

Linux for beginners



Cele mai importante lucruri în acest curs

- 1) Punem întrebări oricând despre orice .
- 2) Căutam împreună pe internet fiecare cerință și rezolvarea ei ca în cazul suntem angajați și avem de rezolvat un task .
- 3) Ne vom concentra pe partea practică și vom repeta anumite comenzi chiar dacă poate le știm cu scopul de a le reține mai ușor .
- 4) Scrieți pe un caiet anumite definiții și detalii pentru a reuși să rețineți mai ușor .
- 5) Cutați explicații video și documentație despre orice nelămurire după curs și veti descoperi și alte lucruri despre subiectul respectiv .

Linux - este o familie de sisteme de operare open-source de tip Unix . Cele mai cunoscute distribuții de linux sunt:

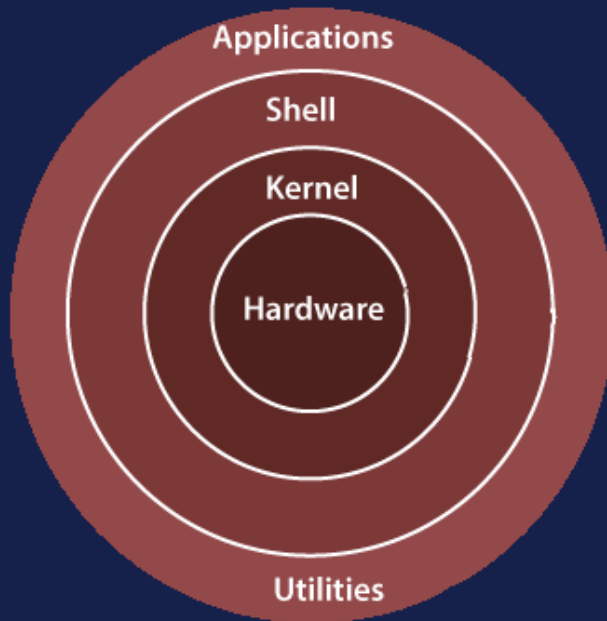
- Ubuntu (sisteme desktop)
- Kali Linux (utilizat pe partea de securitate)
- CentOS/RedHat (pentru configurarea de servere)

Componente principale

- Kernel
- Shell
- Sistemul de fișiere



Kernel-ul este nucleul sistemului de operare. Este un software ce se află în memorie și care îi spune procesorului ce să facă .



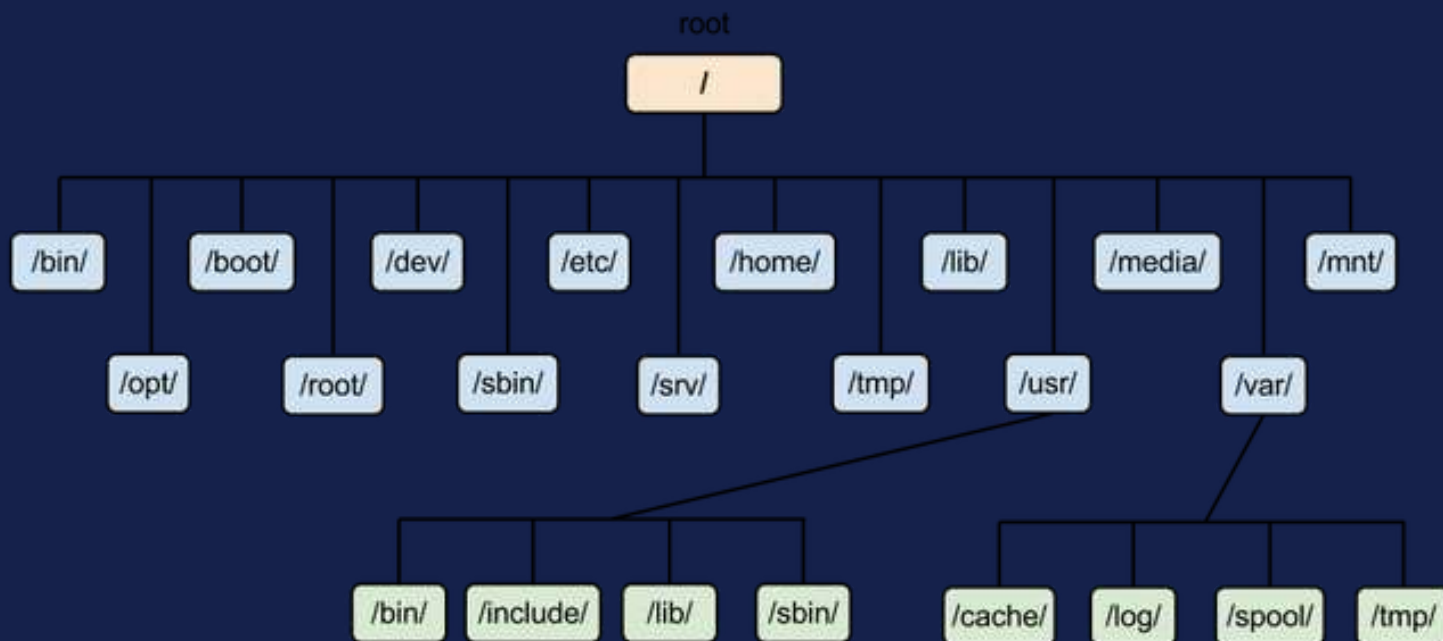
O parte componentă importantă o reprezintă shell-ul . El este interpretorul de comenzi și reprezintă interfața între kernel și utilizator . Shell-ul va interpreta comenzile date de utilizator și apoi trimite acele instrucțiuni către kernel .

```
localhost:~$ bash --version
GNU bash, version 5.0.17(1)-release (i586-alpine-linux-musl)
Copyright (C) 2019 Free Software Foundation, Inc.
License GPLv3+: GNU GPL version 3 or later <http://gnu.org/licenses/gpl.html>

This is free software; you are free to change and redistribute it.
There is NO WARRANTY, to the extent permitted by law.
localhost:~$
```



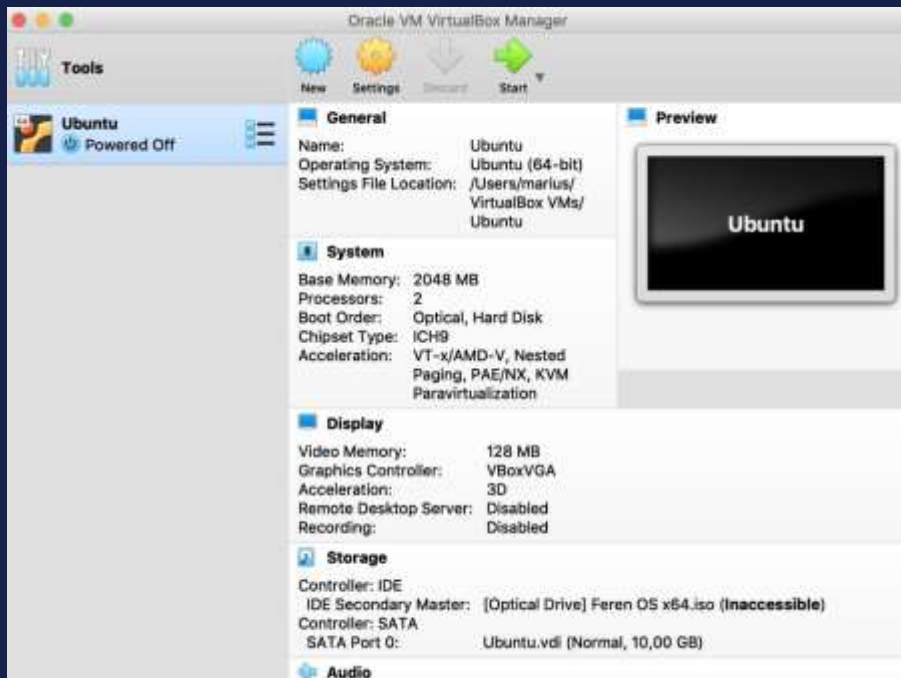
Sistemul de fișiere în linux - Reprezintă o ierarhie de fișiere și directoare. Toate folderele au un rol specific de exemplu: /usr ce conține comenzi executabile sau /home unde sunt directoarele utilizatorilor. Se pot găsi [aici](#) detalii despre toate directoarele din sistemul de fișiere .



Ce este virtualbox ?

Virtual Box este o aplicație care îți propune să vină în ajutorul nostru atunci când avem nevoie rapid și ușor de un server care să îndeplinească anumite specificații, atât hardware, cât și software.

Se poate descărca de aici: [Download](#)



Configurarea serverului de linux

Descărcăm imaginea de linux ubuntu : [Download](#)

Pași pentru [Instalare](#)

A screenshot of a Linux desktop environment. The background is a red and black Ubuntu logo. On the left, there is a sidebar with icons for 'blender', 'visual studio', '18.04 LTS', and 'trash'. A terminal window is open in the center, displaying the following commands and output:

```
ubuntu@ubuntu:~$ uname -a
Linux ubuntu 4.15.0-19-generic #11-ubuntu SMP Tue 4
4 AM; root:x86_64 GNU/Linux
ubuntu@ubuntu:~$ cat /etc/issue
Ubuntu 18.04.1 LTS (development branch) \n \n
ubuntu@ubuntu:~$ cat /etc/lsb-release
DISTRIB_ID=Ubuntu
DISTRIB_RELEASE=18.04
DISTRIB_CODENAME=bionic
DISTRIB_DESCRIPTION="Ubuntu 18.04.1 LTS (development branch)"
ubuntu@ubuntu:~$
```

Structura unei comenzi

Nume_comandă [opțiuni] [arg1] [arg2] ...

Câteva comenzi pentru manipularea fișierelor :

ls -> listează conținutul unui director

pwd -> afișează numele directorului curent în care ne aflăm

cd -> schimbă directorul

touch -> folosit pentru a crea fișiere

mkdir -> folosit pentru a crea directoare

vi , vim, nano -> editoare de text

cat -> afișează conținutul unui fișier

echo -> ne va afișa un text introdus ca argument

exit or logout -> iese din sesiunea curentă



cd .. -> ne va duce mai sus pe arborele de fişiere
cd ~ -> ne duce către directorul principal al unui utilizator
ctrl+l -> va curăţa terminalul de comenzile vechi
history -> putem vedea comenzile vechi listate în ordine
uptime -> spune de cât timp am pornit sistemul
man -> vedem documentaţia pentru fiecare comandă

Exemplu

```
root@Ubuntu:~# ls
file1.txt  file2.txt
root@Ubuntu:~# ls -l
total 0
-rw-r--r-- 1 root root 0 ian 11 12:20 file1.txt
-rw-r--r-- 1 root root 0 ian 11 12:20 file2.txt
root@Ubuntu:~# la -la
total 28
drwx----- 4 root root 4096 ian 11 12:20 .
drwxr-xr-x 24 root root 4096 nov 12 13:37 ..
-rw----- 1 root root 135 nov 12 14:07 .bash_history
-rw-r--r-- 1 root root 0 ian 11 12:20 file1.txt
-rw-r--r-- 1 root root 0 ian 11 12:20 file2.txt
root@Ubuntu:~# ls -lah
total 28K
drwx----- 4 root root 4,0K ian 11 12:20 .
drwxr-xr-x 24 root root 4,0K nov 12 13:37 ..
-rw----- 1 root root 135 nov 12 14:07 .bash_history
-rw-r--r-- 1 root root 0 ian 11 12:20 file1.txt
-rw-r--r-- 1 root root 0 ian 11 12:20 file2.txt
```

Exemplu

```
root@Ubuntu:/bin# uptime
12:22:52 up 11 min,  2 users,  load average: 5,05, 5,15, 3,17
root@Ubuntu:/bin# cd /var/
backups/  crash/    local/    log/      metrics/  run/      spool/
cache/    lib/      lock/     mail/     opt/      snap/     tmp/
root@Ubuntu:/var# pwd
/var
root@Ubuntu:/bin# cd /var/log/
root@Ubuntu:/var/log# pwd
/var/log
root@Ubuntu:/var/log# cd /var/log/apt/
root@Ubuntu:/var/log/apt# pwd
/var/log/apt
```

Calea relativă și absolută

- . (punct) reprezintă directorul curent în care se află utilizatorul .
- .. Reprezintă directorul părinte al directorului curent în care se află utilizatorul .

Calea absolută este atunci când se specifică locația unui fișier sau director din directorul rădăcină (/) .

Calea relativă este calea legată de prezentul director de lucru (pwd). Începe din directorul curent și niciodată cu (/) .

```
demo@Ubuntu: ~  
File Edit View Search Terminal Help  
demo@Ubuntu:~$ cd .  
demo@Ubuntu:~$ pwd  
/home/demo  
demo@Ubuntu:~$ ls  
Desktop Documents Downloads Music Pictures Public Templates Videos  
demo@Ubuntu:~$ mkdir var  
demo@Ubuntu:~$ ls  
Desktop Documents Downloads Music Pictures Public Templates var Videos  
demo@Ubuntu:~$ cp ./Desktop/fisier.txt ./var/  
demo@Ubuntu:~$  
demo@Ubuntu:~$ cp ./Desktop/fisier.txt /var/  
backups/ crash/ local/ log/ metrics/ run/ spool/  
cache/ lib/ lock/ mail/ opt/ snap/ tmp/
```



Alte comenzi folosite pentru manipularea fișierelor

cp -> copiază fișiere/foldere în locația specificată

rm -> șterge fișierul dat ca argument

rmdir nume_director -> șterge directorul dat ca argument (doar dacă directorul este gol)

rm -rf nume_director -> șterge directorul dat ca argument chiar dacă nu este gol

mv -> poate muta/redenumi fișiere și foldere

Comenzi rapide cu CTRL

ctrl+c: întrerupeți (kill) procesul curent din prim-plan care rulează în terminal. Acest lucru trimite semnalul SIGINT către proces .

ctrl-z: suspendă procesul curent .

ctrl+u: șterge linia de comandă .

ctrl+s: întrerupe toate comenzile pe ecran .

ctrl+u: șterge cuvântul de dinaintea cursorului .

ctrl+k: curăță toată linia de comenzi ce urmează după cursor .

ctrl+a: mută cursorul la începutul liniei de comandă pe care scrii .

Exemplu

```
demo@Ubuntu:~/Desktop$ ls
demo@Ubuntu:~/Desktop$ touch file1
demo@Ubuntu:~/Desktop$ cp file1 file2
demo@Ubuntu:~/Desktop$ ls
file1  file2
demo@Ubuntu:~/Desktop$ rm file1
demo@Ubuntu:~/Desktop$ ls
file2
demo@Ubuntu:~/Desktop$ mkdir folder1
demo@Ubuntu:~/Desktop$ ls
file2  folder1
demo@Ubuntu:~/Desktop$ rmdir folder1/
demo@Ubuntu:~/Desktop$ ls
file2
demo@Ubuntu:~/Desktop$ cd ..
demo@Ubuntu:~$ pwd
/home/demo
```

Ce reprezintă fiecare element din construcția:
demo@Ubuntu:~/Desktop ?

Exemplu

```
demo@Ubuntu:~$ cd Desktop/  
demo@Ubuntu:~/Desktop$ ls  
file2  
demo@Ubuntu:~/Desktop$ mkdir folder3  
demo@Ubuntu:~/Desktop$ ls  
file2  folder3  
demo@Ubuntu:~/Desktop$ cp file2 folder3/  
demo@Ubuntu:~/Desktop$ ls  
file2  folder3  
demo@Ubuntu:~/Desktop$ rm file2  
demo@Ubuntu:~/Desktop$ ls  
folder3  
demo@Ubuntu:~/Desktop$ find . -name file2  
./folder3/file2
```

Permisiunile și utilizatorii în Linux

Utilizatorii sunt conturi care pot fi folosite pentru a se autentifica într-un sistem. Fiecare utilizator este identificat printr-un număr unic de identificare sau UID de către sistem . În cadrul laboratoarelor de mai devreme am folosit userul creat de noi cu numele “demo” .



Ce este sudo?

Este un acronim pentru “superuser” și reprezintă o comandă ce îți dă drepturi superioare fără a avea nevoie să îți schimbi userul. În funcție de setările din fișierul **/etc/sudoers** se controlează cine poate rula astfel de comenzi cu drepturi superioare .

De exemplu, nu toți utilizatorii pot rula comenzi cu sudo. În cazul de mai jos utilizatorul demo poate rula comenzi cu sudo.

```
demo@Ubuntu: ~  
File Edit View Search Terminal Help  
Defaults      env_reset  
Defaults      mail_badpass  
Defaults      secure_path="/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/  
sbin:/bin:/snap/bin"  
  
# Host alias specification  
  
# User alias specification  
  
# Cmnd alias specification  
  
# User privilege specification  
root    ALL=(ALL:ALL) ALL  
demo    ALL=(ALL:ALL) ALL  
# Members of the admin group may gain root privileges  
%admin   ALL=(ALL) ALL  
  
# Allow members of group sudo to execute any command  
%sudo    ALL=(ALL:ALL) ALL  
  
# See sudoers(5) for more information on "#include" directives:  
  
#includedir /etc/sudoers.d
```



Comanda su: Aceasta poate fi folosită pentru a deveni alt user (switch user) . Exemplu: su – new_user

```
root@Ubuntu:~# ls
file1.txt  football.txt  snap
root@Ubuntu:~# su demo
demo@Ubuntu:/root$ ls
ls: cannot open directory '.': Permission denied
demo@Ubuntu:/root$ cd ~
demo@Ubuntu:~$ ls
Desktop  Documents  Downloads  Music  Pictures  Public  Templates  var  Videos
demo@Ubuntu:~$
```

Cum creăm un user?

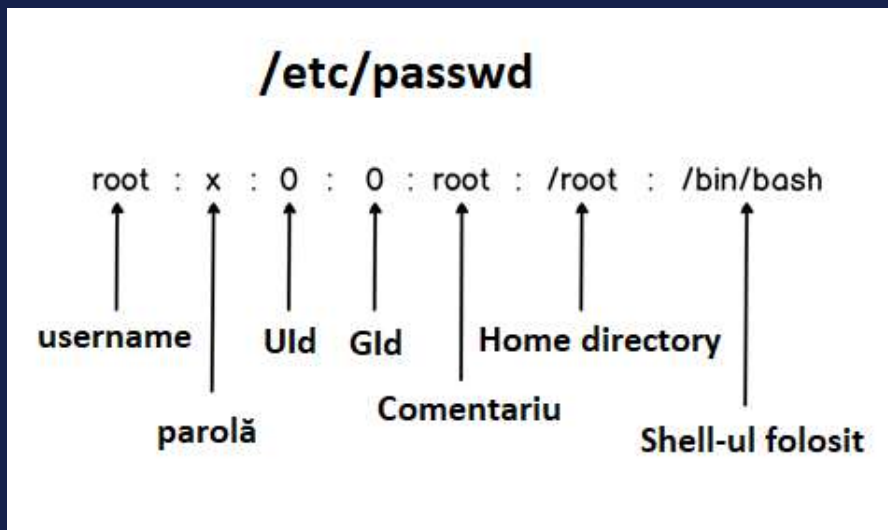
Pentru a crea un nou user: **adduser new_user**

Pentru a vedea noul user: **cat /etc/passwd**

Pentru a șterge userul: **userdel new_user**

Pentru a schimba parola userului: **sudo passwd new_user**

Poți schimba parola altui user? Dar propria parolă ? În primul caz încearcă această comandă cu sudo în față. De exemplu: **sudo passwd new_user**



Ce sunt grupurile în linux?

Un grup în Linux este o colecție de conturi cărora li se pot acorda permisiuni speciale pe sistem.

Să presupunem, de exemplu, că un user 'George' nu poate avea acces la fișierul cu parole passwords.txt . Userul 'Mihai' care este în grupul 'admins' are acces pe fișierul passwords.txt . Pentru a avea acces și userul 'George', este suficient să fie adăugat pe acel grup, astfel va putea citi fișierul de parole.

Ce sunt grupurile în linux?

Cum vedem grupurile curente?

comanda: **cat /etc/group**. Acest fișier conține o listă cu numele grupului, parola, id-ul grupului, lista de membri ai acelui grup .

Cum creăm un grup?

comanda: **sudo groupadd -g id nume_grup**

ex: **sudo groupadd -g 10000 studenti**

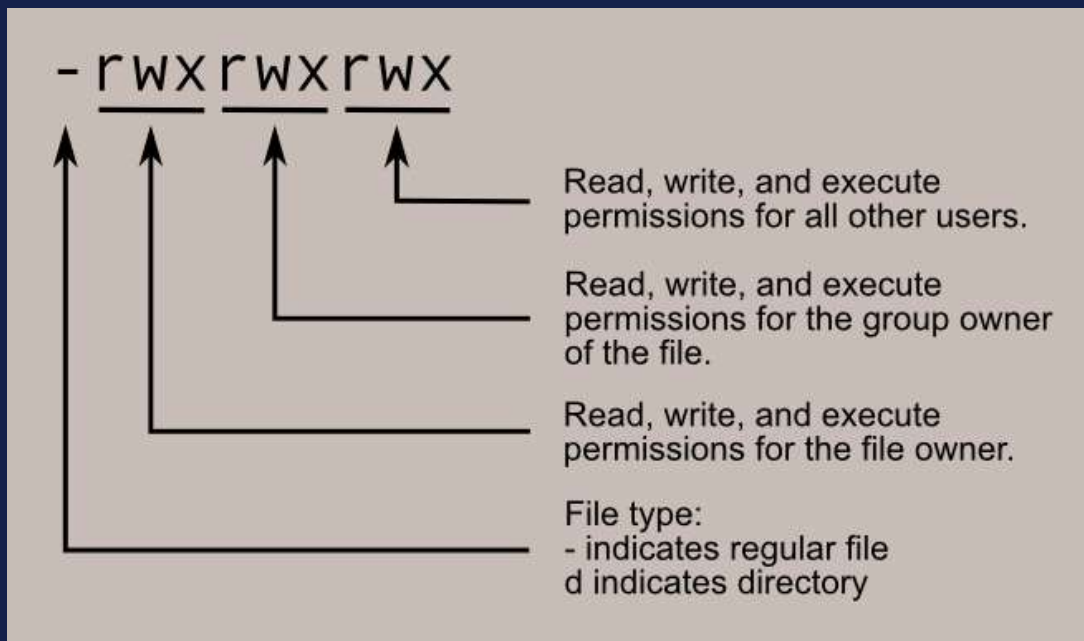
Evitarea folosirii userului de root

"**Root user**" este numele de utilizator sau contul care, în mod implicit, are acces la toate comenzile și fișierele pe un Linux. De asemenea, mai este numit "**superuser**" sau "**root account**". Este de preferat să nu se utilizeze direct acest user, deoarece cu cât avem acces mai restrictiv cu atât limităm eventualele defecțiuni pe care le putem produce din neatenție .

Este de preferat să creăm un user și să lucrăm de pe el, la nevoie îi putem acorda mai multe privilegii .

Chmod și chown

În termen scurt, chown este folosit pentru a schimba calitatea de proprietar al unui fișier în timp ce chmod este pentru schimbarea biților modului de fișier .



Comanda chmod

Primele trei caractere arată permisiunile pentru utilizatorul care deține fișierul (permisiuni de utilizator), cele trei caractere din mijloc sunt pentru membrii grupului fișierului (permisiuni de grup), iar ultimele pentru restul (others)

În cazul în care documentul este un folder, se va afișa un “d” la început .

R - permisiune de citire (read)

W - permisiune de scriere (write)

X - permisiune de execuție (execute)

--- -> fără nicio permisiune

rwX -> permisiuni complete

- (minus) -> scoate permisiunea

+ (plus) -> adaugă permisiunea

```
File Edit View Search Terminal Help
demo@Ubuntu:/tmp/test$ ls -lah
total 8,0K
drwxrwxr-x  2 demo demo 4,0K ian 29 16:58 .
drwxrwxrwt 16 root root 4,0K ian 29 17:00 ..
-rwxrwxr-x  1 demo demo   0 ian 29 16:57 file.txt
-rw-r--r--  1 root root   0 ian 29 16:58 rootfile2.txt
--wxr----x  1 root root   0 ian 29 16:58 rootfile.txt
```

- 0: (000) Fără permisiuni
- 1: (001) Permisiuni de execuție
- 2: (010) Permisiuni de scriere
- 3: (011) Permisiuni de scriere și execuție
- 4: (100) Permisiuni de citire
- 5: (101) Permisiuni de citire și execuție
- 6: (110) Permisiuni de citire și scriere
- 7: (111) Permisiuni de citire, scriere și execuție

Example:

`chmod u=rwx file`

`chmod a+x new_script.sh` (a = all)

`chmod u-x new_script.sh`

`chmod o-rx folder/`

`chmod -R o-rwx test/`

`chmod 553 file.txt`



Comanda chown

chown [OPTIONS] USER[:GROUP] FILE(s)

Comanda chown vă permite să schimbați proprietatea (ownership) utilizatorului și/sau a grupului asupra unui anumit fișier, director sau link simbolic .

Chown poate accepta ca parametrii id-uri sau nume de grup/utilizatori

```
chown mike:admin_group file2.txt
```

```
chown :admin_group file2.txt
```

Poți schimba și pentru un folder:

```
chown mike:admin_group FOLDER1
```

Poți schimba recursiv:

```
chown -R mike:admin_group FOLDER1
```



Comanda pipe '|'

Pipe ('|') este o comandă ce permite să folosiți două sau mai multe comenzi, astfel încât ieșirea unei comenzi să servească drept intrare pentru următoarea .

```
root@Ubuntu:~# cat file1.txt
1111111111111111
2222222222222222
3333333333333333
this is a text more testing grep
This TExt is for case sensitive example
root@Ubuntu:~# cat file1.txt | grep text
this is a text more testing grep
root@Ubuntu:~# cat file1.txt | grep TEXT
root@Ubuntu:~# cat file1.txt | grep TExt
This TExt is for case sensitive example
root@Ubuntu:~# cat file1.txt | grep -i TExt
this is a text more testing grep
This TExt is for case sensitive example
root@Ubuntu:~# cat file1.txt | grep -i '1\|2\|3'
1111111111111111
3333333333333333
root@Ubuntu:~# cat file1.txt | grep -i '1\|2\|3'
1111111111111111
2222222222222222
3333333333333333
root@Ubuntu:~# cat file1.txt | grep -i '1\|2\|3' | sort
1111111111111111
2222222222222222
3333333333333333
root@Ubuntu:~# cat file1.txt | grep -i '1\|2\|3' | sort -r
3333333333333333
2222222222222222
1111111111111111
```

Comanda grep

Comanda grep caută într-un fișier un anumit model de caractere și afișează toate liniile care conțin acel model

```
root@Ubuntu:/var/log# cat auth.log | grep "create session"
Feb  6 19:56:47 Ubuntu su[2813]: pam_systemd(su:session): Cannot create session: Already running in a session
Feb  6 19:56:55 Ubuntu su[2823]: pam_systemd(su:session): Cannot create session: Already running in a session
Feb  6 20:00:40 Ubuntu su[2866]: pam_systemd(su:session): Cannot create session: Already running in a session
root@Ubuntu:/var/log# grep "create session" auth.log
Feb  6 19:56:47 Ubuntu su[2813]: pam_systemd(su:session): Cannot create session: Already running in a session
Feb  6 19:56:55 Ubuntu su[2823]: pam_systemd(su:session): Cannot create session: Already running in a session
Feb  6 20:00:40 Ubuntu su[2866]: pam_systemd(su:session): Cannot create session: Already running in a session
root@Ubuntu:/var/log#
```

grep '111' file1.txt

grep -i 'tHiS' file1.txt

grep -i "root" /etc/passwd

grep -c 'football' file1.txt

grep 'users' -r .



Comanda find

Comanda “find” poate fi folosită pentru a găsi fișiere și directoare. Se pot căuta fișiere atât după nume, data creării, proprietar, permisiuni sau data modificării .

`find /home/demo/ -name secret_file.txt`

`find /home/demo/ -name *.txt`

`find /home/demo/ -type f -name "*" -> doar fișierele`

`find . -mtime +8 -> cu ultima modificare mai veche de 8 zile`

```
root@Ubuntu:/home/demo# find -name secretfile.txt
./Desktop/secretfile.txt
root@Ubuntu:/home/demo# find -name "secret*"
./secret1231.txt
./Desktop/secretfile.txt
```


Comanda locate

Comanda locate caută în sistemul de fișiere fișiere și directoare al căror nume corespunde unui model dat. Sintaxa comenzii este ușor de reținut, iar rezultatele sunt afișate aproape instantaneu. Diferența față de comanda “find” este că această comandă caută într-o bază de date actualizată periodic și la cerere, de aceea căutarea e și mai rapidă.

exemplu: locate nume.txt

Pentru a actualiza baza de date: updatedb

Arhivarea și dezarhivarea fișierelor

Cele mai comune programe pentru arhivarea fișierelor și directoarelor sunt:

.tar si .gz

tar -cf arhiva.tar file1 file2 -> creează o arhivă

tar -tvf arhiva.tar -> Vezi ce conține arhiva

tar -xvf arhiva.tar -> extragem fișierul arhiva.tar

tar -xvf arhiva.tar -C ./New_Folder/ -> extragem fișierul arhiva.tar în "New_Folder"

tar -zcvf arhiva.tar.gz file1 file2

tar -zxvf arhiva.tar.gz

Un fișier TAR este ceea ce ați numi o arhivă, deoarece este doar o colecție de fișiere multiple reunite într-un singur fișier. Și un fișier GZ este un fișier comprimat arhivat folosind algoritmul gzip.



Comenzile shutdown și reboot

shutdown -h 1 -> sistemul se va închide peste 1 minut

shutdown -h 20 -> sistemul se va închide în 20 de minute după comandă

shutdown now -> închide sistemul după executarea comenzii

shutdown -h 17:30 -> se va închide la ora curentă 17.30

reboot now -> restartează sistemul după executarea comenzii

Aceste comenzi sunt executate doar cu privilegii de pe anumiți utilizatori (sudo)

Metacaractere

Exemple:

>	Redirecționare ieșire
>>	Redirecționare ieșire (adaugare)
<	Redirecționare de intrare
*	Inlocuieste zero sau mai multe caractere
?	Inlocuieste un caracter
[]	înlocuire a oricarui caracter între paranteze
	pipe
&&	Comanda "si"
&	Ruleaza comenzi in background
#	Comentare
\$	Extinde valoarea unei variabile

```
echo "redirectează acest text" > output.txt
```

```
echo "redirectează acest text v2" >  
output.txt
```

```
echo "redirectează acest text v3" >>  
output.txt
```

```
wc -l < file.py
```

```
more < /etc/passwd
```

```
head < file.py
```

```
ls f*
```

```
ls folder?
```

```
ls ba* && cat /tmp/output.txt
```

```
#scrie orice aici
```

```
x=4 && echo $x
```



Redirectarea output-ului unei comenzi

Ce este output-ul ? Când rulați o comandă, aceasta produce un rezultat la ieșire. Acest rezultat este fie răspunsul pe care îl așteptăm, fie este un status sau cod de eroare dacă nu am executat comanda corect .

Uneori, poate doriți să stocați rezultatul unei comenzi într-un fișier sau într-o variabilă pentru a fi utilizată într-o operațiune ulterioară .

```
demo@Ubuntu: ~/Documents
File Edit View Search Terminal Help
demo@Ubuntu:~/Documents$ cat config.txt
Lista IP-uri
2.2.2.2/32
demo@Ubuntu:~/Documents$ cat config2.txt
cat: config2.txt: No such file or directory
demo@Ubuntu:~/Documents$ echo "3.3.3.3/32" > config.txt
demo@Ubuntu:~/Documents$ cat config.txt
3.3.3.3/32
demo@Ubuntu:~/Documents$ echp "4.4.4.4/32" >> config.txt

Command 'echp' not found, did you mean:

  command 'ecp' from deb ecere-dev
  command 'echo' from deb coreutils

Try: apt install <deb name>

demo@Ubuntu:~/Documents$ echo "4.4.4.4/32" >> config.txt
demo@Ubuntu:~/Documents$ cat config.txt
3.3.3.3/32
4.4.4.4/32
```

stdin, stdout și stderr

stdin, stdout și stderr sunt trei fluxuri de date create atunci când lansați o comandă Linux .

stdin -> comanda introdusă ca input (0)

stdout -> textul de se afiseaza la ieșirea atunci când dăm o comandă (1)

stderr -> eroarea ce ne apare când introducem o comandă gresită (2)

Pentru stdin putem încerca comanda: read

Creați un fișier **script.sh** și copiați urmatorul text:

```
#!/bin/bash
```

```
echo "fișierul nu există dar voi încerca să îl afișez"
```

```
cat fisier.txt
```

Executați script-ul cu comanda: **bash script.sh**

Verificați și următoarele comenzi:

```
bash script.sh 1> corect.log
```

```
bash script.sh 2> error.log
```



Fișierul dev/null

Redirectarea catre /dev/null a unei comenzi va elimina orice este scris pe acesta și va returna EOF(end of file) la citire.

```
echo 'Hello world'
```

```
echo 'Hello world' > /dev/null ( stdout )
```

```
cat xqrrqe > /dev/null -> ( stderr )
```

```
bash script.sh > /dev/null 2>/dev/null (Discard error messages)
```

Pentru a nu scrie de fiecare dată `bash script.sh`, putem sa îi dăm fișierului `script.sh` drepturi de execute astfel: `chmod +x script.sh`

Acum îl vom putea rula direct cu comanda: `./script.sh`

Putem analiza fișierul `script.sh` cu comanda de listare să vedem ce este diferit acum. Ce se poate observa ?



Procesele în linux

ps – cu aceasta comanda putem vedea procesele ce rulează în timp real pe sistemul nostru

Cele mai folosite **ps -ef** sau **ps -aux**

a = arată procesele pentru toți utilizatorii

u = afișați utilizatorul/proprietarul procesului

x = arată și procesele care nu sunt atașate la un terminal

Pentru a căuta un anumit proces:

ps aux | grep firefox

Procesele în linux

Pentru a închide un proces:

Kill -9 process_id -> putem inchide procesul instant (trimiți un semnal de SIGKILL)

Cand ne gandim la ps din linux, ne putem gândi la echivalența lui cu Ctrl+alt+del din windows si lista de procese de acolo.

Soft links și hard links

Softlinks:

`ln -s source_file symbolic_link`

ex: `ln -s Desktop/fisier_important.txt Downloads/`

`ln -s Desktop/folder1 ~/linked_folder`

`unlink fisierul_linkuit`

Putem compara acest proces cu shortcut din windows. Ce se întâmplă dacă ștergem sursa sau fișierul/folderului ?

Hardlinks:

`ln fisier_sursa fisier_destinatie_hardlinked`

Care e diferența între ele ? Dacă le ștergem sursa se va șterge și destinația ?
Hardlink-urile au același inode number (`ls -li`)



Comanda file

file — comanda file este utilizată pentru a determina tipul unui fișier.

```
file document.txt
```

```
file *
```

```
file dir/*
```

```
file -F ":" file.py
```

Comenzile more, head și tail

Comanda more este utilizată pentru a vizualiza fișierele text în promptul de comandă, afișând câte un ecran în cazul în care fișierul este mare (de exemplu fișiere mari de log-uri).

more file.txt

cat file.txt | more

tail -> comanda tail se folosește pentru a afișa ultimele linii ale unui fișier

tail -3 -> afișează ultimele 3 rânduri

tail -f afișează în timp real

head -> este complementarul comenzii tail. În mod implicit, imprimă primele 10 rânduri ale fișierelor specificate.

head -3 fisier.txt -> afișează primele 3 rânduri



Comanda word counter

După cum sugerează și numele, este folosit în principal în scop de numărare. Este folosit pentru a afla numărul de linii, numărul de cuvinte, numărul de octeți și caracterele din fișierele specificate în argumentele fișierului .

`wc fisier.txt` -> prima coloană arată numărul de linii prezente, a doua coloană arată numărul de cuvinte prezente în fișier, a treia coloană arată numărul de caractere prezente în fișier și a patra coloană este numele fișierului care este dat ca argument .

Comanda word counter

w - numărul de cuvinte

m - numărul de caractere

l - numărul de linii

wc -w file.py -> afisează numărul de cuvinte

wc -m file.py -> afisează numărul de caractere

wc -l file.py -> afisează numărul de linii

```
demo@Ubuntu:/tmp$ cat fisier.txt
1 abc xyz
2 curs de linux
3
4
5
demo@Ubuntu:/tmp$ wc fisier.txt
 5 10 32 fisier.txt
```

Instalarea pachetelor in linux

Sistemul de gestionare a pachetelor Ubuntu este derivat din același sistem folosit de distribuția Debian GNU/Linux.

Apt - comanda apt este un instrument din linia de comandă, care funcționează cu Advanced Packaging Tool (APT) al Ubuntu, efectuând funcții precum instalarea de noi pachete software, actualizarea pachetelor software existente, actualizarea indexului listei de pachete și chiar actualizarea întregului Ubuntu.

Instalarea unui pachet: **sudo apt install vim**

Dezinstalarea unui pachet: **sudo apt remove vim**

Upgrade de pachete: **sudo apt upgrade**



Comanda `sudo apt-get update` : Indexul pachetelor APT este în esență o bază de date a pachetelor disponibile din depozitele definite în `/etc/apt/sources.list` și `/etc/apt/sources.list.d`

```
demo@demo-VirtualBox:~$ sudo apt update -y
[sudo] password for demo:
Hit:1 http://ro.archive.ubuntu.com/ubuntu bionic InRelease
Get:2 http://ro.archive.ubuntu.com/ubuntu bionic-updates InRelease [88,7 kB]
Get:3 http://security.ubuntu.com/ubuntu bionic-security InRelease [88,7 kB]
Get:4 http://ro.archive.ubuntu.com/ubuntu bionic-backports InRelease [74,6 kB]
Get:5 http://ro.archive.ubuntu.com/ubuntu bionic-updates/main amd64 DEP-11 Meta
data [297 kB]
Get:6 http://ro.archive.ubuntu.com/ubuntu bionic-updates/universe amd64 DEP-11
Metadata [301 kB]
Get:7 http://ro.archive.ubuntu.com/ubuntu bionic-updates/multiverse amd64 DEP-1
1 Metadata [2.468 B]
Get:8 http://ro.archive.ubuntu.com/ubuntu bionic-backports/universe amd64 DEP-1
1 Metadata [9.276 B]
Fetched 861 kB in 2s (376 kB/s)
```


Comenzile du și df

du: este un program standard Unix folosit pentru a estima utilizarea spațiului de fișiere. De obicei se folosește împreună cu **-h** pentru a afișa în format human readable (Mb/Gb) și nu în Kb. Care e diferența între ele ?

```
demo@Ubuntu:~/Desktop$ du -h
4,0K      ./folder1
8,0K      ./config
4,0K      ./folder2
24K       .
demo@Ubuntu:~/Desktop$ du
4          ./folder1
8          ./config
4          ./folder2
24         .
demo@Ubuntu:~/Desktop$
```

df este o comandă standard Unix folosită pentru a afișa cantitatea de spațiu disponibil pe disc pentru sistemele de fișiere pe care utilizatorul care invocă are acces adecvat de citire

```
demo@Ubuntu:~/Desktop$ df -h
Filesystem      Size  Used Avail Use% Mounted on
udev            630M     0  630M   0% /dev
tmpfs           131M   1,3M  130M   1% /run
/dev/sda1       20G   6,9G   12G  37% /
tmpfs           653M     0  653M   0% /dev/shm
tmpfs           5,0M   4,0K   5,0M   1% /run/lock
tmpfs           653M     0  653M   0% /sys/fs/cgroup
demo@Ubuntu:~/Desktop$ df
Filesystem      1K-blocks      Used Available Use% Mounted on
udev            644168         0     644168   0% /dev
tmpfs           133616       1308     132308   1% /run
/dev/sda1       20509264  7145848  12298560  37% /
tmpfs           668064         0     668064   0% /dev/shm
tmpfs           5120          4         5116   1% /run/lock
tmpfs           668064         0     668064   0% /sys/fs/cgroup
```

Comanda sed

Comanda SED în UNIX înseamnă editor de flux și poate îndeplini o mulțime de funcții pe fișier, cum ar fi căutarea, găsirea și înlocuirea, inserarea sau ștergerea. Totuși, cea mai comună utilizare a comenzii SED în UNIX este pentru înlocuire sau pentru găsire și înlocuire .

sed 's/george/mihai/' fisier.txt

sed -i 's/george/mihai/' fisier.txt dacă vrem să și înlocuim cuvântul

Conexiunea SSH

SSH sau Secure Shell este un protocol de comunicare în rețea care permite a două computere să comunice .

Pentru asta vom folosi comanda putty. SSH folosește portul 22 pentru asta.

Calculatorul A se poate conecta la Calculatorul B prin protocolului SSH. Astfel, calculatorul A poate face asta scriind în linia de comandă:
ssh userB@adresa_ipB și se introduce parola utilizatorului userB

Uneori firewall-ul instalat pe serverul B poate bloca conexiunea. De asemenea serverul B trebuie să aibă serviciul SSH instalat și pornit .



Telnet

Telnet este un protocol de rețea vechi care este utilizat pentru a se conecta la sisteme de la distanță printr-o rețea TCP/IP. Se conectează la servere și echipamente de rețea prin portul 23.

Centos: `yum install telnet telnet-server -y`

Ubuntu: `apt install telnet -y`

Dacă serviciul de ssh este pornit (pe portul 22) putem verifica și cu comanda: `telnet localhost 22`

Telnet transferă datele în text simplu, în timp ce în SSH datele sunt trimise în format criptat printr-un canal securizat.

Servicii în linux

Un serviciu este un program care rulează în fundal în afara controlului interactiv al sistemului. Pe de altă parte, în sisteme precum Unix sau Linux, serviciile sunt cunoscute și ca demoni. Uneori, numele acestor servicii sau demoni se termină cu litera d

Cum vedem toate serviciile din sistemul nostru?

sudo systemctl

Putem verifica dacă serviciul ssh este pornit pe sistem:

sudo systemctl | grep ssh sau cu comanda **systemctl status ssh.service**



Servicii în linux

De asemenea, putem opri sau porni servicii:

sudo systemctl start/stop [service_name]

O opțiune importantă este să pornim anumite servicii la bootare. Acest lucru se rezolvă prin executarea comenzii de enable pe acel serviciu:

sudo systemctl enable/disable [service_name]

Comenzile curl și wget

Curl este un instrument de linie de comandă pentru a transfera date către sau de la un server, utilizând oricare dintre protocoalele acceptate (HTTP, FTP, IMAP, POP3, SCP, SFTP, SMTP, TFTP, TELNET, LDAP sau FILE).

ex: `curl www.google.ro`

De asemenea, putem descarca un fișier:

`curl -L https://example.com/poza.png -o file.png`

Wget - este o comandă Linux care ne ajută să descărcăm fișierele de pe web folosind protocoalele HTTP, HTTPS și FTP.

ex: `wget https://example.com/poza.png`

`wget -O file.zip http://example.com/file-1.zip`



Cronjob

Cron este numit după cuvântul grecesc „Chronos” care este folosit pentru timp. Este un proces de sistem care va efectua automat sarcini conform programului specific. Este un set de comenzi care sunt utilizate pentru rularea sarcinilor obișnuite de programare.

De este util?

Este util pentru a sterge periodic fisiere de log și să nu încarcăm sistemul, pentru a trimite email-uri periodic sau face back-up zilnic la o bază de date [minut] [ora] [ziua din luna] [luna din an] [ziua din saptamana] [comanda]

Cronjob

Exemple:

`crontab -e` -> editeaza fisierul de crontab și adaugam :

`* * * * date > /tmp/ora` (la fiecare minut scriem în fișierul ora.txt)

`*/5 * * * * echo "test" > /tmp/ora` (la fiecare 5 minute scriem în fișierul ora.txt)

`crontab -l` -> afișează crontab-ul curent

<https://crontab.guru/>

Comanda free

Comandă free afișează cantitatea totală de spațiu liber disponibil împreună cu cantitatea de memorie utilizată.

Se poate folosi cu argumentul `-h` pentru a avea output-ul în format human readable

```
demo@Ubuntu:/tmp$ free
              total        used        free      shared  buff/cache   available
Mem:      1336132      638180      145896         1600       552056       542704
Swap:      969960       24692      945268
demo@Ubuntu:/tmp$ free -h
              total        used        free      shared  buff/cache   available
Mem:          1,3G       623M       142M         1,6M       539M       530M
Swap:         947M        24M       923M
```

Alias

Aliasurile sunt ca comenzile rapide personalizate folosite pentru a reprezenta o comandă (sau un set de comenzi) executată cu sau fără opțiuni personalizate.

```
alias ram="free -h"
```

Alias-urile pot fi temporare sau permanente. În exemplul de mai sus este prezentată varianta temporară. Pentru a seta alias-ul permanent este nevoie sa se adauge comanda de alias în ~/.bashrc .

Ce sunt log-urile?

Logurile oferă o cronologie a evenimentelor pentru sistemul de operare, aplicațiile și sistemul Linux și sunt un instrument valoros de depanare atunci când întâmpinați probleme. Când apar probleme, analiza fișierelor de log este primul lucru pe care trebuie să-l facă un administrator.

Directorul **/var/log** este locația unde se afla majoritatea log-urilor din linux. Acolo avem loguri pentru kernel, sistem, diferite servicii instalate de noi etc



Ce este adresa IP?

O adresă de protocol Internet (adresă IP) este o etichetă numerică care este conectată la o rețea de calculatoare. O adresă IP îndeplinește două funcții principale: identificarea interfeței de rețea și adresarea locației.

O adresă IP este de obicei scrisă în cifre zecimale, formatate ca patru câmpuri de 8 biți separate prin puncte .

An IPv4 address (dotted-decimal notation)

172 . 16 . 254 . 1



10101100 . 00010000 . 11111110 . 00000001



One byte=Eight bits



Thirty-two bits (4 x 8), or 4 bytes

Comenzi utile de networking

ifconfig -> este utilizat pentru a configura sau vizualiza configurația unei interfețe de rețea

traceroute -> utilizat pentru depanarea rețelei

ping -> verifică conectivitatea între două noduri

ip add -> afișează interfețele de rețea de pe server

netstat -> afișează informații despre conexiunile din rețea

route -> afișează și modifică tabela de rutare

hostname -> comanda hostname în Linux este folosită pentru a obține numele DNS (Domain Name System) și pentru a seta numele de gazdă al sistemului

Porturi

Un port este un punct virtual unde încep și se termină conexiunile de rețea. Porturile sunt bazate pe software și gestionate de sistemul de operare al unui computer. Fiecare port este asociat cu un anumit proces sau serviciu.

Cele mai cunoscute porturi:

Protocol	Port
SSH	22
Telnet	23
DNS	53
HTTP	80
FTP	20,21
SMTP	25

Ping

Comanda ping este folosită pentru a verifica dacă o rețea este disponibilă și dacă o gazdă (host-ul) este accesibilă. Cu această comandă, puteți testa dacă un server este în funcțiune și de asemenea, ajută la depanarea diferitelor probleme de conectivitate.

Ping [option] [hostname] or [IP address]

Sfârșit

