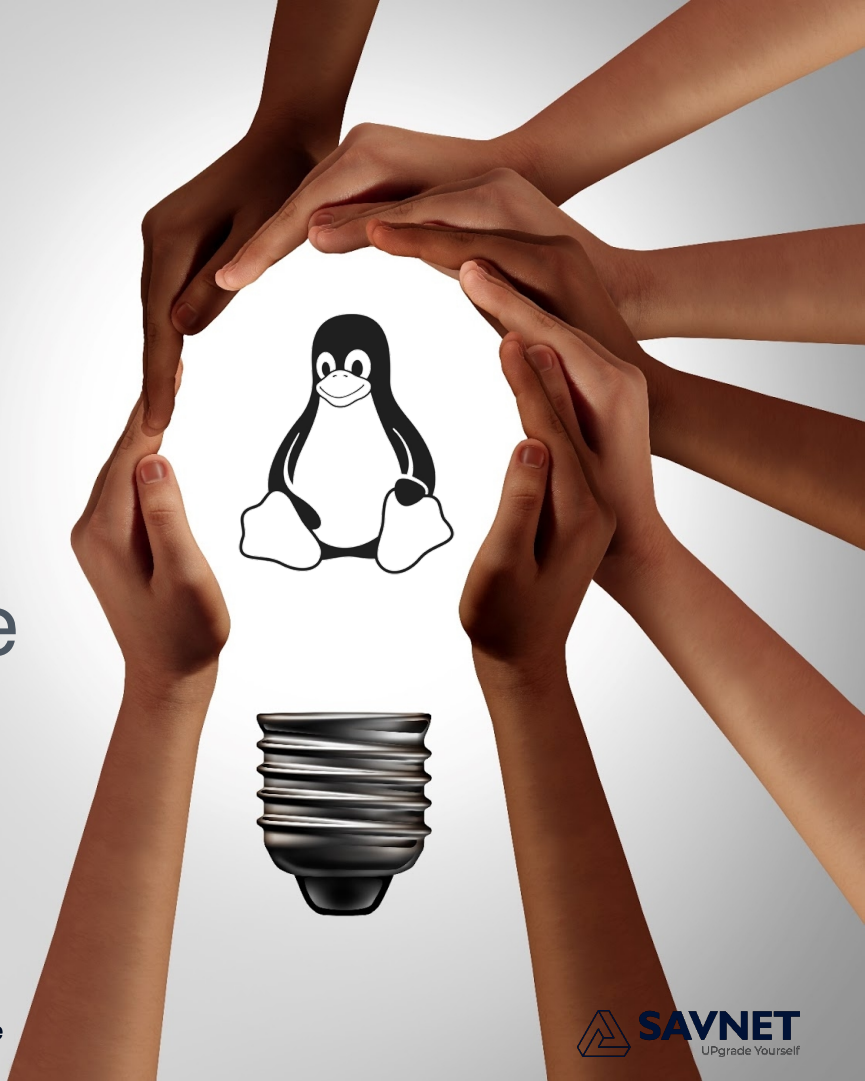


Linux

Administrare și Securitate



Ce vom învăța în acest curs?

- Structura sistemului de fișiere Linux (FHS)
- Comenzi de navigare în sistem
- Montarea sistemelor de fișiere, unelte utile
- Căutarea și localizarea fișierelor în sistem
- Demo: Workflow practic



Capitolul 2:

Navigarea în Sistemul de Fișiere



Structura sistemului de fișiere Linux

Filozofia Linux "Everything is a file"



În Linux, totul este reprezentat și accesat ca fișier
Date, devices, procese, directoare = toate sunt fișiere
Aceleași operații (read/write) pentru toate tipurile

Structura directoarelor:

Windows: "My Computer" - top level

Linux: root directory - / (slash character)

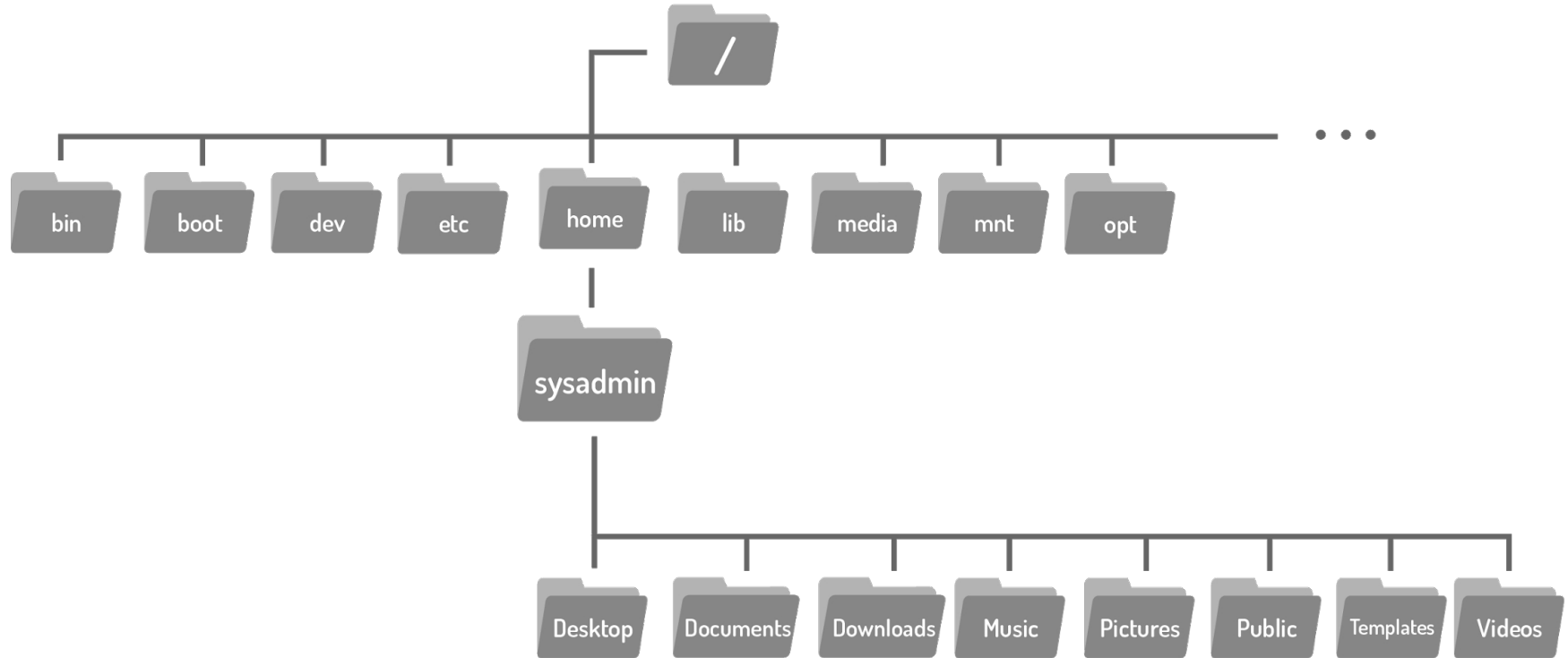
Root - punctul de start al întregului filesystem

Vizualizare conținut root:

ls /

Output conține directoare cu nume descriptive: /boot, /home, /var, /usr, /etc

Filozofia Linux "Everything is a file"



Structura FHS - Filesystem Hierarchy Standard

Directoare de sub root (/):

Linux organizează totul într-o ierarhie unică pornind de la / (root)

FHS = standard pentru locații directoare în toate distribuțiile

Fiecare director are rol specific și bine definit

Directoare esențiale pentru boot:

/bin /sbin /boot /lib

Directoare configurare și date:

/etc /home /root

Directoare device și system info:

/dev /proc /sys /run

Directoare user programs, date servicii:

/usr /opt /srv

Directoare variable data și temporary:

/var /tmp /media /mnt

```
~$ tree -L 1 /  
/  
├── ~  
├── bin -> usr/bin  
├── boot  
├── cdrom  
├── dev  
├── etc  
├── gost.so  
├── home  
├── lib -> usr/lib  
├── lib32 -> usr/lib32  
├── lib64 -> usr/lib64  
├── libx32 -> usr/libx32  
├── lost+found  
├── media  
├── mnt  
├── opt  
├── proc  
├── root  
├── run  
├── sbin -> usr/sbin  
├── snap  
├── srv  
├── swapfile  
├── sys  
├── tmp  
├── usr  
└── var
```

Scurtă detaliare

Configurări și setup:

/etc → toate config files (nginx.conf, ssh_config, fstab)

/boot → kernel și boot files

Logs și troubleshooting:

/var/log → toate log-urile sistem și aplicații

/proc, /sys → info runtime despre sistem și hardware

Programe și executabile:

/bin, /sbin → esențiale pentru boot

/usr/bin, /usr/sbin → restul programelor

/usr/local/bin → instalări manuale admin

Device-uri și hardware:

/dev → access la toate device-urile (disk, network, terminal)

User data:

/home/username → fișiere personale utilizatori

/root → fișiere personale root user

Avantajele FHS - De ce această organizare?

Separarea preocupărilor:

Fiecare director = scop clar definit

/etc = configs | /bin = executabile | /var = date variabile

Backup selectiv:

/etc → backup (configs mici, critice)

/home → backup (user data)

Consistent între toate distribuțiile (troubleshooting standardizat)

Securitate:

Permisiuni diferite per director

/usr read-only în producție

/home izolat per user

Portabilitate:

Scripts funcționează pe orice distro

Admins pot schimba distribuția ușor

Mount-uri separate:

/home pe partition separată → păstrezi data la reinstalare

Home Directory

Locație și scop:

Majoritatea distribuțiilor: /home/ sub root directory

Fiecare user → propriul subdirector: /home/username

User are control complet: create/delete files și subdirectories

Alte directoare protejate prin file permissions

Simbol special pentru home:

Tilda ~ - shortcut pentru home directory

Nume director - nume user (ex: /home/sysadmin)

Comportament implicit:

La deschidere shell → automatic în home directory

Workspace personal user, izolat de sistem



Comenzi de navigare: pwd, cd, ls, tree

Paths - Căi Absolute și Relative

Definiție path:

Listă de directoare separate prin / character

Două tipuri: absolute și relative

Absolute Paths:

Specifică locația exactă pornind de la root

Întotdeauna încep cu / character

Exemplu: `/home/sysadmin` → Start la root / → intră în home → apoi în sysadmin

Independent de current directory

Relative Paths:

Direcții relative la locația curentă în filesystem

NU încep cu /

Exemplu: `cd sysadmin` (relative) - `cd /home/sysadmin` (absolute)

Depind de directorul curent

Path Shortcuts ■ și ■ ■

Double dot (..) - Parent Directory:

Reprezintă directorul imediat superior (one level up)

`cd ..` → urcă un nivel în ierarhie

Utilizare în paths lungi:

`cd ../../etc` → urcă două nivele, apoi intră în /etc

```
/var/www$ ls -la
total 12
drwxr-xr-x  3 root root 4096 iun 26  2025 .
drwxr-xr-x 15 root root 4096 iun 26  2025 ..
drwxr-xr-x  2 root root 4096 iun 26  2025 html
```

Single dot (.) - Current Directory:

Reprezintă directorul curent

Nu e util pentru `cd`, dar esențial pentru alte comenzi

`./script.sh` → execută script din directorul curent

`cp file.txt .` → copiază în directorul curent

Directorul curent și Navigare

`pwd` - **Print Working Directory:**

Afișează locația curentă în filesystem

Output: calea absolută către directorul curent

Change Directory:

`cd Documents` → navighează în subdirectorul Documents

`cd` → return la home directory (fără argumente)

`cd /var/log` → calea absolută către /var/log

Prompt-ul shell afișează directorul curent (de obicei în albastru/verde)

Eroare dacă directorul nu există: "No such file or directory"

Listare fişiere - ls

Unul din cele mai puternice tools de navigare filesystem

Poate afişa informaţii detaliate despre fişiere

Multiple opţiuni pentru customizare output

Utilizare de bază:

ls → listează conţinutul directorului curent

ls /var/log → listează conţinut director specificat

ls /home /etc → listează multiple directoare

Default: sortare alfabetică după nume

Display: doar numele fişierelor, fără metadata

Fișiere Ascunse

Orice fișier/director care începe cu . (dot character)
Este default: omite hidden files

```
-rw----- 1 cristi cristi 79812 ian 15 22:53 .bash_history
-rw-r--r-- 1 cristi cristi 220 iul 18 2023 .bash_logout
-rw-r--r-- 1 cristi cristi 4202 nov 17 22:55 .bashrc
```

Afișare hidden files:

`ls -a` → afișează ALL files, inclusiv hidden

Tipuri comune hidden files:

Fișiere personalizate: .bashrc, .profile, .vimrc

Config directories: .ssh/, .config/

Cache și temporary: .cache/, .local/

Customizare comportament shell, programe, Linux

Nu poluează listingul normal cu config files

Protecție împotriva ștergerii accidentale

Metadata Fişiere

```
drwxr-xr-x  7 root root 4096 iun 26 2025 addons
drwxr-xr-x  2 root root 4096 aug 25 20:47 registry
drwxr-xr-x 17 root root 4096 apr  8 2025 vim82
```

Long display format:

`ls -l` → afişează metadata detaliată

`ls -l /var/log` → long listing pentru director specificat

Metadata afişată:

File Type: primul caracter (d=directory, -=regular, l=link)

Permissions: următoarele 9 caractere

Hard Link Count: număr de link-uri către fişier

User Owner: owner user account

Group Owner: owner group

File Size: dimensiune în bytes

Timestamp: ultima modificare conţinut

File Name: nume fişier/director

File Types:

d = directory | - = regular file | l = symbolic link

s = socket | p = pipe | b = block file | c = character file

Human Readable Sizes și Sorting

Dimensiuni human-readable:

`ls -lh` → file sizes în KB, MB, GB (nu bytes)

Exemplu: 1561400 bytes → 1.5M

Combinație -l (long) + -h (human-readable)

Sortare după criterii:

`ls -lS` → sort după size (descrescător)

`ls -lt` → sort după modification time (cel mai recent primul)

`ls -ltr` → sort după time, reversed (cel mai vechi primul)

`ls -l --full-time` → timestamp complet (ore, minute, secunde)

Reverse sorting:

`ls -lr` → reversează ordinea default (alfabetic)

`ls -lSr` → sort după size, reversed (cel mai mic primul)

```
~$ ls -ld Downloads/
drwxr-xr-x 24 cristi cristi 65536 ian 16 08:59 Downloads/
~$ ls -ldh Downloads/
drwxr-xr-x 24 cristi cristi 64K ian 16 08:59 Downloads/
```

Listare Directoare și Recursivitate

Directory info (nu conținut):

`ls -d` → info despre directorul curent (.)

`ls -ld /etc` → metadata despre /etc directory itself

Fără `-d`: afișează conținutul

Cu `-d`: afișează info despre director ca entity

Recursive listing:

`ls -R` → afișează ALL files în director + subdirectories

`ls -R /etc` → tot tree-ul sub /etc

Display ierarhic: director → conținut → subdirector → conținut

Util pentru overview complet structură filesystem

Poate produce output foarte lung pe directories mari

tree - Vizualizare ierarhică directoare

Afișează structura directoarelor ca un arbore vizual
Alternativă la ls -R (mai ușor de citit)
Foarte util pentru overview rapid structură filesystem

Instalare (dacă lipsește):

apt install tree (Debian/Ubuntu) yum install tree (RHEL/CentOS)

Sintaxă de bază:

tree → arată tot sub directorul curent, nelimitat
tree -L 3 /var/log → limitează la 3 nivele adâncime

Opțiuni utile:

tree -d → doar directoare (fără fișiere)
tree -h → file sizes human-readable (KB, MB, GB)
tree -p → afișează permissions (drwxr-xr-x)
tree -u → afișează owner (user:group)
tree --du → disk usage per directory

```
~$ tree -L 1 /  
/  
├── bin -> usr/bin  
├── boot  
├── cdrom  
├── dev  
├── etc  
├── gost.so  
├── home  
├── lib -> usr/lib  
├── lib32 -> usr/lib32  
├── lib64 -> usr/lib64  
├── libx32 -> usr/libx32  
├── lost+found  
├── media  
├── mnt  
├── opt  
├── proc  
├── root  
├── run  
├── sbin -> usr/sbin  
├── snap  
├── srv  
├── swapfile  
├── sys  
├── tmp  
├── usr  
└── var
```



Montarea unui sistem de fișiere

Conceptul de Montare (Mounting)

Filesystem-ul trebuie "atașat" la un director din ierarhie pentru a fi accesibil

Mount point = directorul unde devine vizibil conținutul

În Linux: totul sub / (un singur tree)

În Windows: C:, D:, E: (litere separate)

Exemplu:

USB stick cu fișiere → conectat fizic → INVIZIBIL pentru sistem

`mount /dev/sdb1 /mnt/usb` → ACUM accesibil la `/mnt/usb/`

Fișierele USB apar în `/mnt/usb/`

Device: `/dev/sdb1` (partition fizică)

Mount point: `/mnt/usb` (director în filesystem)

`mount` = attach | `umount` = detach

Comenzi de bază - mount și umount

Vizualizare mount-uri existente:

`mount` → listează toate filesystem-urile montate

`lsblk` → tree view device-uri și mount points

Montare manuală:

`mount /dev/sdb1 /mnt/usb` → montează USB la `/mnt/usb`

`mount -t nfs server:/share /mnt/network` → montează network share

Demontare:

`umount /mnt/usb` → demontează USB (poate fi scos în siguranță)

Verificare spațiu:

`df -h` → spațiu disponibil pe toate filesystem-urile montate

`du -sh /var/log` → spațiu ocupat de un director

Exemplu practic - Montare USB stick

Workflow complet:

1. Identificare device:

`lsblk`

→ Output: sdb1 (USB stick detectat)

2. Creare mount point (dacă nu există):

`mkdir /mnt/usb`

3. Montare:

`mount /dev/sdb1 /mnt/usb`

4. Utilizare:

`ls /mnt/usb` → vezi fișierele de pe USB

`cp /mnt/usb/config.txt /etc/` → copiere fișier

5. Demontare înainte de scoatere:

`umount /mnt/usb`

→ Acum poți scoate USB-ul în siguranță

Montare persistentă - /etc/fstab

Pentru mount automat la boot:

Fișierul /etc/fstab conține configurarea mount-urilor permanente

La pornire sistem → toate mount-urile din fstab sunt aplicate automat

Exemplu linie fstab (demo):

Device → Mount point → Filesystem type → Options → Dump → Pass
`/dev/sdb1 /mnt/backup ext4 defaults 0 2`

Exemple din industrie - Telecom/Networking

NFS pentru configurări centralizate:

NMS (Network Management System) → configs pe shared storage

```
mount -t nfs config-server:/configs /mnt/nms-configs
```

Beneficiu: backup centralizat, acces de pe multiple servere

Log aggregation:

Router/Switch-uri → trimit logs pe server centralizat

```
mount -t nfs log-collector:/var/log/network /mnt/logs
```

Beneficiu: analiză centralizată, compliance, troubleshooting

TFTP server pentru provisioning:

Switches/Routere downloadează configs prin TFTP

```
mount -t nfs storage:/tftp-root /tftpboot
```

Beneficiu: auto-provisioning echipamente noi (ZTP - Zero Touch Provisioning)



Căutarea și localizarea fișierelor în sistem

locate - Găsește orice fișier rapid

Caută în database pre-built, update nightly

Mult mai rapid decât find pe filesysteme mari (seconds vs minutes pe TB+)

Sintaxă de bază:

locate filename → toate path-urile care conțin "filename"

locate -c pattern → număr de matches (count only)

locate -b 'filename' → exact basename, nu substring

locate -i password → case-insensitive search

Update database:

sudo updatedb → rebuild database instant (altfel update nightly)

Database location: /var/lib/mlocate/mlocate.db sau /var/lib/locate/locatedb

Use cases:

Căutare config files: locate nginx.conf găsește instant toate locațiile

Security audit: locate -i password sau locate .ssh pentru fișiere sensibile

Quick file check: locate -c "*.log" → câte log files există

Limitări importante:

Fișiere create astăzi NU apar până mâine (database update nightly)!!!

Pe sisteme fresh install, primul locate poate eșua - rulează updatedb prima dată

Nu găsește fișiere noi până la următorul updatedb

Când folosești locate:

Știi numele fișierului sau pattern

Fișierul există de cel puțin 1 zi

Vrei rezultate instant (seconds)

find - Căutare live pe filesystem

Caută direct pe filesystem (real-time), mai lent dar vede TOT (inclusiv fișiere noi)
Extrem de flexibil: filter după nume, size, permissions, timestamp, type

Sintaxă de bază:

find /path -name "pattern" → caută după nume
find /etc -name "*.conf" → toate configs din /etc
find /var/log -type f -size +100M → fișiere mari (100MB+)
find /var/log -mtime -7 → modificate în ultimele 7 zile
find /home -perm 777 → world-writable (security risk)

Combinare criterii multiple:

find /var/log -name "*.log" -size +50M -mtime +30 → logs vechi ȘI mari
find /etc -type f -name "*.conf" -mtime -1 → configs modificate azi

Acțiuni pe rezultate:

find /tmp -mtime +7 -delete → șterge fișiere vechi
find /var/log -name "*.log" -exec gzip {} \; → comprimă toate logs
find /backup -mtime +90 -exec rm {} \; → șterge backup-uri vechi

Use cases telecom:

Security audit SUID: find / -perm /4000
Logs umflate disk: find /var/log -size +500M
Cleanup old files: find /tmp -mtime +30 -delete
Config orphan: find /etc -name "*backup*" -mtime +180

Când folosești find:

Fișiere create recent (astăzi/acum)
Criterii complexe (size + time + permissions)
Acțiuni pe rezultate (delete, compress, move)

whereis - Unde e instalată comanda?

Caută în locații standard pentru comenzi și man pages

Sintaxă și exemple:

whereis ls → ls: /bin/ls /usr/share/man/man1/ls.1.gz

whereis python → găsește TOATE instalările Python din sistem

whereis nginx → binary + man pages + source files (când există)

Ce găsește:

Executabilul, man page-urile (.gz comprimate), source files

Man pages sunt comprimate cu gzip → extensia .gz

whereis vs which:

whereis python vs which python - whereis găsește tot, which doar ce e în PATH

whereis = toate locațiile standard | which = doar primul executabil din PATH

Use cases:

Util când ai multiple versiuni instalate și vrei să le vezi pe toate

Debugging PATH issues - vezi ce e instalat vs ce rulează efectiv

Verificare man pages disponibile pentru o comandă

Limitări:

Căutare limitată la locații standard: /bin, /usr/bin, /usr/share/man, etc.

Nu găsește executabile în locații non-standard sau în /usr/local/bin



Demo: Workflow

Ce am învățat în acest capitol?

Navigarea în Sistemul de Fișiere	Cum ne deplasăm și explorăm: pwd (unde suntem), cd (navigare), ls (listare), tree (vizualizare ierarhică)
Montarea Sistemelor de Fișiere	Cum facem accesibile dispozitivele și partițiile: mount, umount, /etc/fstab pentru mounting persistent
Localizarea Fișierelor	Cum găsim fișiere rapid: locate (căutare în database), find (căutare live cu criterii complexe), whereis (localizare comenzi)



Q&A



Activități pentru acasă



Ce urmează?



Gestionarea Fișierelor și Directoarelor

Înțelegerea Hardware-ului Computerului

Introducere in scripting

•

