alan bir ürünü) ara derlenmiş veya uygulama hali ile 1. ve 2. Bölüm'deki şartlar dahilinde ve aşağıda sıralanan yöntemlerden birisine uygun olarak kopyalayabilir ve dağıtabilirsiniz.

a. Yaygın olarak yazılım dağıtımında kullanılan bir ortam üzerinde, yukarıda 1. ve 2. Bölüm'de bulunan şartlar dahilinde, bilgisayar tarafından okunabilir kaynak kodlarının tamamı ile birlikte dağıtmak.

b. Herhangi bir üçüncü şahsa, fiziksel olarak dağıtımını gerçekleştirme masraflarından daha fazla ücret almayarak, yaygın olarak yazılım dağıtımında kullanılan bir ortam üzerinde, yukarıda 1. ve 2. Bölüm'de bulunan şartlar dahilinde, bilgisayar tarafından okunabilir kaynak kodlarının tamamını dağıtacağınıza dair en az üç yıl geçerli olacak yazılı bir taahhütname ile birlikte dağıtmak.

c. Size verilmiş olan ilgili kaynak kodunu dağıtma taahhütnamesi ile birlikte dağıtmak. (Bu alternatif yalnızca ticari olmayan dağıtımlar için ve yalnızca siz de yazılımı ara derlenmiş veya uygulama biçiminde ve yukarıda b) bölümünde anlatılan şekli ile bir taahhütname ile birlikte almış iseniz geçerlidir.)

Bir eserin kaynak kodu, esere değiştirme yapmak için en uygun yöntem ve imkan anlamında kullanılmaktadır. Uygulama biçiminde bir eser için, kaynak kodu, içerdği bütün parçalar için ilgili kaynak kodları, ilgili arayüz tanım dosyaları ve derleme ve yükleme işlemlerinde kullanılan bütün betikler anlamında kullanılmaktadır. Bir istisna olarak, dağıtılan kaynak kodu, genelde uygulamanın üzerinde çalışacağı işletim sisteminin ana parçaları (derleyici, çekirdek v.b.) ile birlikte dağıtılan herhangi bir bileşeni, eğer ilgili bileşen, uygulama ile birlikte dağıtılmıyorsa, içermek zorunda değildir.

Eğer uygulama veya ara derlenmiş biçimde yazılımın dağıtımını belli bir yere erişim ve oradan kopyalama imkanı olarak yapıyorsa, aynı yerden, aynı koşullar altında kaynak koduna erişim imkanı sağlamak, üçüncü şahısların ara derlenmiş ve uygulama biçimleri ile birlikte kaynak kodunu kopyalama zorunlulukları olmasa bile kaynak kodunu dağıtmak olarak kabul edilmektedir.

4. Yazılım'ı bu Lisans'ta sarıh olarak belirtilen şartlar haricinde kopyalayamaz, değiştiremez, ruhsat hakkını veremez ve dağıtamazsınız. Buna aykırı herhangi bir kopyalama, değiştirme, ruhsat hakkı verme, veya dağıtımda bulunma hükümsüzdür ve böyle bir teşebbüs halinde bu Lisans altındaki bütün haklarınız iptal edilir. Sizden, bu Lisans kapsamında kopya veya hak al-

miş olan üçüncü şahıslar, Lisans şartlarına uygunluklarını devam ettirdikleri sürece, ruhsat haklarını muhafaza edeceklerdir.

5. Bu Lisans sizin tarafınızdan imzalanmadığı için bu Lisans'ı kabul etmek zorunda değilsiniz. Fakat, size Yazılım'ı veya onu baz alan ürünleri değiştirmek veya dağıtmak için izin veren başka bir belge yoktur. Eğer bu Lisans'ı kabul etmiyorsanız bu eylemler kanun tarafından sizin için yasaklanmıştır. Dolayısıyla, Yazılım'ı (veya onu baz alan bir ürünü) değiştirmeniz veya dağıtmanız bu Lisans'ı ve Lisans'ın Yazılım'ı veya ondan iştikak etmiş bütün eserleri kopyalamak, değiştirmek ve dağıtmak için getirdiği şart ve kayıtları kabul ettiğiniz manasına gelmektedir.

6. Yazılım'ı (veya onu baz alan herhangi bir ürünü) yeniden dağıttığınız her defada alıcı, ilk ruhsat sahibinden otomatik olarak Yazılım'ı bu şartlar ve kayıtlar dahilinde kopyalamak, değiştirmek ve dağıtmak için ruhsat almaktadır. Alıcının burada verilen hakları kullanmasına ek birtakım kısıtlamalar getiremezsiniz. Üçüncü şahısları bu Lisans mucibince hareket etmeğe mecbur etmek sizin sorumluluk ve yükümlülüğünüz altında değildir.

7. Eğer bir mahkeme kararı veya patent ihlal iddiası veya herhangi başka bir (patent meseleleri ile sınırlı olmayan) sebep sonucunda size, bu Lisans'ın şart ve kayıtlarına aykırı olan birtakım (mahkeme kararı, özel anlaşma veya başka bir şekilde) kısıtlamalar getirilirse, bu sizi bu Lisans şart ve kayıtlarına uyma mecburiyetinden serbest bırakmaz. Eğer aynı anda hem bu Lisans'ın şartlarını yerine getiren hem de diğer kısıtlamalara uygun olan bir şekilde Yazılım'ı dağıtamıyorsanız, o zaman Yazılım'ı dağıtamazsınız. Örneğin, eğer bir patent lisansı direkt veya endirekt olarak sizden kopya alacak olan üçüncü şahısların bedel ödemeksizin Yazılım'ı dağıtmalarına hak tanııyorsa o zaman sizin hem bu koşulu hem Lisans koşullarını yerine getirmenizin tek yolu Yazılım'ı dağıtmamak olacaktır.

Eğer bu bölümün herhangi bir parçası herhangi bir şart altında uygulanamaz veya hatalı bulunur ise o şartlar dahilinde bölümün geri kalan kısmı, bütün diğer şartlar altında da bölümün tamamı geçerlidir.

Bu bölümün amacı sizin patent haklarınızı, herhangi bir mülkiyet hakkını ihlal etmenize yol açmak veya bu hakların geçerliliğine muhalefet etmenizi sağlamak değildir; bu bölümün bütün amacı kamu lisans uygulamaları ile oluşturulan serbest yazılım dağıtım sisteminin bütünlüğünü ve işlerliğini korumaktır. Bu sistemin tutarlı uygulanmasına dayanarak pek çok kişi bu sistemle dağıtılan geniş yelpazedeki yazılımlara katkıda bulunmuştur; yazılımı-

nı bu veya başka bir sistemle dağıtmak kararı yazara aittir, herhangi bir kullanıcı bu kararı veremez.

Bu bölüm Lisans'ın geri kalanının doğurduğu sonuçların ne olduğunu açıklığa kavuşturmak amacını gütmektedir.

8. Eğer Yazılım'ın kullanımı ve/veya dağıtımı bazı ülkelerde telif hakkı taşıyan arayüzler veya patentler yüzünden kısıtlanırsa, Yazılım'ı bu Lisans kapsamına ilk koyan telif hakkı sahibi, Yazılım'ın yalnızca bu ülkeler haricinde dağıtılabileceğine dair açık bir coğrafi dağıtım kısıtlaması koyabilir. Böyle bir durumda bu Lisans bu kısıtlamayı sanki Lisans'ın içerisine yazılmış gibi kapsar.

9. Free Software Foundation zaman zaman Genel Kamu Lisansı'nın yeni ve/veya değiştirilmiş biçimlerini yayınlayabilir. Böyle yeni sürümler mana olarak şimdiki haline benzer olacaktır, fakat doğacak yeni problemler veya kaygılara cevap verecek şekilde detayda farklılık arzedebilir.

Her yeni biçime ayırdedici bir sürüm numarası verilmektedir. Eğer Yazılım bir sürüm numarası belirtiyor ve “bu ve bundan sonraki sürümler” altında dağıtılıyorsa, belirtilen sürüm veya Free Software Foundation tarafından yayınlanan herhangi sonraki bir sürümün şart ve kayıtlarına uymakta serbestsiniz. Eğer Yazılım Lisans için bir sürüm numarası belirtmiyor ise, Free Software Foundation tarafından yayınlanmış olan herhangi bir sürümün şart ve kayıtlarına uymakta serbestsiniz.

10. Eğer bu Yazılım'ın parçalarını dağıtım koşulları farklı olan başka serbest yazılımların içerisinde kullanmak isterseniz, yazara sorarak izin isteyin. Telif hakkı Free Software Foundation'a ait olan yazılımlar için Free Software Foundation'a yazın, bazen istisnalar kabul edilmektedir. Kararımız, serbest yazılımlarımızdan iştikak etmiş yazılımların serbest statülerini korumak ve genel olarak yazılımların yeniden kullanılabilirliğini ve paylaşımını sağlamak amaçları doğrultusunda şekillenecektir.

Garanti Yoktur

11. Bu yazılım ücretsiz olarak ruhsatlandığı için, yazılım için ilgili kanunların izin verdiği ölçüde herhangi bir garanti verilmemektedir. Aksi yazılı olarak belirtilmediği müddetçe telif hakkı sahipleri ve/veya başka şahıslar yazılımı “olduğu gibi”, aşikar veya zimnen, satılabilirliği veya herhangi bir amaçla uygunluğu da dahil olmak üzere hiçbir garanti vermeksizin dağıtmaktadırlar. Yazılımın kalitesi veya performansı ile ilgili tüm sorunlar size aittir.

Yazılımdaki herhangi bir bozukluktan dolayı doğabilecek olan bütün servis, tamir veya düzeltme masrafları size aittir.

12. İlgili kanunun icbar ettiği durumlar veya yazılı anlaşma haricinde herhangi bir şekilde telif hakkı sahibi veya yukarıda izin verildiği şekilde yazılımı değiştiren veya yeniden dağıtan herhangi bir kişi, yazılımın kullanımı veya kullanılamaması (veya veri kaybı oluşması, verinin yanlış hale gelmesi, sizin veya üçüncü şahısların zarara uğraması veya yazılımın başka yazılımlarla beraber çalışmaması) yüzünden oluşan genel, özel, doğrudan ya da dolaylı herhangi bir zarardan, böyle bir tazminat talebi telif hakkı sahibi veya ilgili kişiye bildirilmiş olsa dahi, sorumlu değildir.

Şart ve Kayıtların Sonu

Bu Şartlar Yeni Yazılımlara Nasıl Uygulanır?

Eğer yeni bir yazılım geliştiriyor ve bunun kamuya en fazla düzeyde yarar sağlamasını istiyorsanız, yazılımınızı herkesin dağıtıp, değiştirebileceği özgür yazılım haline getirmenizi öneriyoruz.

Bu koşulları uygulamak için yazılıma aşağıdaki bildirimleri ekleyin. En sağlıklı yöntem her kaynak kodu dosyasının başına bu bildirimi ekleyerek garanti olmadığına dair bilginin verildiğinden emin olmaktır; her dosya en azından “copyright” (telif hakkı) satırını ve bildirimin tam metninin nerede bulunabileceğine dair bilgi içermelidir.

{yazılımın adını ve ne yaptığını anlatan bir satır.}

Copyright (C) {yıl} {yazarın adı}

This program is free software; you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation; either version 2 of the License, or (at your option) any later version.

This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with this library; if not, write to the Free Software Foundation,

Inc., 59 Temple Place, Suite 330, Boston, MA 02111-1307 USA

Size normal ve elektronik posta ile nasıl ulaşılabileceğine dair bilgi eklemeyi unutmayın.

Eğer yazılımınız interaktif ise, interaktif kipte başlatıldığı zaman gösterilen kısa bir bildirim koyun.

Gnomovizyon version 69, Copyright (C) *yıl yazarın adı*

Gnomovizyon comes with ABSOLUTELY NO WARRANTY; for details type `show w'. This is free software, and you are welcome to redistribute it under certain conditions; type `show c' for details.

Gnomovizyon sürüm 69, Telif hakkı (C) *yıl yazarın adı*

Gnomovizyon için HİÇBİR GARANTİ verilmemektedir; detaylar için `show w' yazın. Bu bir serbest yazılımdır ve belli koşullar altında yeniden dağıtılabilir; detaylar için `show c' yazın.

Örnekte verilen `show w' ve `show c' komutları GNU Genel Kamu Lisansı'nın ilgili bölümlerini göstermelidir. Elbette kullanılan komutlar daha farklı olabilir veya yazılımınıza uyan başka yöntemlerle bu bildirim yapılabilir.

İşvereninizin (eğer programcı olarak çalışıyorsanız) veya, eğer öğrenci iseniz, okulunuzun telif haklarından feragat ettiklerine dair bir feragatname imzalamalarını isteyebilirsiniz. Aşağıda bir örnek yer almaktadır, isimleri değiştirin:

Gereksizİşler, A.Ş., Mehmet Herhangibiri tarafından yazılmış `Abidik-Gubidik' yazılımında (kapıkolu çevirmekte kullanılan bir yazılım) olabilecek bütün telif haklarından feragat eder.

{*Yön Etici İmzası*}, 1 April 1990

Yön Etici, Gereksizİşler Yetkilisi

Bu Genel Kamu Lisansı yazılımınızın serbest olmayan yazılımların içerisine dahil edilmesine imkan tanımaz. Eğer yazılımınız bir kitaplık ise, serbest olmayan yazılımların kitaplığınıza bağlanmasına imkan tanımak isteyebilirsiniz. Eğer yapmak istediğiniz bu ise, bu Lisans yerine GNU Kısıtlı Genel Kamu Lisansı'nı kullanabilirsiniz.

Çeviren: Deniz Akkuş Kanca, 2001

Translated by: Deniz Akkuş Kanca, 2001

Sonsöz

Bu kitap, LINUX dünyasına “ilk adımınızı” atmanızı sağladıysa ne mutlu bize.

LINUX ve doğal olarak UNIX dünyasındaki yolculuk uzun, heyecanlı, keyifli ve yorucu bir deneyimdir. İyi bir UNIX sistem yöneticisi olabilmek için yıllarca çalışmak gerekir. İyi bir dış hekimi, iyi bir cerrah olabilmek için nasıl zamana ve en önemlisi mesleğin araçlarını iyi tanıma ve kullanma becerilerini geliştirmeye gereksinim varsa; UNIX dünyasında da işletim sisteminin araçlarını iyi öğrenmeye ve kullanma becerilerini geliştirmeye gerek vardır.

Bu kitabı okumakla LINUX öğrenmediniz. Olsa olsa korkmamayı öğrendiniz. Bu kitapta bir iki satırla anlatılan birçok konunun her biriyle ilgili yüzlerce sayfalık kitaplar var. Onları da bir okuyun; ondan sonra tekrar görüşelim. LINUX/UNIX sistem yöneticiliği bir meslektir. Tek bir kitap; hele giriş düzeyinde bir kitap okumakla bir meslek edinmeyi ummuyordunuz herhalde...

Tüm bilişim konularında olduğu gibi bir işletim sistemini kullanmayı okuyarak öğrenemezsiniz. Denemeli, uğraşmalı, araştırmalı, aç ve uykusuz kalmalısınız. Eğer LINUX dünyasına ciddi bir şekilde girmek istiyorsanız size tek bir öneride bulunabiliriz: bilgisayarınızdaki LINUX dışındaki tüm işletim sistemlerini silin. Eskiden beri diğer işletim sistemiyle yapmakta olduğunuz işleri LINUX ile yapmaya çalışın. Unutmayın; aklınıza gelebilecek her türlü sorunun çözümü LINUX'ta vardır; bütün mesele bu çözümü bulup uygulayabilmektedir.

LINUX işletim sistemine bir kez alıştıktan ve onu yönetmeyi öğrendikten sonra diğer işletim sistemleri; özellikle de Windows ailesinin sunucu özelliğine sahip olan sürümleri size çok zayıf gelecektir. Bugüne kadar UNIX/LINUX dünyasını bırakıp Windows ortamına geçen profesyonelle hiç rastlamadık ama tersini çok gördük. Gözlem süresi yazarlardan genç olanı için 3-4 yıl, ihtiyar olanı için Windows'un tarihi kadar uzundur.

UNIX dünyasına hoşgeldiniz.

Ömer Ayfer - Can Uğur Ayfer