

# GNU/Linux

---

## 2. Ders



# VirtualBox Önemli Klavye Kısayolları

---

- Host Tuşu = klavyenin sağındaki Ctrl
- Sanal Makine çalıştıktan sonra:
  - Host + Home: Seçenekler Menüsünü Açar
  - Host + F : Ekran Modu
  - Host + A : Makine Çözünürlüğüne göre pencereyi ayarlama
  - Host + C : Pencere uzatma ayarı

# VirtualBox Önemli Klavye Kısayolları

---

- Host +Q : Sanal Makineyi Kapat
- Host +S : Ayarlar Penceresi
- Host + T : Makinenin anlık görüntüsünü kaydet (Snapshot)
- Host + N : Oturum Bilgisi İstatistikleri



# Terminal

- Terminal, İşletim sistemimizi CLI Shell'i ile kullanmamızı sağlayan arayüzdür.
- GNU/Linux dağıtımları Shell olarak genelde «Bash» shellini kullanır.
- Windows'ta bunun karşılığına CMD ya da Powershell diyebiliriz.
- Windows'ta şuana kadar hiç CMD'ye girmeden hayatta kalabiliyorsanız, aslında GNU/Linux dağıtımlarının bir çoğunda da terminale hiç uğramadan hayatta kalabilirsiniz.

# O Zaman Neden Terminal Kullanımını Öğreniyoruz?

- Kullanığımız işletim sistemine tamamen hakim olmak, istediklerimizi grafik arayüzünün hantallığına takılmadan yapmak ve işlerimizi çok daha hızlı görmek ve otomatikleştirmek açısından terminal kullanımı önemlidir.
- Ayrıca Bilgisayar Mühendisleri olarak burada çok işimiz oluyor : ))
- Mesela Git komutu gibi komutları kullanmak için terminal kullanmalıyız. Ayrıca herhangi bir programı/yazılımı derlerken terminal kullanmamız gerektiği gibi faktörler terminal kullanımına alışmanız açısından önemlidir.

# O Zaman Neden Terminal Kullanımını Öğreniyoruz? -devam-

- Yani ş uanda öğrencekleriniz size anlamsız ve hantallık gibi gözükecek olsa da ileriki mesleki hayatınızda çokça rahat etmenizi sağlayacaktır.
- Ne kadar erken, o kadar iyi...

# Shell Nedir?

- Geçen hafta anlatmıştık fakat tekrar edelim:
  - Shell (kabuk) işletim sistemlerinin servislerini kullanmak için kullanılan bir kullanıcı arabirimidir[kullanıcı arabirimi(UI)], bir programdır.
  - Çekirdek ile kullanıcı arasındaki iletişimi sağlar.
  - Birçok Shell çeşidi mevcuttur:
    - Bash
    - sh
    - zsh
    - csh
    - ksh
    - tcsh ...

# Temel Komutlar

- Terminal içinde kullanacağınız birçok uygulama ve komut bulunmakta.
- Fakat biz bu eğitim boyunca sadece temel düzeydeki komutları göstereceğiz.



# ls (list)

Herhangi bir dizinin içeriğini görmek için kullandığımız komuttur:

```
metin@DESKTOP-0L11T3Q:~$ ls  
images test
```

Git eğitiminde bahsettiğimiz gibi GNU/Linux'da dosyanın gizli olması için başına «.» konması yeterlidir. Yukarıda yazdığımız gibi ls komutu tüm dosya ve dizinleri döndürmedi.

Gizli dosya ve dizinleri de görebilmemiz için -a ya da --all seçeneğini/parametresini kullanmamız gerekir:

```
metin@DESKTOP-0L11T3Q:~$ ls -a  
.  .aws  .bash_history  .bashrc  images  .lessht  .profile  test  
.. .azure .bash_logout  .docker  .landscape .motd_shown .sudo_as_admin_successful
```

# ls (list)

Örneğin dosyaların oluşturulduğu tarihe göre listelenmiş bir şekilde sıralamak istiyoruz. O zaman bu şekilde seçenekleri kullanabiliriz:

```
metin@DESKTOP-0L11T3Q:~$ ls -alt
total 44
drwxr-x--- 6 metin metin 4096 Dec 10 14:08 .
-rw----- 1 metin metin   20 Dec 10 14:08 .lessht
drwxr-xr-x 2 metin metin 4096 Dec 10 14:02 test
-rw-r--r-- 1 metin metin    0 Dec 10 13:37 .motd_shown
-rw----- 1 metin metin 1263 Oct 21 23:11 .bash_history
drwxr-xr-x 2 metin metin 4096 Jul 25 22:53 images
drwxr-xr-x 5 metin metin 4096 Jul 25 22:12 .docker
lrwxrwxrwx 1 metin metin   25 Jul 25 22:12 .azure -> /mnt/c/Users/Metin/.azure
lrwxrwxrwx 1 metin metin   23 Jul 25 22:12 .aws -> /mnt/c/Users/Metin/.aws
-rw-r--r-- 1 metin metin    0 Jul 25 22:09 .sudo_as_admin_successful
drwxr-xr-x 2 metin metin 4096 Jul 25 22:09 .landscape
drwxr-xr-x 3 root  root  4096 Jul 25 22:07 ..
-rw-r--r-- 1 metin metin  220 Jan  6  2022 .bash_logout
-rw-r--r-- 1 metin metin 3771 Jan  6  2022 .bashrc
-rw-r--r-- 1 metin metin  807 Jan  6  2022 .profile
metin@DESKTOP-0L11T3Q:~$ |
```

# **man** (manual)

- Her komut için bu seçenekleri tek tek öğrenecek miyiz???
- Evet.
- Peki ezberleyecek miyiz? HAYIR!
- Nerdeyse tüm kullandığımız komutların bir dokümantasyon/manual sayfası vardır. Bunları okumamız için «man» komutunu kullanabiliriz.

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 `-cftuvSUX` nor `--sort` is specified.

Mandatory arguments to long options are mandatory for short options too.

`-a`, `--all`

do not ignore entries starting with `.`

`-A`, `--almost-all`

do not list implied `.` and `..`

`--author`

with `-l`, print the author of each file

`-b`, `--escape`

print C-style escapes for nongraphic characters

`--block-size=SIZE`

with `-l`, scale sizes by SIZE when printing them; e.g., `'--block-size=M'`; see SIZE format below

`-B`, `--ignore-backups`

do not list implied entries ending with `~`

Manual page ls(1) line 1 (press h for help or q to quit)

- Q tuşuna basarak bu manual sayfasından çıkabilirsiniz.

# man (manual)

## - devam

- Bu manual oldukça detaylıdır.
- Öneri olarak; eğer man sayfaları kadar detaylı açıklamaya ihtiyacınız olmadığını düşünüyorsanız man komutu yerine «tldr» uygulumasını kullanabilirsiniz.
  - [github.com/tldr-pages/tldr](https://github.com/tldr-pages/tldr)
    - tl;dr : «too long; didn't read»

```
metin@DESKTOP-0L11T3Q:~$ tldr ls
```

```
ls
```

```
List directory contents.
```

```
More information: https://www.gnu.org/software/coreutils/ls.
```

```
- List files one per line:
```

```
ls -1
```

```
- List all files, including hidden files:
```

```
ls -a
```

```
- List all files, with trailing '/' added to directory names:
```

```
ls -F
```

```
- Long format list (permissions, ownership, size, and modification date) of all files:
```

```
ls -la
```

```
- Long format list with size displayed using human-readable units (KiB, MiB, GiB):
```

```
ls -lh
```

```
- Long format list sorted by size (descending):
```

```
ls -lS
```

```
- Long format list of all files, sorted by modification date (oldest first):
```

```
ls -ltr
```

```
- Only list directories:
```

```
ls -d */
```

```
metin@DESKTOP-0L11T3Q:~$ |
```

# cd (change directory)

- Bulunulan dizini değiştirmeye yarar

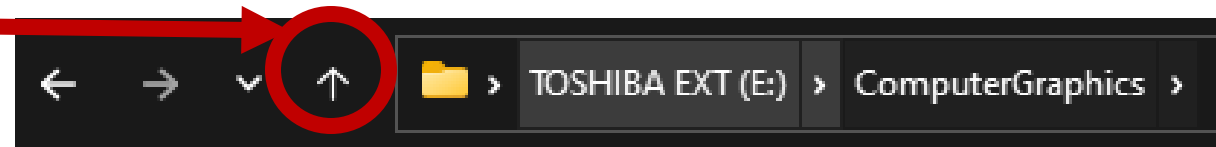
```
metin@DESKTOP-0L11T3Q:~$ ls  
images test  
metin@DESKTOP-0L11T3Q:~$ cd images  
metin@DESKTOP-0L11T3Q:~/images$ |
```

# cd (change directory)

- Bir üst dizine çıkmak istiyorsanız `cd ..` Komutunu kullanmalısınız.

```
metin@DESKTOP-0L11T3Q:~/images$ cd ..  
metin@DESKTOP-0L11T3Q:~$ |
```

- «.» → şuanda bulunduğumuz dizini belirtir.
- «..» → bir üst dizini (parent) belirtir.



# **pwd** (print working directory)

- Hangi dizinin içinde olduğumuzu döndürür. (path)

```
metin@DESKTOP-0L11T3Q:~$ pwd  
/home/metin  
metin@DESKTOP-0L11T3Q:~$ |
```



# mkdir (make directory)

- Yeni bir dizin oluşturmamızı sağlar:

```
metin@DESKTOP-0L11T3Q:~$ mkdir nutella
metin@DESKTOP-0L11T3Q:~$ ls
images  nutella  test
metin@DESKTOP-0L11T3Q:~$ |
```

# touch

- Boş dosya oluşturmamıza ya da var olan dosyaların değiştirilme zamanını düzenlememize yarar.

```
metin@DESKTOP-0L11T3Q:~/test$ touch yazget
metin@DESKTOP-0L11T3Q:~/test$ ls -al
total 8
drwxr-xr-x 2 metin metin 4096 Dec 10 14:58 .
drwxr-x--- 9 metin metin 4096 Dec 10 14:55 ..
-rw-r--r-- 1 metin metin    0 Dec 10 14:58 yazget
metin@DESKTOP-0L11T3Q:~/test$ touch -d "-4 hour" yazget
metin@DESKTOP-0L11T3Q:~/test$ ls -al
total 8
drwxr-xr-x 2 metin metin 4096 Dec 10 14:58 .
drwxr-x--- 9 metin metin 4096 Dec 10 14:55 ..
-rw-r--r-- 1 metin metin    0 Dec 10 10:59 yazget
```

# cat (concatenate)

- Dosyanın içeriğini terminale yazdırır:

```
metin@DESKTOP-0L11T3Q:~/test$ ls  
yazget  
metin@DESKTOP-0L11T3Q:~/test$ cat yazget  
Yazılım Geliştirme Topluluğu
```

# tac

- Terminale son satırı başta olacak şekilde TERSten yazdırır.

```
metin@DESKTOP-0L11T3Q:~/test$ cat alfabe
a
b
c
d
e
f
g
1
2
3
metin@DESKTOP-0L11T3Q:~/test$ tac alfabe
3
2
1
g
f
e
d
c
b
a
```

# echo

- Terminale string olarak herhangi bir ifadeyi yazdırmak için kullanılır:

```
metin@DESKTOP-0L11T3Q:~/test$ echo Haydaaaa  
Haydaaaa
```

- Ancak daha çok dosya içine veri yazmak için kullanılır:

```
metin@DESKTOP-0L11T3Q:~/test$ echo "yeni girdi" >> yazget  
metin@DESKTOP-0L11T3Q:~/test$ cat yazget  
Yazılım Geliştirme Topluluğu  
yeni girdi
```

Append: sona veri ekle

```
metin@DESKTOP-0L11T3Q:~/test$ echo "Tüm içeriği sil ve bunu yaz." > yazget  
metin@DESKTOP-0L11T3Q:~/test$ cat yazget  
Tüm içeriği sil ve bunu yaz.
```

(over)write:  
üstüne yaz

# less

- more komutuna ve vim editörüne benzer, dosya görüntüleme komutudur. Tüm dosyayı, dosyayı açtığı gibi hepsini okumak zorunda olmadığından büyük dosyaları dahi açarken hızlıdır.
- `less [filename]`

# mv (move)

- Dosya ya da dizinlerin taşınmasını sağlar:
- Aynı zamanda dizin ya da dosyaları yeniden adlandırma için de kullanılabilir:

```
metin@DESKTOP-0L11T3Q:~/test$ mv yazget ..  
metin@DESKTOP-0L11T3Q:~/test$ ls  
alfabe  
metin@DESKTOP-0L11T3Q:~/test$ ls ..  
images nutella ödev_yarım_kalan test yazget
```

```
metin@DESKTOP-0L11T3Q:~$ ls  
Homework images  
metin@DESKTOP-0L11T3Q:~$ mv Homework/ Ödev  
metin@DESKTOP-0L11T3Q:~$ ls  
images Ödev  
metin@DESKTOP-0L11T3Q:~$ |
```

# cp (copy)

- Dosya ya da dizinlerin kopyalanmasını sağlar:

```
metin@DESKTOP-0L11T3Q:~/Ödev$ ls
dosya.txt
metin@DESKTOP-0L11T3Q:~/Ödev$ cp dosya.txt dosya_copy.txt
metin@DESKTOP-0L11T3Q:~/Ödev$ ls
dosya_copy.txt  dosya.txt
metin@DESKTOP-0L11T3Q:~/Ödev$ |
```

- Dizinlerin kopyalanması için -r seçeneği kullanılır:

```
metin@DESKTOP-0L11T3Q:~$ ls
Dizin  images  Ödev
metin@DESKTOP-0L11T3Q:~$ cp -r Ödev/ Dizin/
metin@DESKTOP-0L11T3Q:~$ ls Dizin
Ödev
```



# rm (remove)

- Dosya ya da dizinlerin silinmesini sağlar:

```
metin@DESKTOP-0L11T3Q:~$ ls
Dizin  images  Ödev  silinecek_dosya
metin@DESKTOP-0L11T3Q:~$ rm silinecek_dosya
metin@DESKTOP-0L11T3Q:~$ ls
Dizin  images  Ödev
```

```
metin@DESKTOP-0L11T3Q:~$ rm -r Dizin
metin@DESKTOP-0L11T3Q:~$ ls
images  Ödev
```

# wget

- İnternet protokolleri üzerinden dosya indirme aracıdır.

```
metin@DESKTOP-0L11T3Q:~$ wget https://raw.githubusercontent.com/ymozer/GMYI/main/README.md
--2022-12-10 16:22:39-- https://raw.githubusercontent.com/ymozer/GMYI/main/README.md
Resolving raw.githubusercontent.com (raw.githubusercontent.com)... 185.199.108.133, 185.199.111.133, 185.199.109.133, ...
Connecting to raw.githubusercontent.com (raw.githubusercontent.com)|185.199.108.133|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 1216 (1.2K) [text/plain]
Saving to: 'README.md'

README.md                                100%[=====>] 1.19K --.-KB/s   in 0s

2022-12-10 16:22:40 (24.6 MB/s) - 'README.md' saved [1216/1216]

metin@DESKTOP-0L11T3Q:~$ cat README.md
**!! This repo is under development !!**
# GMYI
GMYI can collect infected user's system data and analyze it for displaying pop-up ads on the desktop.
## TO-DO's
- [ ] Convert every data collection function to Pandas DataFrame
- [ ] Command line flags for running certain functions.
- [ ] Get browser data (Edge, Chrome, Chromium, Firefox, Opera)
  - [ ] Cookies
  - [ ] Passwords ??
  - [ ] History
- [ ] Test GMYI on other systems.
- [ ] USB devices information function
- [ ] Screenshot/Recording ability as well as access to microphone and camera.
- [ ] Keylogger ability
- [x] Make requirement.txt.
- [ ] Make an executable.
- [ ] Make a database that will store infected system's information.
- [ ] Make a webserver for receiving information from infected systems.
- [ ] Analyze "user types" according to collected data.
- [ ] Implement the pop-up window on desktop for displaying personalized advertisement.
- [ ] Reverse shell to infected Windows computer.
- [ ] Obfuscation
- [ ] ...

## How to Spread/Run GMYI ??
GMYI can sit on some website that can downloaded and executed by rubber ducky script.
* Rubber Ducky
* Social Engineering (e-mail)
* ??

## Flowchart
![Flowchart](/Media/GMYI_flowchart.png)
metin@DESKTOP-0L11T3Q:~$ |
```

!!

- Son çalıştırılan komutu tekrar çalıştırır.

```
metin@DESKTOP-0L11T3Q:~$ echo "Tekrar"  
Tekrar  
metin@DESKTOP-0L11T3Q:~$ !!  
echo "Tekrar"  
Tekrar
```

# history

- Önceden yürütülen komutları liste olarak döndürür.
- `history` | `less` komutuyla daha düzgün şekilde görüntüleyebiliriz:

```
metin@DESKTOP-0L1T3Q: ~  
4 sudo apt update  
5 sudo apt upgrade  
6 shutdown now  
7 sudo shutdown now  
8 ls  
9 docker  
10 docker run -ti --rm -v c:/Users/youruser/datasets:/datasets opendronemap/odm --project-path /dataset  
s project  
11 docker run -ti --rm -v C:\Users\Metin\Desktop\drone_dataset opendronemap/odm --project-path /dataset  
s project  
12 docker run -ti --rm -v C:\Users\Metin\Desktop\drone_dataset opendronemap/odm --project-path C:\Users  
\Metin\Desktop\drone_dataset project  
13 docker run -ti --rm -v C:\Users\Metin\Desktop\drone_dataset o--project-path C:\Users\Metin\Desktop\dr  
one_dataset project  
14 docker run -ti --rm -v C:\Users\Metin\Desktop\drone_dataset o --project-path C:\Users\Metin\Desktop\  
drone_dataset project  
15 docker run -ti --rm -v C:\Users\Metin\Desktop\drone_dataset --project-path C:\Users\Metin\Desktop\d  
rone_dataset project  
16 docker run -ti --rm -v C:\Users\Metin\Desktop\drone_dataset images --project-path C:\Users\Metin\Des  
ktop\drone_dataset project  
17 ls  
18 cd ..  
19 ls  
20 cd ..  
:|
```

# history - devam

- history listesindeki herhangi bir komutu tekrar çalıştırmak için komutların solundaki gözüken numaraların başına ! koyarak terminalde çalıştırabiliriz:
- Örneğin

\$ !18 → history listesindeki 18. komutu çalıştırır.

# Terminal Kısa Yolları

CTRL + A	Terminalde yazılı olan ifadenin en başına gider
CTRL + E	Terminalde yazılı olan ifadenin en sonuna gider
CTRL + L ya da «clear» komutu	Terminal temizlenir
CTRL + D	Terminalden çıkılır
CTRL + Shift + C	Kopyala
CTRL + Shift + V	Yapıştır
CTRL + C	Çalışan komutu ya da uygulamayı durdurur
CTRL + U	Satıra yazılan her şeyi siler

# Terminal Üstünde Çalışan Metin Editörleri

## vim

```
metin@DESKTOP-0L1T13Q: ~  
root:x:0:0:root:/root:/bin/bash  
daemon:x:1:1:daemon:/usr/sbin:/usr/sbin/nologin  
bin:x:2:2:bin:/bin:/usr/sbin/nologin  
sys:x:3:3:sys:/dev:/usr/sbin/nologin  
sync:x:4:65534:sync:/bin:/bin/sync  
games:x:5:60:games:/usr/games:/usr/sbin/nologin  
man:x:6:12:man:/var/cache/man:/usr/sbin/nologin  
lp:x:7:7:lp:/var/spool/lpd:/usr/sbin/nologin  
mail:x:8:8:mail:/var/mail:/usr/sbin/nologin  
news:x:9:9:news:/var/spool/news:/usr/sbin/nologin  
uucp:x:10:10:uucp:/var/spool/uucp:/usr/sbin/nologin  
proxy:x:13:13:proxy:/bin:/usr/sbin/nologin  
www-data:x:33:33:www-data:/var/www:/usr/sbin/nologin  
backup:x:34:34:backup:/var/backups:/usr/sbin/nologin  
list:x:38:38:Mailing List Manager:/var/list:/usr/sbin/nologin  
irc:x:39:39:ircd:/run/ircd:/usr/sbin/nologin  
gnats:x:41:41:Gnats Bug-Reporting System (admin):/var/lib/gnats:/usr/sbin/nologin  
nobody:x:65534:65534:nobody:/nonexistent:/usr/sbin/nologin  
systemd-network:x:100:102:systemd Network Management,,,:/run/systemd:/usr/sbin/nologin  
systemd-resolve:x:101:103:systemd Resolver,,,:/run/systemd:/usr/sbin/nologin  
messagebus:x:102:105:nonexistent:/usr/sbin/nologin  
systemd-timesync:x:103:106:systemd Time Synchronization,,,:/run/systemd:/usr/sbin/nologin  
syslog:x:104:111:/home/syslog:/usr/sbin/nologin  
_apt:x:105:65534:nonexistent:/usr/sbin/nologin  
tss:x:106:112:TPM software stack,,,:/var/lib/tpm:/bin/false  
uuidd:x:107:113:/run/uuidd:/usr/sbin/nologin  
tcpdump:x:108:114:nonexistent:/usr/sbin/nologin  
sshd:x:109:65534:/run/sshd:/usr/sbin/nologin  
pollinate:x:110:111:/var/cache/pollinate:/bin/false  
landscape:x:111:116:/var/lib/landscape:/usr/sbin/nologin  
metin:x:1000:1000:Metin:/home/metin:/bin/bash  
fnupd-refresh:x:112:120:fnupd-refresh user,,,:/run/systemd:/usr/sbin/nologin  
"/etc/passwd" [readonly] 32L, 1712B  
1,1 All
```

## nano

```
metin@DESKTOP-0L1T13Q: ~  
GNU nano 6.2 /etc/passwd  
root:x:0:0:root:/root:/bin/bash  
daemon:x:1:1:daemon:/usr/sbin:/usr/sbin/nologin  
bin:x:2:2:bin:/bin:/usr/sbin/nologin  
sys:x:3:3:sys:/dev:/usr/sbin/nologin  
sync:x:4:65534:sync:/bin:/bin/sync  
games:x:5:60:games:/usr/games:/usr/sbin/nologin  
man:x:6:12:man:/var/cache/man:/usr/sbin/nologin  
lp:x:7:7:lp:/var/spool/lpd:/usr/sbin/nologin  
mail:x:8:8:mail:/var/mail:/usr/sbin/nologin  
news:x:9:9:news:/var/spool/news:/usr/sbin/nologin  
uucp:x:10:10:uucp:/var/spool/uucp:/usr/sbin/nologin  
proxy:x:13:13:proxy:/bin:/usr/sbin/nologin  
www-data:x:33:33:www-data:/var/www:/usr/sbin/nologin  
backup:x:34:34:backup:/var/backups:/usr/sbin/nologin  
list:x:38:38:Mailing List Manager:/var/list:/usr/sbin/nologin  
irc:x:39:39:ircd:/run/ircd:/usr/sbin/nologin  
gnats:x:41:41:Gnats Bug-Reporting System (admin):/var/lib/gnats:/usr/sbin/nologin  
nobody:x:65534:65534:nobody:/nonexistent:/usr/sbin/nologin  
systemd-network:x:100:102:systemd Network Management,,,:/run/systemd:/usr/sbin/nologin  
systemd-resolve:x:101:103:systemd Resolver,,,:/run/systemd:/usr/sbin/nologin  
messagebus:x:102:105:nonexistent:/usr/sbin/nologin  
systemd-timesync:x:103:106:systemd Time Synchronization,,,:/run/systemd:/usr/sbin/nologin  
syslog:x:104:111:/home/syslog:/usr/sbin/nologin  
_apt:x:105:65534:nonexistent:/usr/sbin/nologin  
[ File '/etc/passwd' is unwritable ]  
^G Help ^O Write Out ^W Where Is ^M Cut ^T Execute  
^X Exit ^R Read File ^_ Replace ^U Paste ^J Justify ^_ Location M-U Undo  
^C Go To Line ^M-E Redo
```

# Dosya Sistem Hiyerarşisi

- GNU/Linux sistemlerdeki dosya sistem hiyerarşisi Windows işletim sisteminden oldukça farklıdır.
- GNU/Linux'ün hiyerarşisinin en üstünde root(kök) «/» bulunur. Tüm sistemin referans noktası burasıdır.
- Alt dizinler ya da dosyalar bu root'tan ayrılan dallar ya da yapraklar olarak düşünebilirsiniz.
- GNU/Linux sistemlerde her zaman bir dizin içinde olursunuz. Yani ağaç yapısı dışına çıkamazsınız.
- Şimdi dosya sistemimiz içerisindeki önemli dizinlerimize bakalım



# Ubuntu Dosya Sistemi

kullanır

Filesystem Hierarchy Standard(FHS)

- **/bin** Sistem komutlarını içerir. Kullandığımız çoğu komut bu dizin içindedir. Örneğin: ls, mount, rm vb.
- **/boot** Sistemi başlatmak için gereken dosyaları içerir, bunların içinde: Linux kerneli, RAM disk görüntüsü ve bootloader ayar dosyaları.
- **/dev** Tüm aygıtların dosyalarını içerir. Bu dosyalar normal dosyalar değil de harddisk gibi sistem üstündeki çeşitli donanım aygıtlarını içerir.
- **/etc** Sistem geneli konfigürasyon/ayar dosyalarını içerir. Bunlar tüm kullanıcılar için sistemin davranışını etkiler.
- **/home YA DA ~** Kullanıcıların Home dizinlerini içerir.
- **/lib** Çok önemli DLL(dynamic libraries ) ve Kernel modüllerini içerir.
- **/media** harici cihazların «mount» (takılma-bağlanma) noktasıdır. Örneğin harddiskler ya da çıkarılabilir medyalar(flash USB, CD, DVD...)
- **/mnt** burası da mount (takılma) noktasıdır, fakat özellikle «geçici takılmış» cihazlar içindir. Örneğin Ağ dosya sistemleri.
- **/opt** sisteme sonradan eklenmiş, GNU/Linux dosya hiyerarşisine uymayan yazılımları depolamak için kullanılır. Bu yazılımlar package manager tarafından yönetilmez (indirme/kaldırma/düzenleme) ve sistemden yalıtılmış halde bu dizin altında tutulur.
- **/proc** Kernelin proseslere bilgi göndermesini sağlayan sanal dosya sistemi mekanizmasıdır.
- **/root** superuser'ın home dizinidir. Sistemin /home/ dizininin ulaşılabilir olmadığı zamanlarda da sistemin açılmasına olanak sağlamak için /home/ içinde değildir.
- **/var** is dedicated to variable data, such as logs, databases, websites, and temporary spool (e-mail etc.) files that persist from one boot to the next. A notable directory it contains is /var/log where system log files are kept.

# Ubuntu Dosya Sistemi - devam

- Root altında yer alan diğer dizinler aşağıdadır:
  - /run
  - /sbin
  - /srv
  - /sys
  - /tmp
  - /usr
- <https://help.ubuntu.com/community/LinuxFilesystemTreeOverview>

# Tam Yol (Absolute Path)-Göreceli Yol (Relative Path)

Bir dosyanın ya da dizinin yerini tarif ederken iki çeşit yolu kullanırız. Bunlar tam yol ve göreceli yoldur.

1. Tam Yol: root dizininden başlar ve hedef gösterdiğimiz dizine kadar devam eder. Eğer tam yol içeren bir komut yazarsak hangi dizinde olursak olalım çalıştırıldığında aynı sonucu verecektir. Çünkü referans aldığı ilk yol root'tur
2. Göreceli Yol : Bu yolun her zaman geçerliliği yoktur. İçerisinde bulunan dizine göre değişiklik gösterir.

# Paket Yönetici (Package Manager) Nedir?

- Bilgisayar programlarını kurma, yükseltme, ayarlama ve kaldırma işlemlerini otomatikleştiren yazılım araçlarıdır.
- Tek bir komutla istediğiniz uygulamayı, kullandığınız dağıtımını repo(-sitory)larından direkt indirip kurabilirsiniz.
- Her paket yöneticisini her dağıtımda kullanmayız. Mesela Arch Linux pacman kullanırken Debian APT kullanır. Diğer Örnekleri:
  - portage → Fedora
  - dnf(yum) → Gentoo
  - Zypper → OpenSUSE
  - Homebrew → macOS
- Kullanımları farklılık gösterse de işleveleri aynıdır.

# Installer – Package Manager Farkı

- Windows kullanırken bir yazılım kuracağımız zaman şu aşamaları izleriz:
  - Tarayıcı açarız
  - İstedğimiz programı arama motorundan ararız
  - Sitelerden uygun olanı buluruz
  - Program kurucu yazılımını indiririz
  - Çalıştırırız
  - Kurma konumunu belirtiriz ...
- GNU/Linux sistemlerinin kullandığı paket yöneticisi sitemleriyle bu sistemi karşılaştırdınca ne kadar ilkel olduğunu görebiliyoruzdur sanırım.

# Installer – Package Manager Farkı

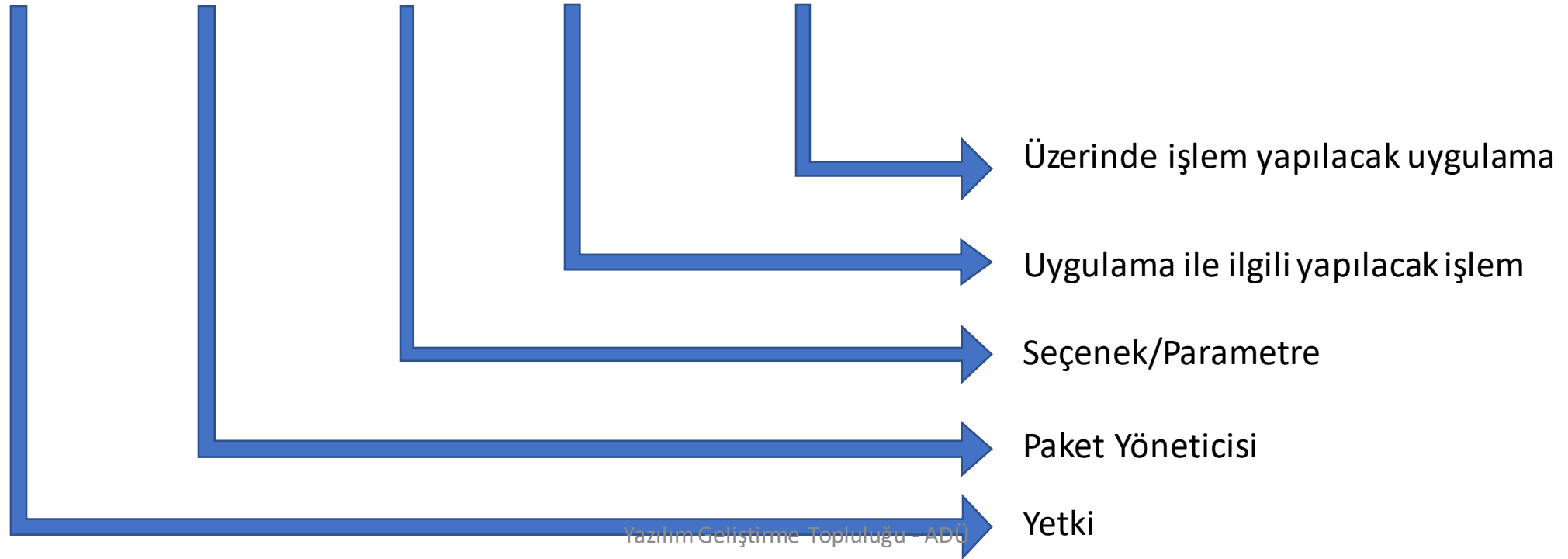
	Package Manager	Installer
Birlikte gelir	İşletim sistemiyle birlikte gelir	Her programla ayrı ayrı gelir
Kurulum bilgisinin konumu	Bir tane merkezi kurulum veri tabanı	Kurucu programın kendisine bağlıdır. İşletim sistemi dosyalarıyla birlikte olabileceği gibi utgulmalar klasörü içerisinde de olabilir.
Bakım Detayları	Tüm paketlere hep birlikte bakım olasıdır.	Sadece birlikte verildiği ürüne bakım mümkündür.
Geliştiricisi	Bir paket yöneticisi tedarikçisi	Birden çok tedarikçi
Paket Formatı	Çok bilindik birkaç paket formatı	Uygulama sayısı kadar format olabilir.
Paket Formatının Uyumluluğu	Paket yöneticisi desteklediği sürece uyumluluğu vardır.	Installer her zaman arşivlenmiş (sıkıştırılmış) formatı ile uyumludur.

# APT ve APT-GET Kullanımı

Paket kurulum için:

```
sudo apt install [PACKAGE NAME]
```

```
sudo apt-get -y install firefox
```



# Update ve Upgrade

- `sudo apt-get update`
  - Updates the package database
- `sudo apt-get upgrade`
  - Upgrades an already installed package
- `sudo apt-get update && sudo apt-get upgrade -y`
  - Update your entire system packages



# Paketleri Arama

- apt-cache search [ARAMA TERİMİ]
  - Searches for packages similar to the search term
- apt-cache pkgnames [ARAMA TERİMİ]
  - Searches for a specific package
- apt-cache showpkg [PAKET]
  - Shows version, dependencies etc for the package

# Paket Kurma

- `sudo apt-get install [paket-1] [paket-2] ...`
  - Install one or several programs simultaneously
- `sudo apt-get install [paket] --no-upgrade`
  - Only install an not already program
- `sudo apt-get install [paket] -- only-upgrade`
  - Only upgrade an already installed program
- `sudo apt-get install [paket]=[versiyon numarası]`
  - Install a specific version of a program

# Kurulmuş Paketleri Kaldırma

- `sudo apt-get remove`
  - Remove a program
- `sudo apt-get purge`
  - Remove a program and its configuration files

# REPO

- Paket yöneticisi ile birlikte uygulamaların nasıl kurulduğunu öğrendik fakat bu paketler nereden geliyor??
- Git'te olduğu olduğu gibi paket yöneticilerinde de REPO diye bir kavram bulunmaktadır. Her dağıtımın kendine ait repo'su vardır. Paket yöneticisi ile indirilen ve kurulan uygulamalar bu repo'lardan gelir.
- Dağıtımımızdaki repo'ları görüntüleyebilmek ve düzenleyebilmek için:
  - nano /etc/apt/sources.list
- Ubuntu Repolarındaki Paketleri aramak istiyorsanız:
  - <https://packages.ubuntu.com/jammy/>

# Repolarda olmayan uygulamalar

- ...