

# CVIČENÍ 2

ZOS

L. Pešička, 2024



# OBSAH

- Příkaz mount, /etc/fstab, /etc/mtab
- Nejdůležitější podadresáře v /
- Přístupová práva

# POJMY - OPAKOVÁNÍ A ROZŠÍŘENÍ

## Příkazový interpret

- Program, který čeká na vstup uživatele
- Bud' vykoná zadaný příkaz nebo zobrazí chybové hlášení.
- Příkladem příkazového interpretu je `/bin/bash`
- Existují ale i další (zkuste `cat /etc/shells`, `man shells`)

## Kolona příkazů

- Umožňuje výstup jednoho procesu poslat na vstup dalšího procesu.
- Používá se roura, symbol `|`
- `cat nakup.txt | grep rohlik | wc -l`

# POJMY II.

## Přesměrování výstupu do souboru

- místo aby výstup procesu šel na obrazovku, uloží se do nějakého souboru
- Používá se symbol `>`
- `cat nakup.txt | grep rohlik | wc -l > seznam.txt`
- Pokud bychom chtěli připisovat na konec souboru, pak `>>`

# ZÁKLADNÍ STRUKTURA SOUBOROVÉHO SYSTÉMU

- jeden kořen /
- `ls -l /`
- více pevných disků – připojeny jako podstrom od nějakého adresáře, např.:

| Oblast pevného disku | Místo připojení | Obsah                       |
|----------------------|-----------------|-----------------------------|
| /dev/sda1            | /               | Linuxový systém             |
| /dev/sdb1            | /home           | Domovské adresáře uživatelů |

Výhoda – pokud uživatelé zaplní přidělený disk, systém může stále rozumným způsobem fungovat

# PŘÍKAZ MOUNT

- zadejte příkaz **mount** a pozorujte výstup
- zobrazí odkud a **kam** jsou připojené filesystemy  
příklady výpisu příkazu mount:  
`/dev/sda2 on /var type ext3 (rw)`  
`/dev/mapper/vg_local-root on / type xfs`
- umožní připojit další filesystem, příklad připojení:  
`mount -t ext3 /dev/sda4 /mnt/data`  
(typ fs, co připojujeme, kam)

# UKÁZKA PŘÍKAZU MOUNT

```
eryx2> mount  
/dev/md0 on / type xfs (rw)  
tmpfs on /lib/init/rw type tmpfs (rw,nosuid,mode=0755)  
proc on /proc type proc (rw,noexec,nosuid,nodev)  
sysfs on /sys type sysfs (rw,noexec,nosuid,nodev)  
procusb on /proc/bus/usb type usbfs (rw)  
udev on /dev type tmpfs (rw,mode=0755)  
tmpfs on /dev/shm type tmpfs (rw,nosuid,nodev)  
devpts on /dev/pts type devpts (rw,noexec,nosuid,gid=5,mode=620)  
/dev/md4 on /usr type xfs (rw)  
/dev/md2 on /var type xfs (rw)  
/dev/md3 on /var/cache/openafs type ext3 (rw)  
AFS on /afs type afs (rw)  
eryx2>
```

zařízení `/dev/md0` je připojeno jako kořen `/`, typ fs je `xfs`, čtení i zápis

..

zařízení `/dev/md2` je připojeno do adresáře `/var`, typ fs je `xfs`, práva `rw`  
Starší a přehlednější výpis, na eryxu v roce 2023 vypadá výpis jinak

# /ETC/FSTAB — JINÝ STROJ, UUID

```
root@debian0ra:~# cat /etc/fstab
# /etc/fstab: static file system information.
#
# Use 'blkid' to print the universally unique identifier for a
# device; this may be used with UUID= as a more robust way to name devices
# that works even if disks are added and removed. See fstab(5).
#
# <file system> <mount point>   <type>  <options>          <dump>  <pass>
proc          /proc             proc    defaults           0        0
# / was on /dev/sda1 during installation
UUID=f99eeaba-6a8c-4753-bf18-37e12e968e8c /                ext3      errors=remount
-ro 0         1
# swap was on /dev/sda5 during installation
UUID=7dbd2964-f84f-4ab5-afdf-bc2af5a94f3f none             swap      sw
0             0
/dev/scd0     /media/cdrom0     udf,iso9660 user,noauto       0        0
root@debian0ra:~#
```

na některých systémech se místo tvaru /dev/sda1 používá UUID (stálejší)



# DŮLEŽITÉ SOUBORY (!)

- **/etc/fstab**
  - Informace o filesystemech, které se do systému připojují.
  - Které diskové oblasti se připojují a kam po startu systému
  - Zkuste man fstab
- **/etc/mtab**
  - Právě v této chvíli připojené disky.

# SOUBOR /ETC/FSTAB

```
eryx2> cat /etc/fstab
# /etc/fstab: static file system information.
#
# <file sys>      <mount point>  <type>  <options>          <dump>  <pass>
/dev/md0          /              xfs      defaults            0        1
/dev/md4          /usr           xfs      defaults            0        2
/dev/md2          /var           xfs      defaults            0        2
/dev/md3          /var/cache/openafs  ext3     defaults            0        2
/dev/md1          none           swap     sw                  0        2
#/dev/hda         /mnt/cdrom     auto     ro,user,nosuid,noauto 0 0
#/dev/fd0         /mnt/floppy    auto     defaults,user,noauto 0 0
#147.228.52.152:/afs/zcu.cz/public/linux-fai/mirror/debian/i386 /mnt/mirror nfs
ro,rsiz=8192,wsiz=8192 0 0
eryx2>
```

každý fs na samostatné řádce

1. sloupec – blokové zařízení nebo remote fs, lze i LABEL nebo UUID
2. sloupec – bod připojení (mount point), pro swap none
3. sloupec – typ fs (ext2, ext3, xfs, smbfs aj., swap, auto – autom. detekovat)
4. sloupec – options (man 8 mount), viz další slide
5. sloupec – dump (zda by měl být zálohovaný, většinou 0 – ignore)
6. sloupec – pass – použít fsck (0 – přeskočit, 1 - jako první, 2 - jako další)

# VYBRANÉ OPTIONS V /ETC/FSTAB

| Příklad volby | popis  |
|---------------|--|
| atime         | Udatuje inode <b>access time</b> při přístupu k souboru (při přístupu se změní údaj, zpomaluje vytížený systém, jsou různé workarouny) |
| noatime       | Neupdatuje čas přístupu pro daný fs  |
| noexec        | Nelze přímo spouštět binární programy z tohoto fs  |
| user, nouser  | Zda může běžný uživatel připojit zařízení  |
| auto, noauto  | Zařízení bude automaticky připojeno (při bootu)  |
| ro            | Read-only  |
| rw            | Read-write   |
| defaults      | rw, suid, exec, auto, nouser, async (např.)  |

# ADRESÁŘ /BIN

- V některých distribucích je /bin symbolický link na /usr/bin
- základní uživatelské programy
  - např. **ls**, **cp**
  - příkazové interprety **sh**, **bash** ...
  - příkaz **man** pro nápovědu k příkazům (někdy v /usr/bin)
- další adresáře pro spustitelné programy:
  - /usr/bin
  - /usr/local/bin
  - /sbin , /usr/sbin

Čím se liší /bin a /sbin?

# ADRESÁŘ /SBIN

- programy pro správu systému
- mění nastavení, mohou systém poškodit
- spouštěny při startu nebo je spouští uživatel root
- `ip`, `ifconfig` - nastavení síťového rozhraní
- `mke2fs` – formát diskové oblasti na `ext2`, `ext3`, ...
  - Symbolické linky `mkfs.ext3`, `mkfs.ext4`, ...

# ADRESÁŘ /DEV

## zařízení

- se vším se zachází jako se **souborem**
    - např. porty, pevné disky, USB disky, scannery
    - např. console, tty
  - `ls -l /dev/sd* ... disky`
  - `ls -l /dev/tty* ... terminály`
  - **hlavní a vedlejší číslo zařízení**
  - hlavní .. typ
  - vedlejší .. rozlišuje zařízení stejného typu
  - bloková (b), znaková (c)
- 
- Bloková zařízení – pracujeme po blocích dat, často můžeme přeskakovat, vracet se k předchozímu bloku atp.
  - Znakové zařízení – pracujeme po jednotlivých znacích

# UKÁZKA /DEV

```
eryx2> ls -l /dev/sd*  
brw-rw---- 1 root disk 8, 0 2012-08-16 16:11 /dev/sda  
brw-rw---- 1 root disk 8, 1 2012-08-16 16:11 /dev/sda1  
brw-rw---- 1 root disk 8, 2 2012-08-16 16:11 /dev/sda2  
brw-rw---- 1 root disk 8, 5 2012-08-16 16:11 /dev/sda5  
brw-rw---- 1 root disk 8, 6 2012-08-16 16:11 /dev/sda6  
brw-rw---- 1 root disk 8, 7 2012-08-16 16:11 /dev/sda7  
brw-rw---- 1 root disk 8, 8 2012-08-16 16:11 /dev/sda8  
brw-rw---- 1 root disk 8, 16 2012-08-16 16:11 /dev/sdb  
brw-rw---- 1 root disk 8, 17 2012-08-16 16:11 /dev/sdb1  
brw-rw---- 1 root disk 8, 18 2012-08-16 16:11 /dev/sdb2  
brw-rw---- 1 root disk 8, 21 2012-08-16 16:11 /dev/sdb5  
brw-rw---- 1 root disk 8, 22 2012-08-16 16:11 /dev/sdb6  
brw-rw---- 1 root disk 8, 23 2012-08-16 16:11 /dev/sdb7  
brw-rw---- 1 root disk 8, 24 2012-08-16 16:11 /dev/sdb8  
eryx2> █
```

b znamená blokové zařízení  
červeně – hlavní číslo zařízení  
zeleně – vedlejší číslo zařízení

# ADRESÁŘ /ETC

systémové konfigurační soubory

důležitý adresář pro správu systému

- `/etc/passwd`
  - login, \* místo šif.hesla, UID, GID, poznámka (jméno už.)
  - domovský adresář, příkazový interpret
- `/etc/shadow` .. soubor se šifrovanými hesly
  - `ls -la /etc/shadow` .. práva nemá běžný uživatel
- `/etc/group` .. skupiny a kdo je jejich členem



# PŘIDÁNÍ UŽIVATELE (UKÁZKA)

```
root@debian0ra:~# adduser kokrhac
Přidávám uživatele „kokrhac“...
Přidávám novou skupinu „kokrhac“ (1002)...
Přidávám nového uživatele „kokrhac“ (1001) se skupinou „kokrhac“...
Domovský adresář „/home/kokrhac“ již existuje. Nekopíruji z „/etc/skel“.
Zadejte nové UNIX heslo:
Opakujte nové UNIX heslo:
passwd: heslo bylo úspěšně změněno
Měním informace o uživateli kokrhac
Zadejte novou hodnotu, nebo stiskněte ENTER pro použití implicitní hodnoty
    Celé jméno []: Tomas Marny
    Číslo místnosti []: UL401
    Telefon do zaměstnání []: 12345
    Telefon domů []: 54321
    Ostatní []: hokejista
Jsou informace správné? [A/n] A
root@debian0ra:~#
```

adduser kokrhac

deluser kokrhac

Můžete vyzkoušet na svém domácím Linuxu, jinak alespoň man adduser

# UKÁZKA /ETC/PASSWD

```
root@debian0ra:~# cat /etc/passwd | grep kokrhac
kokrhac:x:1001:1002:Tomas Marny,UL401,12345,54321,hokejista:/home/kokrhac:/bin/b
ash
root@debian0ra:~#
```

- uživatelské jméno
- šifrované heslo: obvykle je zde x - znamená informace je jinde 😊
- uid – číslo uživatele
- gid - skupina
- další blok od : do : .. poznámka
- domovský adresář
- shell uživatele

# /ETC

- `/etc/profile` .. vykonáván příkazovým interpretem
- `man bash`:
  - `/etc/profile`
  - `~/.bash_profile` (login shell)
  - `~/.bashrc` (nové terminálové okno např. v Gnome)
- `/etc/motd` .. uvítací zpráva  
(zobrazí se po úspěšném loginu, viz `man motd`)
- `/etc/issue` .. zobrazí se před loginem (`man issue`)

# /ETC

- **/etc/fstab** .. kam se připojují fs při startu
- **/etc/mtab** .. aktuálně připojené fs

Zkratka fs = file system = souborový systém

# ADRESÁŘ /HOME

- domovský adresář uživatelů  
(např. na vaší instalaci lokálního Linuxu)
- může být i jinde – ve škole např. na AFS  
(distribuovaný souborový systém)
- konkrétní adresář je uveden v `/etc/passwd`
- `cat /etc/passwd | grep pesicka`

# ADRESÁŘ /LIB

- sdílené knihovny pro běh programů
- **ldd** */bin/ls*
  - zobrazí potřebné knihovny pro běh programu ls
  - užitečné, pokud programu chybí nějaká knihovna

# ADRESÁŘ /TMP

- adresář pro pracovní soubory
- zvláštní chování adresáře (tzv. sticky bit)
  - Vytvořené soubory může smazat vlastník souboru nebo root
  - Nechceme, aby nám jiný uživatel mazal naše soubory
- Může být symbolický link do `/var/tmp`

# ADRESÁŘ /USR

většina programů se instaluje do /usr

- /usr/bin .. příkazy systému
- /usr/include .. hlavičkové soubory C
- /usr/lib .. knihovny
- /usr/local .. programy, které nejsou součástí distribuce OS



# ADRESÁŘ /VAR

soubory, které za běhu často mění svojí velikost

- /var/log .. logy
- /var/spool .. tiskové fronty (pamatovat!)
- /var/tmp .. tmp adresář
- /var/lock .. zámky (někdy link do /run/lock)

# ADRESÁŘ /PROC

- obsahuje informace o systému
  - adresář je virtuální – není na disku, vytváří jej jádro
  - „okno“ do systému
  - podadresáře dle PIDu procesu
- 
- `cat /proc/cpuinfo`
  - `cat /proc/version`
  - `cat /proc/meminfo`

# /ROOT, /BOOT, /OPT

- **/root**
  - Domovský adresář uživatele root
- **/boot**
  - Jádro OS, zavaděče
  - initrd – read only RAM disk (viz man initrd)
  - vmlinuz – Linux kernel
- **/opt**
  - Komerční sw, některé nestandardní balíčky

# PŘÍSTUPOVÁ PRÁVA

- `cd /bin ; ls -l cp`
- `-rwxr-xr-x 1 root root 34664 Mar 18 2002 cp`
  - 1. sloupec – `dir/file`, přístupová práva
  - 2. sloupec – počet odkazů na soubor
  - 3. sloupec – vlastník (uživatel)
  - 4. sloupec – vlastník (skupina)
  - 5. sloupec – velikost
  - dále – čas modifikace
  - název souboru

# ZMĚNA VLASTNÍKA UŽIV. SKUPINA

- **chown** *uživatel* **soubor**
  - Příkaz mění vlastníka souboru
- **chgrp** *skupina* **soubor**
  - Příkaz mění skupinového vlastníka
- **chown** *uziv.skupina* **soubor**
  - Příkaz umí změnit obojí
- vyzkoušejte si na vlastním Linuxu pod rootem 😊
- na eryxu zkuste jen `man chown`, `man chgrp`

# PŘÍSTUPOVÁ PRÁVA (CHMOD)

- r,w,x – čtení, zápis, spuštění
- s – nastav ID uživatele nebo ID skupiny pro vykonávání proces poběží s právy vlastníka souboru místo uživatele
- t – sticky bit  
soubory v adresáři může smazat jejich vlastník nebo root na /tmp adresář, aby si uživatelé nepřepisovali soubory
- **chmod** práva soubor
  - u,g,o,a – vlastník, skupina, ostatní, všichni
- chmod a+r soubor
- chmod o-w soubor

# PŘÍSTUPOVÁ PRÁVA

- `chmod ug+r program`
  - Uživateli a skupině se přidá právo read
- `chmod go-r soubor`
  - Skupině a ostatním se odebere právo read
- `chmod a=rx soubor`
  - Vlastník, skupina i ostatní budou mít jen právo číst a spustit
  - Nastaví práva na uvedenou množinu

# PŘÍSTUPOVÁ PRÁVA ČÍSELNĚ

- $(r,w,x)$  – trojice bitů 4,2,1
- $--- = 0$ ,
- $rw- = 4+2 = 6$
- $rw x = 4+2+1 = 7$
  
- Tři trojice: vlastník, skupina, ostatní
- `chmod 777 soubor.txt`
- `chmod 640 soubor2.txt`



# PŘÍSTUPOVÁ PRÁVA - ADRESÁŘ

- **r** – soubory mohou být vypsaný ls, musí být i **x**
  - **w** – soubory mohou být vytvářeny a rušeny, musí být i **x**
  - **x** – adresář může být prohledáván
- 
- **--x ..** soubory z adresáře lze použít, lze se přepnout cd, ale nelze vypsat jejich název, musíme ho znát

# STICKY BIT

sticky bit využívá typicky tmp  
adresář

- sticky bit **na adresář**
- soubory uvnitř adresáře mohou být přejmenovány a odstraněny jen **vlastníkem** souboru, vlastníkem adresáře a rootem
- ostatní Vám nebudou mazat Vaše soubory (jen root)
- vhodné pro */tmp*

# STICKY BIT - SOUBOR

- význam spíše historický
- aplikace by měla zůstat v paměti (swapu), aby ji mohl někdo další rychle spustit
- viz snaha některých sw pod Windows – při startu systému nahrát do paměti svoje knihovny, aby spuštění vlastní aplikace bylo rychlé
- Linuxové jádro ignoruje sticky bit nastavený na souborech, používá se na adresářích, např. tmp

dnes se sticky bit používá typicky jen na adresář

# PŘESMĚROVÁNÍ VSTUPU, VÝSTUPU, CHYBOVÉHO VÝSTUPU

- `ls > vystup.txt`
- `ls >> vystup.txt` (>> připojí na konec souboru)
- `more < dopis.txt`
- `ls > vystup.txt 2> chyby.txt`
- `ls 2>&1`
  - chybový výstup na standardní výstup
- Jak pomocí `ls` vygeneruji chybový výstup?
- `ls nakup.txt neni.txt`
  - `nakup.txt` bude existovat, ale `neni.txt` nebude
- `ls nakup.txt neni.txt > vystup.txt 2> chyby.txt`

# SPECIÁLNÍ ZAŘÍZENÍ

- `cp s1 /dev/null`
- `cp s2 /dev/full`
- `less < /dev/random`

zařízení `/dev/zero` .. generuje nuly

zařízení `/dev/null` .. nekonečný koš

zařízení `/dev/full` .. plný koš

# DALŠÍ PŘÍKAZY

- `cat file1 file2 file3 > spojeny_soubor`
- `head soubor`
- `tail soubor`

# LOKACE PROGRAMU A NÁPOVĚDY, DODATKY

- **whereis** ip (najde umístění příkazu/nápovery)
- **which** ip (najde umístění příkazu)
- **locate** ifconfig (často není příkaz locate k dispozici)
- **apropos** fork (hledá v nápovědě)

další důležité příkazy:

- **passwd**
- **sync**