Linux faydalı komutlar

**Ubuntu** ve **debian**'nın paket yöneticisi **apt** dir. **CentOS** ve **Amazon Linux**'ün paket yöneticisi **yum** , **red-hat’**in paket yöneticisi **rpm, OS (Operating System). RPM distros: RHEL, CentOS, Oracle Linux, DEB distros: Debian, Ubuntu and Linuz Mint**

**apt** : advanced packaging tool

**yum** : yellowdog updater modified

sudo apt/yum update –y : -y soru sormadan devam et, yükle demek, \*\* -tavsiye edilen

sudo apt/yum upgrade

sudo apt/yum install git : git yükle

yum –y install <package name> (install without asking for confirmation)

yum check-update

yum update \*\* -tavsiye edilen

yum search git (search for packages)

sudo apt/yum autoremove git –y : konfigürasyon dosyaları ile beraber tamamen siler

sudo apt/yum info git

sudo yum/apt list : yüklenebilecek paketlerin listesini verir

sudo apt list --installed: yüklenmiş paketlerin listesini verir

sudo yum list installed: yüklenmiş paketlerin listesini verir

yum list installed | grep ssh (check for install ssh package)

sudo yum install git-2.14.5-1.amzn2 –y : istediğimiz versiyonu yükleriz.

sudo yum --showduplicates list git : dublikasyonları gösterir.

sudo yum --showduplicates list httpd

yum/apt/dnf search <string> serach for string

yum/apt/dnf info [package]

yum/apt/dnf install –y [package]

yum/apt/dnf remove package

yum/apt/dnf upgrade [package]

apt-get install –y <package> (install, remove, or update software packages)

apt-cache search <package> (search for packages)

apt-get Show package = display information about package

apt-get remove package = remove package, leaving configuration

apt-get purge package = remove package, deleting configuration

rpm –qa = list all install packages

rpm –qf /path/to/file = list the file’s package

rpm –ql package = list package’s files

rpm –ivh package.rpm = install package

rpm –e package = erase (uninstall) package

./first.sh ile çalıştırmak yerine pwd yapıp PATH’İ buluruz, sonra PATH=$PATH:/home/ubuntu (pwd yolu) ile çalıştırabiliriz.

export PATH=$PATH:$HOME # or current working directory export PATH=$PATH:$PWD

Ekranı yeşillendirmek için

PS1='${debian\_chroot:+($debian\_chroot)}\[\033[01;32m\]\u@\h\[\033[00m\]:\[\033[01;34m\]\w\[\033[00m\]\$ '

UBUNTU TREE   alias tree="find . -print | sed -e 's;[^/]\*/;|\_\_\_\_;g;s;\_\_\_\_|; |;g'"

veya

sudo apt update -y && sudo apt install tree –y

veya

sudo apt update

sudo apt upgrade

sudo apt install tree -y

. (this directory)

.. (parent directory)

cd (home directory)

cd – (go to previous directory, bir önceki çalıştığımzı dosyaya gider)

**updateler :**

sudo apt update -y && sudo apt upgrade -y

sudo apt update –y

sudo mandb

mkdir –help (mkdir’in ne olduğunu gösterir)

echo "bugun carsamba" > file3 (file3 diye dosya oluştur, içine “bugun carsamba” yaz)

echo "hello today" >> file5 (mevcut file5 dosyasının içine “hello today” yazısını alta ekle)

date – şu anki tarih ve saati gösterir

rm -i file4 = file4’ü silmeden önce bilgi ver

rm –f file4 = file4’ü silmeden önce bilgi verme

rmdir dir1 = içi boş klasörü siler

rm –r dir1 = klasörü siler

rm –rf dir1 = klasörü zorla (forcefully) siler

echo "su an saat `date`" > date.txt = date.txt oluştur ve tarih ve zamanı içine yaz

mv ./folder1/file1 ./folder2/file3 - taşı ve ismini değiştir

mv folder1/file2 folder1/file8 - dosyanın ismini değiştirmek

cp ./folder1/file1 ./folder2/file5 - kopyala

cp sadece file ları kopyalıyormuş eğer folderları copyalamak istersek cp -r ile başlamak gerekiyormuş bu şekilde dolu boş fark etmiyor dosyayı komple taşıyor

nano siir.txt : nano’da siir.txt oluşurup nano’da açar

head -20 siir.txt : ilk 20 satırı getirir

tail -20 siir.txt : son 20 satırı getirir.

more siir.txt

less siir.txt , q ile çıkılır

touch file{1..100} = 1den 100e kadar dosya oluşturur (file1, file2, file3 …)

touch file{1..100}.txt = 1den 100e kadar .txt uzantılı dosya oluştur(file1.txt, file2.txt ..)

touch file10.txt{,.eski} = komutunu ile yeni bir dosya oluşturur = file10.txt.eski

mkdir folder{1..100} = 1den 100e kadar klasör oluşturur (folder1, folder2, folder3 ..)

# echo {1..100..5} #1den 100e kadar 5 artırarak yazdırır

# # 1 6 11 16 21 26 31 36 41 46 51 56 61 66 71 76 81 86 91 96

# echo {5..100..5} #5den 100e kadar 5 artırarak yazdırır

# # 5 10 15 20 25 30 35 40 45 50 55 60 65 70 75 80 85 90 95 100

# echo {01..10} #01den 10a kadar yazdırır

# # 01 02 03 04 05 06 07 08 09 10

echo {a,b,c}{1..5}

#a1 a2 a3 a4 a5 b1 b2 b3 b4 b5 c1 c2 c3 c4 c5

passwd : kendimiz için

passwd username = başka kullanıcının password değiştirmek (sadece root değiştirmeye yetkilidir.)

passwd –l username (to lock a user account)

passwd –u username (to unlock a user account)

**wildcard**

**? tek bir karakter**

rm file? = file olup yanında 1 karakter olan dosyaları sil

rm file?? = file olup yanında 2 karakter olan dosyaları sil

rm file??? = file olup yanında 3 karakter olan dosyaları sil

rm –rf folder? = klasör olup, yanında 1 karakter olan klasörleri sil

rm –rf folder?? = klasör olup, yanında 2 karakter olan klasörleri sil

rm –rf folder??? = klasör olup, yanında 3 karakter olan klasörleri sil

ls ? = tek karakterli dosyaları listele

ls ?? = iki karakterli dosyaları listele

ls ??? = üç karakterli dosyaları listele

ls ???? = dört karakterli dosyaları listele

ls a?2 = tek karakterli a ile başlayıp 2 ile biten dosyaları listele

ls –F (File ları göster)

ls –lF (File ları listeler)

ls - - color (colorize the output)

ls –r (reverse the order)

ls –R (list file recursively-tekrarlı)

ls –t (sort by time, most recent first)

ls –d (list directory names, not contents)

ls –latr (hepsi karışık)

ls –lt | head (list up to ten of the most recently modified files in the current directory)

ls -l -S ./Fruits (-S option sorts the files in descending order of file size.)ls -l -S -r /Fruits (the -r option the sort order is reversed.)

ls [a-z]\* = a ile z arasındaki herşeyi listele

ls [0-9]\* = 0 ile 9 arasında başlayan herşeyi listele

ls [a, z]\* = a veya z ile başlayan herşeyi listele

ls [a-z][ab]\* = a ile z arasındakilerle başlayıp ikinci harfi a veya b, sonrası önemli değil olan herşeyi listele

ls [abc] = a veya b veya c ile başlayanları listele

ls [abc]?? = a veya b veya c ile başlayan ve 2 karakterli dosyaları listele

ls \*? = tek karakterli bütün herşeyi listele

ls \*\? = herhangi bir şey ile başlayıp sonu ? işareti ile biten herşeyi listele

ls -r [a,c,z]????

ls -r [a,c,z]?????

**List Files Without Using `ls`**

# display the files and directories that are in the current directory

**printf "%s\n" \***

# display only the directories in the current directory

**printf "%s\n" \*/**

# display only (some) image files

**printf "%s\n" \*.{gif,jpg,png}**

touch {ocak,subat,mart}-{2020..2025} ocak 2020, ocak 2021, ocak 2022 vb….

rm ocak\*.txt = ocak ile başlayan bütün txt dosyaları sil

rm ocak\* = ocak ile başlayan bütün dosyaları sil

**\* yüzlerce karakter, bütün karakterler**

ls \*mp3 = mp3 ile biten tüm dosyaları getir

ls a\* = a ile başlayan tüm dosyaları getir

ls [[:alpha:]][[:digit:]]\* = alfabeden herhangi bir harfle başlayıp, ikinci karakteri sayı olan, devamı bütün karakter olan dosyaları listele

ls [[:upper:]][[:digit:]]\* = alfabeden büyük bir harfle başlayıp, ikinci karakteri sayı olan, devamı bütün karakter olan dosyaları listele

ls [[:lower:]][[:digit:]]\* = alfabeden küçük bir harfle başlayıp, ikinci karakteri sayı olan, devamı bütün karakter olan dosyaları listele

ls [[:alnum:]]\* = alnum =alphanumeric, harf yada rakam ile başlayan, devamı bütün karakter olan dosyaları listele

ls \*[[:space:]]\* = başlangıcı ve bitişi herhangi bir şey olup, içinde boşluk bulunan dosyaları listele

USING DİRECTORİES LİSTING FILES (21.11.2022)

sudo apt update -y && sudo apt install tree –y (tree komutunu yükleme, diğerlerini update et)

etc/ = sistem konfigürasyon dosyaları buradadır.

chmod u=rw,g=rw,o=rw file1

chmod ugo=rw file1

chmod a=rwx file1 (ugo yerine a (all) diyebiliriz.)

chmod a +rwx myfile

chmod a=r file5

chmod u+w,o-r,g-r file2

chmod +x exercise1.sh

chmod go=+r myfile

chmod go -r myfile

read=4

write=2

execute=1

754 = (rwx)(r-x)(r--)

chmod 754 file1

chmod 775 file1

chmod 541 file1

chmod 761 file2

chmod 070 html.txt

chmod 444 l.txt

chmod 700 cas.txt html.txt java.txt (birden fazla dosyaya aynı anda yetki verilebiliyor)

chmod –-reference=cas.txt js.js (cas.txt’yi referans al, cas.txt izin gruplarını js.js’e uygula.)

ping -c 3 google.com, Google 3 kere ping attı. –c count

ping Google.com (sonsuza kadar ping atar, durdurmak için ctrl + c yaparız.)

ssh = secure Shell

İt is a protocol used to securely connect to …….

wget command : uzaktan dosyaları indirmek için kullanılır.

wget -O latest-hugo.zip https://github.com/gohugoio/hugo/archive/master.zip

curl command = kaynak kodlarını ekranda görmemizi sağlar

curl Google.com

**user1:x:1004:006::/home/user1:/bin/bash (ll /etc/passwd)**

Field 1 = username

Field 2 = password (stored in /etc/shadow)

Field 3 = UID (User ID) = 0 for root

Field 4 = GID (Group ID) = 0

Field 5 = Comments = comment or empty

Field 6 = Home Directory = /root, /home/user

Field 7 = Default Shell = /bin/bash and /sbin/nologin

**user1:x:1004: (grep user /etc/group)(cat /etc/group)**

Field 1 = group name

Field 2 = password placeholder

Field 3 = GID (Group identifier or group ID)

Field 4 = Blank and jdoe = group members

**man** ls

man 1 ls : ls manuel sayfasının 1nci sayfasını göster

man –k keyword = aradığımız komutun tam adını bilmiyorsak

**info** ls

info <name>

help <name>

**–help** : gives a short explanation about how to use the command and a list of available options

ls - -help

command –h = help olarak açıklar

**whatis** : gives e brief information about the command

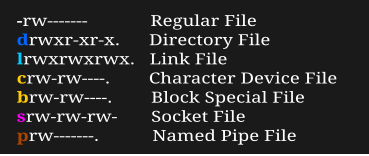
whatis ls

**apropos** : apropos searches the manual pages for a keyword or regular expression.

apropos passwd = passwd nerede geçiyorsa onu bulur

man -k passwd = passwd nerede geçiyorsa onu bulur

The file types:



Color\_File Type

blue \_directories

red\_compressed archives

white\_text files

pink\_ images

cyan \_links

yellow\_devices

green\_executables

flashing red\_broken links

**head** : display the first ten lines of a file.

head /etc/passwd

head -4 /etc/passwd : sadece 4 satırı göster

**tail** : display the last ten lines of a file.

tail /etc/passwd

tail -3 /etc/passwd : sadece son 3 satırı göster

**cat** is short for concatenate. One of the basic uses of cat is to concatenate files into a bigger (or complete) file.

cat part1 part2 part3 > all

You can use cat to create flat text files. Type the cat > winter.txt command as shown in the screenshot below. You will see that cat can be used to copy files.

* create files

cat > winter.txt

It is very cold today!

Today is rainy (çıkış için = ctrl +d)

* copy files

cat winter.txt > cold.txt

cat -n file.txt (print the contents with line numbers)

cat <<END >file

Hello, World.

END

**Using cat**

Option Details

-n Print line numbers

-v Show non-printing characters using ^ and M- notation except LFD and TAB

-T Show TAB characters as ^I

-E Show linefeed(LF) characters as $

-e Same as -vE

-b Number nonempty output lines, overrides -n

-A equivalent to -vET

-s suppress repeated empty output lines, s refers to squeezecat --number file (print line numbers before each line. Alternatively, -n does the same thing)

**tac** count : count’u geriden göster

**more** myFile

**less** myFile (çıkış için = q)(more a göre less daha güçlüdür)

**FIND** is an command for searching file(s) and folder(s) using filters such as size , access time , modification time.

find . : lists all files found in the current directory.

find /home -name fil\\* : if we want to find all the files that start with “fil” in the /home directory, we type the following command. Use the -name argument to find files that fit a specific pattern.

find /home -name clarusway.txt

**find –name** (name isimli dosya arar)

find /sbin –name makedev

**find –iname** (case sensitive olmadan name isimli dosya arar)

find /sbin –iname makedev

**find –ls** (aranılan dosyaları listeler)

find . –name s\* -ls

**find -mtime days** (find files that are days old)

find . –mtime +10 –mtime -13

**find -size num** (find file that are of size num)

find . –size +1M

find . –size +1G

**find –newer file** (find files that are newer than file)

find . –type d –newer file.txt

**find . -exec file {} \;** (run command against all the files that are found)

The find command is case sensitive. If you want to search for a word to be case insensitive, use the -iname option with the find command.

find . -iname fil\\*

find /home -type d -name linux-lessons = tipi d(irectory) olup ismi linux-lessons olanları ara

find /home -type f -name linux-lessons = tipi f(ile) olup ismi linux-lessons olanları ara

**locate** komutu, find komutundan daha hızlıdır.

Locate sales

The **grep**, which stands for "global regular expression print," is used to search text. It searches the given file for lines containing a match to the given strings or words.

GREP :(Globally search a Regular Expression and Print).

grep 'word' filename : This means search any line that contains the word in filename on Linux.

The Major difference is FIND is for searching files and directories using filters while GREP is for searching a pattern inside a file or searching process(es)

find dosya buluyor, grep dosyanin icinde geziniyor, kelime, metin arıyor

grep “Start” quotes.txt : quotes.txt içinde Start ile başlayanları getir

grep –v “Start” quotes.txt : quotes.txt içinde Start ile başlamayanları getir

cat words | grep kar | grep –v ^kar | grep –v kar$ (^kar : kar ile başlayanlar, kar$ : sonu kar ile bitenler)

| **Command** | **Description** |
| --- | --- |
| grep -i | Returns the results for case insensitive strings |
| grep -n | Returns the matching strings along with their line number |
| grep -v | Returns the result of lines not matching the search string |
| grep –c  grep –ni | Returns the number of lines in which the results matched the search string |

grep –Eo = sonucu birebir örtüşenleri göster

grep –g 0 sonuc hata verirse, devam et, hata çıkarma

grep –ci user secret.txt

grep –ni user secret.txt

grep –i user secret.txt

**grep user /etc/passwd | cut -d: -f1,6 | sort | tr ":" " " | column –t**

Go into "Futsal\_Women" directory again from inside "Swimming\_Men" directory and list the content of "Futsal\_Women".

cd ../../Futsal/Futsal\_Women ; ls –al

Olivia plays in two teams Futsall and Basketball. Append "Basketball\_Power-Forward" line into Olivia'sinformation file inside "Futsal\_Women" without overwrite using echo command and absolute path.

echo Basketball\_Power-Forward >> ~/ABC\_College/Futsal/Futsal\_Women/Olivia

cp ~/ABC\_College/Futsal/Futsal\_Women/Olivia ./ = Olivia’yı buraya kopyala

**hostname** = Display hostname of the system.

**hostname -f** = Displays Fully Qualified Domain Name (FQDN) of the system.

**whoami** = Username of the users logged in at the terminal

**who** = sistemi kimin ne zaman kullandığını gösterir

**w** = sisteme kimlerin ne zaman erişip ne işlem yaptığını gösterir.

**last** = Who recently used the system.

**last root** = When was the last time root logged in as user.

**lastb** = Shows all bad login attempts into the system.

**/etc/passwd** = sistemdeki kullanıcıların bilgilerini gösterir.

**useradd[username]** : kullanıcı ekleme (1nci yöntem)

-m home directory oluşsun

-d home directory adresi belirttiğim gibi olsun

-c description, ismini tanımlarız

sudo useradd aydin –m –d /home/aydin\_home –c “Aydin – Devops Engineer”

tail -1 /etc/passwd = girip girmediğini kontrol ederiz.

**cat /etc/default/useradd** (useradd’in uygulama standardını gösterir.)

**adduser[useradd]** : kullanıcı ekleme (2nci yöntem)

sudo su adduser Ahmet

**userdel** Ahmet : kullanıcı siler

userdel –r Ahmet : force ile kullanıcı siler

ls /home = home’daki kullanıcıları listeler

**usermod –[option][value][username]** = kullanıcı bilgilerini modifiye eder

sudo usermod -c "aws solution architect" Ahmet

bir kişiye grup eklemek için = **usermod –g sap user2**

bir kişiyi birden fazla gruba (multiple groups) eklemek için **usermod –G sales user2**

**tail -5 /etc/passwd** (bu komutla kontrol ederiz, düzeltmenin yapılıp yapılmadığını)

**passwd [username]** = password update edilir.

passwd aydin

passwd –l usernam = (to lock a user account)

passwd –u username = (to unlock a user account)

passwd –e = (set password expiry immediately)

chage –l username = (check users current account aging and settings)

passwd –n 7 –x 28 –w 5 username = (configure password aging for username)

-n = mindays = 7 (minimum number of days between password change)

-x = maxdays = 28 (maximum number of days between password change)

-w = warndays = 5 (number of days of warning before password expires)

NEWUSR=$1

useradd -m -U $NEWUSR

PASSWD=$(date | md5sum | cut -c1-8)

echo $PASSWD | passwd $NEWUSR

man passwd (man pages of passwd to view other options)

sudo tail -5 /etc/shadow : kullanıcı şifreleri kriptolu olarak burada muhafaza edilir

grep ^PASS /etc/login.defs = password’ün geçerlik süresini, ne kadar süre önceden değiştirilmemesi gerektiği ve expire date’den ne kadar önce kullanıcıya bilgi verileceğini bildirir.

sudo –l = (sudo erişim izinlerini gösterir)

sudo visudo = (sudo erişim izinlerini değiştirmek için vim komutu)

sudo su devops -y

$ exit

**su user** (root kullanıcıysak başka kullanıcıdan geçersek şifre gerekmez)

**su – user** (root kullanıcıysak başka kullanıcıdan geçersek şifre gerekmez)

**su devops** = devops kullanıcısına geç (başka kullanıcıdan geçersek şifreyi bilmemiz lazım)

**su - devops** = devops kullanıcısına geç (başka kullanıcıdan geçersek şifreyi bilmemiz lazım)

Password:

$ whoami

devops

$ exit

groups = grupları listeler

groups aydin = aydin’in hangi grupta olduğunu gösterir.

sudo groups Jason = Display groups that Jason belongs to

tail -5 /etc/group = Display groups

**sudo groupadd Linux**

**groupadd –g 5000 linuxadm (**5000 grup id’li linuxadm grubu oluşturmak)

**groupmod –g 6000 linuxadm (**grup id’sini 6000 olarak modifiye etmek)

sudo tail /etc/group = grupları listeler

**usermod –a –G devops mustafa** = devops grubuna mustafayı ekliyor (-a : append, -G:Group)

**groupmod –n [newname][oldname] :** grubun ismini değiştirebiliriz

sudo groupmod -n aws-cloud aws

**groupdel[groupname]**

**gpasswd –[option][username][groupname]**

**gpasswd –d mustafa devops** = mustafayı devops gruptan sileriz (-d = delete)

**gpasswd –a mustafa devops** = mustafayı devops grupa ekleriz (-a = append)

sudo gpasswd -a devops aws

tail /etc/group

sudo amazon-linux-extras list

sudo amazon-linux-extras enable nginx1

sudo chmod -R 777 /usr/share/nginx/html = -R altındaki her şey demek

 EXERCISE:

​

  - Create a file named `countries.csv`.

​

```bash

cat << EOF > countries.csv

Country,Capital,Continent

USA,Washington,North America

France,Paris,Europe

Canada,Ottawa,North America

Germany,Berlin,Europe

EOF

```

  - Cut only 'Continent' column | Remove header | Sort the output | List distinct values | Save it to 'continent.txt' and display on the screen.

  cat countries.csv | cut -d',' -f3 | tail -4 | sort | uniq

**DEVAMI ====🡺**

sudo nano /etc/sudoers = group’lara veya user’lara root yetkileri verme

sudo chown ersin:idareciler ersin/

sudo chown suleyman:suleyman Disksdb1

chown owner1 filename

chgrp grp\_owner filename

chgrp -R grp\_owner dir-name

!60 = to re-execute a command (uzun bir kodu tekrar yazmak yerine, kısaca çalıştırır)

!! = to re-execute a command (uzun bir kodu tekrar yazmak yerine, kısaca çalıştırır)

sudo authconfig –passminlen=10 –update (şifreler minimum 10 karakter olacak şekilde update ediyoruz.)

pwscore = gireceğiniz şifrenin ne kadar güvenli olduğunu söyler 1-100 arası

**Environment Variables**

HOSTNAME = stores hostname (echo $HOSTNAME)

HISTFILE = define files for storing history (echo $HISTFILE)

HOME = sets the home directory name (echo $HOME)

LOGNAME = stores the login name (echo $LOGNAME)

PWD = stores the current directory location (echo $PWD)

SHELL = holds the absolute path to primary shell file (echo $SHELL)

VRI=”This is a test”

echo $VRI (This is a test)

unset VRI = atamayı iptal ederiz.

echo $VRI ( )

**connecting from one server to another :**

with ip : ssh [user@191.1.1.1](mailto:user@191.1.1.1)

with hostname : ssh user@hostname

ssh centos

*if logging in with the same user name as the current user, user@ is not required:* ssh hostname

**Transferring and copying files over the network**

**- Transferring files with SCP :**

*Scp : secury copy and is part of SSH süite. It allows you to transfer a file securly from one server to another*

scp file1 root@hostname:/tmp

*(to copy file1 from current location to another server under the directory /tmp)*

scp file1 centos:/tmp

scp test username@host:~

ssh username@host (login to host as username with secure shell)

**- Connecting through** **sftp:**

*SFTP is installed as part of the SSH suite. It allows secure transfer of files as well as secure authentication. Like SSH, SFTP also operates on the same port 22*

sftp centos

sftp> put file1 /tmp (uploading file1 to /tmp/file1)

sftp> ls /tmp

sftp> exit (logout from the centos server)

ftp host (start a file transfer session with host)

**wc command to count lines, words and characters**

wc file1 (/etc/profile)

76 252 1795

(76: number of lines, 252: number of words, 1750:number of characters (or bytes))

wc –l /etc/profile (-l prints line count)(-w prints Word count) (-n prints all the line with numbers)(-c prints bytes count)(-m prints character count)

**netstat** : print network connections, routing tables, interface statistics, masquerade connections, and multicast memberships

**ifconfig** : configure a network interface

**Viewing and monitoring processes**

**ps** (report a snapshot of the current processes)

ps –e (display all processes)

**ps –l**

**ps –ef** (lists all processes)

ps –eH (display a process tree)

**ps -efl**

ps –ef | grep sshd

ps aux (lists all processes in alternative format (BSD))( List all the processes by all users on the current system)

**ps -u root** (List all of the processes and commands root is running)

**ps -U root** (List all of the processes and commands root is running)

ps –u username (display user’s processes)

**ps -G user1/root** (List all of the processes and commands associated with that group)

**ps –eaf** (kimin ne zaman hangi programı kullandığını görebiliriz)

-a=all

-e=every

-f=full-format

-F=Extra full format

-l=long format

Ekran açıklaması:

UID=USER ID (owner of the process)

PID=Process ID number

PPID=Process ID of the parent process

C=Processor utilization for the process

STIME=Process start date or time

TTY=Terminal the process belongs to

TIME=Aggregated execution time fort he process

CMD=Name of command or program being executed.

**pstree** (display processes ina tree format )

**top** : it displays the statistics in real-time and helps to identify performance issues on the system. Sistemde neler olduğunu gösteren, troubleshooting göstergesi ( çıkış : q veya ctrl+c)

Ekran açıklaması:

PID=PROCESS ID

USER=Name of the effective user (owner of the process)

PR=Priority

NI=Nice value (-20 is the highest value, +19 is the lowest value, 0 default value)

VIRT=Virtual memory size

RES=resident memory size

SHR=shared memory size

S=process status (D:uniterruptible sleep, R:running, S:sleeping, T:traced or stopped, Z:zombie)

%CPU=the share of cpu time used by the process since last update

%MEM=share of physical memory used

TIME+= total cpu time used by the task in hundredths of a second

COMMAND=command name or command line

**nice** = Nice value (-20 is the highest value, +19 is the lowest value, 0 default value)

**nice -2 top** (to run the top command at a lower priority with nice value of +2)

**nice --10 top** (to run the top command at a hihger priority with nice value of -10)

**Use htop Instead of top**

htop is a newer, faster version of top command. You should use htop if your system has it. Compare to top , htop has the following advantages:

* You can scroll the process list vertically and horizontally to see all process and complete command line parameters.
* CPU/MEM/SWAP are presented in a better visual way
* It is faster
* Easier way of killing a process, F9 key then select the process
* Easier way of renicing a process

**Schedule and automate tasks (jobs) with Cron**

Cron = cron is a Linux utility which can schedule a command or a script to run automatically at a specified time and date.

Crontab= A cron job is the scheduled task itself. Cron jobs can be very useful to automate repetitive tasks. It is a program that can create, read, update, and delete your job schedules. Cron’un çalışması için önce indirilmesi gerekiyor. İndirilip indirilmediğini **apt list installed | grep cron** ile kontrol ederiz. İndirilmediyse indiririz (sudo apt install cron).

Crontab format:

\* \* \* \* \* command

| | | | +-- Day of the Week (0-6)

| | | +---- Month of the Year (1-12)

| | +------ Day of the Month (1-31)

| +-------- Hour (0-23)

+---------- Minute (0-59)

0 7 \* \* 2 /home/user/sales/backup = run every Tuesday at 07:00

0 7 \* \* 1 /home/user/sales/backup = run every Monday at 07:00

0,15,30,45 \* \* \* \* /pot/acme/bin/15-min-check = run every 15 minutes

0,30 \* \* \* \* /pot/acme/bin/15-min-check = run every 30 minutes

\*/15 \* \* \* \* /pot/acme/bin/15-min-check = run every 15 minutes

0-4 \* \* \* \* /opt/acme/bin/first-five-mins = run for the first 5 minutes of the hour

crontab –e = cron sistem açılır, komutumuzu yazarız, edit your cron jobs

crontab –l = yazdığımız komutu listeler, list your cron jobs

crontab –r = yazdığımız komutu siler, remove all of your cron job

crontab file = install a new crontab from file

crontab -u username –l = list another user's crons

crontab -u username –e = edit another user's crons

**At –f /home/user/script.sh now + 3 hours** (to run a script script.sh from a user’s home directory 3 hours from now)

yum list installed at (check if installed)

yum install at (to install)

**Switching users**

su user

su – user

whoami (display the effective username)

sudo –l (list available commands)

sudo –u root <command> (run command as root)

sudo –u user <command> (run command as user)

sudo su (switch to the superuser account)

sudo su - (switch to the superuser account with root’s environment)

sudo su – username (switch to the username account)

sudo –s (start a shell)

sudo –u root –s (same as sudo -s)

sudo –u user –s (start a shell as user)

visudo (editör, edit the /etc/sudoers file)

mkdir –p 1/2/3 (iç içe alt folderlar oluşturmak)

chmod g+r file1

chmod g-r file1

chmod g+rw, o+x file1

chmod a=r file1

chown user1 file1 (change ownership of file to user1)

chgrp user1 file1 (change group ownership of file to user1)

chown user1:user1 file1 (ownership ve group ownership’i tek komutla değiştirmek)

cp –i file1 newfile1 (kopyalamadan önce sor)

grep –v hello file1 (file1’de hello hariç herşeyi gösterir.)

grep –i “a” file1 (file1’de incase sensitive herşeyi gösterir.)

grep foo ~/Desktop/bar (To find the word foo in the file bar )

grep –v foo ~/Desktop/bar (To find all lines that do not contain foo in the file bar )

grep "\*foo" ~/Desktop/bar (To use find all words containing foo in the end (Wildcard Expansion))

alias (alias olan komutları gösterir)

alias word='command' (create an alias)

alias print\_things='echo "foo" && echo "bar" && echo "baz"' (multiple commands with &&)

alias –p (list all aliases)

alias now='date'

alias cls=’clear’

unalias now

unalias slc

unalias (alias olarak oluşturulan komutu kaldırır)

unalias –a (remove all aliases)

zipping files = gzip, gunzip (unzip file) , bzip2, bunzip2 (unzip file)

gzip example

cat file1 file2 file3 | gzip > combined.gz

cat hello.txt.gz howdy.txt.gz > greetings.txt.gz gunzip greetings.txt.gz

gunzip example.gz (uzantısı .gz)

bzip2 example

bunzip2 example.bz2 (uzantısı .bz2)

archieving files = tar (tape archive)(tar command creates, appends, updates lists and extracts files to and from a single file called tar file.)

options=

-c (creates a tarball)

-v (verbose mode)

-f (specifies a tarball name)

-t (lists contents of a tarball)

-x (extracts from a tarball)

-z (compresses a tarball with gzip command)

tar cvf example.tar example (istediğimiz isimde oluşturmamızı sağlar.)

tar tvf example.tar (içinde ne tar olduğunu gösterir.)

tar xvf example.tar (untar the file)

tar cvzf example.tar.gz example (tar yapıp dosyayı zipler, ismi biz belirleriz.)

tar xvzf example.tar.gz (ziplenmiş tarlı dosyayı, untar yapar, açar)

hostname –I (ip adresimizi gösterir.)

ip addr (tüm kullandığımız mevcut ip adresleri gösterir)

ip a (kullandığımız ip adresleri gösterir)

ip r (kullandığımız route ipleri gösterir)

**id** kullanıcı bilgisini gösterir

**id user2** (user2’ın sahip olduğu (user, group id) bilgileri gösterir.)

**id -u user3** (user3’ün kullanıcı (uid) numarasını verir)

**id -g user3** (user3’ün grup (gid) numarasını verir)

**id -G user3** (user3’ün dahil olduğu tüm grupları (groups) verir)

**getent passwd** (passwd’lerin kayıtlı olduğu veri tabanındaki tüm verileri getirir.)

getent passwd | grep user2

**getent group** (group veri tabanındaki tüm verileri getirir)(**cat /etc/group** ile aynı)

**getent protocols**

**getent hosts**

**getent networks**

grep user2 /etc/passwd (user2 bilgierini görebiliriz)

ping google.com (pingi durdurmak için = ctrl + c)

ping –c 3 127.0.0.1 (belirtilen adrese 3 ping gönderir)

ping –i 5 127.0.0.1 (her pingi 5 saniyede bir gönderir)

ping –s 100 127.0.0.1 (default package size’ı 56 byte(64)’tan 100 byte çıkarır.)

ping –w 3 127.0.0.1 (ping gönderme 3 saniye içinde duracak, bu sürede ne kadar ping gönderebilirse gönderecek)

ping –c 3 –i 0.5 127.0.0.1 (karışık gönderme)

**Start and kill specific processes**

kill –l (display a list of signals)

kill [-sig] pid (send a signal to a process)

kill 123

kill -15 123

kill -9 123

kill –TERM 123

pkill -f test.py

kill $(pgrep -f 'python test.py')

It is always adviced to use the kill command with SIGTERM rather than SIGKILL.

Signal name Signal value Effect

SIGHUP 1 Hangup

SIGINT 2 Interrupt from keyboard

SIGKILL 9 Kill signal (kill -9 pid)

SIGTERM 15 Termination signal

ctrl +c (kill the foreground process)

ctrl + z (suspend the foreground process)

**arkada çalışan programları bulmak ve kill etmek**

./long-running-program & (arkada çalışan programlar)

veya jobs

ps –p 81 (o programın pid’sini bulmak)

ctrl +c (o programı öldürmek)

NOTE: **&>** is known to work as desired in both Bash and Zsh.

echo 'hello' &> /dev/null

**Disk Usage**

du (estimates file usage)

du –k (display sizes in kilobytes)

du –k file1

du –h (display sizes in human readbale format)

**df :** report file system disk space usage

**free :** display amount of free and used memory in the system

**Input-Output Types**

I/O Name Abbreviation File Descriptor

Standart Input stdin 0

Standart Output stdout 1

Standart Error stderr 2

**Redirection**

**>** Redirects standard output to a file. Overwrites (truncating) existing contents.

**>>** Redirects standard output to a file. Appends to any existing contents.

**<** Redirects input from a file to a command.

& - used with redirection to signal that a file descriptor is being used

2>&1 - Combine stderr and standard output

2>file - Redirect standard error to a file.

>/dev/null - Redirect output to nowhere

ls here not-here 2> /dev/null here

ls here not-here > /dev/null 2>&1

grep -r hello /sys/ &> /dev/null

**Comparing files**

diff file1 file2 - compare two files

sdiff file1 file2 - side-by-side comparision

vimdiff file1 file2 - highlight differences in vim

**! Syntax (Shell)**

!N (Repeat command line number N)

!! (Repeat the previous command line)

!string (Repeat the most recent command starting with “string”)

!^ (represents the first argument) !^ = !:1 !:2 !:3

!$ (represents the last argument)

!l (list files)

files.txt sorted-files.txt notes.txt, !^=files.txt !$=notes.txt

$ find /data/app/oracle/oradata/prod/ -type f -size +500M (To search for files exceeding the specifed size, such as 500MB)

$ find /data/ -name ".\*" –print (For hidden files, you can use .\*)

$ ln -s test test2 (#create a symbolic link named test2 to test)

$ whois [www.google.com](http://www.google.com) (#show whois information for google.com)

$ dig [www.google.com](http://www.google.com) (#show DNS information for google.com)

$ host [www.google.com](http://www.google.com) (#show DNS information for google.com)

$ nslookup [www.google.com](http://www.google.com) (#show DNS information for google.com)

$ killall firefox (#kill all firefox processes)

$ ssh-keygen (#generate ssh key pair)

$ ssh-copy-id username@host (#copy ssh public key to host as username)

$ lsof (#list open files)

$ awk '{print $1}' test (#print first column of test)

$ sed 's/Hello/Hi/g' test (#replace Hello with Hi in test)

$ last (#shows list of last logged in users)

$ last username (#shows list of last logged in users)

$ lastlog (#shows list of last logged in users)

$ lastlog –u username (#shows list of last logged in users)

# Generate random password

password=$(date +%s%N | sha256sum | head -c32)

# # Create new user account with generated password

sudo useradd $username -c "$name" -m $username

echo $username:$password | chpasswd

sudo passwd $username

# Force password change on first login

chage -d 0 $username