

Souborové systémy

Eliška Jégrová

24. 11. 2025

Obsah

Adresářová struktura

Souborové systémy

Inode a odkazy

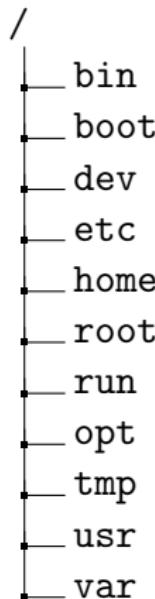
Zařízení a souborové systémy

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



- ▶ Tenthle základ najdeš (s drobnými rozdíly) na všech linuxových 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 rozdelení 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 jadra 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:
 - ▶ sem si ukládají uživatelé 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

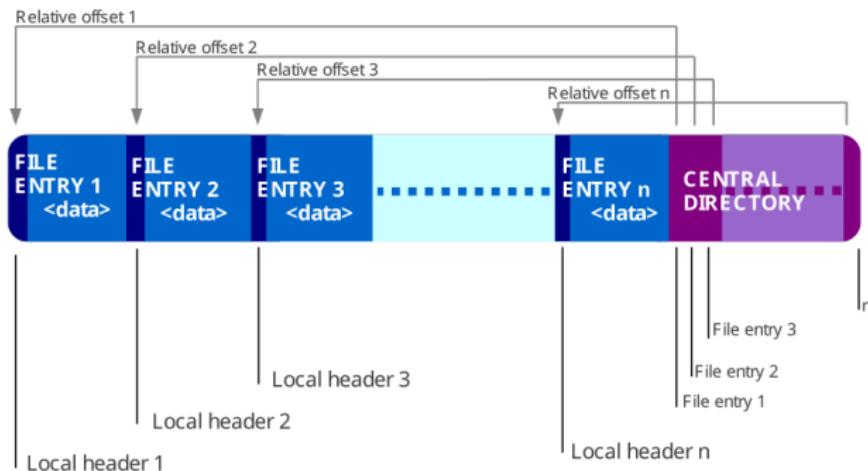
- ▶ /var – „proměnná“ data systému:
 - ▶ logy (/var/log),
 - ▶ maily, spool tiskáren,
 - ▶ databáze a pracovní data služeb.
- ▶ /tmp – dočasné soubory:
 - ▶ používají ho aplikace jako odkladiště,
 - ▶ obsah se často maže po restartu.
- ▶ /run – běhová data po startu systému:
 - ▶ PID soubory, sockety, různé „stavové“ informace,
 - ▶ existují jen za běhu (po rebootu se vytváří znova).
- ▶ /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 naskládaná **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 -i 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 „rozbitý“.
- ▶ Na rozdíl od hard linku může symlink:
 - ▶ mířit i na jiný souborový systém,
 - ▶ mířit na adresář.

Samostatná práce 1 – inode a odkazy

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

```
$ ls -i 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 znova se podívej na inode.
6. Zkus smazat data.txt a sleduj, co se stane s data2.txt a data-link.txt.

Disk jako soubor

- ▶ Data jsou fyzicky uložená 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 oddílu (např. /dev/sda1).

Připojené souborové systémy

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

```
$ findmnt  
TARGET SOURCE FSTYPE OPTIONS  
/ /dev/vda3[/root] btrfs rw,relatime,seclabel,  
compress=zstd:1,discard=async,  
space_cache=v2,subvolid=256,subvol  
|-/boot /dev/vda2 btrfs rw,relatime,seclabel  
|-/tmp tmpfs tmpfs rw,nosuid,nodev,  
seclabel,nr_inodes=1048576,inode64
```

- ▶ případně jaký souborový systém je připojený na daném adresáři.

```
$ findmnt /  
TARGET SOURCE FSTYPE OPTIONS  
/ /dev/vda3[/root] btrfs rw,relatime,seclabel,..
```

df – využití souborových systémů

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

```
$ df -h
```

Filesystem	Size	Used	Avail	Use%	Mounted on
/dev/vda3	19G	4,8G	14G	26%	/
tmpfs	2.0G	0	2.0G	0%	/tmp

- ▶ Přepínač -h (human-readable) ukazuje velikosti v MB/GB místo v blocích.

lsblk – přehled disků a oddílů

- ▶ Příkaz lsblk zobrazí bloková zařízení a jejich strukturu:

```
$ lsblk
NAME   MAJ:MIN RM  SIZE RO TYPE MOUNTPOINTS
sr0     11:0    1  2,2G  0 rom  /run/media/eliska/Fedora
zram0   251:0   0  3,8G  0 disk [SWAP]
vda     253:0   0   20G  0 disk 
|---vda1 253:1   0    1M  0 part 
|---vda2 253:2   0    1G  0 part /boot
|---vda3 253:3   0   19G  0 part /home
                           /
```

Mount point

- ▶ **Mount point** je adresář, do kterého se připojí souborový systém.
- ▶ Bez mountu zařízení existuje, ale není nikde vidět ve stromu souborů.
- ▶ Ruční připojení souborového systému:

```
# mount zdroj cílový_adresář
```
- ▶ Příklad (USB disk s jedním oddílem):

```
# mkdir /mnt/usb
# mount /dev/sdb1 /mnt/usb
```
- ▶ Typ souborového systému (-t ext4, -t vfat, ...) se většinou pozná automaticky.

umount – odpojení

- ▶ Odpojení souborového systému:

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

- ▶ Lze odpojit i podle zdroje:

```
# umount /dev/sdb1
```

- ▶ Pokud je souborový systém používán (otevřený soubor, shell v tom adresáři), umount selže.

- ▶ Po odpojení findmnt /mnt/test nic nenajde.

Připojení při startu

- ▶ Soubor `/etc/fstab` říká, které souborové systémy se připojí automaticky při startu.
- ▶ Každý řádek popisuje jeden souborový systém:

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

- ▶ **zdroj** – napr. `/dev/sda2` nebo UUID,
- ▶ **cílový adresář** – mount point,
- ▶ **typ** – typ souborového systému,
- ▶ **options** – základní volby (`defaults` apod.).
- ▶ Po zápisu do souboru spusť `mount -a`, kterým se uplatní změny a disk se připojí, případně se vypíše chyba.
- ▶ **Chybná konfigurace způsobí nenastartování systému po restartu**

Cvičení v hodině – mount/umount

1. Stáhni si alpine.iso z

<https://alpinelinux.org/downloads/>

2. Připoj disk k virtuálce:

```
$ sudo mount -o ro alpine.iso /mnt
```

3. Zobraz si adresář /mnt:

```
$ ls /mnt
```

4. Podívej se, co je právě připojeno:

```
$ findmnt
```

5. Odpoj iso:

```
$ sudo umount /mnt
```

6. Zobraz si adresář /mnt:

```
$ ls /mnt
```

Samostatná práce 2 – mount a přehled FS

1. Zjisti, jaká bloková zařízení a oddíly má tvoje virtuálka. (lsblk)
2. Zobraz přehled souborových systémů a jejich využití. (df)
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.

Samostatná práce 3 – findmnt, mount, umount

1. Vytvoř adresář /mnt/test a podle návodu na cvičení do něj připoj souborový systém. Ověř:
`$ findmnt /mnt/test`
2. Souborový systém opatrně odpoj pomocí umount a zkontroluj, že findmnt /mnt/test už nic nevrací.
3. Otevři /etc/fstab a vlož záznam pro připojení alpine.iso.
4. Spusť příkaz přo ověření správně nastaveného /etc/fstab.
(mount -a)
5. Pokud máš zkontrolováno od kouče, restartuj virtuálku a přesvědč se, že se alpine připojilo. (nepovinný krok)
6. Odeber z /etc/fstab záznam o alpine.iso