

## Ayrıntılı Linux Terminal Komutları

# Router'ın IP'sini verir

```
route -n // ya da : netstat -r -n
```

# Kaynak koddan Linux'ta kurulum yapar.

```
./configure // Windows'ta next next yaparkenki ayarlamalar gibi
```

```
make // Compile yapar.
```

```
make install // Install yapar.
```

# Sitemi kapatır.

```
shutdown
```

# Sistemi kapatır.

```
halt
```

# Sistemi kapatır.

```
poweroff
```

# Sistemi reboot edecektir.

```
shutdown -r
```

# Sistemi reboot edecektir.

```
reboot
```

# Yerel ağda tüm terminallere broadcast yaparak "Sistemi kapatmamız

# gerek..." mesajını yayınlar ve sistemi parametrede belirtildiği gibi

# 10 dakika içinde kapatır.

```
shutdown +10 "Sistemi kapatmamız gerek. İşlerinizi 10 dakikada halledin"
```

# Çalışmakta olan servislerin isimlerini ve kullandıkları port

# numaralarını gösterir.

```
cat /etc/services
```

# Bir linux makinasındaki ethernet kartını disable etmeye yarar.

```
ifconfig eth0 down
```

# Bir linux makinasındaki ethernet kartını aktif hale getirmeye yarar.

```
ifconfig eth0 up
```

# Bir linux makinasındaki ethernet kartına geçici ip yapılandırması

# atar.

```
ifconfig eth0 192.168.130.134 netmask 255.255.255.0
```

# Router'ın hal-i hazırda kullandığı yönlendirme tablosunu ekrana basar.

```
route
```

# Ethernet kartını dinler ve tüm trafiği ekrana anlık olarak sunar.

```
tcpdump -i eth0
```

# Belirli bir IP adresine giden ve gelen trafiği gösterir.

```
tcpdump -i eth0 host www.includekarabuk.com

# Belirli bir IP adresinin belirli bir portu dinlenir ve trafik ekrana
# anlık olarak yansıtılır.
tcpdump -i eth0 tcp port 80 and host www.includekarabuk.com

# Belgenin ilk 3 satırını yazdırır.
head -n 3 dosya.txt

# Belgenin son üç satırını yazdırır.
tail -n 3 dosya.txt

# Belgenin son beş satırını yazdırır ve her 3 saniyede bir bu olayı gün-
# celler (Anlık veri alan log dosyalarında kullanışlıdır).
tail -n 5 -f logDosyasi.txt

# var dizini altındaki dosyaları listeler.
ls /var

# Gizli dosyalar dahil bütün dosyaları listeler.
ls -al

# O an sistemde bulunan kullanıcıların kullanıcı isimlerini, hangi
# uçbirimlerde çalıştıklarını ve sisteme giriş tarih ve saatlerini
# gösterir.
who

# CPU, RAM kullanımı ve çalışan process'leri görüntüler.
htop

# Process'leri sıralar.
ps

# ps komutunun sunduğu çıktıyı [anlık] olarak sunan bir arayüz.
top

# Aktarılan birinci komutun output'undaki bash kelimesinin geçtiği
# satırı ya da satırları ekrana basar:
ps aux | grep bash

# pid değerine göre process'i sonlandırır.
kill pidNumber // pid numarası ps aux | grep processAdi ile öğrenilir.

# pid değerine göre değil de direk ismiyle process yok etmeye yarar.
killall processAdi // e.g. killall httpd

# Process takıldı ve sonlanmıyorsa -9 parametresi girilir.
kill -9 pidNumber

# file1.txt içeriğini file2.txt'e yazar (overwrite)
cat file1.txt > file2.txt
```

# file1.txt içeriğini file2.txt içindekilerin sonuna ekler (append)

```
cat file1.txt >> file2.txt
```

# file1.txt ve file2.txt içeriğini alt alta file3.txt'ye yazar

# (overwrite)

```
cat file1.txt file2.txt > file3.txt
```

# Sisteme hasan isimli kullanıcı ekler.

```
useradd hasan
```

# Sistemdeki hasan adlı kullanıcının şifresini değiştirir.

```
passwd hasan
```

# Sistemdeki hasan adlı kullanıcının hesabını siler.

```
userdel hasan
```

# Sistemdeki hasan adlı kullanıcının hesabı ilişkili olduğu /home dizinin-

# deki klasörle beraber silinir.

```
userdel -r hasan
```

# hasan adlı kullanıcının hesabını geçici süreliğine kilitler.

```
usermod -L hasan // L: Locked
```

# hasan adlı kullanıcının hesabı açılır.

```
usermod -U hasan // U: Unlocked
```

# Yeni bir grup oluşturur.

```
groupadd sistemYoneticileri
```

# Varolan bir grubu siler.

```
groupdel sistemYoneticileri
```

# Gruba kullanıcı ekler.

```
gpasswd -a hasan // a: add
```

# Gruptan kullanıcı siler.

```
gpasswd -d hasan // d: delete
```

# Belirli bir kullanıcının dahil olduğu grubu (grupları) gösterir.

```
groups hasan
```

# Dosyanın (Klasörün) sahibini değiştirir.

```
chown kullanıcıİsmi dosyaİsmi
```

```
chown -R kullanıcıİsmi klasorİsmi
```

# Dosyanın (klasörün) ait olduğu grubu değiştirir.

```
chgrp yeniGrupİsmi dosyaİsmi
```

# Dosya satırlarını alfabetik sıralı olarak ekrana basar.

```
sort dosyaİsmi
```

# Aranan stringin bulunduğu satırı ya da satırları ekrana basar.

```
cat deneme.txt | grep "arananAnahtar"
```

```
# Aranan stringin kaç tane satırda bulunduğuun sayısını ekrana basar.  
cat deneme.txt | grep -c "arananAnahtar"
```

```
# Aranan string büyük küçük harf oluşuna bakmaksızın aranır ve bulunan  
# satırlar ekrana basılır.  
cat deneme.txt | grep -i "arananAnahtar"
```

```
# Aranan stringin bulunduğu satırları göstereceği gibi çıktıdaki her  
# satırın başında bir de kaçınıcı satırda oldukları bilgisi (sayısı)  
# gösterilir.  
cat deneme.txt | grep -n "arananAnahtar"
```

```
# Dosyadaki satırların sayısını ekrana basar.  
cat deneme.txt | wc -l // -l : line
```

```
# Dosyadaki kelimelerin sayısını ekrana basar.  
cat deneme.txt | wc -w // -w : word
```

```
# Dosyadaki karakter sayısını ekrana basar.  
cat deneme.txt | wc -c // -c : character
```

```
# Dizindeki öğeler satır satır sıralanır. wc ile de satır sayısı  
# sayılır. Böylece dizindeki öğe sayısı bulunmuş olur.  
ls -l | wc -l
```

```
# Mevcut dizin ve aşağısındaki tüm dosyaların içeriğini tarar ve  
# "merhaba" string'i bulunan satırları dosya ismi : "aranan string"  
# şeklinde ekrana basar.  
grep -r "merhaba"
```

```
# Belirtilen pattern'a uygun string'leri tarar ve bulunduğunda ekrana  
# basar.  
cat deneme.txt | grep -E "ps au*" // ps aux satırları ekrana gelir.
```

```
# Sed gelen / giden içerikleri düzenleyerek belirli hedefe  
# çıkılmaya yarayan bir stream editor'dür. -i parametresi  
# (--in-place) düzenlemenin başka hedefe doğru çıkılanmasını  
# değil de argumanda belirtilen dosya üzerine çıkılanmasını  
# sağlar. s variable'ı select'tir ve string içeriği seçer.  
# sed komutu aşağıda apache2.conf dosyasındaki ServerName  
# localhost string'ini ServerName deneme olarak düzenler  
# (günceller).  
sed -i "s/ServerName localhost/ServerName deneme/" /etc/apache2/apache2.conf
```

```
# Sed gelen / giden içerikleri düzenleyerek belirli hedefe  
# çıkılmaya yarayan bir stream editor'dür. -i parametresi  
# (--in-place) düzenlemenin başka hedefe doğru çıkılanmasını  
# değil de argumanda belirtilen dosya üzerine çıkılanmasını  
# sağlar. 7i variable'ı 7 numaralı satır arasında (inline'ı)  
# seçer. sed komutu aşağıda apache2.conf dosyasının 7nci satı-
```

```
# rına "# edited by hasan fatih" string'ini ekler ve mevcut
# devam eden apache2.conf içeriklerini aşağı doğru kaydırarak
# dosyayı düzenler (günceller).
sed -i "7i # edited by hasan fatih" /etc/apache2/apache2.conf

# Uzun ve kompleks komutlara bir kod ad atamaya ve o uzun kodu atanana
# kısa kod ile çalıştırmaya yarar.
alias kısaAd='ps aux | grep bash'
kisaAd // ps aux | grep bash çalışır.

# Atanan alias'ı silmeye yarar.
unalias kısaAd

# alias'ları ~/.bashrc dosyasına yerleştirirseniz sisteme her giriş-
# çıkışınızda yeniden aynıını tanımlamak zorunda kalmazsınız.
[Aynı kodlamayı ~/.bashrc dosyasının en altına koyduktan sonra
bash komutunu çalıştırmalısın. Böylece alias aktifleşir]
nano ~/.bashrc
bash

# Bulunulan dizine ait GUI penceresi açar:
nautilus .

# Dosya Sistemi Penceresi Açar:
gnome -open /dosyayolu

# PDF Dosyasını Açar:
evince file2open.pdf

# Belirtilen dosyanın klasör mü değil mi ve değilse hangi türde bir
# dosya olduğunu belirtir.
file filename

# ls,nano gibi temel komutlar dizisini yükler:
sudo apt-get install --reinstall coreutils

# ls,nano gibi temel komutlar dizisini yükler:
vertx_home=/anto/vertx
export PATH=$PATH:$vertx_home/bin

# .class uzantılı dosyalar metin belgelerinde görüntülenemezler.
# Assembly dilindedirler. Dolayısıyla şu şekilde disassembly edilebilir:
javap -c UpdateProfile // UpdateProfile.class

# Debian paketlerini yüklemeyi sağlar:
dpkg -i packetName.deb // or dpkg --install packetName.deb

# Dosyanın içeriğini aşağı doğru kaydırarak gösterir.
more file.txt

# Dosyanın içeriğini hem aşağı doğru hem de yukarı doğru kaydırarak
# gösterir.
```

less file.txt

# Dosyanın en başındaki ilk 10 satırı gösterir.  
head file.txt

# Dosyanın en sonundaki son 10 satırı gösterir.  
tail file.txt

# Stdin'den aldığı input'un çıktısını hem dosyaya  
# hem de stdout'a basmaya yarar (duplicate eder).  
echo "123" | tee ~/Desktop/sample.txt // 123 hem dosyaya hem ekrana  
// basılır.

# Kullanılan donanım modellerini gösterir.  
lspci -nnk // k çıktıya kernel bilgisini, nn ise  
// donanım ID'sini ekler. Donanım ID'si  
// donanımı internette araştırmak için  
// kullanışlı olabilir.

# Kullanılan ekran kartı modelini gösterir.  
lspci -nnk | grep VGA -A3 // VGA ile hardware info list'teki ilgili  
// satır eşleştirilir. -A 3 ile de eşleşen  
// satırdan sonraki 3 satırı daha ekrana ver  
// denir. ( -A, yani (A)fter)

# Kullanılan ethernet kartı modelini gösterir.  
lspci -nnk | grep net -A2 // net ile hardware info list'teki ilgili  
// satır eşleştirilir. -A 2 ile de eşleşen  
// satırdan sonraki 2 satırı daha ekrana ver  
// denir. ( -A, yani (A)fter)

# FTP sunucusuna login olmayı sağlar.  
ftp includekarabuk.com // Gelen ekrana kullanıcı adı  
// ve sonra şifre girilir.

# Bulunulan dizinin altındaki belirli isimde dosyaları topluca silmeyi sağlar.  
find . -type f -name "\*error\*" // Recursive olarak tüm alt dizinlerdeki error  
// string'ine sahip dosyaları listeler.  
// Not: Silme işlemine geçmeden önce silincek  
// listeyi incelemek ve yanlış dosya arada kay-  
// nayıp silinmesini önlemek için kullanılır.

find . -type f -name "\*error\*" -delete // Recursive olarak tüm alt dizinlerdeki error  
// string'ine sahip dosyaları siler.

// Benim Not: Bu komutu /var/www/includekarabuk\_inw/  
// dizininde kullandın ve 1GB'ı aşan ve farklı  
// farklı dizinlerde birden fazla kez bulunan  
// error.log dosyalarını topluca silmede kullandın.

Referans:

<https://askubuntu.com/questions/443830/delete-all-files-whose-filenames-contain-a-particular-string>