
Table of Contents

| | |
|--|------|
| Giriş | 1.1 |
| 01-Özgür Yazılım Felsefesi ve Temel Kavramlar | 1.2 |
| 02-GNU/Linux İşletim Sisteminin Yapısı | 1.3 |
| 03-Komut Satırı (Kabuk) ve Temel Komutlar | 1.4 |
| 04-Kabuk ve Özellikleri | 1.5 |
| 05-Kullanıcı ve Grup Yönetimi | 1.6 |
| 06-Paket Yönetim Sistemi | 1.7 |
| 07-Süreçler ve Servisler | 1.8 |
| 08-Sistem Kayıtları | 1.9 |
| 09-Depolama Aygıtlarının Yönetimi | 1.10 |
| 10-Zamanlanmış Görevler | 1.11 |
| 11-Temel TCP/IP Bilgisi ve Ağ Yönetimi | 1.12 |
| 12-DNS Teknolojisine Giriş | 1.13 |
| 13-Güvenli Uzaktan Erişim | 1.14 |
| 14-Yedekleme, Arşivleme ve Sıkıştırma | 1.15 |
| 15-Web Teknolojisine Giriş | 1.16 |
| 16-Veritabanı Servislerine Giriş | 1.17 |
| 17-Apache/PHP/MySQL Kurulumu ve Örnek Uygulama | 1.18 |
| 18-Bash Betik (Script) Yazımına Giriş | 1.19 |

Mustafa Akgül 9. Özgür Yazılım Yaz Kampı (20 Temmuz - 4 Ağustos 2018) "**GNU/Linux Sistem Yönetimi 1. Düzey**" ders notlarım.

Fatih Kaan Salgır

Özgür Yazılım

http://www.cagataycebi.com/free_articles/freedom/freedom_brief.html

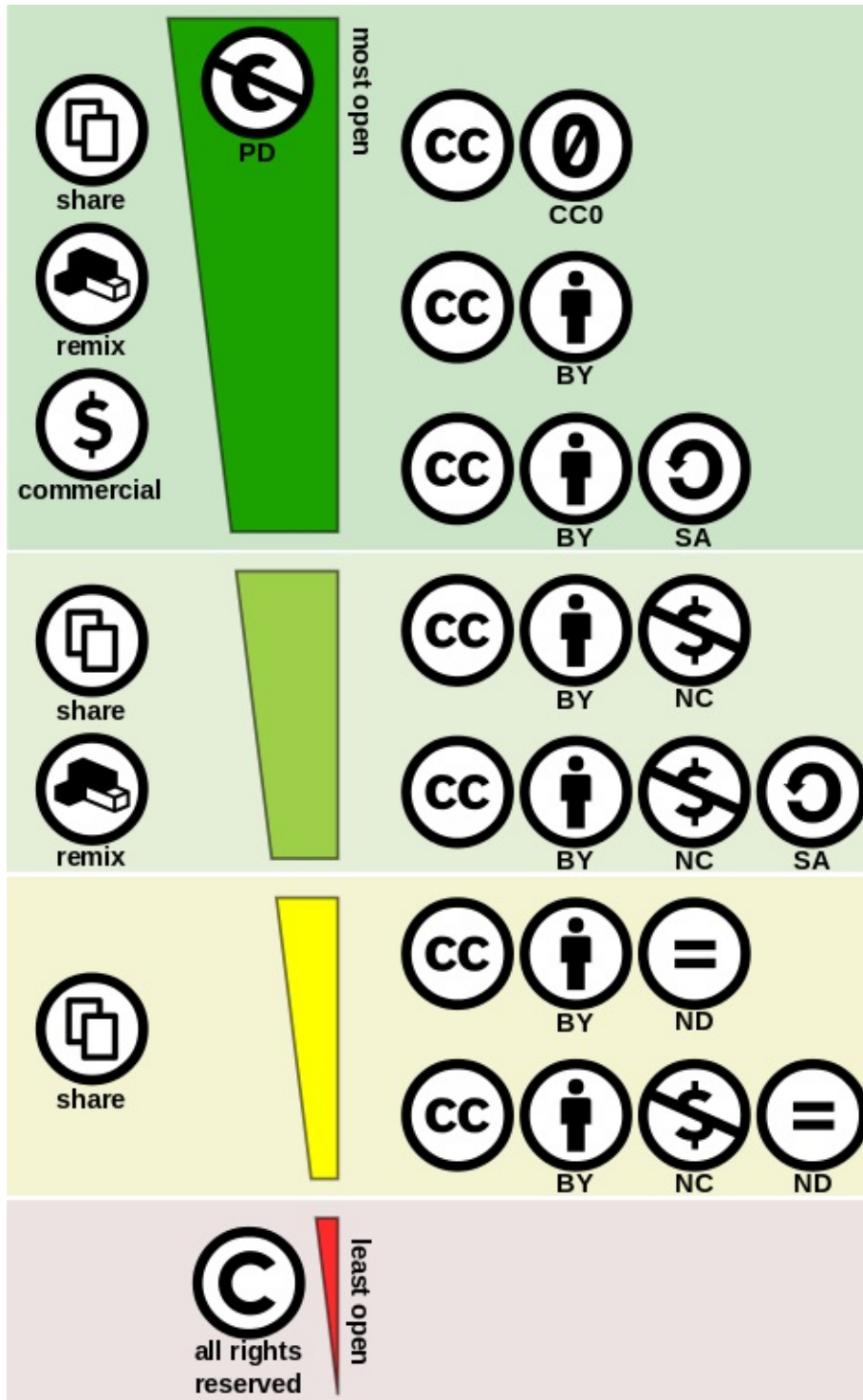
<http://www.gnu.org/philosophy/free-sw.tr.html>

4 temel hak;

1. Her türlü amaç için programı çalıştırma özgürlüğü.
2. Programın nasıl çalıştığını inceleme ve değiştirme (kaynak koduna erişim)
3. Yeniden dağıtma ve paylaşma
4. Geliştirme ve gelişmiş haliyle paylaşma

Copyleft

Creative Commons

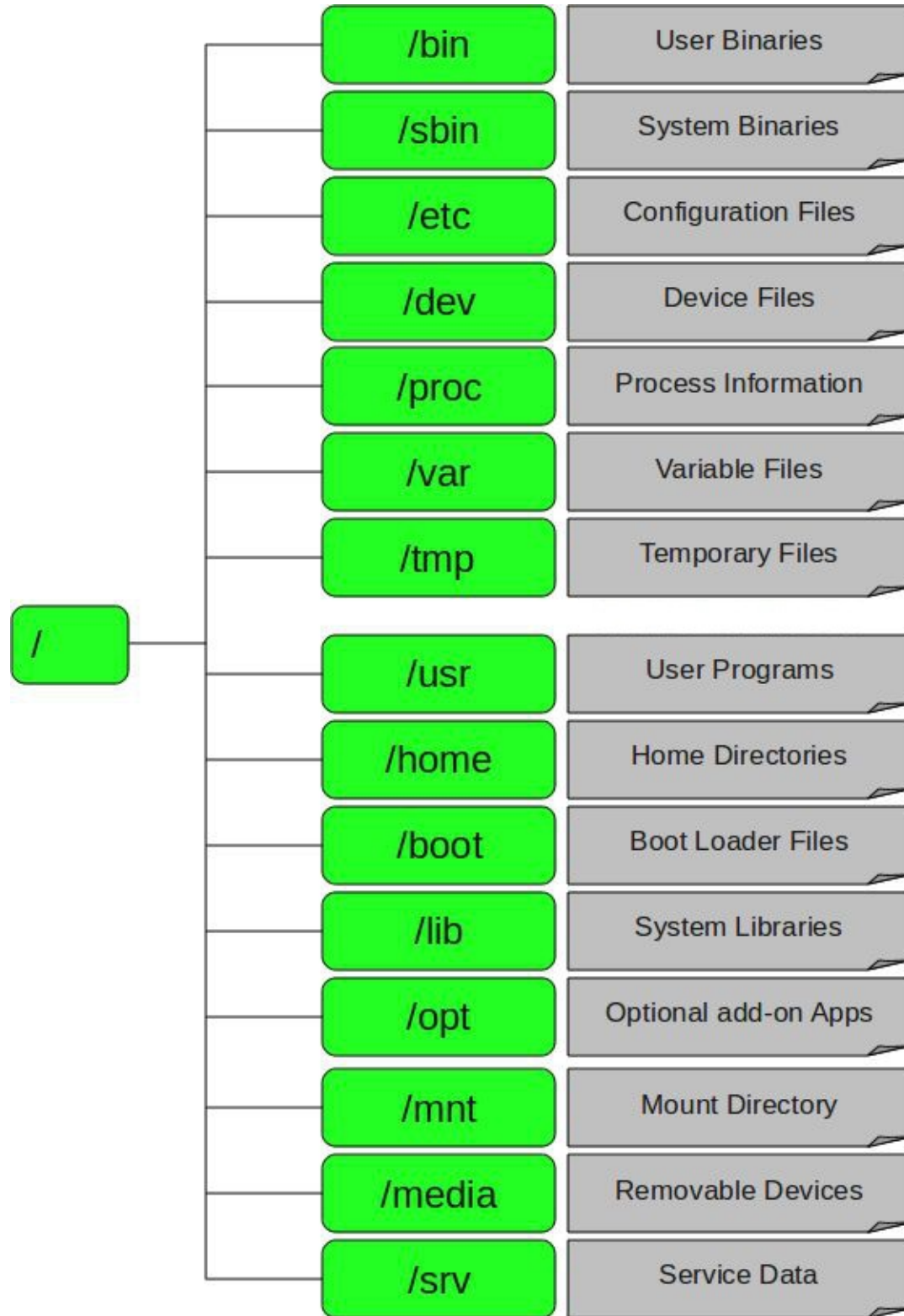


GNU/Linux İşletim Sistemi Yapısı

Açılış sistemi;

1. BIOS
2. MBR
3. GRUB
4. Kernel
5. Init
6. Runlevel programs

-
- -: file
 - l: link
 - d: directory
 - s: socket
 - c: character
 - b: block

Dosya Sistemi

[Daha Detaylı Tablo](#)

- İşletim sistemi kurulumu yaparken /boot klasörü için ayrı kurulum yapılması daha güvenlidir.

Komut Satırı (Kabuk) ve Temel Komutlar

- **komut 1>log.txt 2>hatalog.txt:** komutun çıktı logları log.txt, hatalı çıktıları hatalog.txt ye yazdırır.
- **sudo su:** Root kullanımına geçme. **exit** komutu ile root kullanıcısından çıkılır.
- **curl:**bağlantıyı terminalde gösterir.
- **wget:**bağlantıyı indirir.
- **zip:** zipleme komutu **unzip** dosyayı çıkarır.
- **du:** (disk usage)
- **sort**
- **tail**
- **head**
- **more**
- **less**
- **script:** Terminalde yazılanları dosyaya kaydeder.
- **wc:** (word count)
- **which:** Programın konumunu gösterir.
- **whereis:** Programın binary, source ve manual dosyalarının konumlarını gösterir.
- **&&:** AND [-a]
- **||:** OR [-o]
- **df:** (disk free) Bağlı diskleri gösterir.
- **w:** Kullanıcı hakkında ayrıntılı bilgi verir.
- **who:** Kullanıcı hakkında bilgi verir.
- **last:** Son sisteme giriş kayıtlarını görüntüler.
- **/dev/null:** Buraya gönderilen dosya ya da çıktılar silinir.
- **/dev/zero:** Sonsuz sayıda 0'lar oluşturabilen dizin.

Çıktıdan Sütun-Satır Seçme

Ayrııcı varsa **cut**, boşluklar varsa **awk** komutu kullanılır.

```
awk '{print $[sütun sayısı]}'
```

```
ls -l /home/near/ | awk '{print $1}'
```

```
cut [dosya] | -d"[ayırıcı]" -f[sütun]
```

```
cut etc/passwd | cut -d":" -f1-3,6
```

- **-f:** field
- **uniq:** Çıktıda aynı satırlardan sadece birer tanesini gösterir.

sed: Çıktıdan satır seçmek için kullanılır.

Dosyadan 3. satırı seçmek için;

```
sed -n 3p dosya.txt
```

Dosyadan 'abc' geçen bütün satırları seçmek için;

```
sed -n '/abc/p' dosya.txt
```

Yardımcı Komutlar

- **nl:** Çıktıda her satırı numaralandırır.
-

- **grep:** Dosyadan aranan karakterleri getirir.

```
grep -Rin "[aranacak karakterler]" [dizin-dosya 1] [dizin-dosya 2]
```

- -v: invert matches (çıktıdan belli satırı çıkarmak için kullanılabilir.)
 - R (recursive): Dizinleri de aramaya katar.
 - i (ignore case) : Küçük büyük harfe dikkat etmez.
 - n (line number)
-

- **find:** Dosya özelliklerine göre arama yapar. Çıktı olarak sadece dosya isimlerini getirir.

```
$ find [aranacak_dizin] -name [] -type []
```

- iname: ignore case

access modify change değerlerini görüntüleme ve bu değerle göre arama yapma;

```
stat [dosya]
```

```
$ find -(amc)min -[sayi] [dizin]
```

```
$ find -(amc)time +[sayi] -(amc)time -[sayi] [dizin]
```

find ile istenilen dosyalar bulunduktan sonra -exec parametresi girilir, find ile bulunan dosyalar yerine '{}' yazılır ve \; ile komut kapanır.

```
$ find ... -exec ... [komut] '{}' \;
```

home daki txt uzantılı dosyaları /tmp/yedek/ dizinine kopyalama;

```
$ find /home/near/ -iname "*.txt" 2>/dev/null -exec cp -v '{}' /tmp/yedek/ \;
```

```
$ locate a.txt
>...a.txt
>...sirala.txt
>...yaka.txt
```

Dosya ararken ya da listelerken kullanılabilir. (Wildcards)

- ?: 1 karakterlik
- *: karakter sınırı yok

```
rm -rf /home/deneme
```

```
rm /home/deneme -rf
```

(alttaki daha güvenli, dizin yazarken yanlışlıkla boşluk bırakılması durumunda bir güvenlik önlemi daha ekler.)

Kullanıcı Yönetimi

Kullanıcı değiştirme;

```
su [kullanici]
```

Kullanıcıyı çevresel değişikliklerle birlikte değiştirme;

```
su - [kullanici]
```

Komutlar

- id [user] [group]: Kullanıcı ve group hakkında bilgi verir.
- sudo: Komutu farklı bir kullanıcı üzerinden çalıştırır.
- usermod: diğer kullanıcıların hesaplarında değişiklik yapmayı sağlar. (parola, home dizini gibi)
- groups: Kullanıcının gruplarını gösterir.
- gpasswd: Kullanıcıyı gruba ekleme ve çıkarma.
- addgroup/groupadd
- groupdel/delgroup
- useradd
- userdel

etc/sudoers: sudo kullanıcılarının yetkilerini belirler.

Kullanıcı terminal ayarları (renk,bilgisayar adı gibi) `.bashrc` dosyasında tutulur. Dosya düzenlenerek kullanıcıya özel komutlar oluşturulabilir.

`alias isim='echo "Fatih Kaan Salgır"'`

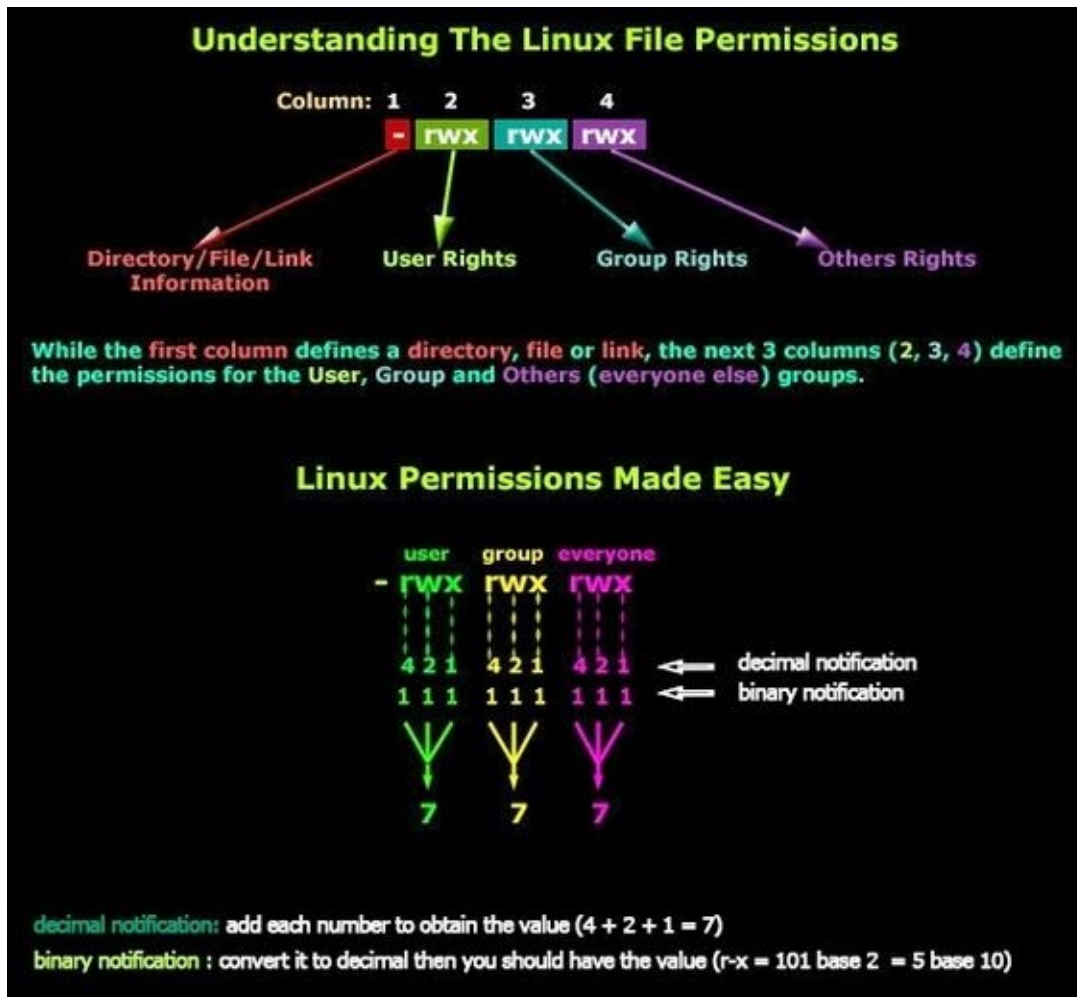
olarak düzenlendiğinde;

```
$ isim
>>>Fatih Kaan Salgır
```

-
- Kullanıcılara özel uygulama kofigürasyon dosyaları kendi `/home/.config` dizinlerinde saklanır.
-

- `etc/passwd` dosyasında tüm kullanıcıların kullanıcı adı, parola olup olmadığı, kullanıcı-grup id'leri, home dizinleri ve kullandıkları shell dizini bulunur.

İzinler



yetki vermek ya da almak için;

```
$ chmod [(ugo)(+)(rwx)] [dizin]
```

user, group, others // read, write, execute

- /bin/ls altında çalıştırılabilir dosyalar bulunur (komutlar), bu komutlardan diğer kullanıcıların çalıştırma yetkisi alınabilir.

r=4

w=2

x=1

```
$ chmod [xxx] [dosya_adi]
```

-R: recursive (dizin altındaki tüm dosyaların yetkisini değiştirme)

- root kullanıcılarının bile yetkilerini engellemek için *chattr* (change attribute) komutu kullanılır.

```
$ chattr (-+ )i [dosya]
```

i: immutable

a: append (sadece ekleme yapılabilir.)

- *etc/passwd* immutable yapılarak sisteme yeni kullanıcı ekleme kapatılabilir.

attribute görüntüleme;

```
$ lsattr [dosya]
```

user ve group değiştirme;

```
$ chown user:group [dosya]
```

suid (2) - sgid (4)

etc/passwd ve /shadow dosyalarında kullanıcıların write izni yoktur.

```
-rw-r--r--  1 root root    2469 Jul 21 18:26 passwd
-rw-r-----  1 root shadow  1276 Jul 21 18:26 shadow
```

Ancak kullanıcı *passwd* komutuyla parolasını değiştirebilir. Çünkü *passwd* komutu *suid* sayesinde diğer kullanıcılar için de çalışır. Dosya sahibinin yetkilerini anlık olarak kullanıcıya verir.

```
-rwsr-xr-x 1 root root 59640 Jan 25 18:09 ./passwd*
```

sticky bit (1)

```
chmod +t
```

/tmp dizini;

```
drwxrwxrwt 15 root root    4096 Jul 26 12:17 tmp/
```

```
chmod xnnn
```

x: stickybit, suid, guid değerlerini toplamı girilerek kullanılabilir.

apt

- **apt install**
- **apt search**
- **apt update**
- **apt upgrade**
- **dpkg -l:** Kurulu paketleri listeler.

Servis Yönetimi

SysV: Kernel yüklendikten sonra init başlar. Init belli scriptleri çalıştırarak sırayla sistem servislerini başlatır.

```
$ chkconfig
$ service
```

Systemd: Servisler paralel olarak başlatılabilir. Sadece talep edilen servisler başlatılabilir.

d: daemon (arkada çalışan)

/usr/lib/systemd/system: RPM ile gelen dosyalar

/run/systemd/system/: çalışan dosyalar

/etc/system.d/system/: kullanıcı tanımlı

```
$ systemctl
```

| SysV | Systemd |
|-----------------------------------|------------------|
| init0: halt (shutdown -h) | poweroff.target |
| init1: single | rescue.target |
| init2: multi/terminal | multiuser.target |
| init3: multi-user with networking | multiuser.target |
| init4: undefined | |
| init5: X11 | graphical.target |
| init6: reboot (shutdown -r) | reboot.target |

- Uzak sunucu *init1 single mod (troubleshooting modu burada başlar)* ile başlatılırsa bazı servisler çalışmayacağından bağlantı kopar.

```
$ systemctl (start-stop-status) service
$ systemctl list-units --type target
$ systemctl get-default
$ systemctl is-enabled
$ systemctl isolate poweroff.target
```

Güvenlik modeli;

chkconfig on/off

Debian, Suse --> apparmor RHEL --> selinux

/etc/sysconfig/selinux: ayar dosyası (SELINUX=0 yapılırsa devre dışı)

Durum görüntüleme;

```
$ getenforce
```

Çalışan sistemde değişiklik;

```
$ setenforce [0-1]
```

/var/log/audit/audit.log: log dosyaları

/proc dizini altında processler vardır. top, ps komutları bu dizinleri okuyarak gösterir.

Komutlar

- **jobs:** Çalışan işlemlerin listesini verir.
- **bg:** Durdurulmuş işlemleri arkaplanda devam ettirir.

```
bg [job]
```

- **fg:** Durdurulmuş işlemi önplanda devam ettirir.

```
$ fg
```

En son durdurulmuş ya da arka plana atılmış işlemi devam ettirir.

uptime: Sistemin ne zamandır açık olduğunu ve son 1, 5 ve 15 dakikalık *system load averages* gösterir.

- **ps:** Mevcut işlemlerin-süreçlerin raporunu verir.

```
ps auxf
```

f: ağaç yapısı

Sisteme sorun durumunda kontrol komutları;

- **top:** İşlemci durumu (htop)
- **free:** Ram bilgileri
- **df:** Disk bilgileri
- **du:** Dosya bazında disk bilgisi

Tüm dizinlerin boyutlarını inceleme;

```
$ du /*
```

Zombie Süreç: Parent süreçlerin altında child süreçler bulunur. Bazı durumlarda child processler ölür ancak parent'a bu bilgi bir şekilde gidemez. Bu child sürece zombie süreç denir. Zombie süreci öldürmek için parentı öldürmek gerekir.

```
$ kill
```

```
$ kill -9
```

```
$ killall
```

```
$ killall -u [user]
```


Kullanıcı işlemleri +19'dan -20'ye (düşük öncelik > yüksek öncelik) kadar öncelik sıralaması yapabilir.

- **nice:** yeni işlem başlatarak öncelik belirler.

```
$ nice --12 large-job
```

--: ikinci -,eksiyi temsile eder. Pozitif değer için + girilmez.

- **renice:** Çalışmakta olan işlem için öncelik belirler.

```
$ renice -3 -p 1134
```

Öncelik sırasını -3 olarak değiştirir.

etc/init.d dizininde servisler bulunur.

Sistem Kayıtları (Loglar)

| Facility | Priority |
|-----------|----------|
| kern | emerg |
| user | alert |
| mail | info |
| syslog | warning |
| user | crit |
| auth | err |
| ... | debug |
| local 0-7 | |

/etc/rsyslog

Logları yönetir.

rsyslog = syslog + module

rsyslog.conf dosyası düzenlenerek log sunucusu haline getirilebilir.

```
...
# provides UDP syslog reception
#module(load="imudp")
#input(type="imudp" port="514")

# provides TCP syslog reception
#module(load="imtcp")
#input(type="imtcp" port="514")
...
```

izinler:

```
# Set the default permissions for all log files.
#
$FileOwner syslog
$FileGroup adm
$FileCreateMode 0640
$DirCreateMode 0755
$Umask 0022
$PrivDropToUser syslog
$PrivDropToGroup syslog
```

/etc/rsyslog.conf: ayar dosyaları tutulur. Ancak bu yapının değiştirilmesi pek istenmez. Bu yüzden bir de *rsyslog.d* dizini bulunur. Değişiklikler buraya eklenir.

| Facility Örn. | Hedef |
|---------------|--------------------------|
| mail.alert | /var/log/mail.alert.log |
| mail.* | /var/log/mail.log |
| | /var/log/httpd/httpd.log |

| | |
|----------|--------------------|
| local0.* | @syslogsunucu:port |
|----------|--------------------|

```
logger -p mail alert
```

Log dosyası oluşturma

1-rsyslog.conf'a facility ve dizini eklenir.

```
local5.*    /var/log/deneme.log
```

2-servis yeniden başlatılır.

```
$ service rsyslog restart
```

3-logger ile log gönderilir.

```
$ logger -p local5.info "test logu"
```

/var/log:

- /messages: sistem
- /dmesg: çekirdek logları
- /boot.log: sistem açılış
- /yum.log: paket yüklemeleri
- /cron: zamanlanmış görevler
- /last
- /wtmp: kullanıcı güvenliği
- /btmp

logrotate: Belli aralıklara göre logların başka işlemlerden geçmesi gerekir. (Taşınması, sıkıştırılması ve zamanlanması gibi.)

/etc/logrotate.d/: Belli zaman aralıklarıyla logların arşivlenmesini yapar.

last: sistem loglarını görüntüler. (/var/log/wtmp)

lastb: yanlış parola girme gibi güvenlik ihlali loglarını gösterir. (/var/log/btmp)

Merkezi log sistemleri

- syslog
- graylog
- logstash

Disk Yönetimi

MBR: Master Boot Record (primary extended) (max 4'e kadar bölünebilir, primary ve extended seçilir.)

GPT: GUID Partition Table (512 MB) (daha güvenli)

- /etc/fstab: Diskleri bilgisayar açılışında mount eder.

Yeni Disk Sistemi Oluşturma

1-Diski bölümle (fdisk/parted) 2 Diski formatla (mkfs) 3 Diski bağlama (mount+/etc/fstab)

1a-fdisk

Disk yönetim aracı.

```
$ fdisk [/dev/sdb]
```

- n: new partition
- p: print partition table
- w: write changes !

1b-parted

Diskleri listele;

```
parted -l
```

- select: disk seçimi
- mklabel: bölümlendirme seçimi (MBR/GPT)
- mkpart: (ext4/fat32)
- print
- unit: MB/GB/TB

2-Diski formatlama

oluşturduğumuz disk hala formatlı değil.

```
$ mkfs -t ext4 /dev/sdb1
```

3-Diski bağlama

```
$ mount /dev/sdb1 /mnt/
```

disk bağlandı. Görüntülemek için;

```
$ df -hT
```

Bilgisayar yeniden başladığında bağlantının kalıcı olması için */etc/fstab* a yazılır.

```
...
/dev/sdb1    ext4    defaults    0 0
```

Yardımcı Komutlar

List block devices:

```
$ lsblk
```

mount/unmount;

```
$ mount [disk(/dev/sda1)] [bağlanılacak_dizin]
$ umount
```

fsck

Linux dosya sistemlerinin kontrol ve tamir işlemlerini gerçekleştirir.

Diskte hata taraması yapar. Örneğin badblock. Diskteki badblock karantına altına alınabilir ancak bulaşıcıdır. Diskin en kısa sürede değiştirilmesi gerekir.

LVM

Link oluşturma

1-Hard link

Linklenen tüm dosyalar aynı inode numarası ile kaydedilir. Diskte her biri ayrı ayrı yer kaplamaz. Biri değişince diğerleri de değişir ancak biri silinince diğerleri silinmez. Hard link oluşturulurken tüm linkler için full path zorunludur. Hard link sadece aynı disk üzerinde oluşturulabilir. Çünkü inode numarası çağırır ve bu diske özeldir.

```
$ ln [dosya_1] [dosya_2] [dosya_3]
```

- Neden etc/passwd dosyası hard link oluşturulmaz?

Oluşturulan dosyanın user:group değerleri farklı olacağından buna izin verilmez.

2-Soft link (Symbolic link)

Windows'taki kısayola benzer. Linklenen dosya inode numarasını değil, ana dosyayı gösterir. Farklı disklere soft linkler yapılabilir.

```
$ ln -s [ana_dosya] [link_dosyasi_1] [link_dosyasi_2]
```

- Eğer soft link oluşturulurken full path yerine bağlı bir adres kullanılırsa, link dosyası taşındığında çalışmaz.

cron

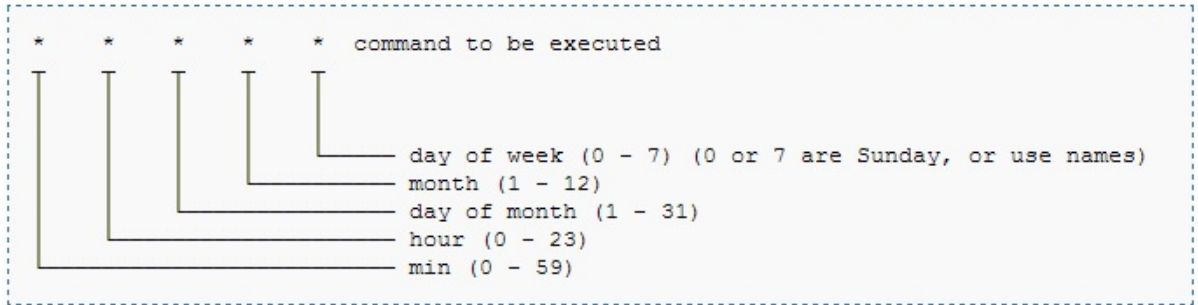
Cron oluşturma;

```
$ sudo crontab -e
```

```
#
#
* * * * * "date >> /tmp/cron.txt"
```

eklediğimiz cron her dakika cron.txt'ye tarih yazar.

| Entry | Description | Equivalent To |
|------------------------|---|---------------|
| @yearly (or @annually) | Run once a year at midnight in the morning of January 1 | 0 0 1 1 * |
| @monthly | Run once a month at midnight in the morning of the first of the month | 0 0 1 * * |
| @weekly | Run once a week at midnight in the morning of Sunday | 0 0 * * 0 |
| @daily | Run once a day at midnight | 0 0 * * * |
| @hourly | Run once an hour at the beginning of the hour | 0 * * * * |
| @reboot | Run at startup | @reboot |



- ***/***: 10 (dk-saat...)'da bir.
- **6-18/3**: 6 ile 18 arasında her 3 saatte bir.
- **12,18**: 12'de ve 18'de

çevresel değişkenleri sisteme tanımlama;

```
export
```

history komutuna tarih saat bilgileri ekleme;

/etc/profile.d/histdate dosyası oluşturulur ve aşağıdaki satır eklenir.

```
export HISTTIMEFORMAT="%d%m%y %T "
```

ekleme;

```
source /etc/profile.d/histdate
```

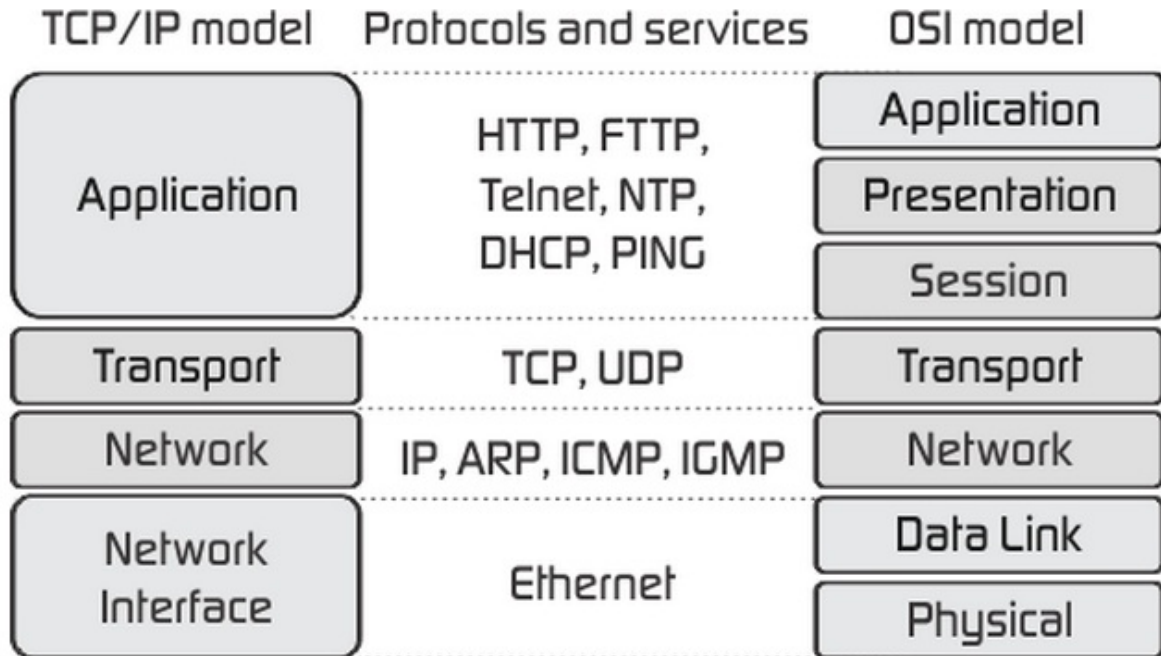
kontrol;

```
$ $HISTTIMEFORMAT  
>%d%m%y %T
```

-
- **source:** Bir dosyadaki komutu mevcut shell de çalıştırır.
 - **set:** Tanımlanmış çevresel değişikliklerinin listesi görüntülenir.

Temel TCP/IP Bilgisi ve Ağ Yönetimi

| TCP/IP modeli - OSI referans modeli |
|---|
| Uygulama (HTTP, HTTPS, FTP, DNS , DHCP, SSL, SSH, IRC...) |
| Sunum |
| Oturum (SMB, NFS) |
| Ulaşım (TCP, UDP) |
| Ağ (IPv4, IPv6, ICMP) |
| Veri (ethernet, wifi) |
| Donanım (RS-232, fiber optik) |



IPv4 = $4 \times 8 = 32$

IPv6 = $8 \times 16 = 128$

1. Dış (Public) IP
2. Özel (Private) IP
3. subnet (alt ağ)
4. broadcast (yayın)

192.168.0.0 (IP ağ adresi) - 255.255.0.0 (alt ağ adresi): 16 (8: Private ip)

gateway

ARPANET

CIDR (Classes Inter-Domain Routing)

TCP (Transmission Control Protocol): Verilerin bütünlüğünü korumak temelli kullanılır. Three handshake yöntemi ile çalışır, UDP'ye göre daha yavaş.

UDP (User Data Protocol): Veri kaybı önemsizdir, genelde tek yönlü çalışır. Oyun servisleri ve tv servisleri örnek verilebilir. Paketleri daha küçük.



DHCP (Dynamic Host Configuration Protocol): Otomatik ip dağıtma sistemi. Sunucu ve istemci olması gerekir.

Statik (elle) IP

hosts

13 root rare server (TLD)

IP bilgisi;

```
ip a (ip addr)

route
```

Erişim kontrolü bilgileri;

```
traceroute
```

-n: IP adresleri ve host name'leri eşleştirmeye çalışmaz.

-i: interface

-T: TCP

Ağ kullanım bilgileri;

```
iftop -i enp0s3
```

/etc/services

telnet: TCP port kontrolü (güvensiz)

no route to host: DROP

unable connect to host: REJECT

TCP/UDP port kontrolü: **netcat**

listening;

```
nc -lu 8000
```

-u: udp

```
nc 192.168.56.101 -u 8000
```

yaparak bilgi gidip gitmediği kontrol edilebilir.

firewall'u kapatmak için;

```
systemctl stop firewalld
```

İstatistik verileri (hop):

```
mtr  
  
ethtool -i enp053
```

Ağ kartı ışığını yakıp söndürme;

```
ethtool -p enp053  
  
netstat (nettools) -nt(u)lp | grep ssh
```

n: isim (numeric)

t: tcp

l: listening

p: show PID name

```
tcpdump -i enp053
```

-w: deneme.pcap

-r: deneme.pcap

(wiresharck, air-crack-engine pcap dosyalarını okuyabilir)

Ağ taraması;

```
nmap -sP 192.168.56.0/24
```

-f:

-p:

-A: işletim sistemi bilgileri

Network Manager

Terminal ağ yönetim uygulaması

```
nmtui
```

Bilgi alma;

```
nmcli
```

DNS Teknolojisine Giriş

kamp.linux.org.tr;

tr --> org --> linux --> kamp

Alan adı çözümleme;

/etc/resolv.conf: Dns sunucu tanımları

/etc/hosts: Yerel adresler

/etc/nsswitch: Dns ve yerel adresler arasında önceliği belirler.

<https://www.iana.org/domains/root/servers>

TTL(time to leave) : Geçerlilik süreleri, daha hızlı cevap

DNS kayıtları;

- A
 - CNAME
 - MX
 - SDA
 - NS
 - SPF
-

Hosts dosyası test ve kontrol amaçlı yereldeki web sitesi düzenlemelerinde kullanılabilir.

service network restart = systemctl restart network

Domain bilgileri;

```
whois [domain]
```

IP/isim dns sorgulama;

```
nslookup [domain] -query (-q) a: ns(name server), mx(mail exchange), any ...
```

```
dig [domain]
```

```
host [domain]
```

isim --> IP (forward lookup)

IP --> isim (reverse lookup)

dns primary (8.8.8.8) - secondary (8.8.4.4):

FQDN: www.deneme.com

host: www

domain: deneme.com

ssh (Secure Shell)

OpenSSH

port:22

-x: x11 (grafik bağlantı)

```
ssh kullanıcı_adi@sunucu
```

karşıda komut çalıştırma;

```
ssh kullanıcı_adi@sunucu "komut"
```

-p: 22 (default port) nin dışında port vermek için

-v: verbose

genel ayarlar: /etc/ssh/

kullanıcı: /home/near/.ssh/

ssh_config: istemci ayarları

sshd_config: sunucu ayarları

systemctl status sshd: çalışan sunucu servisi

Sunucuya bağlanmak için;

```
$ ssh kullanıcı_adi@sunucu_ip
```

Sunucuya parola girmeden bağlanabilmek için ssh key dosyaları (anahtar seti) sunucuya kopyalanır. Yoksa;

```
$ ssh-keygen
```

komutu ile oluşturulur.

Sunucuya kopyalamak için;

```
$ ssh-copy-id kullanıcı_adi@sunucu_ip
```

-idnetify file ~/.ssh/id_rsa: başka dizindeki anahtar seçimi

Sunucuya özel konfigürasyon ayarları yapmak için /home/x/.ssh/config dosyası oluşturulur.

```
host sunucu
user kullanıcı_adi
hostname 178.128.244.33
```

Parola değiştirme;

```
$ passwd
```

Sunucu - Bilgisayar Dosya Kopyalama

yerel --> sunucu

```
scp dosya_adi kullanıcı@sunucu:dizin
```

sunucu --> yerel

```
scp kullanıcı@sunucu:dosya /home/near
```

rsync

Sunucudan bilgisayara;

```
$ rsync -turv sunucu:[sunucu_dizini] [bilgisayar_dizini]
```

Bilgisayardan sunucuya;

```
$ rsync -turv sunucu:[bilgisayar_dizini] [sunucu_dizini]
```

- t: preserve modification times
- u: update (dosya güncelleme)
- r: recursive
- v: verbose
- a: metadataları da kopyalar (kullanıcı izinleri)

FTP (file transfer protocol)

SFTP (secure file transfer protocol)

Uzak sunucuya anahtar taşıyarak bağlanma;

```
ssh-add
```

- -l: list agent
- -D: delete

Yerel port tünelleme

Uzaktaki sunucuda güvenlik gibi sorunlardan dolayı erişilemeyen web hizmetine tünelleme yaparak yakın sunucudan erişebiliriz.

```
ssh -L 2201:192.178.56.101:22(uzak_sunucu) 192.168.56.102(yakin_sunucu)
```

2201: yakın sunucuda açılacak port

22: uzak sunucuyu aktarmak istediğimiz port

Uzak (ters) port tünelleme

(Dışarıdan ulaşılamayan uzak sunucudan ters tünelleme yapılır, böylece istemci yakın sunucu aracılığıyla artık uzak sunucuya bağlanabilir. Teknik destek amaçlı kullanılabilir. (NOT: yakın sunucu ile istemci aynı bilgisayar olabilir.)) Uzak makineden aşağıdaki komut çalıştırılır.

```
$ ssh -R 2201:127.0.0.1:22 192.168.56.102
```

..102: yakın

..localhost: uzak (ters tünelleme açan bilgisayar)

22: uzak sunucu portu

2201: yakın sunucu (kendi belirlediğimiz)

yakın sunucudan uzak sunucunun bağlanıp bağlanmadığı kontrol edilir;

```
$ netstat -ntlp | grep 2201
```

yakın sunucudan uzak sunucuya bağlanmak için;

```
$ ssh -p 2201 127.0.0.1
```

grafik arayüzü ile bağlanma;

ssh bağlantısı kurulan istemcide xorg-server kurulu olmalıdır.

```
$ ssh -X kullanıcı@sunucu
```

sertifika: LPIC, SUSE, RHEL

Sıkıştırma

ZIP

GZIP

- gzip
- gunzip
- zcat: İçeriğini görüntüler.
- zmore
- zless
- zgrep
- -k: keep
- -1-9: fast to best

BZIP2

- bzip2
- bunzip2
- bzip2

XZ

Arşivleme

Klasörleri sıkıştırabilmek için önce arşivlemek gerekir. Arşivlerken eski dosyalar saklanır.

arşivleme:

```
$ tar -cvf [arsiv_olusturulacak_dizin.tar] [/arsive_eklenecek_dizinler]
```

arşivi görüntüleme:

```
$ tar -tvf [dosya adı]
```

arşivden çıkarma:

```
$ tar -xvf [arsiv_dizini] [cikarilacak_dosya(girilmezse hepsini çıkarır)] -C [/çıkartılacak_dizin]
```

- -c: create
- -v: verbose
- -f: file
- -x: extract

aynı zamanda arşivleme ve zipleme:

```
$ tar -cvzf arxiv.tar.gz [dizin]
```

- z: gzip
- j: bzip2
- J: xz

arşive dosya ekleme; (arşive dosya ekleyebilmek için sıkıştırılmamış olmalıdır.)

```
$ tar -rvf [arsiv] [eklenecek_dosyalar]
```

arşivi bölme; (herhangi bir dosyayı bölmek için de kullanılabilir.)

```
$ split -b [parca_bouyutu_K-M-G] [arsiv.tar.gz] ["arsiv.tar.gz.part"]
```

Bölünen dosyaları birleştirme;

```
$ cat "...part*" > a.iso
```

Web Teknolojisine Giriş

İstemciler, sunucular, uygulama ve veritabanı

HTTP: 80 numaralı porttan çalışır.

HTTPS: 443 numaralı porttan çalışır.

URL: <https://kamp.linux.org.tr>

İstemci: Chrome, Firefox

Teknolojiler:

- HTML
- CSS
- JS

Sunucular: Apache, Nginx, IIS, Google

Teknolojiler:

- PHP
 - Apache
 - mail.php
 - PHP-FPM
- .Net
- Java
 - Tomcat
 - JBoss
 - Wildfly
 - Webklogic
 - Glassfish
- Python
- Ruby on Rails
 - Unicorn
 - Passenger

Veritabanı

- İlişkisel veritabanı
 - MySQL
 - Oracle
- İlişkisel olmayan veritabanı
 - MongoDB
 - Redis
 - Cassandra
- Aarama veritabanı
 - Elastic search
 - Solar
- Nesne Önbelleği veritabanı
 - Redis
 - Memeache

- - - - -

Sayfa Bütünlüğü;

- GET: adres satırında
- POST: sayfa kaynağında
- COOKIE: oturum yönetimi, kullanıcıya özel sayfa, sunucu takibi
- SSL (Secure Shell Layer): OpenSSL
- sslabs: sertifika doğrulama sitesi
- letsencrypt: sertifika oluşturma

List of HTTP status codes

- 1xx (informational response)
- 2xx (success)
- 200: OK
- 3xx (redirection)
- 301: moved permanently
- 302: found
- 304: not modified
- 4xx (client errors-istemci hata kodlari)
- 400: bad request
- 401: unauthorized
- 403: forbidden
- 404: not found
- 451: Unavailable for legal reasons
- 5xx (Server errors)
- 500: internal server error
- 502: bad gateway
- 503: service unavailble
- 504: gateway timeout

[tüm kodlar](#)

Uzaktan erişemediğimiz sunucuya port açarak ulaşma;

```
ssh -D 6868 centos
```

firefox socks proxy port eklendiğinde istemciden ulaşılabilir.

Konsoldan web sayfası içeriklerini görüntüleme;

```
# yum install elinks  
  
# links 192.168.56.101
```

nginx kurulumu;

```
$ yum install http://nginx.org/packages/centos/7/noarch/RPMS/nginx-release-centos-7-0.el7ngx.noarch.rpm  
  
$ yum install nginx  
  
$ systemctl start nginx  
  
$ systemctl enable nginx
```

Güvenlik duvarına ekleme;

```
$ firewall-cmd --permanent --add-service=http
```

kontrol;

```
$ firewall-cmd reload
```

yum bilinmeyen paket kaldırma;

```
$ yum -qa | grep nginx  
  
$ yum remove ...
```

DiskDuplicator (convert and copy file)

biçimlendirme;

```
$ dd if=/dev/null of=/dev/sdb
```

İlişkisel Veritabanı

primary key/secondary key

SQL (Structured Query Language): veritabanlarında kullanılan dil

- DDL: Data Defination Language
- DML: Data manipulation language

Nümerik veri tipleri: integer, float

Metin veri tipleri: character/char, varchar, text

Tarih/saat veri tipleri: date, time, timestamp

Mantıksal veri tipleri: boolean (true/false)

DML (Data Manipulation Language)

SELECT: sorgulama

INSERT: veri girişi

DELETE: silme

UPDATE: güncelleme

```
kolon1,kolon2 FROM tablo1 WHERE kolon1=1;
... ORDER BY kolon2;
... GROUP BY kolon1,kolon2;
```

ASC/DESC: ascending/descending

WHERE: = , > , < , >= , <= , <> / ! (eşit değildir) , AND , OR

PostgreSQL kurulum;

```
# sudo yum install https://download.postgresql.org/pub/repos/yum/10/redhat/rhel-7-x86_64/pgdg-centos10-10-2.noar
ch.rpm
# sudo yum install postgresql10-server postgresql10-contrib
# /usr/pgsql-10/bin/postgresql-10-setup initdb
# systemctl start postgresql-10
# systemctl enable postgresql-10
```

postgresql oturum açma;

```
# su - postgresql
$ psql
```

```
# su - postgresql
$ psql
# CREATE DATABASE dvdrental;
\q
```

```
wget http://www.postgresqltutorial.com/wp-content/uploads/2017/10/dvdrental.zip
unzip dvdrental.zip
pg_restore -d dvdrental dvdrental.tar
```

```
psql
\c dvdrental
```

```
\dt
```

```
SELECT * from film;
:Space:
:q:
```

Apache kurulumu;

```
# yum install httpd  
  
# systemctl start httpd  
  
# systemctl enable httpd
```

modül ekleme;

```
# yum install mod_ssl(ornek modul)  
  
# systemctl reload httpd
```

sürüm bilgisi;

```
# apachectl -version  
  
# rpm -qa | grep httpd  
  
# httpd -v
```

- /etc/httpd/conf/httpd.conf
- /etc/httpd/conf.d/*.conf
- /etc/httpd/conf.modules.d/*.conf

LAMP (linux apache mysql php)

Bash Betikleri

Shebang: #!

#!/bin/bash

- .sh uzantili
- `chmod +x myscript.sh`

```
echo "xxx" | mail -s "Baslik" fatihkaansalgir@gmail.com
```

```
printf "Mesaj"

>Mesaj$
```

```
printf "Mesaj\n"

>Mesaj
>$
```

```
printf "%s\n" "bir" "iki" "uc"

>bir
>iki
>uc
```

```
printf "%.2f\t" 3.5 2.75 6.375

>3.50    2.75    6.37
```

```
printf "%03d\t" 1 3 12 405

>001    003    012    405
```

`\t`: ciktida karakterler arasina *tab* koyar.

- Eşittirin sağında veya solunda boşluk bırakılmaz.

```
degisken="ahmet" #dogru
```

- * ve ? değişkende kullanılmaz.
-

- `_` kullanılabilir.

```
_GREP=/bin/grep #dogru
```

- Değişken çağırma;

```
echo "$_GREP"
>/bin/grep

echo '$_GREP'
>$_GREP
```

- Komut çağırmak için: `komut` veya (komut)

```
echo "Bulundugum yer: $(pwd)"
```

```
echo "Gunun tarihi: $ date +%d/%m/%y "
```

```
Degisken="Ahmet geldi"
```

```
karakter=7
```

```
echo ${Degisken:0:Karakter}

>Ahmet g
```

-
- Son komutun/programın döndüğü değer: `$?`
 - Betiğin PID: `$$`
 - Kullanıcı: `$USER`
 - Hangi dizindeyim: `$PWD`
 - Kullandığım path: `$PATH`

```
./betik.sh ;
```

- Betiğe geçirilen argüman sayısı: `$#`
- Betiğe geçirilen argümanlar: `$@`
- Betiğin ismi(kendisi): `$0`
- - 1. argüman: `$1`

```
echo "Isminiz?"
read Isim
echo "Merhaba, Isim!"
```

Sayısal karşılaştırma operatörleri

- `eq` (equal)
 - `ne` (not equal)
 - `gt` (grater than)
-

- ge(g)
- lt(lesser than)
- le()

String karşılaştırma operatörleri

- == veya =: esit
- !=: esit değil
- -z: null
- -n: null değil

Dosya karşılaştırma operatörleri

- d: izin ise doğru
- f: dosya ise doğru
- r: okunabilir
- s: 0 byte boyutlu
- w: yazılabilir
- x: çalıştırılabilir
- L: link

test: , []

0: doğru

1: yanlış

```
test a!=b && echo $?  
>0
```

\$((1+2))

- +: toplama
- -: çıkarma
- *: çarpma
- /: bölme
- %: mod alma
- ++: bir arttır
- --: bir azalt
- **: üssü

Diziler

meyveler=(elma armut kivi muz)

echo \${meyveler[0]}: 0. elemanı yazdırır.

echo \${meyveler[@]}: tüm elemanları yazdırır.

echo \${meyveler[*]}: dizideki eleman sayısı

echo \${meyveler[@]:1}: 1. elemandan sonrası

echo \${meyveler[@]:1:2}: 1. ile 2. arası

Koşul yapıları (if, else, elif, case)

```
sayi=100
if [ "$sayi" -eq 100 ]
then
    echo "sayi 100"
fi
```

```
rakam=99
if ["$rakam" -eq 100]; then
    echo "Rakam 100"
else
    echo "Rakam 100 değil"
fi
```

```
if [ -f ~/.bash_profile ]; then
    echo "bash_profile dosyası var."
else
    cp /etc/skul/.bash_profile ~/
fi
```

```
rakam=99
if ["$rakam" -eq 100 ]; then
    echo "Rakam 100"
elif ["$rakam" -gt 100 ]; then
    echo "rakam 100'den büyük"
else
    echo "rakam 100'den küçük"
fi
```

```
BUGUN=$(LC_TIME=en_US.utf8 date + "%a")
case $BUGUN in
    Mon)
        echo ""
    case $BUGUN in
        Tue|Wed|Thu|Fri)
            echo ""
    case $BUGUN in
        Sat|Sun)
            echo ""
    *)
        echo "bir sorun oluştu"
esac
```

Döngü Yapıları (for, while, until)

```
for Degisken in {1..50}
do
    echo "$Degisken"
done
```

```
$ SON=50; for i in $(seq 1 $SON); do echo $i; done
```

```
deger=0
while [ $deger -lt 4 ]
do
    deger=$((deger+1))
    echo "$deger"
done
```

```
deger=0
until [ $deger -gt 4 ]
do
    echo "$deger"
    deger=$((deger+1))
done
```

Çıkış kodları

- 0: başarılı
- 1: genel hatalar
- 2: içsel komut yanlış kullanım
- 126: komut çalıştırılmayan
- 127: komut bulunamadı
- 128: exit'e yanlış parola
- 128+n: ölümcül hata (işlemci süreç kodları)
- 130: komut ctrl+c ile sonlandırıldı.