

## 17.Doküman

## Kurma-Kaldırma-Güncelleme İşlemleri

Konu Etiketleri

güncelleme , apt , apt-cache , apt-get , dpkg , update , upgrade , dist-upgrade , autoclean , autoremove , repo , depo , repository , install , remove , purge , kaynak koddan derleyerek kurulum

# Kurma-Kaldırma-Güncelleme İşlemleri

Linux işletim sistemini kullanacaksak mutlaka bilmemiz gerekenler arasında; sistemi güncelleme, program kurma ve kurulu programları kaldırma gibi işlemleri yerine getirmek var. Bu işlemleri nasıl yerine getirebileceğimizi bu konunun devamında öğreneceğiz. Anlatıma ilk olarak sistemi güncelleme işlemi ile başlayalım.

## Sistemi Güncelleme

Linux sistemleri, kullanıcının ihtiyacı olduğunda, programa kolayca ulaşabilmesini sağlayacak program paketlerini içinde bulunduran kendi paket *depolarına(repository)* sahiptirler. Farklı Linux dağıtımları için bu paketler de farklılık gösterebiliyor. Bu yüzden farklı Linux dağıtımlarının da kendi paketleri üzerinde işlem yapabilmek için farklı komutları vardır.

Dağıtımlar ve kullanılan pakete göre komutları aşağıdaki tabloda verilmiştir.

Dağıtım	Paketler	Komutlar
Debian	.deb	apt , apt-cache , apt-get , dpkg
Ubuntu	.deb	apt , apt-cache , apt-get , dpkg
CentOs	.rpm	yum
Fedora	.rpm	dnf
FreeBSD	.txz	make , pkg

Debian paketlerinin **.deb** uzantılı olduğunu yukarıdaki tabloda gördük. Kullandığımız dağıtım olan **Kali'de Debian tabanlı** olduğu için anlatıma bu doğrultuda devam edeceğiz.

Anlatıma güncelleme işleminde kullanılan komutlarımızı tanıyarak devam edelim.

**apt-get update**

Komutu **sources.list** dosyasına eklemiş olduğumuz repolara bakarak paket listelerini kontrol edip paketlerin son sürümleri ve bağımlılıkları hakkında bilgi almak için bunları **"günceller"**. Yani bu komutumuz; güncelleme işleminden önce, nelerin güncellenmesi gerektiğine bakarak sistemimizdeki sürümünden yüksek sürümleri bulunan yani güncellenmesi gereken doğru paketlerin güncellenmesini sağlıyor. Kısaca bu komutumuzun amacı sadece depolarda yer alan yenilikleri kontrol etmektir.

**apt-get upgrade**

Komutu da **apt-get update** komutunun depolardan kontrol edip bildirmiş olduğu güncellenmesi gereken paketleri en son versiyonlarına günceller.

**apt-get dist-upgrade**

Komutu ise **apt-get upgrade** komutundan farklı olarak sadece güncelleme yapmakla kalmaz, sistemimizdeki gereksiz paketleri de siler.

**apt-get clean**

Komutuyla, kurmak üzere indirmiş olduğumuz paketlerin hepsini silebiliyoruz. Depodan indirmiş olduğumuz tüm paketler ve uygulamanın çalışması için gereken bağımlılıklar **.deb** uzantısı ile arşivlenerek **/var /cache /apt /archives** dizini içerisinde daha sonra tekrar kullanılma ihtimaline karşı tutuluyorlar. İşte bizler de **apt-get clean** komutu yardımıyla eğer internet bağlantımızda sorun yoksa yani bu paketleri tekrar indirirken sorun yaşamayacaksak bu paketleri silerek sistemimizde yer işgal etmelerini önlemiş oluyoruz.

**apt-get autoclean**

Komutu da **apt-get clean** komutuyla benzer şekilde arşivlenmiş paketleri silme işlemini yapar. Fakat burada silinen arşivler bütün arşiv paketleri değil sadece eski sürüm olup artık kullanımda olmayan ve depolardan kaldırılmış paketlerdir.

**apt-get autoremove**

Komutu ise silmiş olduğumuz uygulamadan geriye kalan ve artık ihtiyaç duyulmayan bağımlılıkları kaldırmamızı sağlıyor.

**-y**

**-y** parametresinin göreviyse; çıkacak olan onay sorularına **evet(yes)** cevabını otomatik olarak vermektir. Bu sayede gerekli işlemler bizden onay beklemeden otomatik olarak zaman kaybedilmeden yapılmış olur.

Yani bir bütün olarak eğer sistemimizi güncellemek istersek ilk başta **/etc/apt/sources.list** konumunda yer alan **sources.list** dosyasına kullandığımız versiyona uygun depoları ekliyoruz.

Bunun için [buradaki kaynaktan](#) kullandığınız versiyona uygun olan **repository kaynağını** kopyalayarak **sources.list** dosyasına eklemeniz gerekiyor. Ben Kali 2016.1 sonrası (bu anlatımda kullandığım versiyon 2017.3) versiyonunu kullandığım için aşağıdaki repoları(repository) **sources.list** dosyasına ekliyorum.

```
deb http://http.kali.org/kali kali-rolling main contrib non-free
```

Ancak dediğim gibi sizler kullandığınız versiyona uygun olan repoları seçmelisiniz.

**Kali 1.0 ve sonrası için:**

```
deb http://old.kali.org/kali moto main non-free contrib
```

**Kali 2.0 ve sonrası için:**

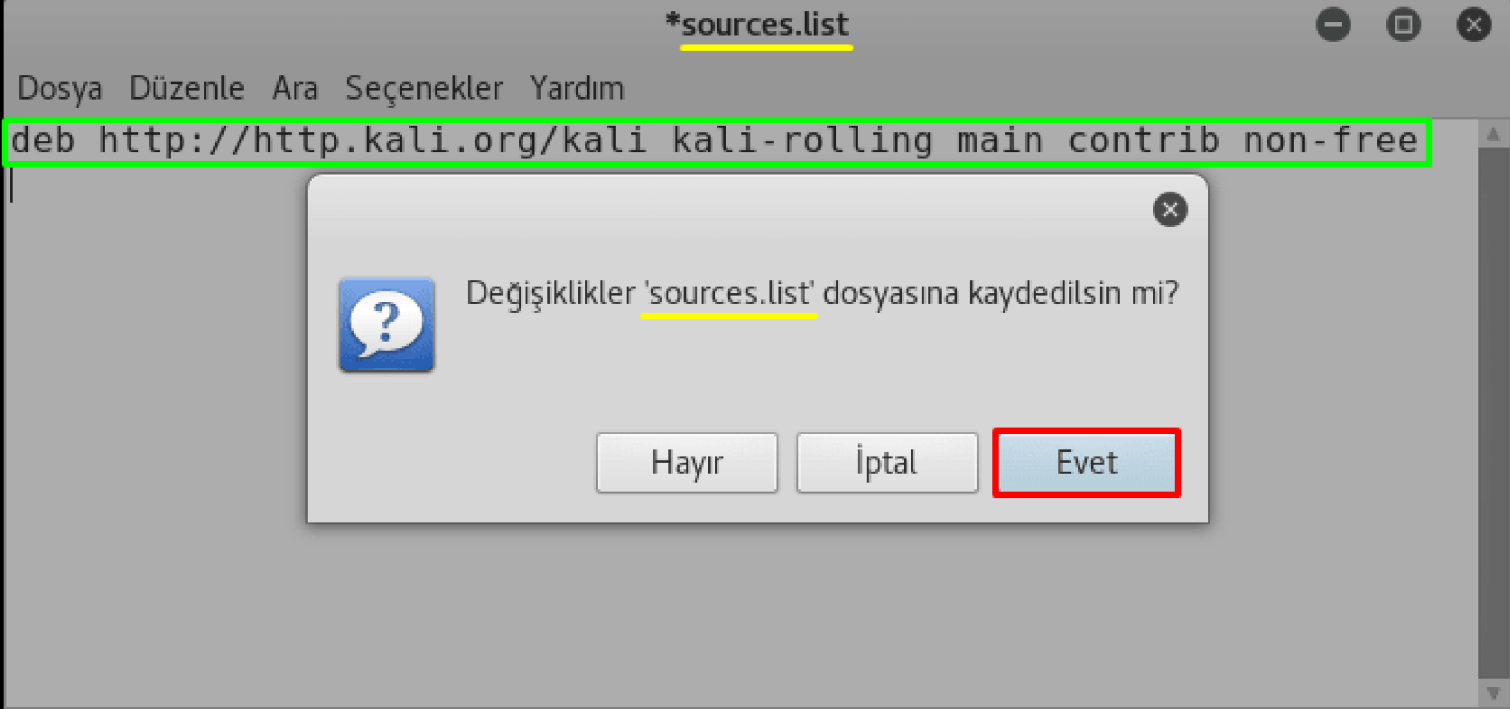
```
deb http://old.kali.org/kali sana main non-free contrib
```

**Kali 2016.1 ve sonrası için:**

```
deb http://http.kali.org/kali kali-rolling main contrib non-free
```

Repoları, **leafpad /etc/apt/sources.list** komutu ile açarak **sources.list** dosyasına ekledim.

```
root@taylan:~# leafpad /etc/apt/sources.list
```



Gerekli depo kaynağını ekledikten sonra sıra geldi güncelleme işlemine, bunun için ilk olarak paket bilgilerini güncellemek üzere **apt-get update**, daha sonra güncel paket bilgileri alınan uygulamaları güncellemek için de **apt-get upgrade -y** komutunu verelim. Ayrıca bu komutları **apt-get update && apt-get upgrade -y** şeklinde birleşik olarak da kullanabiliriz. Özellikle sistemimizi kurulum sonrası ilk defa güncellediğimizde bu güncelleme işlemi internetimizin de hızına bağlı olarak oldukça uzun sürebiliyor. Bunun dışında zaten üst kısımlarda hangi komutun hangi işlevde olduğunu öğrendiğimiz için tek tek kullanımlarına örnek vermiyorum sizler yerine göre ilgili komutları kullanabilirsiniz.

```
root@taylan:~# leafpad /etc/apt/sources.list
root@taylan:~# apt-get update && apt-get upgrade -y && apt-get dist-upgrade -y
0% [Çalışıyor]
```

Güncelleme işlemi boyunca, kullandığımız **-y** parametresi sayesinde herhangi bir soru sorulmadan bütün işlemler otomatik olarak gerçekleşecek ve güncelleme işlemi hızlı bir biçimde tamamlanacaktır.

## Program Kurmak

Linux'ta program kurmak için birden fazla yöntem bulunuyor. Bunlardan bir tanesi kullandığımız dağıtıma uygun programı, **paket yönetim sistemi** ile kurmaktır. Diğer bir yol, programı **kaynak koddan derleyerek** kurmaktır. Diğer seçenek ise dağıtımın kullandığı **depolardan(repository)** otomatik kurulum yapmaktır.

## Depodan Kurulum

Depoda bulunan programların kurulumlarını yaparken **apt-get install program\_adı** komut bütünü kullanılıyor. Unutmayın depodan(repository) kurulum yaparken sisteminizin güncel olması önemlidir. Şayet sisteminizi güncel tutmuyorsanız yani repolarınız güncel değilse **depodan program yükleme** çabalarınız hüsrarla sonuçlanabilir.

Ancak programın yüklenmeme sebebi bir tek güncelleme işlemi ile ilgili değil. Şayet yüklemek istediğimiz program depolarda yer almıyorsa depodan yükleme işlemimiz de haliyle başarısız olacaktır.

Bu yüzden öncelikle kurmak istediğimiz program depolarda yer alıyor mu ona bakalım. Ben örnek olması açısından depolarda *filezilla* aracını araştırıyorum eğer depolarda varsa kurulum yapabiliriz. Depoları kontrol etmek üzere konsola **apt-cache search filezilla** şeklinde komutumu vererek *filezilla* aracını depolarda var mı diye kontrol ediyorum.

Ve gördüğümüz gibi *filezilla* aracı depolarda mevcutmuş. Sizler de bu şekilde depolarda yer alan programları **apt-cache search program\_adı** şeklinde aratarak sorgulayabilirsiniz.

Örnek olması açısından *filezilla* isimli bir programın depodan kurulumunu yapmak için konsola **apt-get install filezilla** komutunu girdim.

```
root@taylan:~/Music#
```

Ve program kurulmuş oldu. Konsola **filezilla** yazarak da bu durumu teyit ettim.

Şayet kurduğunuz programı **kaldırmanız(silmeniz)** gerekirse konsola **apt-get remove program\_adı** şeklinde girdiğinizde ilgili program otomatik olarak sistemden kaldırılacaktır.

```
root@taylan:~#
```

Sadece programı kaldırmakla yetinmeyip bir de programın yapılandırma dosyalarını da sistemden kaldırmak isteyebilirsiniz. **Yapılandırma dosyaları ile birlikte programı kaldırmak için** komutu;

**apt-get --purge remove program\_adı**

```
root@taylan:~# a
```

veya

```
apt-get purge program_adı
```

```
root@taylan:~#
```

şeklinde kullanabilirsiniz.

Bu sayede program yapılandırma dosyaları ile birlikte tamamen silinmiş oluyor.

İşte sizler de bu şekilde depolarda yer alan programları oldukça kolay şekilde kurup kaldırabilirsiniz.

## Paket Yönetim Sistemi İle Kurulum



Bu işlem için kullandığımız dağıtıma uygun derleyiciyi kullanmalıyız. Daha önce Kali'nin **.deb** uzantılı paketleme sistemi olduğunu öğrenmiştik. Bu yüzden biz **.deb** uzantılı kurulum paketlerini açmak için **dpkg** komutunu kullanıyoruz. Sanırım kodun kısaltmasının nereden geldiğini bilirsek daha kolay akılda kalabilir. Kodun kısaltması "**debian package(debian paketi)**" kısaltmasından gelmektedir. Ayrıca **dpkg** komutunu kullanmadan yardımcı bir paket yöneticisi programı(**synaptic**) kullanarak da kurulum işlemlerini yerine getirebiliriz. Konumuza ilk olarak **dpkg** komutu ve kullanımı ile devam edelim.

Örnek olması açısından ben "*master pdf*" adında bir programın kurulumunu ele aldım. Bunun için öncelikle programın **.deb** uzantılı dosyasını programın kendi sitesinden indirdim.

Şimdi programı kurmak için **dpkg -i paket\_adı.deb** şeklinde komutumu giriyorum.

```
root@taylan:~/Downloads#
```

Ve programım kurulmuş oldu.

**Kurduğumuz programı kaldırmak** istersek komutumuzu **dpkg -r program\_adı** şeklinde yani **-r** parametresini ekleyerek kullanıyoruz.

```
root@taylan:~/Downloads# d
```

Program otomatik olarak kaldırılmış oldu. Ancak **/etc** dizini altındaki konfigürasyon dosyaları silinmedi. Eğer bu dosyaları da kaldırmak istersek **-P** parametresini kullanabiliriz.

```
root@taylan:~#
```

Bu sayede programa ait konfigürasyon dosyaları da tamamen kaldırılmış oldu.

Konfigürasyondan bahsetmişken aklınızda bulunsun eğer yüklediğimiz paketin konfigürasyon ayarlarını tekrar yapılandırmamız gerekirse **dpkg-reconfigure paket\_adı** şeklinde komutumuzu kullanabiliriz.

Kurulu olan tüm paketler hakkında bilgi almak için **-l** parametresi kullanılabilir.(Çıktı uzun olduğundan kısa kesilmiştir.)

```
root@taylan:~# dpkg -l
İstenen=bilinmeyen(U)/kur(I)/kaldır(R)/temizle(P)/tut(H)
| Durum=kurulu-değil(N)/kurulu(I)/yapılandırma-dosyası(C)/açılmış(U)
|/yarı-yapılandırılmış(F)/yarı-kurulu(H)/tetikleyici-bekliyor(W)
|/tetikleyici-askıda(T)/ Hata?=(yok)/tekrar-kur(R) (Durum,Hata: büyük-harf=kötü)
||/ Ad Sürüm Mimari Açıklama
+++-----
ii 0trace 0.01-3kali1 amd64 A traceroute tool that can run within an existing
ii aapt 1:7.0.0+r33-1 amd64 Android Asset Packaging Tool
ii accountsservice 0.6.45-1 amd64 query and manipulate user account information
ii ace-voip 1.10-1kali5 amd64 A simple VoIP corporate directory enumeration tool
ii acl 2.2.52-3+b1 amd64 Access control list utilities
ii acpi 1.7-1.1 amd64 displays information on ACPI devices
ii adduser 3.118 all add and remove users and groups
ii adwaita-icon-theme 3.30.0-1 all default icon theme of GNOME
ii afflib-tools 3.7.17-4 amd64 Advanced Forensics Format Library (utilities)
ii aglfn 1.7-3 all Adobe Glyph List For New Fonts
ii aircrack-ng 1:1.4-3 amd64 wireless WEP/WPA cracking utilities
ii albatross-gtk-theme 1.7.4-1 all dark and light GTK+ theme from the Shimmer Project
ii alsa-tools 1.1.3-1 amd64 Console based ALSA utilities for specific hardware
ii amap 5.4-4kali1 amd64 next-generation scanning tool for pentesters
```

Çıktıda yer alan paketlerin sol tarafındaki ifadelerin anlamı:

**ii** : paket normal olarak sisteme yüklendi.

**rc** : paket yüklendikten sonra silindi ancak konfigürasyon dosyaları halen mevcut.

**pn** : paket konfigürasyon dosyaları ile birlikte sistemden kaldırıldı.

Kurulu paketin durumunu öğrenmek için **-s** parametresini kullanabiliriz. Ben örnek olması açısından *leafpad* programı hakkında bilgi almak için komut satırına **dpkg -s leafpad** komutunu giriyorum.

```
root@taylan:~# dpkg -s leafpad
Package: leafpad
Status: install ok installed
Priority: optional
Section: editors
Installed-Size: 454
Maintainer: Paulo Roberto Alves de Oliveira (aka kretcheu) <kretcheu@gmail.com>
Architecture: amd64
Version: 0.8.18.1-5
Depends: libc6 (>= 2.4), libcairo2 (>= 1.2.4), libglib2.0-0 (>= 2.24.0), libgtk2.0-0 (>= 2.10.0), libpango-1.0-0 (>= 1.14.0), libpangocairo-1.0-0 (>= 1.14.0)
Suggests: evince-gtk
Description: GTK+ based simple text editor
 Leafpad is a simple GTK+ based text editor with user interface
 similar to Notepad. It aims to be lighter than Gedit and KWrite,
 and to be as useful as them.
Homepage: http://tarot.freeshell.org/leafpad/
root@taylan:~#
```

Kurulu paketin içeriğini öğrenmek istersek **-L** parametresini kullanırız.

```

root@taylan:~# dpkg -L leafpad
/.
/usr
/usr/share
/usr/share/man
/usr/share/man/man1
/usr/share/man/man1/leafpad.1.gz
/usr/share/doc
/usr/share/doc/leafpad
/usr/share/doc/leafpad/changelog.Debian.gz
/usr/share/doc/leafpad/changelog.gz
/usr/share/doc/leafpad/copyright
/usr/share/doc/leafpad/README
/usr/share/locale
/usr/share/locale/zh_TW

```

Eğer indirmiş olduğumuz **.deb** uzantılı dosyanın içeriğini henüz kurmadan görmek istersek **-c** parametresini kullanabiliriz.

```

root@taylan:~/Downloads# ls
master-pdf-editor-4.3.61_qt5.amd64.deb  rtl8192eu-linux-driver-realtek-4.4.x.zip
rtl8192eu-linux-driver-realtek-4.4.x
root@taylan:~/Downloads# dpkg -c master-pdf-editor-4.3.61_qt5.amd64.deb
drwxrwxr-x root/root      0 2016-08-24 22:03 ./
drwxrwxr-x root/root      0 2016-08-24 22:03 ./usr/
drwxrwxr-x root/root      0 2016-08-24 22:03 ./usr/share/
drwxrwxr-x root/root      0 2016-08-24 22:03 ./usr/share/icons/
drwxrwxr-x root/root      0 2016-08-24 22:03 ./usr/share/icons/hicolor/
drwxrwxr-x root/root      0 2016-09-17 19:31 ./usr/share/icons/hicolor/96x96/
drwxrwxr-x root/root      0 2016-10-09 10:23 ./usr/share/icons/hicolor/96x96/apps/
-rw-rw-r-- root/root    2661 2016-10-04 16:10 ./usr/share/icons/hicolor/96x96/apps/masterpdfedi
tor4.png
drwxrwxr-x root/root      0 2016-08-24 22:03 ./usr/share/icons/hicolor/32x32/
drwxrwxr-x root/root      0 2016-10-09 10:22 ./usr/share/icons/hicolor/32x32/apps/
-rw-rw-r-- root/root    1064 2016-10-04 16:06 ./usr/share/icons/hicolor/32x32/apps/masterpdfedi
tor4.png
drwxrwxr-x root/root      0 2016-08-24 22:03 ./usr/share/icons/hicolor/64x64/
drwxrwxr-x root/root      0 2016-10-09 10:23 ./usr/share/icons/hicolor/64x64/apps/
-rw-rw-r-- root/root    2166 2016-10-04 16:08 ./usr/share/icons/hicolor/64x64/apps/masterpdfedi
tor4.png
drwxrwxr-x root/root      0 2016-08-24 22:03 ./usr/share/icons/hicolor/16x16/
drwxrwxr-x root/root      0 2016-10-09 10:22 ./usr/share/icons/hicolor/16x16/apps/
-rw-rw-r-- root/root     493 2016-10-04 16:03 ./usr/share/icons/hicolor/16x16/apps/masterpdfedi
tor4.png

```

Sistemde kurulmuş ve kaldırılmış tüm paketleri görmek istersek **dpkg --get-selections** komutunu kullanabiliriz.



```

root@taylan:~# dpkg --get-selections
0trace                                install
aapt                                  install
accountsservice                      install
ace-voip                             install
acl                                  install
acpi                                  install
adduser                              install
adwaita-icon-theme                   install
afflib-tools                         install
aglfn                                install
aircrack-ng                          install
albatross-gtk-theme                  install
alsa-tools                           install

```

Bu paket listesini daha sonra kullanmak üzere bir dosya içerisine yedeklemek istersek komutumuzu `dpkg --get-selections >dosya_adı.txt` şeklinde kullanmamız yeterli olacaktır.

```

zenity                                install
zenity-common                        install
zenmap                              install
zerofree                             install
zim                                  install
zip                                  install
zlib1g:amd64                         install
zsh                                  install
zsh-common                           install
root@taylan:~/Downloads# dpkg --get-selections >program_bilgileri.txt
root@taylan:~/Downloads# ls
master-pdf-editor-4.3.61_qt5.amd64.deb  rtl8192eu-linux-driver-realtek-4.4.x
program_bilgileri.txt                  rtl8192eu-linux-driver-realtek-4.4.x.zip
root@taylan:~/Downloads# tail -n 5 program_bilgileri.txt
zim                                  install
zip                                  install
zlib1g:amd64                        install
zsh                                  install
zsh-common                           install
root@taylan:~/Downloads#

```

Yedeklediğimiz bu program listeleri sayesinde başka bir sisteme aynı paketleri tek seferde yüklememiz mümkün. Bunun için komutlar aşağıdaki şekildedir.

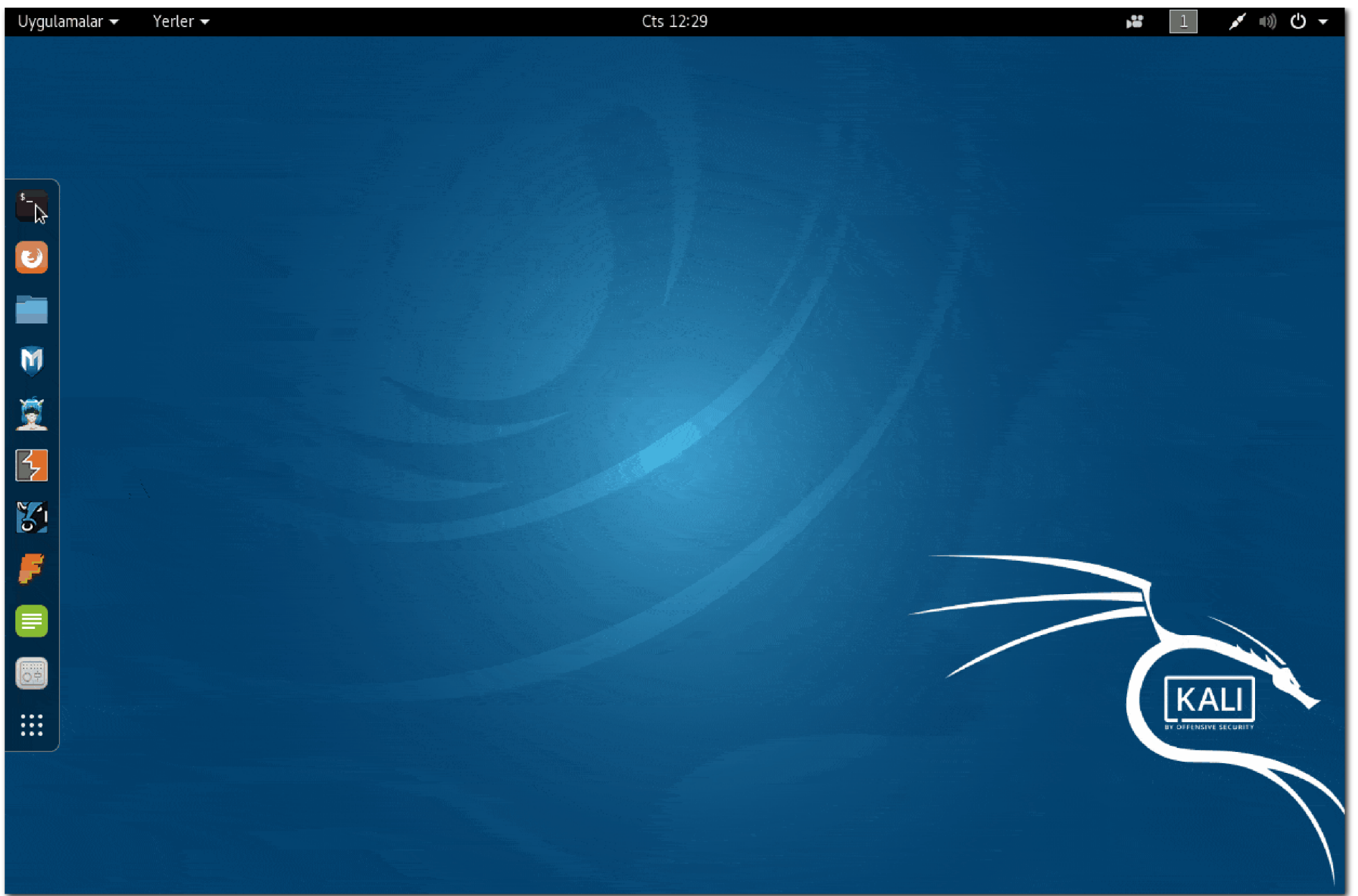
```
dpkg --set-selections > yedek_dosya_adı.txt
```

Daha sonra tanıma işlemi bittikten sonra eksik olanların yüklenmesi için konsola aşağıdaki komutu giriyoruz.

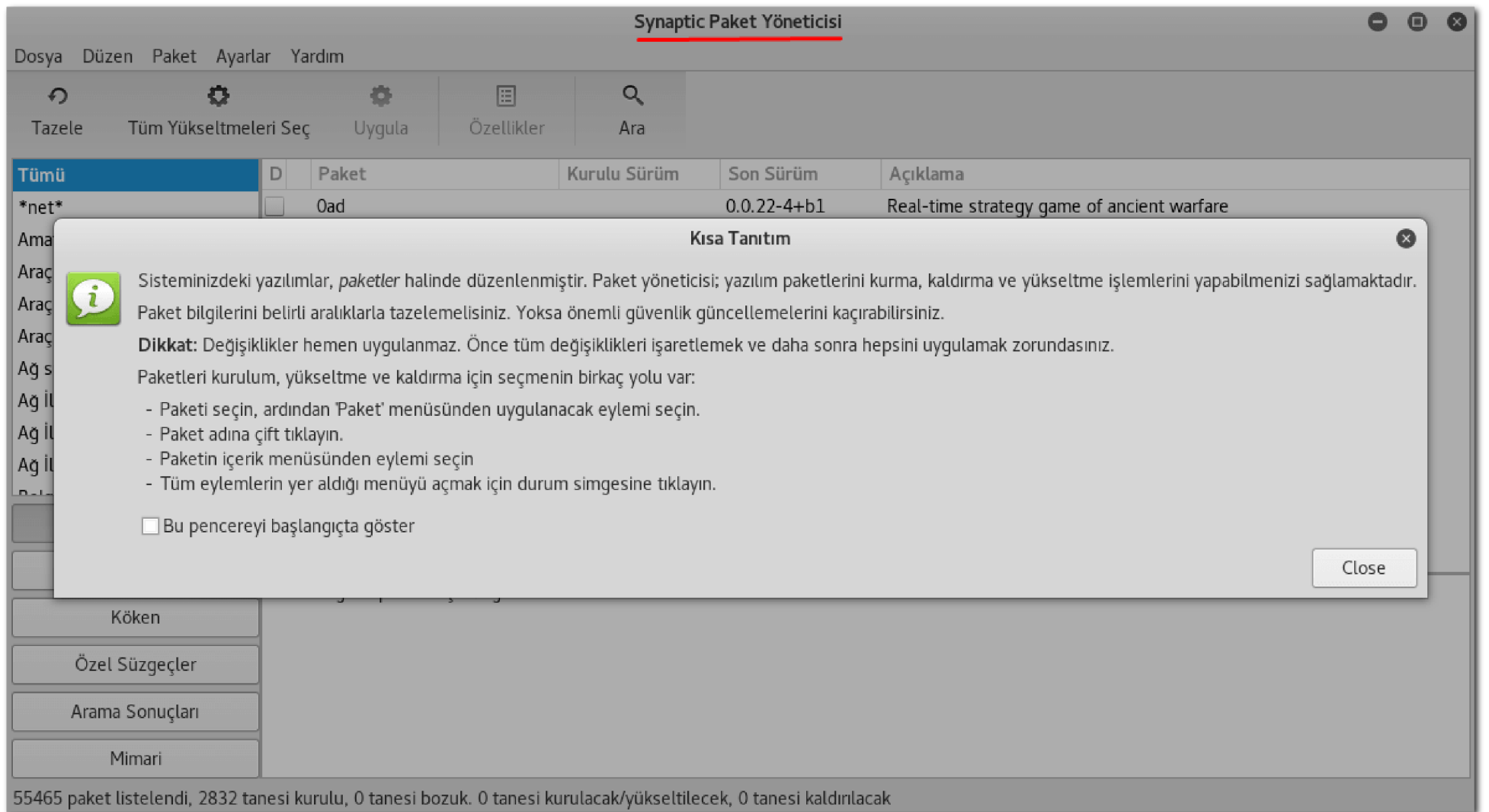
```
apt-get deselect-upgrade
```

Ayrıca bu işlemin uzun süreceğini unutmayın. Yani çok fazla spesifik programlarınız yoksa, bu yöntemi kullanmanızı tavsiye etmem. Bunun yerine kendiniz de pekâla elle kurulumları gerçekleştirebilirsiniz.

Şimdi sıra geldi ikinci yol olan yardımcı paket yöneticisi programı aracılığı ile program kurma-kaldırma-güncelleme işlemlerini yapmaya. Bunun için ilk olarak paket yönetim programı olan **synaptic** programını kurmamız gerek. Komut satırına `apt-get install synaptic` yazarak yardımcı programı depolardan indiriyoruz.



Evet "y" diyerek kurulumu onay veriyoruz ve programımız kurulmuş oluyor. Programı açmak için komut satırına **synaptic** komutunu vermemiz yeterli. Bu program(synaptic) sayesinde depolarda yer alan programları çok kolay şekilde kurup kaldırabilir ve programlar için diğer gerekli işlemleri yerine getirebiliriz. Programın kullanımı oldukça kolay olduğundan keşfini sizlere bırakarak anlatıma devam ediyorum.



## Kaynak Koddan Derleyerek Kurulum

Linux'a uyumlu sürümü bulunan açık kaynak kodlu yazılımların kaynak kod paketleri bu yazılımların internet sitelerinde, genellikle **tar.gz** biçiminde arşivlenmiş olarak indirilmeye sunulurlar. Bu arşivlenmiş olan kaynak kod paketi daha önce gördüğümüz **.deb** uzantılı paketlerin aksine ham şekilde yani derlenmemiş (kurulumu hazır olmayan) şekilde gelirler. Bu derleme işlemini bizim yaparak kurulumu gerçekleştirmemiz gerekir. Ancak her

kurulum aynı olmayabilir. **tar.gz** veya **tar.bz2** benzeri uzantılı paketlerde genellikle programın nasıl kurulacağına dair "**install**", "**readme**", "**configure**" ve benzeri isimlerde yönergeler bulunur. Bu dosyalar okunarak yükleme işlemi gerçekleştirilmelidir. Çünkü her programın gereksinimleri birbirinden farklıdır, yani kurulum için geliştirici ekibin verdiği kurulum yönergesini takip etmek en doğru tercih olacaktır. Ancak biz şimdi burada genel kurulum hakkında bilgi edinelim. Kaynak koddan derleyerek kurulum işleme genel olarak aşağıdaki şekildedir:

İndirdiğimiz program arşivini klasöre çıkartırız.

Ve arşiv içeriğini çıkarmış olduğumuz klasörden komut satırımızı çalıştırırız. İlgili konumda olduğumuzdan emin olduktan sonra;

Programı derlemeden önce gerekli kütüphaneler ve bağımlılıkları kontrol etmek için aşağıdaki komutu kullanırız.

```
./configure
```

Bu işlemin sonunda eksik paketler ve bağımlılıklar çıkarsa bunları tek tek kurmamız gerekecek.

Eğer bu aşamada;

```
bash: ./configure: Böyle bir dosya ya da dizin yok
```

gibi bir hata çıktısı alıyorsanız:

O program **configure** dosyası içermiyordur yani o program için **./configure** aşamasına gerek yoktur. Böyle bir durumla karşılaştığınızda bu aşamayı atlayarak direk **make** aşamasına geçin. Ancak **configure** dosyası varsa mutlaka **./configure** diyerek programın ihtiyacı olan bağımlılıkları kontrol edip eksik olanları yükleyin. Aksi halde program kurulum işlemi başarısız olacaktır. Şimdi devam edelim.

Gerekli kütüphaneler ve bağımlılıkları kontrol edip varsa eksik olanların kurulumlarını yaptıktan sonra, **make** aşaması için derleme yapacak paketleri **apt-get install build-essential** komutu ile kuruyoruz.

Daha sonra **make** diyerek programımızı derlemiş oluyoruz.

Son olarak **make install** komutu ile programı sistemimize kuruyoruz.

Silme işlemleri için de programa göre farklı durumlar ve dosyalar oluşabildiği için; kurulum yaparken olduğu gibi silme işlemi için de aynı şekilde programla birlikte gelen yönergeleri okumanız en doğru silme yönetimini size gösterecektir.

## Alıştırmalar Hakkında

Yalnızca okumak yetmez, öğrendiğiniz bilgilerin kalıcı olabilmesi için bolca alıştırmaya yapmalısınız. Doküman içerisindeki bilgileri pekiştirmek için aşağıdaki alıştırmalar ile başlayabilirsiniz. Elbette burada yer alan alıştırmalar faaliyetleri dışında, konuyu öğrendiğinizi hissetmeye kadar kendiniz de bolca pratik yapmayı da ihmal etmeyin lütfen. Aksi halde öğrendiğiniz bilgiler kısa sürede unutulup gidecektir.

**figlet** isimli aracın depolarda olup olmadığını kontrol edin.

Depolan **figlet** isimli aracı kurun.

Kurmuş olduğunuz **figlet** isimli aracı konsoldan kaldırın(silin).

Paket yönetim sistemini kullanarak **.deb** uzantılı bir kurulum paketini kurun.

Paket yönetim sistemi ile kurduğunuz programı kaldırın.

Sistem kurulu olan tüm paketleri konsoldan listeleyin.

## Geri Bildirimde Bulunun

Sizlere daha verimli bir kaynak sunabilmemiz için, uygulamada veya dokümantasyonlarda yer alan tüm hata ve eksiklerimizi bize bildirebilirsiniz.

Geri Bildirimde Bulunun