

Ghid Practic: 25 de Comenzi Linux pentru un Administrator de Sistem

1. `pwd` – Afișează directorul curent

 Descriere:

Afișează calea absolută a directorului în care te afli în terminal.

 Sintaxă:

```
pwd
```

 Exemplu:

```
pwd
```

 Explicație:

Returnează calea completă a directorului activ, utilă pentru orientare.

 Opțiuni utile:

Această comandă nu are opțiuni suplimentare pentru utilizările de bază.

 Exercițiu:

Deschide terminalul, tastează `pwd` și notează calea afișată. Navighează cu `cd` în alt director și rulează din nou `pwd` pentru a înțelege diferența.

 Sfat:

Folosește `pwd` înainte de comenzi critice precum `rm` sau `cp` ca să fii sigur că lucrezi în directorul potrivit.

2. `ls` – Listează fișierele și directoarele

 Descriere:

Afișează conținutul unui director: fișiere, foldere și detalii despre acestea.

 Sintaxă:

```
ls -lah
```

 Exemplu:

```
ls -lah /etc
```

 Explicație:

Afișează conținutul directorului `/etc` cu detalii extinse, inclusiv fișiere ascunse.

Ghid Practic: 25 de Comenzi Linux pentru un Administrator de Sistem

 Opțiuni utile:

- `-l`: listare detaliată (permisiuni, proprietar, dimensiune)
- `-a`: include fișierele ascunse
- `-h`: formatează dimensiunile în mod „uman” (KB, MB)

 Exercițiu:

Execută `ls -lah ~` pentru a vedea toate fișierele din directorul personal. Observă permisiunile și proprietarii.

 Sfat:

Combină `ls` cu `grep` pentru a filtra fișierele după nume sau extensie.

3. `cd` – Schimbă directorul curent

 Descriere:

Comandă fundamentală pentru navigarea între directoare în terminal.

 Sintaxă:

```
cd /cale/către/director
```

 Exemplu:

```
cd /var/log
```

 Explicație:

Te mută în directorul `/var/log`, unde se află multe dintre fișierele de log ale sistemului.

 Opțiuni utile:

- `cd ..`: urcă un nivel în ierarhia directoarelor
- `cd -`: revine în directorul anterior

 Exercițiu:

Navighează într-un director cu `cd`, apoi folosește `pwd` pentru a verifica noua locație. Testează `cd -` și observă cum revii.

 Sfat:

Folosește `Tab` pentru completarea automată a numelor de directoare și evită greșelile de tastare.

Ghid Practic: 25 de Comenzi Linux pentru un Administrator de Sistem

4. `mkdir` – Creează un nou director

 Descriere:

Creează un director nou în locația specificată.

 Sintaxă:

```
mkdir [opțiuni] nume_director
```

 Exemplu:

```
mkdir -p /tmp/test/subdir
```

 Explicație:

Creează structura întreagă de directoare dacă nu există (`-p`).

 Opțiuni utile:

- `-p`: creează directoare părinți dacă nu există

 Exercițiu:

Creează un lanț de directoare cu `mkdir -p ~/proiecte/linux/test`. Verifică rezultatul cu `ls -R`.

 Sfat:

Folosește `mkdir -p` în scripturi ca să previi erori când directoarele deja există.

5. `rm` – Șterge fișiere și directoare

 Descriere:

Elimină fișiere sau directoare definitiv – nu ajung în coșul de gunoi.

 Sintaxă:

```
rm -rf cale
```

 Exemplu:

```
rm -rf ~/Downloads/vechi
```

 Explicație:

Șterge întregul director `vechi` și tot ce conține, fără confirmări.

Ghid Practic: 25 de Comenzi Linux pentru un Administrator de Sistem

⚙️ Opțiuni utile:

- `-r`: șterge recursiv directoarele
- `-f`: forțează ștergerea fără confirmare
- `-i`: cere confirmare pentru fiecare fișier

🔧 Exercițiu:

Creează un director de test cu fișiere și șterge-l cu `rm -ri`. Observă ce îți cere comanda.

💡 Sfat:

Nu folosi niciodată `rm -rf` / sau în directoare de sistem. Adaugă alias `rm='rm -i'` în `.rc` pentru protecție.

☑️ 6. `cp` – Copiază fișiere și directoare

📄 Descriere:

Comanda `cp` copiază fișiere individuale sau directoare întregi în altă locație.

🔗 Sintaxă:

```
cp [opțiuni] sursă destinație
```

📖 Exemplu:

```
cp -r ~/proiect ~/backup/
```

📖 Explicație:

Copiază tot directorul `proiect` în `~/backup`.

⚙️ Opțiuni utile:

- `-r`: copiază recursiv directoarele și conținutul lor
- `-v`: afișează fiecare fișier copiat
- `-i`: cere confirmare înainte de suprascrisere

🔧 Exercițiu:

Copiază un fișier folosind `cp -i`, apoi încearcă să-l copiezi din nou peste original și observă comportamentul.

💡 Sfat:

Folosește `cp -r` doar pentru directoare. Fără `-r`, comanda va eșua când sursa este un director.

Ghid Practic: 25 de Comenzi Linux pentru un Administrator de Sistem

7. `mv` – Mută sau redenumeste fișiere și directoare

 Descriere:

Mută fișiere/directoare sau le redenumeste, în funcție de destinație.

 Sintaxă:

```
mv [opțiuni] sursă destinație
```

 Exemplu:

```
mv ~/test.txt ~/documente/test_vechi.txt
```

 Explicație:

Fișierul este mutat și redenumit simultan.

 Opțiuni utile:

- `-i`: cere confirmare înainte de suprascriere
- `-v`: afișează fiecare operație

 Exercițiu:

Redenumeste un fișier existent cu `mv vechi.txt nou.txt`. Apoi mută-l într-un alt director.

 Sfat:

Când lucrezi cu fișiere importante, folosește `-i` pentru a evita pierderea datelor prin suprascriere accidentală.

8. `touch` – Creează fișiere goale sau actualizează data

 Descriere:

Creează fișiere goale noi sau actualizează data ultimei modificări.

 Sintaxă:

```
touch nume_fisier
```

 Exemplu:

```
touch raport.txt
```

 Explicație:

Dacă fișierul `raport.txt` nu există, va fi creat. Dacă există, i se actualizează data modificării.

Ghid Practic: 25 de Comenzi Linux pentru un Administrator de Sistem

Opțiuni utile:

- `-c`: nu creează fișierul dacă nu există (doar actualizează)
- `-t`: setează manual data modificării

Exercițiu:

Creează un fișier nou cu `touch test.txt`. Apoi rulează din nou comanda și verifică cu `ls -l` dacă data s-a schimbat.

Sfat:

Comanda `touch` este folosită des în scripturi pentru a verifica sau forța actualizarea fișierelor.

9. `cat` – Afișează conținutul fișierelor

Descriere:

Comanda `cat` (concatenate) afișează conținutul unuia sau mai multor fișiere în terminal.

Sintaxă:

```
cat fișier
```

Exemplu:

```
cat /etc/os-release
```

Explicație:

Afișează conținutul fișierului care conține informații despre sistemul de operare.

Opțiuni utile:

- `-n`: numerotează liniile
- `-b`: numerotează doar liniile non-goale

Exercițiu:

Creează un fișier cu câteva linii de text și folosește `cat -n` pentru a vedea rezultatul.

Sfat:

Pentru fișiere mari, combină `cat` cu `less` sau `grep` pentru o vizualizare mai ușoară.

Ghid Practic: 25 de Comenzi Linux pentru un Administrator de Sistem

✓ 10. `less` – Vizualizează fișiere mari în mod paginat

📄 Descriere:

Permite afișarea fișierelor linie cu linie sau pagină cu pagină. Ideal pentru fișiere mari.

🔗 Sintaxă:

```
less fișier
```

📖 Exemplu:

```
less /var/log/syslog
```

📖 Explicație:

Deschide fișierul de log pentru citire, permițând navigarea cu săgețile.

⚙️ Opțiuni utile:

- `-N`: afișează numerele liniilor
- `-s`: nu rupe liniile lungi
- `/text`: caută un text în fișier

🔗 Exercițiu:

Deschide un fișier mare cu `less`, caută un cuvânt folosind `/`, apoi navighează cu `n` și `N`.

💡 Sfat:

Spre deosebire de `cat`, `less` încarcă doar cât este nevoie în memorie, fiind eficient pentru fișiere mari.

✓ 11. `grep` – Caută texte în fișiere

📄 Descriere:

`grep` este una dintre cele mai puternice comenzi pentru a căuta șiruri de text sau modele în fișiere, cu suport pentru expresii regulate.

🔗 Sintaxă:

```
grep [opțiuni] 'text_căutat' fișier
```

📖 Exemplu:

```
grep -i 'error' /var/log/syslog
```

📖 Explicație:

Caută „error” în fișierul `syslog`, ignorând diferențele dintre litere mari și mici.

Ghid Practic: 25 de Comenzi Linux pentru un Administrator de Sistem

Opțiuni utile:

- `-i`: ignoră diferențele de capitalizare
- `-n`: afișează numărul liniei unde apare textul
- `-r`: caută recursiv în directoare
- `--color=auto`: evidențiază textul găsit

Exercițiu:

Creează un fișier cu mai multe linii, câteva conținând un cuvânt repetat. Caută acel cuvânt cu `grep -n` și observă rezultatele.

Sfat:

Combină `grep` cu `ps`, `dmesg` sau `cat` pentru a filtra rapid rezultatele. Învață expresii regulate de bază pentru a deveni mai eficient.

12. `find` – Caută fișiere și directoare

Descriere:

`find` permite căutarea fișierelor și directoarelor în funcție de nume, mărime, tip, dată etc.

Sintaxă:

```
find [director] [condiție]
```

Exemplu:

```
find /home -name "*.pdf"
```

Explicație:

Caută toate fișierele PDF în directorul `/home` și subdirectoarele sale.

Opțiuni utile:

- `-name`: caută după nume
- `-type f|d`: caută fișiere (f) sau directoare (d)
- `-size +100M`: fișiere mai mari de 100MB
- `-exec cmd {} \;`: execută o comandă pentru fiecare rezultat

Exercițiu:

Caută toate fișierele `.sh` din directorul personal și subdirectoarele lui. Apoi caută toate directoarele cu `-type d`.

Ghid Practic: 25 de Comenzi Linux pentru un Administrator de Sistem

 Sfat:

Combină `find` cu `-exec` pentru a șterge, copia sau modifica fișierele direct în urma căutării.

13. `locate` – Găsește fișiere rapid din index

 Descriere:

Caută fișiere folosind o bază de date indexată, fiind mult mai rapid decât `find`.

 Sintaxă:

```
locate nume_fisier
```

 Exemplu:

```
locate passwd
```

 Explicație:

Returnează toate fișierele din sistem care conțin „passwd” în cale.

 Opțiuni utile:

- `-i`: ignoră diferențele dintre litere mari/mici
- `-c`: afișează doar numărul de potriviri

 Exercițiu:

Rulează `sudo updatedb` apoi caută un fișier cunoscut. Compară rezultatul cu `find`.

 Sfat:

Ideal pentru căutări rapide, dar necesită ca baza de date să fie actualizată periodic (`updatedb`).

14. `df` – Verifică spațiul pe disc

 Descriere:

Arată cât spațiu pe disc este utilizat și cât este disponibil pe fiecare partiție.

 Sintaxă:

```
df -h
```

 Exemplu:

Ghid Practic: 25 de Comenzi Linux pentru un Administrator de Sistem

```
df -h /home
```

 Explicație:

Afișează spațiul total, utilizat și disponibil pentru sistemul de fișiere în care se află `/home`.

 Opțiuni utile:

- `-h`: dimensiuni „umane” (ex: 2G, 450M)
- `-T`: afișează tipul de sistem de fișiere

 Exercițiu:

Rulează `df -h` și analizează cât spațiu mai ai disponibil în fiecare partiție.

 Sfat:

Pentru monitorizare regulată, combină `df -h` cu `watch` pentru actualizare automată la fiecare câteva secunde.

15. `du` – Afișează dimensiunea fișierelor/directoarelor

 Descriere:

Măsoară dimensiunea ocupată de fișiere sau directoare în sistemul de fișiere.

 Sintaxă:

```
du -sh *
```

 Exemplu:

```
du -sh /var/*
```

 Explicație:

Afișează dimensiunea totală a fiecărui subdirector din `/var`.

 Opțiuni utile:

- `-s`: sumă totală, nu pe fiecare fișier
- `-h`: afișare în format „uman”
- `-c`: afișează și totalul general

 Exercițiu:

Rulează `du -sh` într-un director cu multe fișiere și compară dimensiunile afișate cu spațiul raportat de `df`.

Ghid Practic: 25 de Comenzi Linux pentru un Administrator de Sistem

 Sfat:

du este util în identificarea directorilor care ocupă prea mult spațiu. Folosește-l frecvent pe `/var`, `/home` și `/tmp`.

16. `top` – Monitorizare în timp real a proceselor

 Descriere:

Afișează o listă dinamică a proceselor active, similar cu Task Manager din Windows.

 Sintaxă:

```
top
```

 Exemplu:

```
top
```

 Explicație:

Pornește o interfață live care actualizează în timp real procesele și utilizarea CPU/RAM.

 Opțiuni utile:

- `-u utilizator`: afișează doar procesele unui utilizator
- `-n NUMĂR`: rulează un anumit număr de cicluri și se oprește

 Exercițiu:

Rulează `top`, apoi apasă `M` pentru a sorta procesele după consumul de memorie. Apasă `q` pentru a ieși.

 Sfat:

Pentru un mod de afișare mai intuitiv și colorat, încearcă comanda `htop`.

17. `htop` – Versiune interactivă și colorată a `top`

 Descriere:

Un instrument mai vizual și ușor de utilizat decât `top`, cu suport pentru mouse și acțiuni directe.

 Sintaxă:

```
htop
```

 Exemplu:

Ghid Practic: 25 de Comenzi Linux pentru un Administrator de Sistem

```
htop
```

 Explicație:

Pornește o interfață colorată unde poți sorta, căuta și închide procese mai ușor.

 Opțiuni utile:

- `-u` utilizator: filtrează după utilizator
- `-s` coloană: sortează după coloană (ex: CPU, MEM)

 Exercițiu:

Instalează `htop` (dacă nu este deja instalat) cu `sudo apt install htop`. Rulează-l, selectează un proces și apasă `F9` pentru a-l termina.

 Sfat:

`htop` este ideal pentru servere sau VPS-uri când vrei să identifici rapid procesele care consumă resurse.

18. `ps` – Afișează procesele active

 Descriere:

Oferă o fotografie instantanee a proceselor active, utilă pentru scripting și diagnosticare.

 Sintaxă:

```
ps aux
```

 Exemplu:

```
ps aux | grep nginx
```

 Explicație:

Afișează toate procesele și filtrează cele care conțin cuvântul „nginx”.

 Opțiuni utile:

- `a`: include procesele altor utilizatori
- `u`: afișează utilizatorul, timpul, CPU, memorie
- `x`: include procese fără terminal

 Exercițiu:

Caută procesul tău de shell cu `ps aux | grep` și notează PID-ul.

Ghid Practic: 25 de Comenzi Linux pentru un Administrator de Sistem

 Sfat:

Combinatia `ps aux | grep` este extrem de utilă când vrei să verifici dacă un serviciu rulează.

19. `kill` – Trimite semnale către un proces

 Descriere:

Folosită pentru a încheia un proces (sau a trimite alte semnale).

 Sintaxă:

```
kill -9 PID
```

 Exemplu:

```
kill -9 1234
```

 Explicație:

Trimite semnalul `SIGKILL` procesului cu PID 1234 – adică forțază închiderea.

 Opțiuni utile:

- `-9`: terminare imediată (`SIGKILL`)
- `-15`: închidere grațioasă (`SIGTERM` – implicit)
- `-l`: afișează lista semnalelor

 Exercițiu:

Deschide un proces (ex: `sleep 60` într-un alt terminal), identifică PID-ul și închide-l cu `kill`.

 Sfat:

Folosește întâi `kill -15`, pentru a permite procesului să se închidă normal. Dacă nu răspunde, apoi `-9`.

20. `chmod` – Modifică permisiunile fișierelor

 Descriere:

Controlează cine are voie să citească, scrie sau execute un fișier sau director.

 Sintaxă:

```
chmod [mod] fișier
```

Ghid Practic: 25 de Comenzi Linux pentru un Administrator de Sistem

 Exemplu:

```
chmod u+x script.sh
```

 Explicație:

Adaugă drept de execuție pentru utilizatorul care deține fișierul `script.sh`.

 Opțiuni utile:

- `u, g, o, a`: utilizator, grup, alții, toți
- `+, -, =`: adaugă, elimină, setează
- `r, w, x`: citire, scriere, executare

 Exercițiu:

Creează un script și dă-i permisiuni de execuție cu `chmod +x`. Apoi rulează-l și testează dacă funcționează.

 Sfat:

Folosește `chmod` cu înțelegere – nu seta `777` la întâmplare. Este un risc major de securitate.

21. `chown` – Schimbă proprietarul fișierelor

 Descriere:

Comanda `chown` (change owner) modifică utilizatorul și grupul asociat unui fișier sau director.

 Sintaxă:

```
chown [utilizator][:grup] fișier
```

 Exemplu:

```
sudo chown alex:alex document.txt
```

 Explicație:

Fișierul `document.txt` va aparține acum utilizatorului și grupului `alex`.

 Opțiuni utile:

- `-R`: aplică modificarea recursiv (pentru directoare și conținutul lor)
- `--from=USER`: schimbă doar dacă actualul proprietar este cel specificat

 Exercițiu:

Creează un fișier ca `root`, apoi folosește `chown` pentru a-l atribui utilizatorului tău. Verifică cu `ls -l`.

Ghid Practic: 25 de Comenzi Linux pentru un Administrator de Sistem

 Sfat:

Folosește `chown` cu `-R` cu grijă – poți schimba accidental proprietarii la sute de fișiere critice.

22. `systemctl` – Controlează serviciile `systemd`

 Descriere:

Administrează serviciile, sesiunile și unitățile de sistem în distribuțiile moderne Linux care folosesc `systemd`.

 Sintaxă:

```
systemctl [acțiune] nume_serviciu
```

 Exemplu:

```
sudo systemctl restart apache2
```

 Explicație:

Repornește serviciul Apache.

 Opțiuni utile:

- `start`, `stop`, `restart`, `reload`: controlează serviciul
- `enable`, `disable`: activează/dezactivează la boot
- `status`: afișează starea curentă

 Exercițiu:

Rulează `systemctl status ssh` și interpretează ieșirea. Apoi oprește și pornește serviciul dacă este permis.

 Sfat:

Nu confunda `restart` cu `reload`. `reload` nu întrerupe serviciul – doar îl forțează să recitească configurația.

23. `journalctl` – Afișează logurile sistemului

 Descriere:

Instrument pentru citirea jurnalului de sistem gestionat de `systemd`.

 Sintaxă:

```
journalctl [opțiuni]
```

Ghid Practic: 25 de Comenzi Linux pentru un Administrator de Sistem

 Exemplu:

```
journalctl -u ssh
```

 Explicație:

Afișează logurile generate de serviciul `ssh`.

 Opțiuni utile:

- `-xe`: ultimele mesaje, cu detalii extinse
- `-f`: urmărește logurile în timp real (ca `tail -f`)
- `--since="1 hour ago"`: filtrează după timp

 Exercițiu:

Testează `journalctl -xe` după o repornire sau un eveniment important. Apoi folosește `-u` pentru un serviciu anume.

 Sfat:

`journalctl` este o unealtă esențială pentru depanarea problemelor – învață să o folosești împreună cu `grep` pentru filtrare rapidă.

24. `crontab` – Gestionează sarcini automate

 Descriere:

Permite programarea automată a comenzilor care se execută periodic, fără intervenția utilizatorului.

 Sintaxă:

```
crontab -e
```

 Exemplu:

```
0 5 * * * /home/user/backup.sh
```

 Explicație:

Rulează scriptul `backup.sh` în fiecare zi la ora 5:00 dimineața.

 Opțiuni utile:

- `-e`: editează sarcinile utilizatorului curent
- `-l`: listează sarcinile existente
- `-r`: șterge toate sarcinile

Ghid Practic: 25 de Comenzi Linux pentru un Administrator de Sistem

Exercițiu:

Adaugă o sarcină care scrie ora curentă într-un fișier la fiecare 2 minute. Verifică dacă rulează cum trebuie.

Sfat:

Folosește `crontab -l` des pentru a verifica dacă sarcinile sunt corect configurate. Nu uita să incluzi căi absolute în comenzile din cron.

25. ping – Testează conectivitatea de rețea

Descriere:

Trimite pachete ICMP către o adresă IP sau domeniu pentru a verifica dacă este accesibil.

Sintaxă:

```
ping [opțiuni] adresa
```

Exemplu:

```
ping -c 4 google.com
```

Explicație:

Trimite 4 pachete către `google.com` și afișează timpii de răspuns și rata de pierdere.

Opțiuni utile:

- `-c NUMĂR`: numărul de pachete trimise
- `-i SEC`: intervalul dintre pachete
- `-q`: mod silențios – doar sumar
- `-W SEC`: timeout pentru fiecare răspuns

Exercițiu:

Testează `ping -c 5 8.8.8.8` și compară cu `ping -c 5 google.com`. Vezi dacă DNS-ul local funcționează.

Sfat:

`ping` nu înseamnă mereu că ai internet – indică doar dacă gazda e atinsă. Pentru depanare completă, combină cu `traceroute` și `dig`.