

# Souborové systémy

Eliška Jégrová

24. 11. 2025

# Obsah

Adresářová struktura

Souborové systémy

Inode a odkazy

Zařízení a souborové systémy

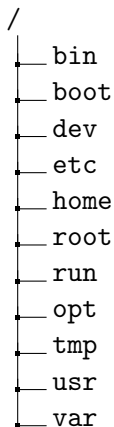
Připojování souborových systémů (mount)

Připojování souborových systémů

# Kořenový adresář /

- ▶ V Linuxu jsou všechny soubory v jednom stromu pod /.
- ▶ Na rozdíl od Windows (disky C:, D:, ...) je tu **jeden společný strom**.
- ▶ Strukturu / popisuje standard **Filesystem Hierarchy Standard (FHS)**.
- ▶ Cíle:
  - ▶ oddělit systémové soubory od dat uživatelů,
  - ▶ rozlišit soubory jen ke čtení vs. pro zápis,
  - ▶ umožnit sdílení částí systému mezi více počítači.

# Adresářový strom – základní přehled



- ▶ Tenhle základ najdeš (s drobnými rozdíly) na všech unixových systémech.
- ▶ Uživatelé pracují nejčastěji s /home, případně s /etc (konfigurace) a logy ve /var/log.

## /bin, /usr/bin – programy

- ▶ /bin – základní příkazy systému (např. bash, ls, cat).
- ▶ /usr/bin – „většina ostatních“ programů.
- ▶ Na moderní Fedoře je /bin jen odkaz na /usr/bin.
- ▶ Historicky bylo rozdělení důležité kvůli sdílení po síti; dnes zůstalo hlavně kvůli kompatibilitě.

## /dev a /boot

- ▶ /dev – speciální soubory reprezentující zařízení:
  - ▶ např. /dev/sda, /dev/tty0, /dev/null.
  - ▶ platí „všechno je soubor“ – čtení/zápis se mapuje na zařízení.
- ▶ /boot – soubory potřebné ke startu systému:
  - ▶ linuxové jádro,
  - ▶ konfigurace zavaděče (bootloaderu),
  - ▶ případně více verzí jádra pro návrat „zpět“.

/etc, /home, /root

- ▶ /etc – konfigurace systému:
  - ▶ např. /etc/passwd, /etc/fstab, /etc/ssh/sshd\_config.
  - ▶ většinou textové soubory, upravuje je jen **root**.
- ▶ /home – domovské adresáře běžných uživatelů:
  - ▶ /home/uzivatel, zkráceně ~.
  - ▶ sem si ukládají vlastní data a konfiguraci.
- ▶ /root – domovský adresář uživatele **root**:
  - ▶ root nemá home v /home, ale právě v /root,
  - ▶ běžný uživatel sem normálně nemá přístup.

## /var, /tmp, /run, /opt

- ▶ /var – „proměnná“ data systému:
  - ▶ logy (/var/log),
  - ▶ maily, spool tiskáren,
  - ▶ databáze a pracovní data služeb.
- ▶ /tmp – dočasné soubory:
  - ▶ používají ho aplikace jako odkladiště,
  - ▶ obsah se často maže po restartu.
- ▶ /run – běhová data po startu systému:
  - ▶ PID soubory, sockety, různé „stavové“ informace,
  - ▶ existují jen za běhu (po rebootu se vytváří znovu).
- ▶ /opt – „volitelné“ aplikace:
  - ▶ typicky software třetích stran,
  - ▶ může mít vlastní podadresář, např. /opt/firma/app.

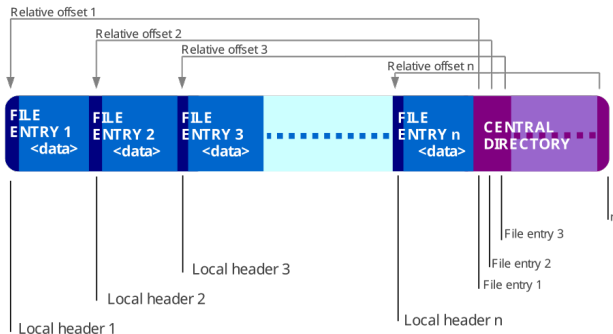


# Co je souborový systém

- ▶ Na disku jsou data jako dlouhá řada jedniček a nul.
- ▶ **Souborový systém** je dohoda, jak z toho udělat:
  - ▶ soubory se jménem,
  - ▶ adresáře (strom),
  - ▶ informace o velikosti, vlastnících, právech, časech změny, ...
- ▶ Umožní:
  - ▶ rychle najít soubor podle cesty,
  - ▶ přidávat/mazat soubory,
  - ▶ kontrolovat přístup pomocí práv.

# Analogie: ZIP archiv

- ▶ ZIP soubor:
  - ▶ uvnitř jsou za sebou naskládána **data souborů**,
  - ▶ na konci je **seznam souborů** (centrální adresář ZIPu).
- ▶ Seznam říká:
  - ▶ jak se soubor jmenuje,
  - ▶ kde v ZIPu data začínají,
  - ▶ jak je soubor dlouhý.



# Inode – informace o souboru

- ▶ Soubor má:
  - ▶ **obsah** (data),
  - ▶ **jméno** (cesta v adresáři),
  - ▶ **metadata** (velikost, práva, čas změny, ...).
- ▶ V Linuxu jsou metadata uložená v **inode**:
  - ▶ typ souboru (běžný, adresář, link),
  - ▶ práva, vlastník, skupina,
  - ▶ velikost, časy změny,
  - ▶ počet odkazů,
  - ▶ odkazy na datové bloky na disku.
- ▶ Ke zjištění inode se používá příkaz `stat`  
`$ stat basnicka.txt`

# Inode vs. jméno souboru

- ▶ **Jméno** souboru je jen záznam v adresáři.
- ▶ Jméno ukazuje na **inode**, inode ukazuje na data.
- ▶ Číslo inode zjistíš např.:

```
$ ls -li soubor.txt  
123456 soubor.txt
```

```
$ stat soubor.txt
```

- ▶ Když soubor přejmenuješ:

```
$ mv soubor.txt novy.txt
```

inode zůstává stejné – změnilo se jen jméno v adresáři.

# Odkazy - Hard link

- ▶ **Hard link** = další jméno, které ukazuje na stejný inode.

- ▶ Vytvoření:

```
$ ln puvodni.txt kopie.txt
```

- ▶ Oba názvy:

- ▶ mají **stejné číslo inode**,
- ▶ ukazují na stejná data,
- ▶ jsou „rovnocenné“ – žádný není „ten původní“.

- ▶ Když jeden z nich smažeš:

```
$ rm puvodni.txt
```

data zůstanou – dokud existuje aspoň jeden hard link na daný inode.

# Odkazy - Symbolický (soft) link

- ▶ **Symbolický link** (symlink, soft link):
  - ▶ zvláštní soubor, který obsahuje **cestu** k jinému souboru,
  - ▶ má své vlastní číslo inode.
- ▶ Vytvoření:  

```
$ ln -s cilovy_soubor link.txt
```
- ▶ Chování:
  - ▶ když čteš `link.txt`, systém přesměruje přístup na cílový soubor,
  - ▶ když se cílový soubor smaže, `link.txt` zůstane, ale je „rozbítý“.
- ▶ Na rozdíl od hard linku může symlink:
  - ▶ mířit i na jiný souborový systém,
  - ▶ mířit na adresář.

# Samostatná práce – inode a odkazy

1. Vytvoř soubor `data.txt` s libovolným obsahem.
2. Zobraz jeho inode:  

```
$ ls -li data.txt
```

```
$ stat data.txt
```
3. Vytvoř hard link `data2.txt`. (`ln`)
4. Zobraz si inode (krok 2) pro `data2.txt` a porovnej s předešlými výpisy.
5. Vytvoř symlink `data-link.txt` (`ln -s`) a znovu se podívej na inode.
6. Zkus smazat `data.txt` a sleduj, co se stane s `data2.txt` a `data-link.txt`.

# Bloková zařízení a souborové systémy

- ▶ Data jsou fyzicky uložena na **zařízeních**:
  - ▶ pevné disky, SSD, USB flashky, virtuální disky ve VM, ISO obrazy. . .
- ▶ V Linuxu jsou vidět jako **bloková zařízení** v `/dev`:
  - ▶ např. `/dev/sda`, `/dev/sda1`, `/dev/vda`, . . .
- ▶ Na zařízení může být:
  - ▶ přímo souborový systém (např. `/dev/sda`),
  - ▶ nebo tabulka oddílů a souborový systém až na oddílů (např. `/dev/sda1`).
- ▶ **Souborový systém** (ext4, xfs, vfat, . . . ) určuje, jak jsou data a adresáře na zařízení uspořádané.



## lsblk – přehled disků a oddílů

- ▶ Příkaz `lsblk` zobrazí bloková zařízení a jejich strukturu:

```
$ lsblk
```

NAME	MAJ:MIN	RM	SIZE	RO	TYPE	MOUNTPOINTS
vda	252:0	0	40G	0	disk	
-vda1	252:1	0	1G	0	part	/boot
-vda2	252:2	0	39G	0	part	/

- ▶ Většinou uvidíš:
  - ▶ disk (TYPE disk),
  - ▶ na něm oddíly (TYPE part),
  - ▶ u nich případně sloupec MOUNTPOINTS – kam jsou připojené.
- ▶ Hodí se před každou prací s mountem – ať víš, jak se zařízení jmenují.

## df – využití souborových systémů

- ▶ df ukazuje, jak jsou souborové systémy velké a zaplněné:

```
$ df -h
```

Filesystem	Size	Used	Avail	Use%	Mounted on
/dev/vda2	39G	15G	22G	41%	/
tmpfs	2.0G	0	2.0G	0%	/tmp

- ▶ Užitečné sloupce:
  - ▶ Filesystem – zařízení nebo název FS,
  - ▶ Size, Used, Avail – velikost a využití,
  - ▶ Mounted on – kam je FS připojený.
- ▶ Přepínač -h (human-readable) ukazuje velikosti v MB/GB místo v blocích.

## findmnt / – co je připojeno jako /

- ▶ Příkaz `findmnt` ukáže, jaký souborový systém je připojený na daném adresáři.
- ▶ Začneme kořenovým adresářem:

```
$ findmnt /
```

```
TARGET SOURCE      FSTYPE OPTIONS  
/        /dev/sda2 ext4    rw,relatime
```

- ▶ Sloupce:
  - ▶ TARGET – kam je FS připojený (adresář),
  - ▶ SOURCE – odkud se bere (např. zařízení nebo jméno),
  - ▶ FSTYPE – typ souborového systému,
  - ▶ OPTIONS – základní připojovací volby.

## findmnt – další příklady

- ▶ Stejným způsobem se můžeme podívat na další mountpointy:

```
$ findmnt /boot
```

```
TARGET SOURCE      FSTYPE OPTIONS
```

```
/boot  /dev/sda1 ext4    rw,relatime
```

```
$ findmnt /home
```

```
TARGET SOURCE      FSTYPE OPTIONS
```

```
/home  /dev/sda3  ext4    rw,relatime
```

- ▶ Dobré otázky pro studenty:
  - ▶ Který souborový systém je připojený na /boot?
  - ▶ Který na /home?

# findmnt – přehled připojených souborových systémů

- ▶ findmnt ukazuje, co je kde připojeno, v podobě stromu:

```
$ findmnt
```

TARGET	SOURCE	FSTYPE	OPTIONS
/	/dev/vda2	ext4	rw,relatime
-/boot	/dev/vda1	ext4	rw,relatime
'-/tmp	tmpfs	tmpfs	rw,nosuid,nodev

- ▶ Přehlednější než čistý výstup mount.
- ▶ Hodí se pro ověření:
  - ▶ jestli se souborový systém připojil tak, jak jsme chtěli,
  - ▶ co všechno je připojené pod /mnt, /media, ...

# Mount point – princip

- ▶ V Linuxu je jeden společný strom adresářů pod /.
- ▶ **Mount point** je adresář, do kterého se připojí souborový systém.
- ▶ Příklady:
  - ▶ / – hlavní souborový systém,
  - ▶ /boot – zvlášť připojený souborový systém,
  - ▶ dočasně můžeme něco připojit např. do /mnt.
- ▶ Příkaz `findmnt` nám pomáhá zjistit, co je kam připojeno.

## mount – ruční připojení

- ▶ Ruční připojení souborového systému:

```
# mount zdroj cílový_adresář
```

- ▶ Typický scénář v návodu:

- ▶ vytvoří se adresář pro připojení, např. /mnt/test,

```
# mkdir /mnt/test
```

- ▶ použije se mount, aby byl obsah vidět v /mnt/test.

- ▶ Po připojení můžeme zkontrolovat:

```
$ findmnt /mnt/test
```

# Mount point – připojení souborového systému

- ▶ V Linuxu je jeden společný strom adresářů pod /.
- ▶ **Mount point** = adresář, do kterého „vložíme“ jiný souborový systém.
- ▶ Příklad:
  - ▶ / – hlavní (kořenový) souborový systém,
  - ▶ /boot – může být jiný FS na jiném oddílu,
  - ▶ USB flashka může být připojena třeba do /run/media/... nebo /mnt/usb.
- ▶ Bez mountu zařízení existuje, ale není nikde vidět ve stromu souborů.



# mount – ruční připojení

- ▶ Základní tvar:

```
# mount zařízení adresář
```

- ▶ Příklad (USB disk s jedním oddílem):

```
# mkdir /mnt/usb
```

```
# mount /dev/sdb1 /mnt/usb
```

- ▶ Po připojení uvidíš obsah flashky v /mnt/usb.
- ▶ Typ souborového systému (-t ext4, -t vfat, ...) se většinou pozná automaticky.
- ▶ Co je kde připojeno, zjistíš např.:  

```
$ findmnt
```

# umount – odpojení

- ▶ Odpojení souborového systému:

```
# umount /mnt/test
```

- ▶ Lze odpojit i podle zdroje:

```
# umount zdroj
```

- ▶ Pokud je souborový systém používán (otevřený soubor, shell v tom adresáři), umount selže.
- ▶ Po odpojení `findmnt /mnt/test` nic nenajde.

# umount – odpojení souborového systému

- ▶ Odpojení:

  - # `umount /mnt/usb`

- ▶ nebo podle zařízení:

  - # `umount /dev/sdb1`

- ▶ Když je souborový systém „zanepřázdněný“:

  - ▶ někdo má otevřený soubor,
  - ▶ máš otevřený shell v tom adresáři,
  - ▶ bude hláška typu „target is busy“.

- ▶ Řešení:

  - ▶ přejít jinam (`cd`) a zavřít programy,
  - ▶ pak zkusit `umount` znovu.

## /etc/fstab – připojení při startu

- ▶ Soubor /etc/fstab říká, které souborové systémy se připojí automaticky při startu.
- ▶ Ukázka jednoduché konfigurace:

```
/dev/sda2  /          ext4  defaults  1 1
/dev/sda1  /boot      ext4  defaults  1 2
```

- ▶ Základní sloupce:
  - ▶ **zdroj** – napr. /dev/sda2 nebo UUID,
  - ▶ **cílový adresář** – mount point,
  - ▶ **typ** – typ souborového systému,
  - ▶ **options** – základní volby (defaults apod.).

## /etc/fstab – co se má připojit při startu

- ▶ Soubor /etc/fstab říká, co a kam se má připojit při startu systému.

- ▶ Každý řádek popisuje jeden souborový systém:

```
/dev/vda2  /          ext4  defaults  1 1
/dev/vda1  /boot      ext4  defaults  1 2
```

- ▶ Důležité sloupce:

- ▶ **co** – zařízení nebo UUID,
- ▶ **kde** – mount point,
- ▶ **typ** – např. ext4, xfs, vfat, ...
- ▶ **options** – nastavení připojení (pro začátek stačí defaults).

- ▶ Nemusíš znát všechny detaily, ale je dobré vědět, že automatické mounty se řídí právě tímto souborem.

## Samostatná práce – findmnt, mount, umount

1. Pomocí `findmnt` / zjisti, jaký souborový systém je připojený na kořenový adresář.
2. Zkus `findmnt /boot` a `findmnt /home` (pokud existuje) a podívej se, jak se liší `SOURCE` a `FSTYPE`.
3. Vytvoř adresář `/mnt/test` a podle návodu na cvičení do něj připoj souborový systém. Ověř:  

```
$ findmnt /mnt/test
```
4. Souborový systém opatrně odpoj pomocí `umount` a zkontroluj, že `findmnt /mnt/test` už nic nevrací.

# Samostatná práce – mount a přehled FS

1. Pomocí `lsblk` zjisti, jaká bloková zařízení a oddíly má tvoje virtuálka.
2. Zobraz přehled souborových systémů a jejich využití:  
`$ df -h`
3. Podívej se, co je právě připojeno:  
`$ findmnt`
4. Otevři `/etc/fstab` (např. `less /etc/fstab`) a najdi, který řádek odpovídá kořenovému souborovému systému `/` a který `/boot`.
5. Pokud máš v prostředí připravený další disk/oddíl, zkus ho ručně připojit do `/mnt/test` a ověř, že se objeví ve výstupu `findmnt`.