

sudo command bir root user gibi komutlar yazmayı sağlar. Ancak verilen çerçeve dahilinde hareket edilir.

sudo yetkisi dahilinde hangi kişilerin hangi komutları çalıştırabileceği /etc/sudoers altındaki bir klasörde listelenir. bu dosya içeriğini sudo cat /etc/sudoers komutuyla görebiliriz:

```
ubuntu@ip-172-31-85-28:~$ cat /etc/sudoers
cat: /etc/sudoers: Permission denied
ubuntu@ip-172-31-85-28:~$ sudo !!
sudo cat /etc/sudoers
#
# This file MUST be edited with the 'visudo' command as root.
#
# Please consider adding local content in /etc/sudoers.d/ instead of
# directly modifying this file.
#
# See the man page for details on how to write a sudoers file.
#
Defaults        env_reset
Defaults        mail_badpass
Defaults        secure_path="/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin:/snap/bin"
Defaults        use_pty

# This preserves proxy settings from user environments of root
# equivalent users (group sudo)
#Defaults:%sudo env_keep += "http_proxy https_proxy ftp_proxy all_proxy no_proxy"
```

Commands	Meaning
sudo -l	List available commands.
<b>sudo command</b>	<b>Run command as root.</b>
sudo -u root command	Run command as root.
sudo -u user command	Run command as user.
<b>sudo su</b>	<b>Switch to the superuser account.</b>
sudo su -	Switch to the superuser account with root's environment.
sudo su - username	Switch to the username's account with the username's environment.
sudo -s	Start a shell as root
sudo -u root -s	Same as above.
sudo -u user -s	Start a shell as user.

sudo -l hangi sudo komutlarını kullanabileceğimizi gösterir.

sudo yazmamız gereken bir komutu sudo yazmadan girdiysek, sudo !! komutu ile önceki komutu sudo ile çalıştırmasını sağlayabiliriz.

sudo command şeklinde komutu sudo ile gireriz.

sudo -u root command ile de bir komutu root olarak çalıştırabiliriz.

sudo -u user command ile bir komutu bir user adına çalıştırırız.

sudo su komutu ile root hesabına gidebilirken sudo su - komutu ile root environment ile root hesabına gideriz.

sudo su - username ile o userın environmenti ile o user ın hesabına gideriz.

sudo -s ve sudo -u root -s komutları root olarak başka bir shell başlatır.  
sudo -u user -s başka bir kullanıcı adına başka bir shell başlatır. bunlar sanal makine ya da başka makinelere bağlanırken kullanılan kodlar.

bir prompt olmadığını düşünelim, whoami komutu o anda hangi kullanıcı ile bağlandığımızı bize gösterir:

whoami

display one-line manual page descriptions.

## whoami

```
clarusway@DESKTOP-UN6T2ES:~$ whoami
clarusway
clarusway@DESKTOP-UN6T2ES:~$
```

who komutu, bağlanmış olduğumuz işletim sistemi birden fazla kullanıcı tarafından aktif olarak kullanılıyor olsun, sisteme kaç kişi hangi saatte hangi ip adres

who

provide with details about **who is logged on the system.**

## who

```
clarusway@DESKTOP-UN6T2ES:~$ who
root pts/0 2019-11-10 23:07 (10.104.33.101)
james pts/1 2019-11-10 23:30 (10.104.33.101)
john pts/2 2019-11-10 23:34 (10.104.33.96)
clarusway pts/3 2019-11-10 23:39 (10.104.33.91)
clarusway@DESKTOP-UN6T2ES:~$
```

w komutunu, tek harfli bir komut, who komutunun verdiği bilginin ötesinde kullanıcıların hangi processleri konuşturduğunu, ne kadardır sistemde olduklarını, kullanmış oldukları CPU değerlerini ve sistem kaynaklarından ne kadarını tüketiyorlar görmek isterseniz kullanırsınız:

w

inform who is logged on and what they are doing.

## w

```
clarusway@DESKTOP-UN6T2ES:~$ w
14:22:38 up 1:52, 0 users, load average: 0.52, 0.58, 0.59
USER TTY LOGIN@ IDLE JCPU PCPU WHAT
root pts/0 14:07 15.00s 0.01s 0.01s top
clarusway@DESKTOP-UN6T2ES:~$
```

id komutu, kullanıcı isimlerinin bir şey ifade etmeyebileceği ve karışabilen kullanıcı isimleri olabileceği için her kullanıcıya, kullanıcının ana grubuna id verilir ve bu komutla bu bilgiler görülür:

id

Display user id, your primary group id, and a list of the groups you belong to.

### **id [username]**

```
clarusway@DESKTOP-UN6T2ES:~$ id
uid=1000(clarusway) gid=1000(clarusway) groups=1000(clarusway)
```

default olarak ilk user a verilen id 1000 dir. ilk grup id de default olarak 1000 dir.

su username komutu başkasının hesabına geçebilmek için kullanılır. ancak o kullanıcının hesabına gitmeye yetkili olmaya, password unu bilmeye ihtiyacımız olacaktır. bu kodu genelde gerçek kişi kullanıcısında değil, tüzel kişilik gibi oluşturulan userlara girilirken kullanılır:

su

enables a shell to be run as another user.

### **su [username]**

```
clarusway@DESKTOP-UN6T2ES:~$ su oliver
Password:
oliver@DESKTOP-UN6T2ES:/home/clarusway$ _
```

su - username kodu ile de o kullanıcının environment variable ları ile o kullanıcıya geçmek için kullanılır.

sudo su ise root hesaba geçmek için kullanılır:

sudo su

The root user does not have a password set on some Linux systems like Ubuntu and Xubuntu. On these Linux systems, You can become root user via sudo su command.

### **sudo su** **sudo su -**

```
clarusway@DESKTOP-UN6T2ES:~$ sudo su
root@DESKTOP-UN6T2ES:/home/clarusway#
```

User Management:

etc/passwd

On Linux, the local user database is /etc/passwd.

```
clarusway@DESKTOP-UN6T2ES:~$ tail -5 /etc/passwd
clarusway:x:1000:1000:,,,:/home/clarusway:/bin/bash
john:x:1002:1002:john,room,work,home,other:/home/john:/bin/bash
oliver:x:1003:1003:oliver,room_1,work_1,home_1:/home/oliver:/bin/bash
aaron:x:1001:1001:aaron,,,:/home/aaron:/bin/bash
james:x:1005:1009:james,,,:/home/james:/bin/bash
clarusway@DESKTOP-UN6T2ES:~$
```

kullanıcı bilgileri, /etc/passwd altında tutulur. Her veri arasında , ler var. Girilmeyen veriler olduğunda , , , şeklinde bir görüntü oluşuyor.

Kullanıcı eklerken kullandığımız komut: (sudo ile çalışacak)

useradd

useradd command is used for creating a new user.

**useradd [username]**

```
root@DESKTOP-UN6T2ES:~# useradd -m -d /home/walter -c "walter clarus" walter
root@DESKTOP-UN6T2ES:~# tail -1 /etc/passwd
walter:x:1006:1006:walter clarus:/home/walter:/bin/sh
root@DESKTOP-UN6T2ES:~#
```

-m is used for forcing the creation of the home directory  
-d is used for setting the name of the home directory  
-c is used for setting a description

-m ile kullanıcıya bir home directory ayarlıyoruz.

-d ile bu home directory nin ismini belirlememi ağlar. kullanıcı ismi yerine farklı isim de verilebilir.

-c ile de kullanıcıyı describe edecek şekilde bir açıklama yazılabilir.

mustafa isminde bir kullanıcı ekleyelim:

/etc/passwd veya tail /etc/passwd komutlarıyla kullanıcıları görebiriliriz:

```
ubuntu@ip-172-31-85-28:~$ sudo useradd mustafa -m -d /home/mustafa_home -c "Mustafa - DevOps Engineer"
ubuntu@ip-172-31-85-28:~$ tail /etc/passwd
tcpdump:x:108:114:/:nonexistent:/usr/sbin/nologin
sshd:x:109:65534:/:run/sshd:/usr/sbin/nologin
pollinate:x:110:1:/:var/cache/pollinate:/bin/false
landscape:x:111:116:/:var/lib/landscape:/usr/sbin/nologin
fwupd-refresh:x:112:117:fwupd-refresh user,,,:/run/systemd:/usr/sbin/nologin
ec2-instance-connect:x:113:65534:/:nonexistent:/usr/sbin/nologin
_chrony:x:114:121:Chrony daemon,,,:/var/lib/chrony:/usr/sbin/nologin
ubuntu:x:1000:1000:Ubuntu:/home/ubuntu:/bin/bash
lxd:x:999:100:/:var/snap/lxd/common/lxd:/bin/false
mustafa:x:1001:1001:Mustafa - DevOps Engineer:/home/mustafa_home:/bin/sh
```



kullanıcı eklemek için bir diğer kod da `sudo adduser` ; bu aslında arka tarafta `useradd` ı kullanan bir komut.

user oluştururken bizi bilgilendiriyor ve password oluşturmamızı sağlıyor. Sonrasında ise ekleyeceğimiz kullanıcının bilgilerini eklemeye olanak sağlıyor. user eklerken kullanışlı bir komut:

ahmet isminde bir user oluşturup `tail` komutuyla oluşturulduğunu görelim:

```
ubuntu@ip-172-31-85-28:~$ adduser ahmet
adduser: Only root may add a user or group to the system.
ubuntu@ip-172-31-85-28:~$ sudo !!
sudo adduser ahmet
Adding user `ahmet' ...
Adding new group `ahmet' (1002) ...
Adding new user `ahmet' (1002) with group `ahmet' ...
Creating home directory `/home/ahmet' ...
Copying files from `/etc/skel' ...
New password:
Retype new password:
passwd: password updated successfully
Changing the user information for ahmet
Enter the new value, or press ENTER for the default
  Full Name []: Ahmet CELIK
    Room Number []: 245
    Work Phone []: 555999666
    Home Phone []: -
    Other []: Jr. Data Enginner
Is the information correct? [Y/n] y
ubuntu@ip-172-31-85-28:~$ tail -5 /etc/passwd
_chrony:x:114:121:Chrony daemon,,,:/var/lib/chrony:/usr/sbin/nologin
ubuntu:x:1000:1000:Ubuntu:/home/ubuntu:/bin/bash
lxd:x:999:100::/var/snap/lxd/common/lxd:/bin/false
mustafa:x:1001:1001:Mustafa - DevOps Engineer:/home/mustafa_home:/bin/sh
ahmet:x:1002:1002:Ahmet CELIK,245,555999666,-,Jr. Data Enginner:/home/ahmet:/bin/bash
```

genelde bir kullanıcı firmadan ayrıldığında kullanıcı user silinmez, ileriye dönük dondurulur. ancak bu şirket politikasıyla ilgilidir.

bir user ı silmek için kullanacağımız komut, `sudo userdel` komutudur. bu komutu `sudo userdel -r` şeklinde kullandığımızda kullanıcının home directory de silinir. -r kullanmazsak kullanıcının homder directory i alır:

```
ubuntu@ip-172-31-85-28:~$ sudo userdel ahmet
ubuntu@ip-172-31-85-28:~$ tail -5 /etc/passwd
ec2-instance-connect:x:113:65534::/nonexistent:/usr/sbin/nologin
_chrony:x:114:121:Chrony daemon,,,:/var/lib/chrony:/usr/sbin/nologin
ubuntu:x:1000:1000:Ubuntu:/home/ubuntu:/bin/bash
lxd:x:999:100::/var/snap/lxd/common/lxd:/bin/false
mustafa:x:1001:1001:Mustafa - DevOps Engineer:/home/mustafa_home:/bin/sh
ubuntu@ip-172-31-85-28:~$ ll /home
ahmet/ mustafa_home/ ubuntu/
ubuntu@ip-172-31-85-28:~$ ls /home
ahmet mustafa_home ubuntu
ubuntu@ip-172-31-85-28:~$
```

`usermod` -option value username ile kullanıcının bilgilerini modifiye edebiliyoruz.

`usermod --help` ile neleri değiştirebildiğimizi inceleyebiliriz.

Aşağıdaki örnekte usermod -c ile user ın description u aws solution architect olarak değiştirilmiş:

## **usermod -[option][value][username]**

```
root@DESKTOP-UN6T2ES:~# tail -1 /etc/passwd
walter:x:1004:1004:walter clarus:/home/walter:/bin/sh
root@DESKTOP-UN6T2ES:~# usermod -c 'aws solution architect' walter
root@DESKTOP-UN6T2ES:~# tail -1 /etc/passwd
walter:x:1004:1004:aws solution architect:/home/walter:/bin/sh
root@DESKTOP-UN6T2ES:~#
```

User password:

User password u sudo passwd username komutuyla değiştirilir. admin yetkisiyle girdiğimde bir kullanıcının şifresini bilmesem de değiştirebiliyorum:

```
ubuntu@ip-172-31-85-28:~$ sudo passwd mustafa
New password:
Retype new password:
passwd: password updated successfully
```

ancak user (burada mustafa) kendi password unu değiştirmek isterse öncelikle kendi şifresini girmesi gerekiyor.

```
$ whoami
mustafa
$ passwd
Changing password for mustafa.
Current password: █
```

password ler /etc/shadow altında tutuluyor. Bu dosya yüksek güvenli bir dosya. Şifreler burada yüksek güvenli bir krypto anahtarı ile tutuluyor. sudo tail /etc/shadow komutuyla son on password e bakalım:

```
sudo tail /etc/shadow
tcpdump*:19327:0:99999:7:::
sshd*:19327:0:99999:7:::
pollinate*:19327:0:99999:7:::
landscape*:19327:0:99999:7:::
fwupd-refresh*:19327:0:99999:7:::
ec2-instance-connect!:19327:0:99999:7:::
_chrony*:19327:0:99999:7:::
ubuntu!:19334:0:99999:7:::
lxd!:19334:0:99999:7:::
mustafa:$y$j9T$Pd1mjt1mJY1qFK8F144aK.$nVt3KoXBunh2UjXPuaEEBJ2G6CA8X8qlbUoC71n6mQ6:19334:0:99999:7:::
ubuntu@ip-172-31-85-28:~$ █
```

an altta mustafa userının passwordunu görüyoruz. Biz bu passwordu 12345 olarak ayarlamıştık ancak burada bir kriptografi anahtarı ile tutulduğunu görüyoruz.

/etc/login.defs dosyası password'un geçerlilik süresi, ne kadar süreden önce değiştirilmemesi gerektiği ve expire date'ten ne kadar önce kullanıcıya bilgi default password ayarlarını tutuyor.

```
clarusway@DESKTOP-UN6T2ES:~$ grep ^PASS /etc/login.defs
PASS_MAX_DAYS    99999
PASS_MIN_DAYS    0
PASS_WARN_AGE    7
clarusway@DESKTOP-UN6T2ES:~$
```

Exercise:

## Exercise

- Create a user named **devops**
- Set **devops** user password as **clarusway**
- Change devops user description as **the best cloud engineer**
- Switch to **devops** user
- Display information of **devops** user
- Go back to previous user
- Delete **devops** user with home directory

### Group Management:

Kullanıcılar çoklu gruplara üye olabilir. Grup üyelikleri /etc/group dosyasında tutuluyor.

groups koduyla sistemde hangi grupların olduğunu görebiliyoruz.

groups username koduyla kullanıcının hangi gruba dahil olduğunu görebiliyoruz.

```
ubuntu@ip-172-31-85-28:~$ groups
ubuntu adm dialout cdrom floppy sudo audio dip video plugdev netdev lxd
```

sudo groupadd groupname komutu ile yeni gruplar ekleyebiliriz, aws ve devops isminde iki grup ekleyelim:

```

ubuntu@ip-172-31-85-28:~$ sudo groupadd aws
ubuntu@ip-172-31-85-28:~$ sudo groupadd devops
ubuntu@ip-172-31-85-28:~$ tail /etc/group
fwupd-refresh:x:117:
admin:x:118:
netdev:x:119:ubuntu
lxd:x:120:ubuntu
_chrony:x:121:
ubuntu:x:1000:
mustafa:x:1001:
linux:x:1002:
aws:x:1003:
devops:x:1004:
ubuntu@ip-172-31-85-28:~$

```

gruplara ait kullanıcıları değiştirebileceğimiz kod: `sudo usermod -a -G groupname username`

-a append yani eklemeyi simgeliyor. -a yazmazsak bu kodla kullanıcıyı istediğimiz gruba ekler ancak diğer bütün gruplardan çıkarır.

```

ubuntu@ip-172-31-85-28:~$ usermod -a -G devops mustafa
usermod: Permission denied.
usermod: cannot lock /etc/passwd; try again later.
ubuntu@ip-172-31-85-28:~$ sudo !!
sudo usermod -a -G devops mustafa
ubuntu@ip-172-31-85-28:~$ tail /etc/group
fwupd-refresh:x:117:
admin:x:118:
netdev:x:119:ubuntu
lxd:x:120:ubuntu
_chrony:x:121:
ubuntu:x:1000:
mustafa:x:1001:
linux:x:1002:
aws:x:1003:
devops:x:1004:mustafa

```

`sudo groupmod -n newname oldname` komutuyla da grup ismini değiştirebiliyoruz. -n fonksiyonu bunu sağlıyor.

**`groupmod -n [newname] [oldname]`**

```

root@DESKTOP-UN6T2ES:~# groupmod -n ubuntu linux
root@DESKTOP-UN6T2ES:~# tail -3 /etc/group
aws:x:1007:walter
python:x:1008:oliver
ubuntu:x:1006:john,james,aaron
root@DESKTOP-UN6T2ES:~#

```

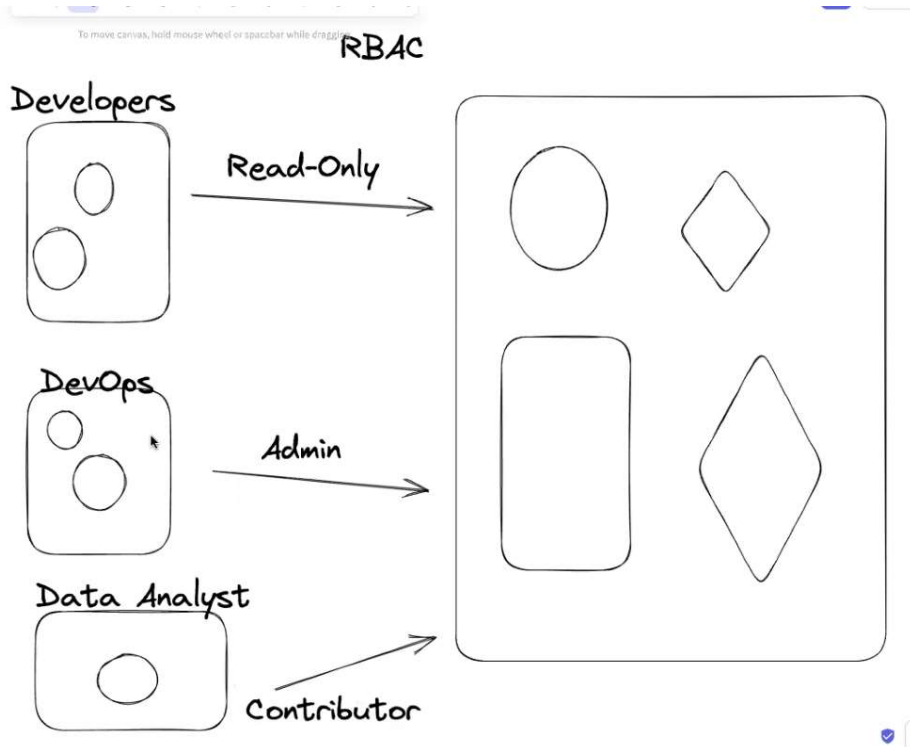


grubu silmek için de sudo groupdel komutu kullanıyoruz:

## groupdel [groupname]

```
root@DESKTOP-UN6T2ES:~# groupdel ubuntu
root@DESKTOP-UN6T2ES:~#
```

aws , azure, google cloud gibi cloud tabanlı sistemlerde de Linux'taki gibi user ve group management mantığı aynıdır. her yerde ufak değişikliklerle role based access hususu geçerlidir. gruplar ve yetkileri belirlenir, userlar gruplara dahil edilir. Aksi halde kullanıcılara tek tek rol atamak işleri çok zorlaştırır. Bu mantığı kavramamız gerekiyor. **RBAC**



gruba user eklerken kullandığımız bir başka komut daha var. bu komutla gruptan user da çıkarabiliyoruz:

sudo gpasswd -a username groupname eklemek için  
sudo gpasswd -d username groupname çıkarmak için kullanılır.

mustafa kullanıcısının devops grubundan önce çıkaralım sonra tekrar ekleyelim:

```
ubuntu@ip-172-31-85-28:~$ sudo gpasswd -d mustafa devops
Removing user mustafa from group devops
ubuntu@ip-172-31-85-28:~$ tail /etc/group
fwupd-refresh:x:117:
admin:x:118:
netdev:x:119:ubuntu
lxd:x:120:ubuntu
_chrony:x:121:
ubuntu:x:1000:
mustafa:x:1001:
linux:x:1002:
aws:x:1003:
devops:x:1004:
ubuntu@ip-172-31-85-28:~$ sudo gpasswd -a mustafa devops
Adding user mustafa to group devops
ubuntu@ip-172-31-85-28:~$
```

Bir user a admin yetkileri vermemiz için user ı sudo grubuna eklememiz gerekir. sudo grubu yukarıda bahsettiğimiz gibi /etc/sudoers altında tutulur.

## Using Package Managers

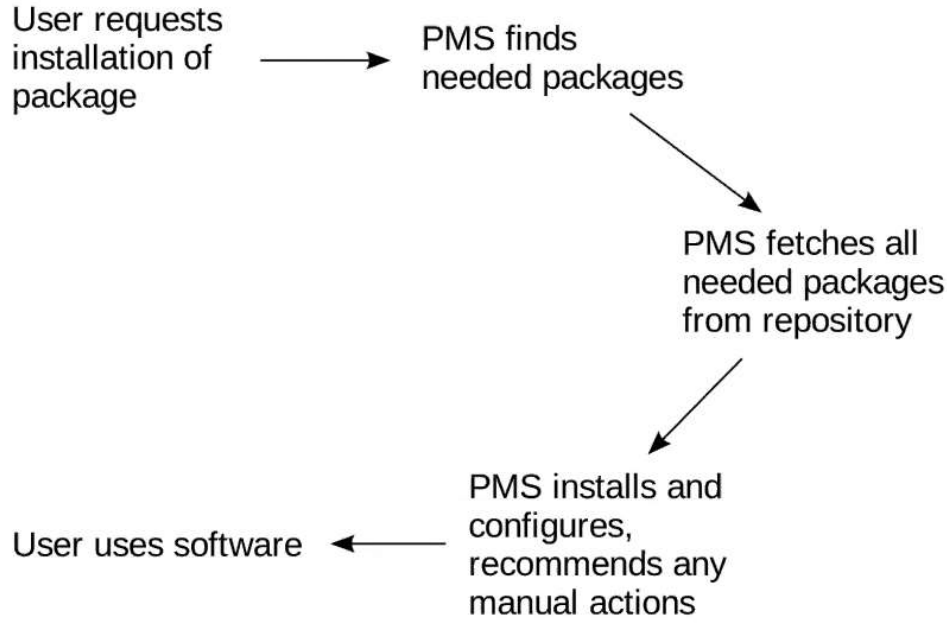
Aslında cep telefonlarımızda uygulama mağazasından uygulama indirirken arka planda package manager çalışıyor.

Önceleri yarı otomatik package managers kullanırken paket yüklerken dependency dediğimiz bağımlı paketlerin yüklü olmasından biz sorumluyduk. şu an package manager lar tam otomatik olarak çalışıyor ve bağımlı paketleri de yüklüyorlar.

Bir package manager (pakey yöneticisi) bilgisayarın işletim sistemine uygun ve uyumlu bir şekilde bilgisayar programlarının install, upgrade, configure ve remove edilmesi işlemlerinin sürecini otomatikleştiren bir yazılım araçları bütünüdür. Bir nevi office boydur ;)



Paketlerin dağıtımlarını, versiyonlarını, archive dosyalarının nerede olduğunu, isimlerini, amaçlarını, versiyon numaralarını, paketin doğru olduğuna dair konulan checksumlarını package managerlar tutarlar.



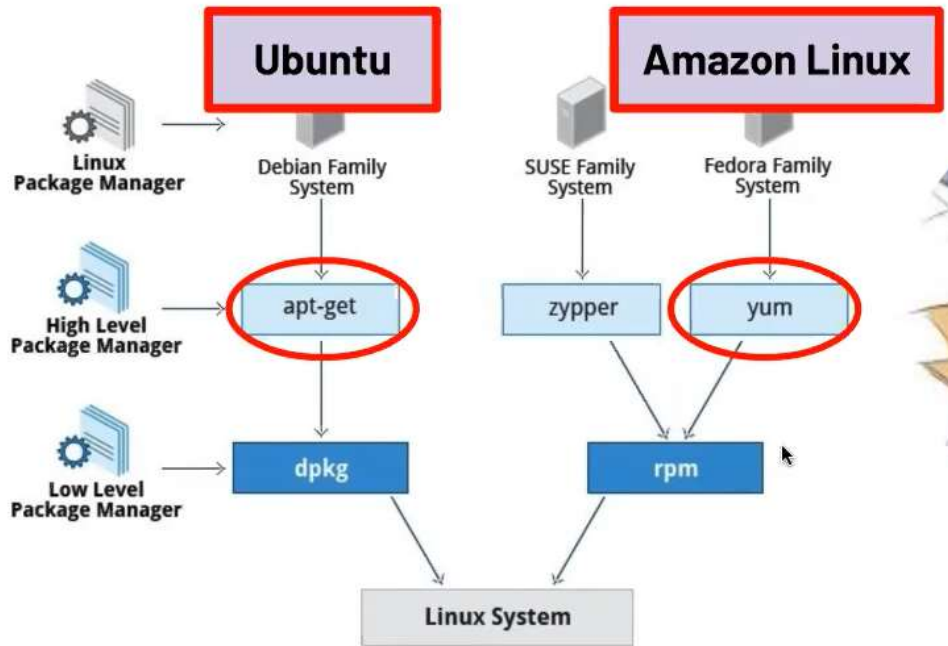
Git gibi bir toolu kullanma ihtiyacımız oldu diyelim, o paketin yüklenmesi için package manager kullanarak otomatik olarak bizim için yapmasını istiyoruz. Package manager bizim için gerekli paketleri bulur, öncesinde yüklenmesi gereken başka paketler varsa indirir, configure eder ve gerekli manuel hareketleri bildirerek kullanıma hazır hale getirir.

P.M (Package Manager) işletim sistemlerine göre farklılık gösterir.

Operating System	Format	Tool(s)
Debian	.deb	apt, apt-cache, apt-get, dpkg
Ubuntu	.deb	apt, apt-cache, apt-get, dpkg
CentOS	.rpm	yum
Fedora	.rpm	dnf
FreeBSD	Ports, .txz	make, pkg

<https://stackoverflow.com/questions/10286459/multiple-package-manager>

apt Ubuntu için kullanılan PM, yum Linux için:



dpkg Debian Linux distroları için main PM dir. yarı otomatiktir. Şu kodlarla kullanılır:

```

$ dpkg -i [package-name] # Installing a package
$ dpkg -r [package-name] # Removing a package
$ dpkg -l                 # Lists installed packages

```

dpkg yerine otomatik olarak bu işlemleri yapmak için debian tabanlı işletim sistemlerinde apt kullanıyoruz:

### APT (Advanced Package Tool)

• The **A**dvanced **P**ackaging **T**ool is what Ubuntu Software Center is built on



- 'apt-get install PACKAGE' will install and organize software
- 'apt-cache list PACKAGE' will search for PACKAGE in the local database
- 'apt-get update' update the local package database

```

$ apt update           # Update the installed packages
$ apt install [package-name] # Install a package and all its dependencies
$ apt remove [package-name] # Remove a package
$ apt purge [package-name]  # Remove a package and its configuration files

```

genelde apt yeterli oluyor, apt-get komutu da mevcut.



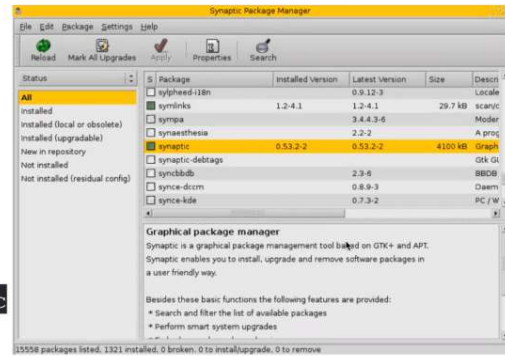
Linux u masaüstü görsel olarak da çalıştırıyorsak, Synaptic PM yi de kullanabiliyoruz:



### Synaptic Package Manager

Synaptic is a graphical package manager and used for installing, upgrading and removing single and multiple packages in a more user-friendly way.

```
sudo apt-get install synaptic
```



Red Hat distrosunda da rpm var, bu da yarı otomatiktir. yum da Red Hat'te tam otomatik olarak çalışır.

```
$ rpm -i [package-name] # Install a package
$ rpm -e [package-name] # Uninstall a package
```

```
$ yum install [package-name] # Install a package
$ yum remove [package-name] # Remove a package
$ yum update [package-name] # Update a package
```

## YUM (Yellowdog Updater Modified)

paket isimleri işletim sistemlerine göre değişir, yani apt-get için farklı isim yum için farklı isim olabilir. bu konuda google a danışmamız gerekiyor. İşletim sistemlerinin web sayfalarında da bu bilgilere ulaşabiliriz.

yum un biraz daha gelişmiş hali olan bir PM de dnf dir (dandified yum). Fedora 22 ve daha yeni distrolarda default olarak kullanılan PM dir. önce yum ile dnf yükleyip sonra kullanıyoruz:

```
$ yum install dnf # Install DNF via yum.
$ dnf --version # Checking DNF version
$ dnf install # Installing a package
```

ancak yum bize yeter de artar.

Olur da farklı işletim sistemleriyle çalışmak durumunda kalırsak bunlar da farklı PM ler:

## ➡ Other RPM tools:

- zypper (openSUSE)
- up2date (Red Hat Enterprise Linux, CentOS 3 and 4, and Oracle Linux)
- urpmi (Mandriva Linux, ROSA Linux, and Mageia)
- apt-rpm (Ark Linux,[11] PCLinuxOS and ALT Linux)
- smart (Unity Linux and Fedora)
- rpmquery (Red Hat Enterprise Linux)

## ➡ Below are a few more notable/interesting package managers.

- Portage: Package manager for Gentoo.
- Pacman: Arch Linux Package manager.
- Nix: A 'Fully Functional/Transactional' package manager.

•**Brew: An Open Source package manager for OSX.**

•**Chocolatey: A package manager for Windows.**



bazı yazılım dillerinin de default PM leri oluyor. bu diller için var olan kütüphaneleri aramak ve install etmek için kullanılıyorlar. python için pip3 (Flask yüklerken kullanmıştık)

Examples: **Python: pip** / Ruby: gem, rubygems / Haskell: cabal / NodeJS: npm

çalıştığımız şirketin kullandığı işletim sistemi ya da yazılım dili hangisiyse googlayarak PM sini ve hangi kodlarla kullanıldığını öğreneceğiz.

PM kodlarını kullanmak için de sudo kullanıyoruz.

Bazı yum kodlar:

## ► Deep Dive into yum



```
$ yum install [package-name]           # Install a package
$ yum -y install [package-name]        # Skip confirmations during installation
$ yum remove [package-name]           # Remove a package.
$ yum erase [package-name]            # Remove a package (an alias to remove).
$ yum autoremove [package-name]       # Remove a package and unused dependencies.
$ yum update [package-name]           # Update a package
$ yum update                           # Update all installed packages
$ yum info [package-name]             # Get information about a package
$ yum list                             # List all installed and available packages
$ yum list [package-name]             # List available matching package(s)
$ yum list installed                   # List installed packages
$ yum --showduplicates list [package-name] # Lists all available versions
$ yum install [package-name]-[version] # Install a specific version
```

Exercise 1:

# Exercise 1

Update **all** installed packages

List all installed packages start with **http**

Find all available packages start with **http**

Install **httpd** if available. (Skip confirmations during installation)

List installed **httpd** package

Remove **httpd**

List installed **httpd** package

Exercise 2:

# Exercise 2

Uninstall **git** with all unused dependencies

Check installed **git**

Find previous available **git** version

Install previous available **git** version

Check installed **git** version

Update **git** to the **latest** version

Check installed **git** version