

- “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.

Kim Korkar LINUX'tan?

LINUX dünyasında; RedHat ve Debian dağıtımları için geliştirilmiş olan 2 önemli “paket” çeşidi vardır. RedHat'ın 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 yapılabilecek bir iş değildir. Ancak, doğal olarak bunun da çözümü var: “Paket Yöneticileri”...

Paket Yöneticileri (Package Managers)

RedHat'ın “rpm” paket yöneticisi; Debian'in ise “dpkg” si var. Mandrake ve Mandriva da RedHat'ın “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 Rpm Drake’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 bilgisayarınıza 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 dağıtımını 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 çalıştıracaklarını 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:

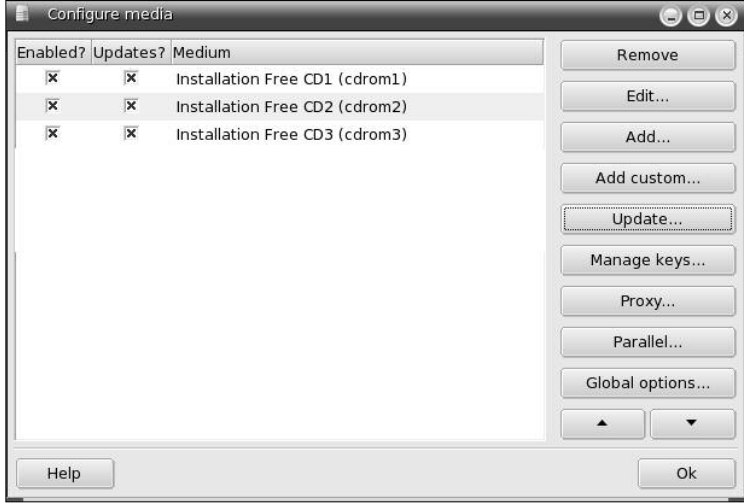


Bu örnekte, Kopete yazılımının yüklenebilmesi için “libkdenetwork2-kopete” kütüphanesinin de kurulması gerektiği belirtilmekte ve bunun için izniniz istenmektedir. Paket yöneticisi kullanmanın en önemli avantajı işte bu tip bağımlılıkların (dependency) otomatik olarak çözümlenmesidir.

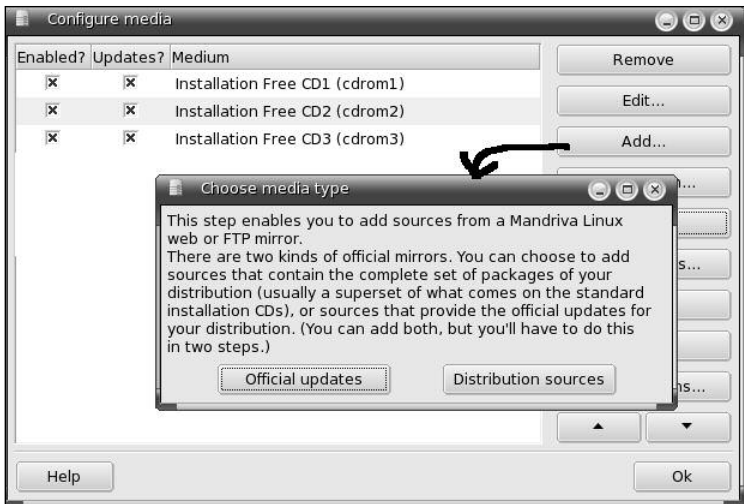
Rpm Drake paket yöneticisi, yeni yüklenecek, kaldırılacak ve en önemlisi güncellenecek paketleri, “media” adı verilen çeşitli kaynaklardan izler. “media” adı verilen bu kaynaklar, kurulum için kullandığınız CD’ler ve İnternet üzerinden erişilebilen Mandriva yazılım depolarıdır (İng.: Repositories).

Kim Korkar LINUX'tan?

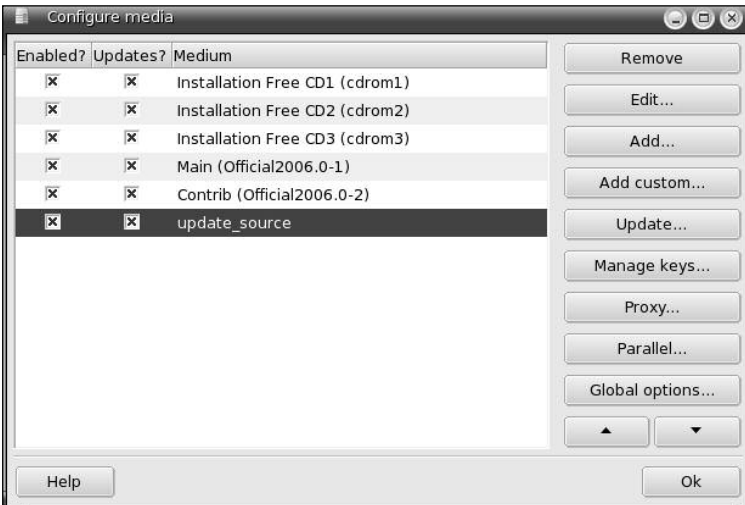
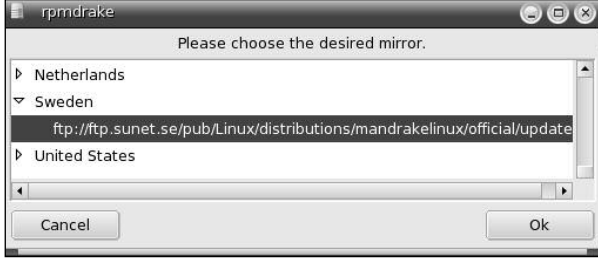
Yazılım ve güncelleme paketlerin indirilip yüklenebileceği kaynaklar rpmdrake yazılımının “Configure Media” işlevi ile belirtilir. CD’den yapılan standart bir kurulumda, bu yazılım kaynakları sadece 3 dağıtım CD’sinden oluşur.



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.



rpmrake 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;



Kim Korkar LINUX'tan?

rpm-drake için yazılım deposu tanımlarını bu şekilde tamamladıktan sonra artık yeni yazılımlar yüklemek; yüklü yazılımların sürüm ve güvenlik güncellemelerini yapmak çok kolaylaşacaktır.

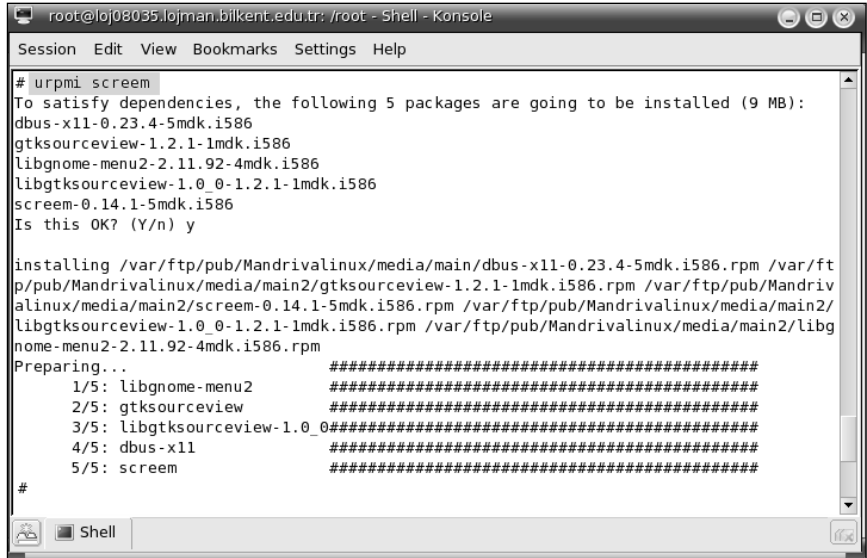
urpmi - Gerçek Paket Yöneticisi

rpm-drake 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 “screen” kurmak isterseniz; bu işi rpm-drake kullanılarak grafik bir arayüz kullanarak yapabileceğiniz gibi root kimliğine sahip olduğunuz bir terminal penceresinden

urpmi screen

komutunu da verebilirsiniz. urpmi, screen 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 screen
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
screen-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/screen-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: screen #####
#
```


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 - Konsolle <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.

Açıkçası, 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 de sendeyse; 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 belirtebilir. 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 üretilecek 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.)

Kim Korkar LINUX'tan?

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 da 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

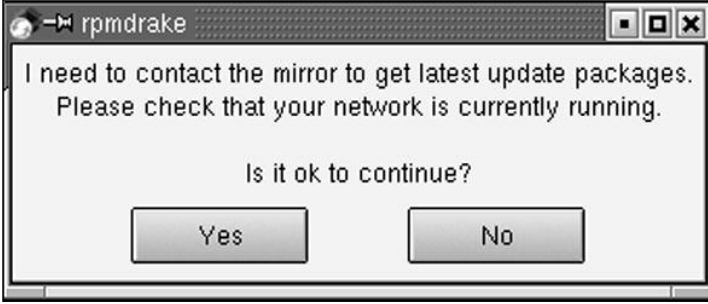
KDE SystemConfiguration → Packaging → Mandriva Update

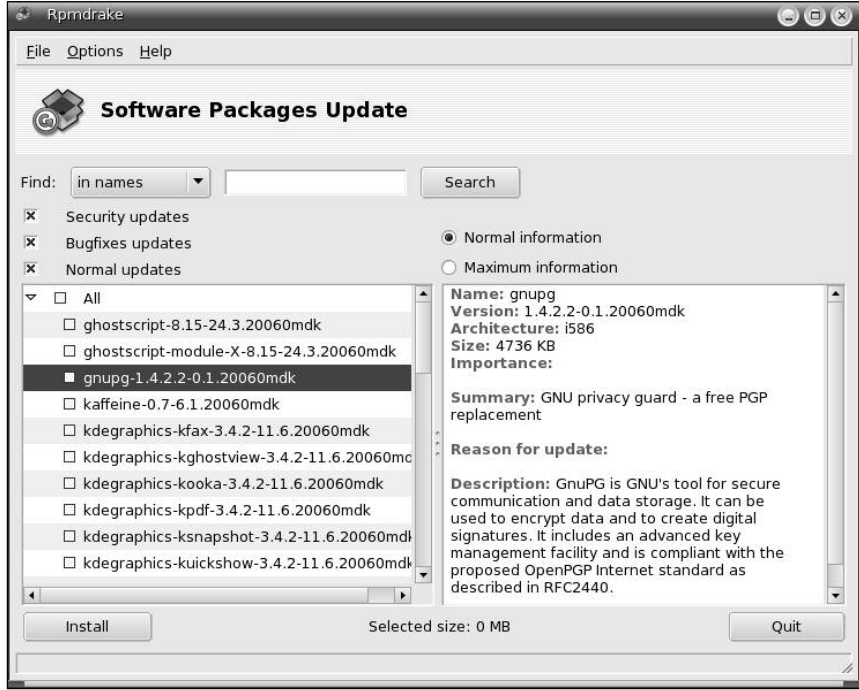
veya Mandriva Linux **Control Center** yazılımını başlatmak için

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 yansuları 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

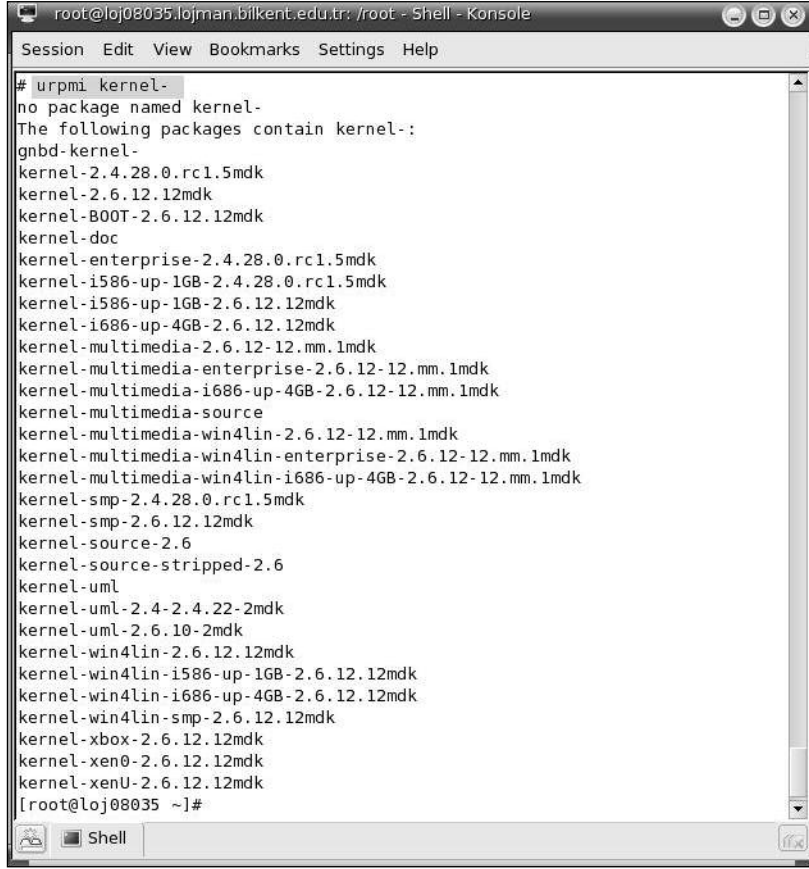
```
# urpmi kernel-
```

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 - Konsolu
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-B00T-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?