

# Používatelia a skupiny

- používateľ (user)
  - z pohľadu systému identifikovaný číslom – UID
  - patrí do jednej primárnej skupiny a 0 alebo viac doplnkových skupín
  - špeciálny používateľ **root** s UID 0 – neobmedzené prístupové práva
- skupina (group)
  - z pohľadu systému identifikovaná číslom – GID
  - obsahuje 0 alebo viac používateľov

# Databáza používateľov

- textový súbor `/etc/passwd`
  - `username:password:uid:gid:full name:home:shell`
    - `username` – používateľské meno, malé písmená, na niektorých systémoch do 8 znakov
    - `password` – zašifrované heslo, \* = nezadateľné heslo
    - `uid, gid`
    - `full name` – plné meno, ďalšie doplňujúce informácie
    - `home` – domovský adresár používateľa
    - `shell` – shell, ktorý sa používateľovi spustí po prihlásení sa (platné shelly sú uvedené v `/etc/shells`)

# Databáza používateľov

- súbor `/etc/passwd` musí byť verejne čitateľný, aby bolo možné mapovať UID na mená (napr. pri `ls`)
- novšie systémy ukladajú do `/etc/passwd` namiesto hesiel `x` a heslá ukladajú do súboru `/etc/shadow`
- ten je čitateľný len **root-om**

# Databáza používateľov

- textový súbor `/etc/shadow`
  - username:password:last changed:min age:max age:warn before:lock after:acc. exp.:reserved
    - last changed – dátum posl. zmeny hesla (dni od 1.1.1970)
    - min age – min. počet dní pred zmenou hesla
    - max age – max. počet dní, dokedy treba zmeniť heslo
    - warn before – varovanie pred expiráciou hesla
    - lock after – počet dní po expirácii hesla, keď sa konto zablokuje
    - acc. exp. – dátum zablokovania konta

# Databáza skupín

- **textový súbor** `/etc/group`
  - `groupname:password:gid:user list`
    - `groupname` – meno skupiny
    - `password` – zašifrované heslo skupiny alebo prázdne
    - `gid` – číslo skupiny
    - `user list` – zoznam používateľov, ktorí majú túto skupinu ako doplnkovú skupinu, mená oddelené čiarkami
- **textový súbor** `/etc/gshadow`
  - `groupname:password::`

# Príklad /etc/passwd

```
root:x:0:0::/root:/bin/bash
bin:x:1:1:bin:/bin:
daemon:x:2:2:daemon:/sbin:
adm:x:3:4:adm:/var/log:
lp:x:4:7:lp:/var/spool/lpd:
sync:x:5:0:sync:/sbin:/bin/sync
shutdown:x:6:0:shutdown:/sbin:/sbin/shutdown
halt:x:7:0:halt:/sbin:/sbin/halt
mail:x:8:12:mail:/:
news:x:9:13:news:/usr/lib/news:
uucp:x:10:14:uucp:/var/spool/uucppublic:
operator:x:11:0:operator:/root:/bin/bash
games:x:12:100:games:/usr/games:
ftp:x:14:50::/home/ftp:
mysql:x:27:27:MySQL:/var/lib/mysql:/bin/bash
gdm:x:42:42:GDM:/var/state/gdm:/bin/bash
nobody:x:99:99:nobody:/:
janko:x:1000:100:Janko Hrasko,,,:/home/janko:/bin/bash
```



# Príklad /etc/group

```
root::0:root
bin::1:root,bin,daemon
daemon::2:root,bin,daemon
sys::3:root,bin,adm
adm::4:root,adm,daemon
tty::5:
disk::6:root,adm
lp::7:lp
mem::8:
kmem::9:
wheel::10:root
floppy::11:root,jerry
mail::12:mail
news::13:news
uucp::14:uucp,jerry
man::15:
games::20:
slocate:x:21:
mysql::27:
gdm::42:
ftp::50:
nobody::98:nobody
```

# Vytvorenie používateľa

- **ručne** – editáciou `/etc/passwd`, `/etc/shadow` a `/etc/group`
- `useradd` `[-c FullName]` `[-m]` `username`
  - `-m` vytvorí aj domovský adresár
  - `-g grp` – primárna skupina
  - `-G grp[, ...]` – doplnkové skupiny
  - `-s shell`, `-e YYYY-MM-DD`, `-f inact`
  - `-D` `[-g grp]` `[-b home]` `[-s shell]` `[-e expiry]` `[-f inact]`
- **skript** `adduser`

# Zrušenie a zmena používateľa

- **zrušenie:**

```
userdel [-r] username
```

- `-r` – zruší aj domovský adresár

- **zmena:**

```
usermod voľby username
```

- `-c FullName`, `-d home [-m]`, `-g grp`,  
`-G g1,g2,...`, `-l newname`, `-s shell`,  
`-e expiry_date`, `-f inact_days`
- `-L`, `-U` – lock, unlock

# Zmena hesla

- zmena hesla: `passwd username`
- zablokovanie hesla: `passwd -l username`
- odblokovanie hesla: `passwd -u username`
- ďalšie prepínače umožňujú nastavenie expiračných atribútov
- ďalšie príkazy na zmenu atribútov používateľa:
  - `chsh` (zmena shellu), `chfn` (zmena FullName), `chage` (zmena expiračných parametrov)

# Vytvorenie a zrušenie skupiny

- ručne – editáciou `/etc/group`
- vytvorenie:  
`groupadd groupname`
- zrušenie:  
`groupdel groupname`
- premenovanie:  
`groupmod -n newname oldname`

# Procesy

- každý proces má priradené
  - UID
    - real – UID používateľa, ktorý proces spustil
    - effective – UID používateľa, s ktorého právami proces beží
    - saved
  - GID
    - real, effective, saved
  - zoznam doplnkových skupín

# Prístupové práva k súborovému systému (FS)

- každý objekt FS má priradené
  - vlastníka (owner) – UID nejakého používateľa
  - skupinu (group) – GID nejakej skupiny
  - prístupové práva
    - read (r) – čítanie súboru/adresára
    - write (w) – zápis do súboru, zmena obsahu adresára (vytvorenie/premenovanie/vymazanie adresárovej položky)
    - execute/search directory – spustenie súboru, použitie adresára v ceste k inému objektu
  - pre vlastníka, pre skupinu a pre ostatných

# Prístupové práva k súborovému systému (FS)

```
jerry@jerryntb:/$ ls -l
```

```
total 80
```

```
drwxr-xr-x  2 root  bin    4096 May  4  2002 bin
drwxr-xr-x  2 root  root   4096 Dec 29 08:28 boot
drwxr-xr-x  2 root  root   4096 May  4  2002 cdrom
drwxr-xr-x  1 root  root           0 Jan  1  1970 dev
drwxr-xr-x 29 root  root   4096 Feb 16 16:10 etc
drwxr-xr-x  2 root  root   4096 May  4  2002 floppy
drwxr-xr-x  5 root  root   4096 Oct 19 13:05 home
drwxr-xr-x  2 root  root   4096 Dec 22 18:40 jet
drwxr-xr-x  3 root  root   4096 May  4  2002 lib
drwxr-xr-x  2 root  root  16384 Apr  6  2002 lost+found
drwxr-xr-x  3 root  root   4096 May  6  2002 mnt
drwxr-xr-x  5 root  root   4096 May  4  2002 old
drwxr-xr-x  3 root  root   4096 Mar 30  2002 opt
dr-xr-xr-x 73 root  root           0 Feb 16 15:05 proc
drwx--x---  3 root  root   4096 May  6  2002 root
drwxr-xr-x  2 root  bin   4096 May  4  2002 sbin
drwxrwxrwt  7 root  root   4096 Feb 16 14:09 tmp
drwxr-xr-x 17 root  root   4096 Apr  1  2002 usr
drwxr-xr-x 14 root  root   4096 Jan 24  2002 var
```

práva  
vlastník  
skupina  
typ

# Prístupové práva k súborovému systému (FS)

- ak je UID vlastníka objektu rovné effective UID procesu, použijú sa práva pre vlastníka
- inak ak je GID skupiny objektu rovné effective GID procesu alebo niektorému GID doplnkovej skupiny procesu, použijú sa práva pre skupinu
- inak sa použijú práva pre ostatných
- ak je effective UID procesu 0, pre proces neplatia obmedzenia prístupu

# Prístupové práva k súborovému systému (FS)

r w x r w x r w x

execute pre ostatných 001

write pre ostatných 002

read pre ostatných 004

execute pre skupinu 010

write pre skupinu 020

read pre skupinu 040

execute pre vlastníka 100

write pre vlastníka 200

read pre vlastníka 400

čísla sú v  
osmičkovej  
sústave

# Prístupové práva k súborovému systému (FS)

- bežný súbor
  - 640 – vlastník môže čítať a písať, skupina čítať, ostatní nič
  - 511 – všetci môžu spúšťať, vlastník aj čítať
- adresár
  - 750 – skupina môže čítať a použiť, vlastník aj meniť, ostatní nič
  - 711 – skupina a ostatní môžu len použiť, ale nie vypísať ani meniť obsah adresára

# Prístupové práva k súborovému systému (FS)

- zmena prístupových práv
  - môže len vlastník (a root)
  - `chmod práva objekt ...`
    - práva ako osmičkové číslo
  - `chmod KtoOpPrávo[, ...] objekt ...`
    - Kto: **u** – vlastník, **g** – skupina, **o** – ostatní, **a** = ugo
    - Op: **+** - pridať, **-** - odobrať, **=** - nastaviť
    - Právo: **r**, **w**, **x**
  - `chmod -R ...` - rekurzívne

# Prístupové práva k súborovému systému (FS)

chmod 640 subor

chmod 711 subor

chmod 711 subor

chmod 711 subor

chmod u=rw,g=r,o= subor

chmod u=rwx,go=x subor

chmod a=rwx,go-rw subor

chmod a=x,u+rw subor

# Prístupové práva k súborovému systému (FS)

- zmena vlastníka
  - môže len root
  - `chown [-R] vlastník[:skp] objekt ...`
    - vlastník – meno používateľa alebo UID
    - skp – meno skupiny alebo GID
- zmena skupiny
  - `chgrp [-R] skupina objekt ...`
    - skupina – meno alebo GID
  - root môže na akúkoľvek, vlastník len na skupinu, ktorej je členom

# Prístupové práva k súborovému systému (FS)

- pri vytvorení súboru alebo adresára je:
  - vlastník = effective UID procesu
  - skupina = effective GID procesu (SysV) alebo skupina rodičovského adresára (BSD)
  - práva sú určené procesom, ktorý objekt vytvára (väčšinou 666 pre súbory a 777 pre adresáre) a upravené hodnotou `umask` – práva nastavené v **umask** sa vo výsledku vypnú
    - **umask** sa dá nastaviť príkazom `umask`
    - `umask 022` – skupine a ostatným sa vypne write
    - `umask 077` – skupine a ostatným sa vypne všetko

# Prístupové práva k súborovému systému (FS)

- Niektoré programy musia mať iné práva ako má používateľ, ktorý ich spustí – napr. `passwd` musí zapisovať do `/etc/passwd`, resp. čítať a zapisovať do `/etc/shadow`.
- **set-UID atribút**
  - `chmod: 4000, u+s, ls -l: rwsr-xr-x`
  - proces bude mať saved UID a effective UID rovné UID vlastníka súboru, z ktorého sa spustí
  - proces môže svoje effective UID prepínať medzi real a saved UID

# Prístupové práva k súborovému systému (FS)

- set-GID atribút
  - `chmod: 2000, g+s, ls -l: rwxrwsr-x`
  - proces bude mať saved GID a effective GID rovné GID skupiny súboru, z ktorého je spustený
  - proces môže svoje effective GID prepínať medzi real a saved GID
- set-GID atribút na adresári (SysV)
  - novovytvorené súbory a podadresáre budú mať rovnakú skupinu ako skupina adresára a podadresáre budú mať nastavený set-GID atribút

# Prístupové práva k súborovému systému (FS)

- sticky bit na adresároch
  - `chmod: 1000, +t, ls -l: rwxrwxrwt`
  - v takomto adresári môže objekt premenovať alebo vymazať len vlastník objektu, vlastník adresára alebo root (samozrejme, len ak mu to ostatné práva dovoľujú)
  - `/tmp` má práva `1777`

# Štruktúra adresárového stromu

- /dev
  - špeciálne znakové a blokové zariadenia
- /etc
  - konfiguračné súbory
- /bin
  - základné systémové programy použiteľné aj bežným používateľom
- /sbin
  - základné systémové programy pre administráciu

# Štruktúra adresárového stromu

- /lib
  - základné systémové knižnice
- /tmp
  - adresár pre dočasné súbory
- /boot
  - kernel a ďalšie súbory pre štart systému
- /proc
  - informácie o systéme, prístup k parametrom bežiaceho systému

# Štruktúra adresárového stromu

- /var
  - meniace sa súbory (zámky (/var/lock), fronty (/var/spool), PID súbory (/var/run), logy (/var/log), mailové schránky (/var/mail, /var/spool/mail), aplikačné súbory (/var/lib/aplikácia), ...)
- /root
  - domovský adresár root-a
- /home
  - domovské adresáre používateľov

# Štruktúra adresárového stromu

- /mnt
  - adresár pre dočasné pripájanie súborových systémov
- /opt
  - adresár pre voliteľné programové subsystémy
- /usr/bin
  - väčšina používateľským programov
- /usr/sbin
  - ďalšie administratívne programy

# Štruktúra adresárového stromu

- /usr/include
  - hlavičkové súbory (.h) pre C/C++ programy
- /usr/lib
  - knižnice
- /usr/share
  - zdieľateľné súbory
- /usr/local
  - hierarchia pre lokálne doinštalovaný SW
- /usr/X11 – X Windows

# Organizácia súborového UNIXového súborového systému

- i-node – informácie o objekte súborového systému
  - typ
  - veľkosť
  - vlastníč, skupina
  - práva
  - čas posledného prístupu, modifikácie a zmeny i-node
  - počet linkov (odkazov) na objekt z adresárov
  - zoznam blokov s obsahom súboru

# Organizácia súborového UNIXového súborového systému

- adresár (directory)
  - meno
  - číslo i-node
- typy objektov súborového systému
  - normálny súbor
  - adresár (d)
  - znakové/blokové zariadenie (c/b)
  - symbolic link (l)
  - pipe/fifo (p)
  - socket (s)

# Organizácia súborového UNIXového súborového systému

- Na jeden objekt môže byť neobmedzený počet odkazov (hard link).
- Objekt súborového systému sa uvoľní, keď
  - počet odkazov z adresárov = 0 a
  - počet odkazov z tabuľky otvorených súborov = 0.
- Jednotlivé odkazy sa líšia len menom, ostatné atribúty sú zdieľané.

# Organizácia súborového UNIXového súborového systému

- Vytvorenie odkazu na objekt:
  - pri vytvorení objektu
  - príkazom `ln`:
    - `ln existujúci kam`
      - vytvorí link `kam`
    - `ln exist1 ... adresár`
      - vytvorí rovnomenné linky v určenom adresári
  - všetky odkazy sú navzájom ekvivalentné, nie je možné určiť, ktorým spôsobom vznikli
  - nemožno vytvoriť link na adresár

# Objekty súborového systému

- Symbolic link (soft link, symlink)
  - špeciálny súbor, ktorý obsahuje cestu (relatívnu alebo absolútnu) k inému objektu
  - bežné operácie okrem vymazania manipulujúce so symbolickým linkom manipulujú s príslušným objektom
  - vymazanie zruší len symbolický link
  - symbolický link nemá prístupové práva
  - **broken link** – symlink na neexistujúci objekt

# Objekty súborového systému

- Vytvorenie symlinku:
  - príkazom ln:  
`ln -s cesta kam`
    - vytvorí symlink kam obsahujúci príslušnú cestu
  - `ln -s cesta1 ... adresár`
    - vytvorí rovnomenné symlinky v určenom adresári
- Relatívna vs. absolútna cesta
  - relatívna vzhľadom k symlinku
  - absolútna začína /

# Objekty súborového systému

- Symlink vs. hard link
  - symlink môže ukazovať na adresár
  - symlink môže ukazovať do iného súborového systému
  - symlink nebráni vymazaniu objektu – neovplyvňuje počet odkazov uvedený v i-node
  - symlink môže ukazovať na neexistujúci objekt
  - symlink je možné odlíšiť od primárneho objektu

# Objekty súborového systému

- Pipe (fifo)
  - jednosmerný prostriedok komunikácie medzi procesmi
    - jeden otvorí na zápis, druhý na čítanie
  - nepomenovaná pipe – nie je objekt FS
  - vytvorenie:  
`mkfifo [-m mode] meno ...`
    - `mode` = práva ako číslo v osmičkovej sústave
    - default práva: 666

# Objekty súborového systému

- Adresár (directory)
  - vytvorenie:  
`mkdir meno ...`
  - vymazanie:  
`rmdir meno ...`
  - nie je možné vymazať adresár, ktorý nie je prázdny

# Objekty súborového systému

- Blokové a znakové zariadenia
  - väčšina zariadení (okrem sieťových interface) je reprezentovaných špeciálnym súborom
  - z pohľadu systému identifikované dvojicou čísel
    - **major number** – skupina zariadení nejakého typu
    - **minor number** – identifikuje konkrétne zariadenie
  - špeciálne súbory sa nachádzajú v **/dev**
  - vytvorenie:  
`mknod [-m mode] meno b|c major minor`

# Blokové a znakové zariadenia

- blokové zariadenia
  - základnou jednotkou je blok
  - napr. disk, CD, disketa, ramdisk
  - môžu obsahovať súborové systémy
- znakové zariadenia
  - základnou jednotkou je znak
  - napr. sériový port, terminál, konzola, tlačiareň, páska

# Vybrané znakové zariadenia

- /dev/null
  - čítanie: prázdny súbor, zápis: zahadzuje
- /dev/zero
  - čítanie: nekonečný súbor plný znakov s kódom 0
- /dev/full
  - čítanie: /dev/zero, zápis: plný disk
- /dev/random
  - výstup generátora náhodných čísel, keď sa vyčerpá entropia, zastaví sa, kým sa nedoplní

# Vybrané znakové zariadenia

- `/dev/urandom`
  - výstup generátora pseudonáhodných čísel – neblokuje sa
- `/dev/tty`
  - aktuálny riadiaci terminál procesu
- `/dev/tty1, /dev/tty2, ..., /dev/vc/1, /dev/vc/2, ...`
  - virtuálne konzoly
- `/dev/console, /dev/tty0`
  - aktuálna konzola

# Vybrané znakové zariadenia

- /dev/ttyS0, /dev/ttyS1
  - sériové porty
- /dev/lp0
  - tlačiareň na paralelnom porte
- /dev/ptmx, /dev/pts/0, ...
  - pseudoterminálové zariadenia
- /dev/tty??, /dev/pty??
  - pseudoterminálové zariadenia

# Vybrané blokové zariadenia

- /dev/hda, /dev/hdb, /dev/hdc, /dev/hdd, ...
  - IDE disky a CD-ROM
- /dev/hda1, /dev/hda2, /dev/hda3, /dev/hda4
  - primárne partitions na /dev/hda
- /dev/hda5, ...
  - logické disky na /dev/hda
- /dev/fd0
  - disketa

# Vybrané blokové zariadenia

- /dev/sda, /dev/sdb, ...
  - SCSI disky
- /dev/sr0, /dev/sr1, ...
  - SCSI CD-ROM
- /dev/sg0, /dev/sg1, ...
  - SCSI generic device
- /dev/c0t1d0s2
  - SCSI zariadenie – ctrl 0, target 1, LUN 0, slice 2

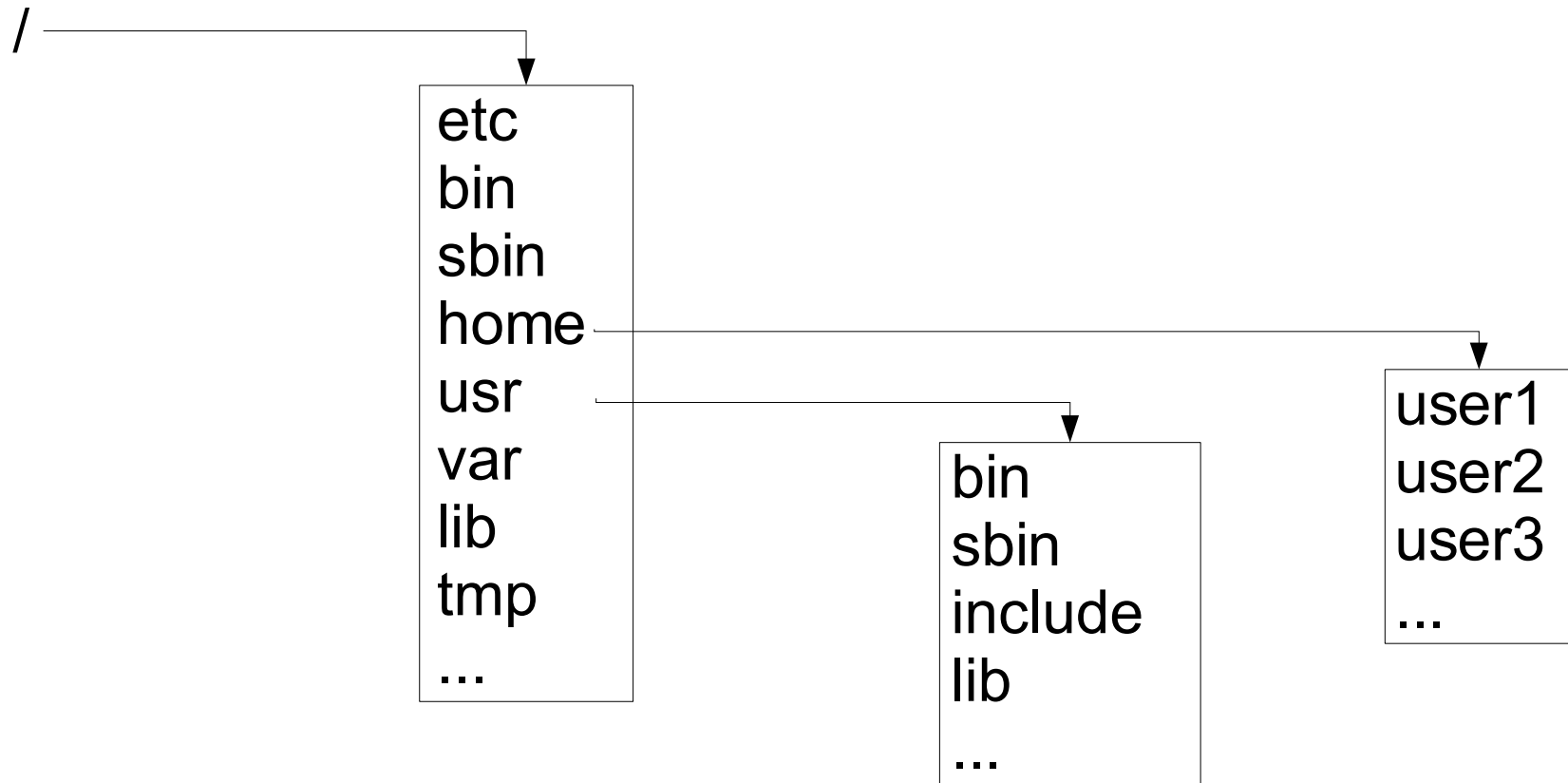
# Vybrané blokové zariadenia

- /dev/ram0, /dev/ram1, ...
  - ramdisky
- /dev/loop0, /dev/loop1, ...
  - loopback – umožňuje pristupovať k súboru ako k blokovému zariadeniu
    - `losetup /dev/loop0 súbor` – pripojenie súboru
    - `losetup -d /dev/loop0` – odpojenie súboru

# Pripájanie súborových systémov do adresárového stromu

- prístup k súborom sa realizuje prostredníctvom jedného adresárového stromu
- súborové systémy sa pripájajú na adresáre (mount point)
- po pripojení súborového systému sa obsah adresára dočasne zneprístupní – nahradí ho koreňový adresár pripojeného súborového systému

# Pripájanie súborových systémov do adresárového stromu



# Pripájanie súborových systémov do adresárového stromu

- pripojenie súborového systému (FS)

```
mount [parametre] blok_zar adr
```

– pripojí súborový systém na blokovom zariadení na uvedený adresár

- `mount -a [-t typ]` – pripojí všetky FS

uvedeného typu uvedené v `/etc/fstab`

bez atribútu `noauto`

- `umount {blok_zar|adr}` – odpojí súborový systém na uvedenom zariadení, resp. pripojený na uvedenom adresári

# Pripájanie súborových systémov do adresárového stromu

- `mount` – vypíše `/etc/mtab` – zoznam pripojených súborových systémov
  - presnejšie údaje sa dajú nájsť v `/proc/mounts`
- parametre pre `mount`:
  - `-t typ` – typ súborového systému
  - `-r, -w` – pripojiť read-only, read-write
  - `-n` – nezapisovať do `/etc/mtab`
  - `-o voľba1[, voľba2, ...]` – rôzne voľby

# Pripájanie súborových systémov do adresárového stromu

- Súbor `/etc/fstab`
  - obsahuje informácie o súborových systémoch využívané príkazom `mount`
    - `mount -a`, `mount adresár`
  - riadky začínajúce `#` sú komentáre
  - položky sú oddelené medzerami a tabulátormi
  - `blok_zar mount-point typ voľby fs_freq pass_no`
    - `fs_freq` – využívajú niektoré zálohovacie programy
    - `pass_no` – poradie kontroly FS pri štarte (0 = bez)

# Pripájanie súborových systémov do adresárového stromu

- voľby pre pripájanie FS (spoločné)
  - `sync`, `async` – synchronne/asynchronne zápisy
  - `[no]atime` – upravovať čas posl. prístupu
  - `[no]auto` – pripájať pri `mount -a`
  - `[no]dev` – FS môže obsahovať zariadenia
  - `[no]exec` – FS môže obsahovať spúšťateľné súbory
  - `[no]suid` – FS môže obsahovať set-UID a set-GID programy

# Pripájanie súborových systémov do adresárového stromu

- `[no]user` – FS môže byť pripojený aj používateľom (použitím `mount` adresár)
  - `user` implikuje `noexec,nosuid,nodev`
- `ro` – FS je read-only
- `rw` – FS je read-write
- `defaults =`  
`rw,suid,dev,exec,auto,nouser,async`
  - užitočné v `/etc/fstab`, pretože položka nemôže byť prázdna
- `remount` – zmena parametrov už pripojeného FS

# Pripájanie súborových systémov do adresárového stromu

- najbežnejšie typy FS v Linuxe
  - proc
    - nemá žiadne blokové zariadenie (uvedie na `none`)
    - interface k systémovým parametrom
    - informácie o bežiacich procesoch
    - pripája sa na `/proc`
  - devfs
    - nemá žiadne blokové zariadenie (uvedie sa `none`)
    - dynamicky generovaný FS obsahujúci blokové a znakové zariadenia
    - určený na pripojenie na `/dev`

# Pripájanie súborových systémov do adresárového stromu

## – devpts

- nemá žiadne blokové zariadenie (uvedie sa `none`)
- obsahuje slave z pseudoterminálovej dvojice, master je `/dev/ptmx`
- pripája sa na `/dev/pts`
- voľby
  - `uid=hodnota` – vlastník vzniknutých zariadení
  - `gid=hodnota` – skupina vzniknutých zariadení
  - `mode=hodnota` – práva vzniknutých zariadení

## – ext2, ext3

- štandardné FS pre Linux
- ext3 je rozšírenie s journalom, lepšia odolnosť

# Pripájanie súborových systémov do adresárového stromu

- msdos, vfat, umsdos
  - FS typu FAT (DOS, Windows)
  - msdos – mená 8+3 znaky (DOS)
  - vfat – “dlhé mená” (Windows)
  - umsdos – umožňuje ukladať objekty s UNIXovými vlastnosťami na FAT
  - dôležité voľby:
    - uid=hodnota – vlastník objektov
    - gid=hodnota – skupina objektov
    - umask=hodnota – ktoré práva budú vypnuté
    - codepage=hodnota – kódová stránka pre krátke mená – napr. 437 (US) alebo 852 (PC Latin2)
    - iocharset=hodnota – znaková sada pre konverziu dlhých mien (uložené sú v Unicode) – napr. iso8859-2

# Pripájanie súborových systémov do adresárového stromu

## – iso9660

- štandardný FS na CD-ROM
- podporuje Rock Ridge aj MS Joliet rozšírenia
- môže byť len read-only
- dôležité voľby
  - norock – nepoužiť Rock Ridge rozšírenia (tieto umožňujú objekty s UNIXovými vlastnosťami – dlhé mená, práva, symlinky, ...)
  - nojoliet – nepoužiť Microsoft Joliet rozšírenia
  - uid=hodnota – vlastník
  - gid=hodnota – skupina
  - mode=hodnota – práva (pre non Rock Ridge CD)
  - map=normal, map=off – úpravy mien (pre non Rock Ridge)

# Príklad /etc/fstab

```
/dev/hda5      swap          swap          defaults          0      0
/dev/hda3      /             ext2          defaults          1      1
/dev/hda6      /mnt/fat      vfat         uid=jerry,umask=077 1      0
/dev/cdroms/cdrom0 /cdrom       iso9660      noauto,user,ro 0      0
/dev/fd0       /floppy      vfat         noauto,user,icharset=iso8859-2 0 0
/dev/sda1     /jet        vfat         noauto,user,icharset=iso8859-2 0 0
none          /dev/pts     devpts       gid=5,mode=620    0      0
none          /proc        proc         defaults          0      0
none          /proc/bus/usb usbdevfs     defaults          0      0
```

# Príklady pre mount

- `mount -t vfat /dev/hda1 /mnt`
  - pripojí FS typu vfat z /dev/hda1 na /mnt
- `mount /cdrom`
  - pripojí FS podľa /etc/fstab na /cdrom
- `mount -o ro,remount /`
  - nastaví koreňový FS na read-only
- `mount -o loop,ro -t iso9660 /tmp/image.iso /mnt`
  - pripojí súbor na /dev/loop? a potom pripojí FS

# Vytváranie súborových systémov

- `mkfs -t typ [fs-options] blok_zar`
  - vytvorí súborový systém určeného typu pomocou volania programu  
`/sbin/mkfs.typ`
  - `mkfs.ext2` → `mke2fs`
    - `mke2fs /dev/hda1` – vytvorí ext2 FS
    - `mke2fs -j /dev/hda1` – vytvorí ext3 FS
  - `mkfs.msdos` → `mkdosfs`
    - `mkdosfs /dev/hda2` – vytvorí FAT (msdos, vfat) FS
    - `mkdosfs -F 32 /dev/hda2` – vytvorí FAT32 FS

# Štartovanie systému

- BIOS/firmware načíta a spustí boot-loader
- boot-loader načíta jadro systému – kernel, odovzdá mu prípadné parametre a spustí ho
- kernel:
  - inicializuje potrebné zariadenia
  - pripojí súborový systém na /
  - spustí `/sbin/init` ako proces 1
- init prečíta `/etc/inittab`

# Štartovanie systému

- prevádzkové úrovne – runlevels (s, 0 – 6)
  - v každej úrovni sa definujú procesy, ktoré majú bežať
  - bežné priradenie úrovní:
    - 0 – ukončiť prácu systému
    - 1, s – single user mode
    - 6 – reboot
    - 2, 3, 4 – multi user mode – význam jednotlivých runlevels sa líši
    - 5 – v Solaris-e na Sun-och = vypnúť počítač

# Štartovanie systému

- Súbor `/etc/inittab`
  - riadky začínajúce `#` sú komentáre
  - položky sú oddelené dvojbodkami
  - `id:runlevels:akcia:proces`
  - `id` – 1 – 2 znaky, jednoznačný identifikátor riadku
  - `runlevels` – ktorých úrovní sa riadok týka
  - `akcia` – čo sa má vykonať
  - `proces` – príkaz, ktorý sa má vykonať

# Štartovanie systému

- akcie:
  - **initdefault** – určuje, ktorý runlevel sa má štandardne použiť, proces sa ignoruje
  - **sysinit** – proces sa spustí pri štarte (ešte pred boot, a bootwait), počká sa na jeho skončenie, runlevels sa ignoruje
  - **bootwait** – proces sa spustí pri štarte, počká sa na dokončenie, runlevels sa ignoruje (nie všade)
  - **boot** – ako bootwait, ale nečaká sa na skončenie

# Štartovanie systému

- **wait** – proces sa spustí pri vstupe do runlevel-u a počká sa na jeho skončenie
- **once** – proces sa spustí pri vstupe do runlevel-u
- **respawn** – proces sa spustí pri vstupe do runlevel-u a po skončení sa spustí znova
- **off** – nerobí sa nič
- **ctrlaltdel** – proces sa spustí pri stlačení Ctrl+Alt+Del

# Štartovanie systému

- **powerwait** – proces sa vykoná pri výpadku napájania, čaká sa na jeho ukončenie
- **powerfail** – ako powerwait, ale nečaká sa
- **powerokwait** – proces sa vykoná pri obnove napájania
- **powerfailnow** – proces sa vykoná pri signalizácii slabej batérie záložného zdroja
- **ondemand** – vykoná sa na požiadanie – pri prechode do špeciálnych runlevel-ov A, B, C, skutočný runlevel sa nezmení

# Štartovanie systému

- Zmena runlevel-u sa dá vyvolať:
  - príkazom `telinit novy_runlevel`
  - príkazom `shutdown`
    - `-h = 0`
    - `-r = 6`
    - inak 1
- Pri zmene runlevel-u `init` zruší procesy, ktoré spustil, a ktoré nemajú bežať v novom runlevel-i

# Štruktúra štartovacích skriptov

- Slackware Linux
  - adresár /etc/rc.d:
    - rc.S spúšťa sa ako sysinit podľa inittab
    - rc.K spúšťa sa pri vstupe do single user mode
    - rc.M spúšťa sa pri vstupe do multi user mode
    - rc.inet1, rc.inet2 inicializujú sieť a niektoré služby
    - rc.local je určený pre spúšťanie doinštalovaných subsystemov
    - rc.sshd, rc.httpd, rc.nfsd, ... spúšťajú príslušné subsystemy z distribúcie

# Štruktúra štartovacích skriptov

- Solaris, Debian Linux, a mnohé iné:
  - adresár `/etc/init.d` obsahuje skripty pre jednotlivé subsystemy, ktoré akceptujú parameter `start`, `stop`
  - adresáre `/etc/rcX.d` pre každý runlevel `X` obsahujú linky tvaru `SnnMeno` a `KnnMeno`, ktoré sa spustia pri vstupe do runlevelu:
    - `nn` = priorita (poradie)
    - `Meno` = názov subsystemu
    - `S` – skript sa spustí s parametrom `start`
    - `K` – skript sa spustí s parametrom `stop`

# Dependency based booting

- usporiadanie poradia štartovacích skriptov automaticky na základe vzájomných závislostí
  - komentáre v hlavičke skriptu
- `inserv skript`
  - pridá skript medzi automaticky spúšťané
- `inserv -r skript`
  - odstráni skript z automaticky spúšťaných

# Dependency based booting

```
### BEGIN INIT INFO
# Provides:          boot_facility_1 [ boot_facility_2 ...]
# Required-Start:   boot_facility_1 [ boot_facility_2 ...]
# Required-Stop:    boot_facility_1 [ boot_facility_2 ...]
# Should-Start:     boot_facility_1 [ boot_facility_2 ...]
# Should-Stop:      boot_facility_1 [ boot_facility_2 ...]
# X-Start-Before:   boot_facility_1 [ boot_facility_2 ...]
# X-Stop-After:     boot_facility_1 [ boot_facility_2 ...]
# Default-Start:    run_level_1 [ run_level_2 ...]
# Default-Stop:     run_level_1 [ run_level_2 ...]
# X-Interactive:    true
# Short-Description: single_line_description
# Description:      multiline_description
### END INIT INFO
```

# Ukončenie práce systému

- príkaz shutdown:

```
shutdown [-t sec] [-arkhc] kedy [správa]
```

- `sec` - čas v sekundách medzi výzvou procesu na skončenie a jeho zrušením
- `-a` – skontrolovať, či je na konzole prihlásený oprávnený používateľ (uvedený v `/etc/shutdown.allow`)
- `-r` – reboot, `-h` – halt, `-k` – len poslať správu
- `-c` – cancel – zrušiť bežiaci shutdown
- `správa` – vypíše sa všetkým prihláseným

# Ukončenie práce systému

- určenie času pre shutdown:
  - `now` – hneď
  - `hh:mm` – v uvedenom čase
  - `+m` – o m minút
- ak nie je `-h` ani `-r`, prejde do runlevel-u 1
- na niektorých systémoch (napr. Solaris):
  - `shutdown [-g sec] [-i runlevel] [správa]`

# Plánovanie úloh

- plánovanie nepravidelných úloh – **at**
- `at čas`
  - načíta zo štandardného vstupu príkazy na vykonanie (ak ide vstup z terminálu, ukončíte ho `Ctrl+D`)
  - v zadanom čase tieto príkazy vykoná použitím `/bin/sh` a výstup pošle používateľovi mailom

# Plánovanie úloh

– špecifikácia času pre at:

- now [+ inkrement]
- čas [+ inkrement]
- čas dátum
- čas = hh:mm | noon | midnight | teatime (16:00)
- dátum = meno\_mesiaca deň [, rok] |  
deň\_v\_týždni | today | tomorrow |  
MM/DD/YYYY | YYYY-MM-DD |  
DD.MM[.YYYY] | DD mmm [YYYY]
- inkrement = počet minutes|hours|days|weeks|months  
next day|week|month|deň\_v\_týždni

# Plánovanie úloh

- príklady:
  - at now + 2 hours – o 2 hodiny
  - at noon tomorrow – zajtra o 12:00
  - at 16:00 next monday – budúci pondelok o 16:00
  - at 23:00 2003-04-15 – 15.4.2003 o 23:00
  - at 23:00 04/15/2003 – 15.4.2003 o 23:00
  - at 23:00 15.4.2003 – 15.4.2003 o 23:00

# Plánovanie úloh

- vypísanie zoznamu naplánovaných úloh
  - `atq`
- zrušenie naplánovanej úlohy
  - `atrm číslo_úlohy`
- plánovanie úlohy na čas, keď bude systém málo vyťažený
  - `batch`
    - načíta príkazy zo štandardného vstupu a vykoná, keď vyťaženie klesne pod 0.8

# Plánovanie úloh

- root môže používať at v každom prípade
- oprávnenie používať systém **at** inými používateľmi je určené dvojicou súborov:
  - ak existuje `/etc/at.allow`, at môžu používať len používatelia, ktorí sú v ňom uvedení
  - inak ak existuje `/etc/at.deny`, at môžu používať všetci, ktorí v ňom nie sú uvedení
  - inak môže at používať len root
- `atrm` môže použiť vlastník úlohy alebo root

# Plánovanie úloh

- plánovanie pravidelných úloh – **cron**
- vykonávanie pravidelných úloh je riadené tabuľkou s položkami oddelenými medzerami
- min hod deň mesiac deň\_v\_týždni príkaz
- položka môže obsahovať číslo, rozsah (2-6), viac čísel alebo rozsahov oddelených čiarkou (2-6,8), \* (akákoľvek hodnota)
- príkaz sa vykoná v každú minútu, ktorá zodpovedá uvedeným hodnotám

# Plánovanie úloh

- príklad časovej špecifikácie pre cron:
  - 10 6 \* \* \* = každý deň o 6:10
  - 10 6 4 \* \* = vždy 4. v mesiaci o 6:10
  - 10 6 \* 4 0 = každú nedeľu v apríli o 6:10
    - 0=nedeľa, 1=pondelok, ..., 6=sobota
  - \* 6 \* \* \* = každú minútu 6. hodiny
  - 10,20 4 \* \* \* = 4:10 a 4:20
  - \* 7-9 \* \* \* = každú minútu 7., 8. a 9. hodiny
  - \*/2 5 \* \* \* = každú druhú minútu 5. hodiny

# Plánovanie úloh

- **zadanie tabuľky systému cron**
  - `crontab súbor [-u username]`
- **editácia existujúcej tabuľky**
  - `crontab -e [username]`
- **vypísanie aktuálnej tabuľky**
  - `crontab -l [username]`
- **vymazanie aktuálnej tabuľky**
  - `crontab -d [username]`

# Plánovanie úloh

- o vykonávanie úloh systému cron sa stará proces `crond`
- tabuľky sa ukladajú v `/var/spool/cron/crontabs`
- o vykonávanie úloh systému `at` a `batch` sa stará
  - buď samostatný proces – `atd`
  - alebo `atrun` spúšťaný pravidelne z cron-u

# Sieť

- najpoužívanejším prostriedkom sieťovej komunikácie v UNIXových systémoch (a dnes už aj vo všeobecnosti) je rodina protokolov TCP/IP
- systém sa k sieti pripája prostredníctvom sieťových interface-ov
  - so stálym spojením – napr. sieťové karty
    - Linux: eth0, Solaris: le0, hme0
  - s dial-up spojením – napr. logický interface pre spojenie protokolom PPP cez modem (ppp0)

# Základy TCP/IP

- na sieťovej vrstve sa používa protokol IP
  - connection less, unreliable
  - zabezpečuje prenos paketov medzi ľubovoľnými dvoma počítačmi v sieti
  - adresy – 4B čísla (1.2.3.4)
  - IP adresa sa skladá z adresy siete a čísla počítača v rámci siete
  - maska určuje, aká časť adresy je adresa siete
    - 4B, najprv samé bin. 1 (sieť), potom 0 (počítač)
    - 255.255.254.0: prvých 23 bitov je adresa siete

# Základy TCP/IP

- namiesto masky sa niekedy uvádza len počet bitov, ktoré sú tvoria adresu siete
- špeciálne IP adresy:
  - 127.0.0.0/8 – loopback (interface lo má zvyčajne adresu 127.0.0.1) – komunikácia v rámci zariadenia
  - číslo počítača = 0 – adresa siete
  - číslo počítača = 1...1 – broadcast – určené všetkým počítačom v určenej sieti
  - 255.255.255.255 – broadcast v lokálnej sieti
  - 224.x.x.x – 239.x.x.x – multicast
  - 240.x.x.x – 255.x.x.x – rezervované
  - 192.168.0.0/16, 172.16.0.0/12, 10.0.0.0/8 – súkromné

# Základy TCP/IP

## – příklady

- 158.195.18.0/255.255.255.0 (24)
  - adresy: 158.195.18.1 – 158.195.18.254
  - broadcast: 158.195.18.255, adr. siete: 158.195.18.0
- 158.195.22.0/255.255.255.128 (25)
  - adresy: 158.195.22.1 – 158.195.22.126
  - broadcast: 158.195.22.127, adr. siete: 158.195.22.0
- 158.195.22.128/255.255.255.128 (25)
  - adresy: 158.195.22.129 – 158.195.22.254
  - broadcast: 158.195.22.255, adr. siete: 158.195.22.128
- 158.195.16.0/255.255.254.0 (23)
  - adresy: 158.195.16.1 – 158.195.17.254
  - broadcast: 158.195.17.255, adr. siete: 158.195.16.0

# Základy TCP/IP

- na linkovej a fyzickej vrstve sa používajú protokoly závislé od interface-u
  - pri sieťach typu broadcast (napr. Ethernet) sa často využíva subsystém ARP (address resolution protokol) na mapovanie IP adresy na fyzickú adresu
  - priamo môžu komunikovať len zariadenia pripojené v jednej sieti na fyzickej alebo linkovej vrstve
  - inak sa komunikuje prostredníctvom **router-ov**

# Základy TCP/IP

- na transportnej vrstve sa používajú protokoly
  - TCP (connection oriented, reliable)
  - UDP (connection less, unreliable)
  - obidva pridávajú k IP adresám zdroja a cieľa ešte čísla **portov** na zdrojovom a cieľovom zariadení
  - štvorica IP adresa zdroja a cieľa a port zdroja a cieľa jednoznačne identifikuje komunikáciu

# Základy TCP/IP

- routovacia tabuľka
  - určuje podľa IP adresy cieľa, kadiaľ sa má odoslať IP paket
  - adresa siete, maska, adr. routera, interface
  - zotriedená podľa masky (od najšpecifickejšej po najvšeobecnejšiu)
  - vyberie sa prvý riadok, ktorý zodpovedá
    - ak nemá uvedený router, pošle cez určený interface priamo
    - ak má uvedený router, pošle cez neho

# Základy TCP/IP

– příklad routovacej tabuľky

- 158.195.18.0 255.255.255.0 eth0
- 127.0.0.0 255.0.0.0 lo
- 0.0.0.0 0.0.0.0 158.195.18.209 eth0

– příklad routovacej tabuľky pri 2 sieťových int.

- 158.195.18.0 255.255.255.0 eth0
- 158.195.16.0 255.255.254.0 eth1
- 127.0.0.0 255.0.0.0 lo
- 0.0.0.0 0.0.0.0 158.195.16.208 eth1

# Konfigurácia TCP/IP

- konfigurácia sieťových interface-ov
  - `ifconfig [interface]` – vypíše nastavenie určeného alebo všetkých aktívnych interface-ov
  - `ifconfig -a` – vypíše nastavenie všetkých int.
  - `ifconfig iface adresa | voľby ...`  
nastaví interface
  - najbežnejšie voľby
    - `netmask maska` – určí masku
    - `broadcast adresa` – určí broadcastovú adresu

# Konfigurácia TCP/IP

- `up` – zapne interface
- `down` – vypne interface
- `pointopoint` `adresa` – nastaví adresu druhej strany pre point-to-point spojenie
- `[-]` `arp` – zapne/vypne ARP subsystém
- `hw` `adresa` – nastaví HW adresu interface-u

## – príklady

- `ifconfig` `iface` `<IP>` `netmask` `<maska>` `broadcast` `<bcast>`  
**nastaví interface**
- `ifconfig` `iface` `<IP>` `netmask` `255.255.255.255`  
`pointopoint` `<IP druhej strany>` **nastaví interface pre point-to-point spojenie**

# Konfigurácia TCP/IP

- manipulácia s routovacou tabuľkou
  - `route [-n]` – vypíše routovaciu tabuľku (s -n sa nesnaží prevádzať IP adresy na mená)
  - `route add -net cieľ netmask maska [gw router] [dev iface]` – pridá záznam pre určitú sieť do tabuľky
  - `route add -host cieľ [gw router] [dev iface]` pridá záznam pre určenú adresu
  - `route add default [gw router] [dev iface]` pridá default (0.0.0.0/0) route do tabuľky

# Konfigurácia TCP/IP

- `route del [-host|-net] cieľ [netmask maska] [gw router] [dev iface]` **zruší záznam z tabuľky**

- **zapnutie routovania v Linuxe**

- je potrebné zapísať 1 do `/proc/sys/net/ipv4/ip_forward`

- **priradenie viac IP adries jednému interface**

- `ifconfig eth0:1 ...` – vytvorí interface `eth0:1` a nastaví ho

# Konfigurácia siete v Debiane

- /etc/network/interfaces
  - auto
    - medzerami oddelený zoznam interfacov, ktoré majú byť automaticky nakonfigurované
  - allow-hotplug
    - zoznam interfacov, ktoré majú byť automaticky nakonfigurované pri pripojení (vzniku)
  - *iface meno-interface rodina-protokolov metóda parametre*
    - konfigurácia konkrétneho interface-u

# Konfigurácia siete v Debiane

- metódy pre rodinu *inet*
  - loopback
    - pre lo
  - dhcp
    - získa konfiguráciu protokolom DHCP
  - static
    - address
    - netmask
    - broadcast
    - gateway

# Konfigurácia siete v Debiane

- spoločné parametre
  - pre-up *príkaz*
  - up | post-up *príkaz*
  - down | pre-down *príkaz*
  - post-down *príkaz*
- skripty v `/etc/network/if-param.d/`
  - vykonajú sa po vykonaní priamo uvedeného príkazu

# Konfigurácia siete v Debiane

- `ifup meno-interface-u`
  - nakonfiguruje zadaný interface
  - `-a` – všetky „auto“ interface-y
- `ifdown meno-interface-u`
  - odkonfiguruje zadaný interface
  - `-a` – všetky „auto“ interface-y

# Sieťové služby

- v súbore `/etc/services` je priradenie názvov služieb číslam portov a protokolov
  - meno      číslo portu/protokol      alias
- mená protokolov sa na ich čísla mapujú pomocou súboru `/etc/protocols`
- programy poskytujúce sieťové služby (démony, servery) buď
  - počúvajú na príslušnom porte – čakajú na požiadavky, alebo
  - ich spúšťa “superserver” `inetd`

# Siet'ové služby

- **inetd** sa riadi súborom `/etc/inetd.conf`
  - medzerami oddelené položky
  - meno typ protokol wait/nowait user prog args
    - meno – názov podľa `/etc/services`
    - typ – **dgram** alebo **stream**
    - protokol – **udp** alebo **tcp**
    - wait = inetd počká na skončenie procesu, kým bude spracovávať ďalšie požiadavky na danom porte
    - nowait = inetd pri ďalšej požiadavke na porte spustí ďalšiu inštanciu programu
    - user – užívateľ, s ktorého UID sa má program spustiť

# Sieťové služby

- prog – cesta k súboru, ktorý sa má spustiť
- args – argumenty (vrátane nultého – meno programu)
- na spúšťanie serverov z inetd sa často využíva program tcpd, ktorý robí kontrolu prístupu k službám
  - riadi sa súbormi `/etc/hosts.allow` a `/etc/hosts.deny`
  - najprv sa konzultuje `hosts.allow`, ak sa nenájde zhoda, tak `hosts.deny`, ak sa nenájde zhoda ani tam, spojenie sa povolí

# Sieťové služby

- štruktúra `/etc/hosts.{allow,deny}`
  - zoznam služieb : zoznam klientov
  - služba:
    - meno[@počítač]
    - ALL
  - klient:
    - ALL
    - [meno@]počítač
  - počítač:
    - IP adresa, meno
    - IP adresa/maska
    - 158.195.
    - .fmph.uniba.sk

# Sieťové služby

- príkaz `netstat`
  - výpis sieťových spojení a otvorených portov, routovacej tabuľky, štatistík, ...
  - najbežnejšie parametre
    - `-n` – IP adresy a čísla portov vypisuje číselne (inak sa snaží konvertovať na meno)
    - `-a` – výpis prebiehajúcich spojení aj otvorených portov čakajúcich na spojenie alebo datagram (inak len prebiehajúce)
    - `-r` – výpis routovacej tabuľky
    - `-p` – výpíše aj číslo príslušného procesu

# IP adresy vs. mená

- **súbor** `/etc/hosts`
  - pôvodné riešenie mapovania medzi IP adresami a menami počítačov
  - medzerami oddelené položky, # uvádza komentár
  - IP\_adresa meno aliasy
  - problém s pridelovaním mien, distribúciou a aktualizáciou údajov vo veľkých sieťach
  - použiteľné v malých sieťach

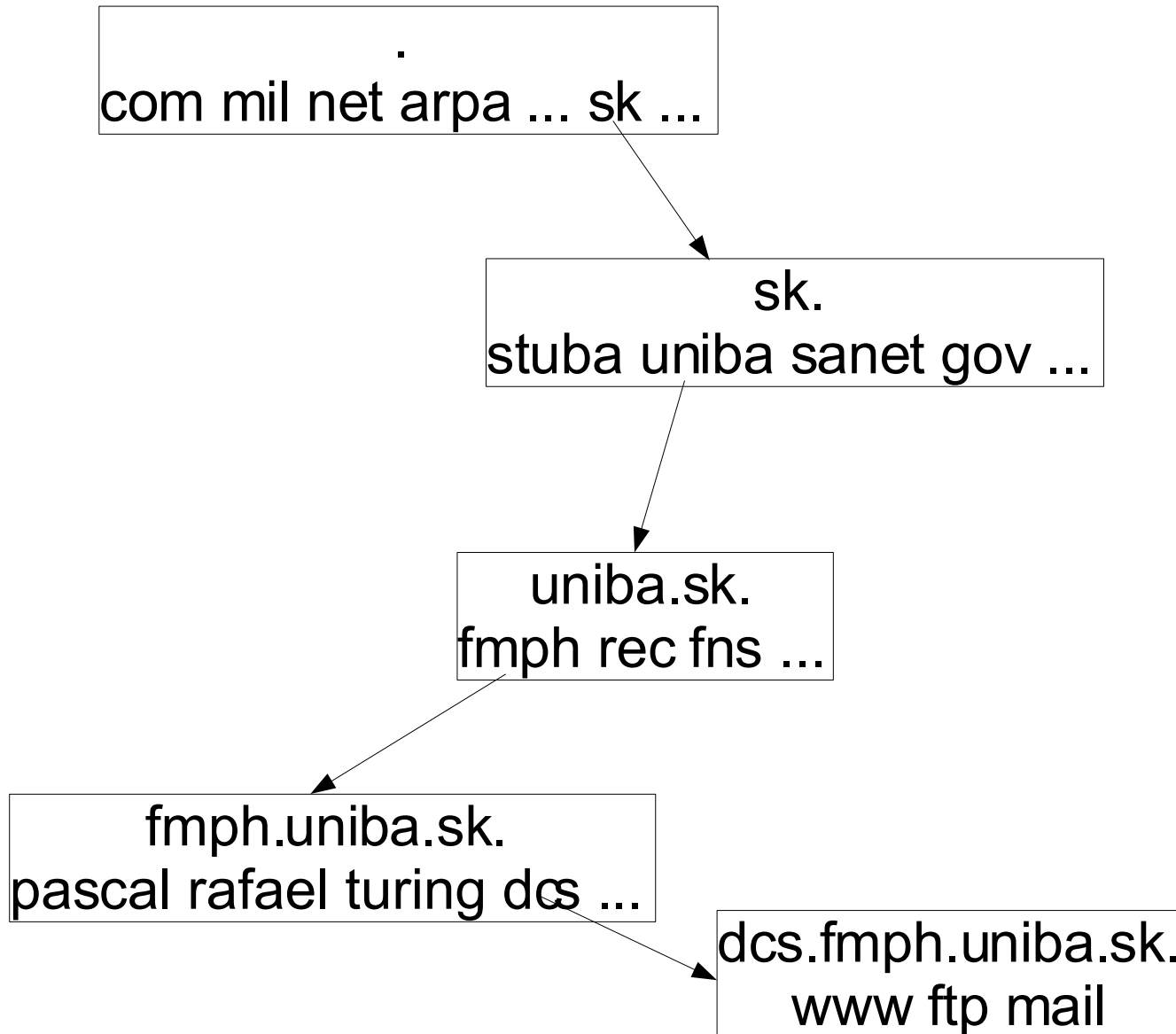
# Príklad /etc/hosts

```
127.0.0.1      localhost
158.195.16.200  cyril.fmph.uniba.sk      cyril
158.195.87.234  jj-ntb.dcs.fmph.uniba.skjj-ntb
158.195.18.163  public.dcs.fmph.uniba.skpublic mail
```

# Domain Name System (DNS)

- DNS – najväčšia distribuovaná databáza na prevod medzi doménovými menami a IP adresami
- doménové meno:
  - meno.doména\_n.doména\_n-1. ... .doména\_1
  - nič nehovorí o fyzickom umiestnení počítača
  - domény 1. (najvyššej úrovne)
    - generické: com, org, net, edu, gov, mil, int, biz, info, pro
    - podľa krajín: sk, cz, at, pl, hu, de, uk, ...
- informácie poskytujú DNS servery

# Domain Name System (DNS)



# Domain Name System (DNS)

- doména – každý uzol doménového stromu
- zóna – spojená časť doménového stromu, o ktorej má name server autoritatívne informácie
  - celá doména
  - časť domény – obsahuje všetky doménové mená okrem tých, ktoré sú v podstromoch, ktoré sú delegované do inej zóny (cez NS záznam)
- DNS server
  - primárny – má informácie o zóne v súbore
  - sekundárny – informácie berie z iného DNS serv.

# Domain Name System (DNS)

- formát zónových súborov:
  - \$ORIGIN doména – špecifikuje doménu, ktorá sa pripája k relatívnym doménovým menám
    - absolútne doménové meno v zónovom súbore vždy končí bodkou
    - relatívne doménové meno – nekončí bodkou
  - \$TTL číslo – určuje time-to-live v sekundách pre záznamy, ktoré ho explicitne neurčujú
    - ak nie je, berie sa TTL z predchádzajúceho záznamu, ktorý ho uviedol

# Domain Name System (DNS)

- doména [TTL] class type data
  - class = `IN` (Internet)
  - doména – doménové meno (rel. alebo abs.), ku ktorému sa záznam vzťahuje
    - `@` - zóna sama
    - medzera – to isté ako v predchádzajúcom zázname
  - type – typ záznamu
  - data – hodnota(y) záznamu
    - závisí od typu
  - TTL – čas, ako dlho sa záznam môže uchovávať v cache-i

# Domain Name System (DNS)

- typy záznamov v DNS

- SOA `name_serv user.host (serial refresh retry expire neg_ttl)`

- informácie o zóne, 1. záznam v súbore
    - `name_serv` – meno primárneho DNS servera
    - `user.host` – e-mailová adresa správcu zóny (. namiesto @)
    - `serial` – sériové číslo – pri každej zmene zóny sa musí zväčšiť
    - `refresh` – čas medzi obnovou údajov na sekundárnych DNS serveroch

# Domain Name System (DNS)

- `retry` – pri neúspešnom refresh-i sa skúša znova po tomto čase
  - `expire` – ak sa nepodarí refresh ani do tohto času, informácie o zóne sa považujú za neaktuálne (a ďalej sa neposkytujú)
  - `neg_ttl` – TTL pre negatívne odpovede (neexistujúca doména)
- `A IP_adresa`
    - priradenie IP adresy k menu
  - `CNAME dom_meno`
    - priradenie iného doménového mena (aliasu)

# Domain Name System (DNS)

- NS `meno.name.servera`
  - odkazuje (deleguje) na iný DNS server
- MX `priorita meno`
  - určenie mail-exchanger-a – počítača, ktorý zabezpečuje príjem mailov pre dané doménové meno
  - pri posielaní e-mailu sa skúšajú od najnižšej hodnoty priority
- PTR `meno`
  - používa sa pri mapovaní z IP adresy na doménové meno
  - meno zvyčajne absolútne

# Domain Name System (DNS)

- mapovanie IP adresy na meno
  - slúži na to doména `in-addr.arpa`.
  - pre adresu A.B.C.D vyhľadáme PTR záznam pre `D.C.B.A.in-addr.arpa`.
  - nie je zabezpečená konzistencia – mapovania z mena na IP adresu a opačne sú na sebe nezávislé
    - nie je možné sa veľmi spoliehať na prevod z IP adresy na meno pre účely autentifikácie

# Príklad zónových súborov

```
$ORIGIN dcs.fmph.uniba.sk.  
$TTL 36000  
@      IN  SOA ns.dcs.fmph.uniba.sk. hostmaster.dcs.fmph.uniba.sk (  
        2003042901  
        7200  
        1800  
        86400  
        600 )  
      IN  NS  ns.dcs.fmph.uniba.sk.  
      IN  MX  50 mail  
      IN  MX  100 cyril.fmph.uniba.sk.  
ns     IN  A   158.195.18.163  
mail   IN  A   158.195.18.163  
public IN  A   158.195.18.113  
www    IN  CNAME public
```

```
-----  
$ORIGIN fmph.uniba.sk.  
dcs     IN  NS  ns.dcs.fmph.uniba.sk.  
ns.dcs  IN  A   158.195.18.163
```

```
-----  
$ORIGIN 18.195.158.in-addr.arpa.  
@      IN  SOA ...  
163    IN  PTR public.dcs.fmph.uniba.sk.
```

```
$ORIGIN 195.158.in-addr.arpa.  
18     IN  NS  ns.dcs.fmph.uniba.sk.
```

# Konfigurácia DNS

- **súbor** `/etc/resolv.conf`
  - obsahuje nastavenie DNS resolvera (klienta)
  - `domain` lokálna\_doména
  - `search` dom1 dom2 ...
    - `domain` a `search` sa navzájom vylučujú
    - určujú domény, v ktorých sa skúšajú hľadať neúplné mená
  - `nameserver` IP\_adresa
    - určuje DNS server, ktorého sa resolver bude pýtať
    - max. 3 záznamy

# Konfigurácia DNS

- **súbor** `/etc/nsswitch.conf`
  - na mnohých systémoch určuje, z akých zdrojov sa berú informácie pre mapovanie medzi menami a číslami – aj medzi menami počítačov a IP adresami
  - `hosts: files dns`
    - najprv `/etc/hosts`, potom DNS
- **konfigurácia pre name server BIND**
  - **súbor** `/etc/named.conf`

# Network Filesystem (NFS)

- štandardný spôsob zdieľania súborov v UNIX-ových systémoch založený na RPC
- prenáša aj informácie o vlastníkoch, skupinách a právach
  - podmienkou správneho fungovania je spoločná databáza používateľov a skupín
  - z bezpečnostného hľadiska možno využiť len v “priateľskom” prostredí
- potrebuje program `portmap`
  - poskytuje info o portoch pre RPC programy

# Network Filesystem (NFS)

- pripojenie disku
  - `mount -t nfs server:cesta kam`
    - pripojí adresár `cesta` na serveri `server` do stromu na adresár `kam`
- parametre špecifické pre NFS
  - `fg/bg` – ak sa nepodarí pripojiť súborový systém, `mount` sa bude pokúšať ďalej (`fg`), alebo sa bude pokúšať ďalej na pozadí (`bg`)
  - `noac` – zakáže cache atribútov súborov

# Network Filesystem (NFS)

- `hard` – ak server prestane odpovedať, proces prístupujúci k súborovému systému sa zastaví a bude čakať na obnovenie spojenia
- `soft` – ak server prestane odpovedať, prístup k súborovému systému skončí chybou
- NFS server
  - normálny proces, alebo súčasť kernelu
  - `rpc.mountd` – zabezpečuje pripájanie FS
  - `rpc.nfsd` – zabezpečuje súborové operácie
  - `rpc.statd`, `rpc.lockd` – zabezpečujú uzamykanie súborov

# Network Filesystem (NFS)

- **súbor** `/etc/exports`
  - kontroluje prístup cez NFS
  - každý riadok obsahuje exportovaný adresár a zoznam klientov, ktorí k nemu môžu pristupovať spolu s voľbami:  
`/home client1(rw) client2(ro,all_squash)`
  - klient môže byť
    - meno, IP adresa, IP adresa/maska
    - meno obsahujúce \* a ?

# Network Filesystem (NFS)

- voľby
  - `ro` – adresár je sprístupnený len na čítanie
  - `rw` – adresár je sprístupnený aj na zápis
  - `root_squash` – prístupy od používateľa 0 sa robia s právami anonymného používateľa
  - `no_root_squash` – prístupy od používateľa 0 sa realizujú s právami root-a
  - `all_squash` – všetky prístupy sa robia s právami anonymného používateľa
  - `sync` – robí synchrónne zápisy na disk

# Network Filesystem (NFS)

- novšie Linux-ové systémy majú NFS server v kerneli
  - komunikuje sa s ním pomocou príkazu `exportfs`
  - `exportfs -r`
    - exportuje podľa `/etc/exports`
- `rpc.nfsd n`
  - `n` = počet threadov, štandardne sa používa 8
  - ak je NFS server v kerneli, `rpc.nfsd` býva len malý program na jeho aktiváciu

# Ďalšie služby

- **syslogd**
  - centrálna správa logov
  - /etc/syslog.conf
- **lpd**
  - správa tlačových front
  - úlohy sa zadávajú príkazmi lpr, resp. lp
- **sendmail**
  - mailový server

# Tlačové systémy

- lpr, lpr-ng
  - popis tlačiarní v /etc/printcap, konfigurácia v /etc/lpd.conf
  - démon lpd
  - lpr, lp – zadávanie úloh
  - lpq, lpstat – zobrazenie stavu a fronty
  - lprm, cancel – rušenie úlohy
  - v /etc/printcap môžu byť definované filtre na konverziu formátu a iné činnosti

# Tlačové systémy

- CUPS (Common UNIX Printing System)
  - administrácia cez web (port 631)
  - podporuje IPP, lpd, inštaláciu windows driverov pre tlač cez sambu
  - démon cupsd
  - konfigurácia v /etc/cups/
  - bohaté možnosti filtrov, driverov tlačiarň a spôsobov pripojenia

# Samba

- implementácia protokolov používaných vo svete MS Windows na zdieľanie súborov a tlačiarňí
- server
  - smbd – poskytuje prístup k súborom a tlačiarňam, autentifikáciu, ...
  - nmbd – zabezpečuje funkcie na preklad Windows mien na IP adresy
  - swat – web rozhranie na konfiguráciu

# Samba

- klient
  - smbclient – program na prístup k súborom a tlačiarňam v štýle ftp
  - nmblookup – nástroj na prístup k službám nmbd
  - súborové systémy smbfs a cifs
    - pripojenie Windows adresárov do UNIX-ového stromu
    - smbmount, smbmount – nástroje na mount/umount aj pre používateľov