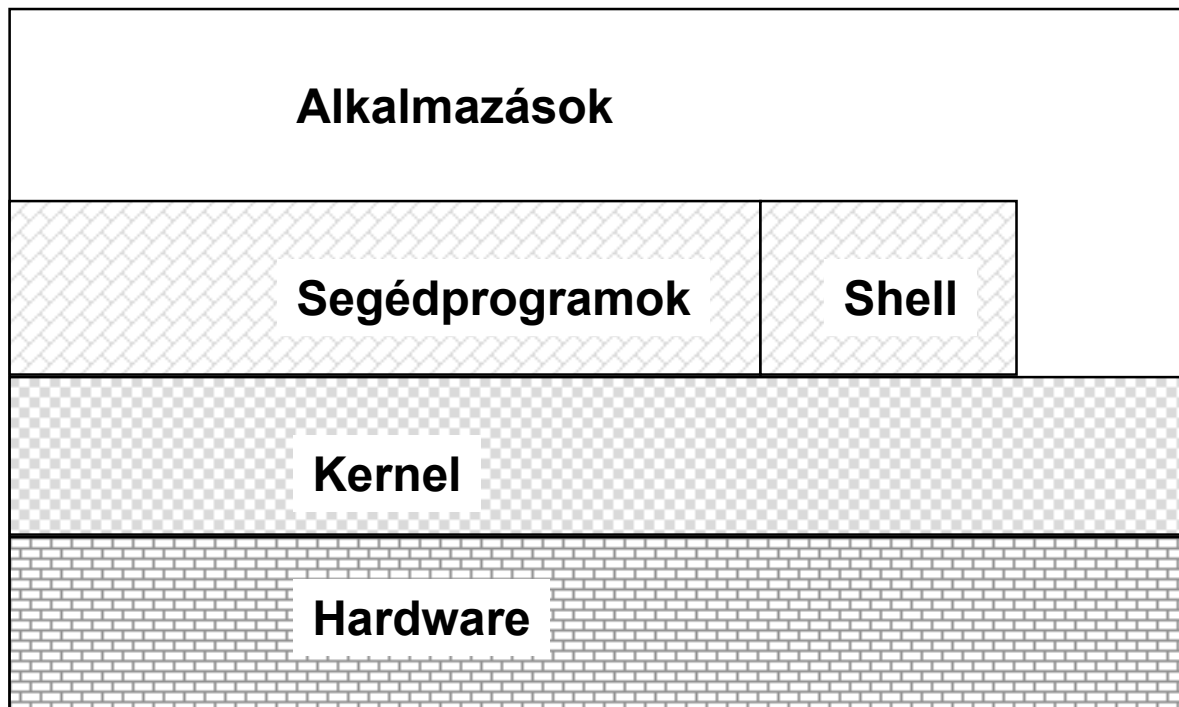


# Unix történet

- ◆ **eredetileg két vonal: System V - BSD**
  - jelenleg összemosódik
- ◆ **megvalósítások gyártónként különböznek**
  - SUN - Solaris
  - IBM - AIX
  - HP - HPUX
  - Compaq - Tru64 UNIX
  - Silicon Graphics - IRIX
  - ingyenes (főleg i386): Linux, (Free) (Net) (Open) BSD
- ◆ **a Unix tulajdonságai**
  - multitasking, multiuser
  - segédprogramok sokasága
  - shell
- ◆ **POSIX**
  - rendszerhívások
  - shell
  - segédprogramok

# A Unix felépítése



# Felhasználók

## ◆ Felhasználók

- password
- login shell
- UID
- home directory
- ulimit
- root, bin, sys, mail, nobody, lpd

## ◆ Csoportok

- GID
- tagok felhasználók
  - » elsődleges csoport
  - » másodlagos csoportok
- root, bin, sys, mail, nobody

# Unix felhasználói felületek

## ◆ soros terminál

- karakteres, soros felület, vezérlő karakterek definiálhatók, echo
- `stty -a`
  - » erase (backspace) `Ctrl-H`
  - » eof `Ctrl-D`
  - » intr (break) `Ctrl-C`
  - » susp (suspend) `Ctrl-Z`
  - » stop (pause) `Ctrl-S`
  - » start `Ctrl-Q`
- üzemmódok: cooked (sört értelmez), raw (nincs értelmezés)

## ◆ X-Window rendszer

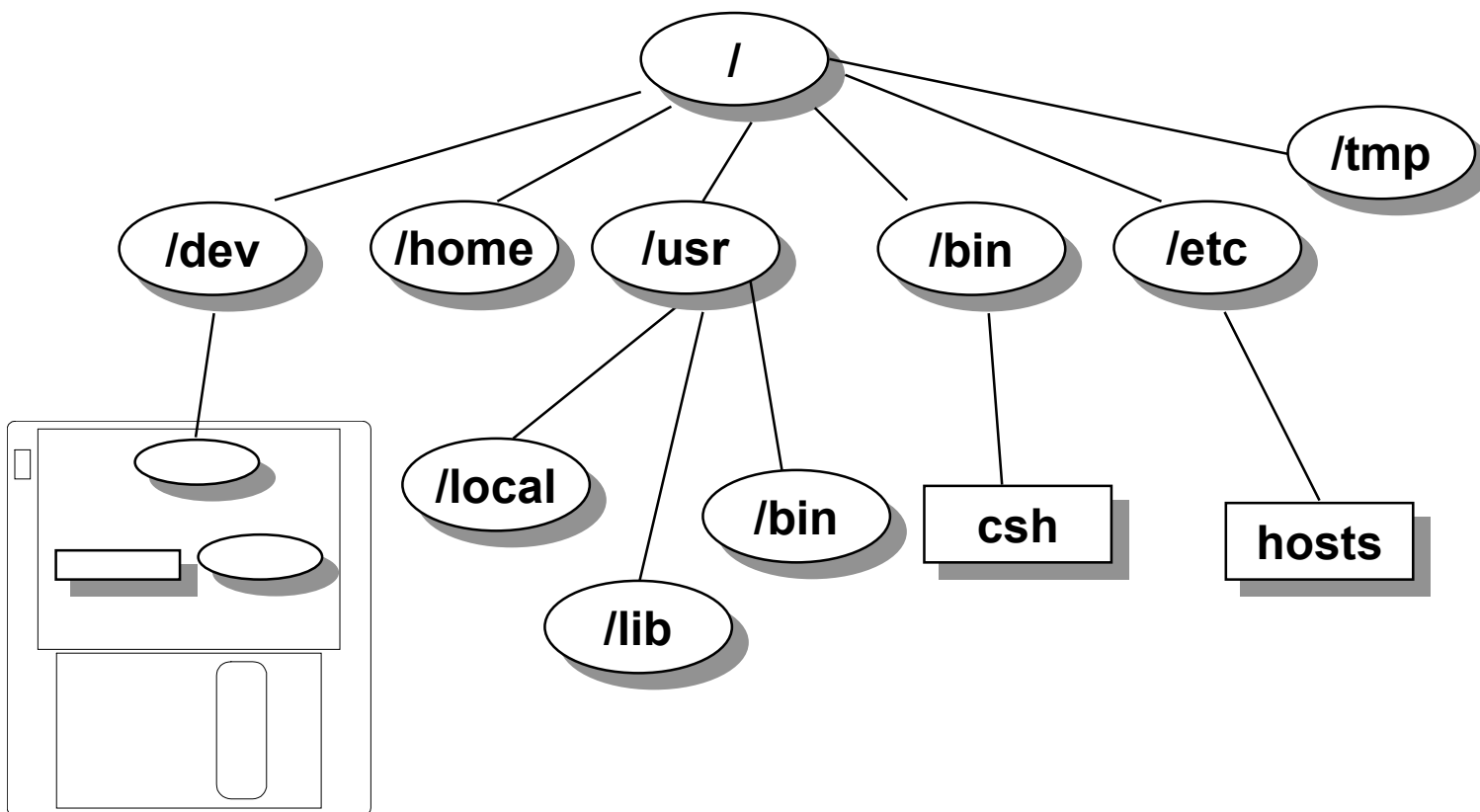
- ablakozós, többtaskos felület (terminál ablakok)
- window manager-ek
  - » beállítások a felhasználó ízlése szerint

## ◆ man - online help

- `man command`

# Logikai file rendszer

## egységes directory struktúra



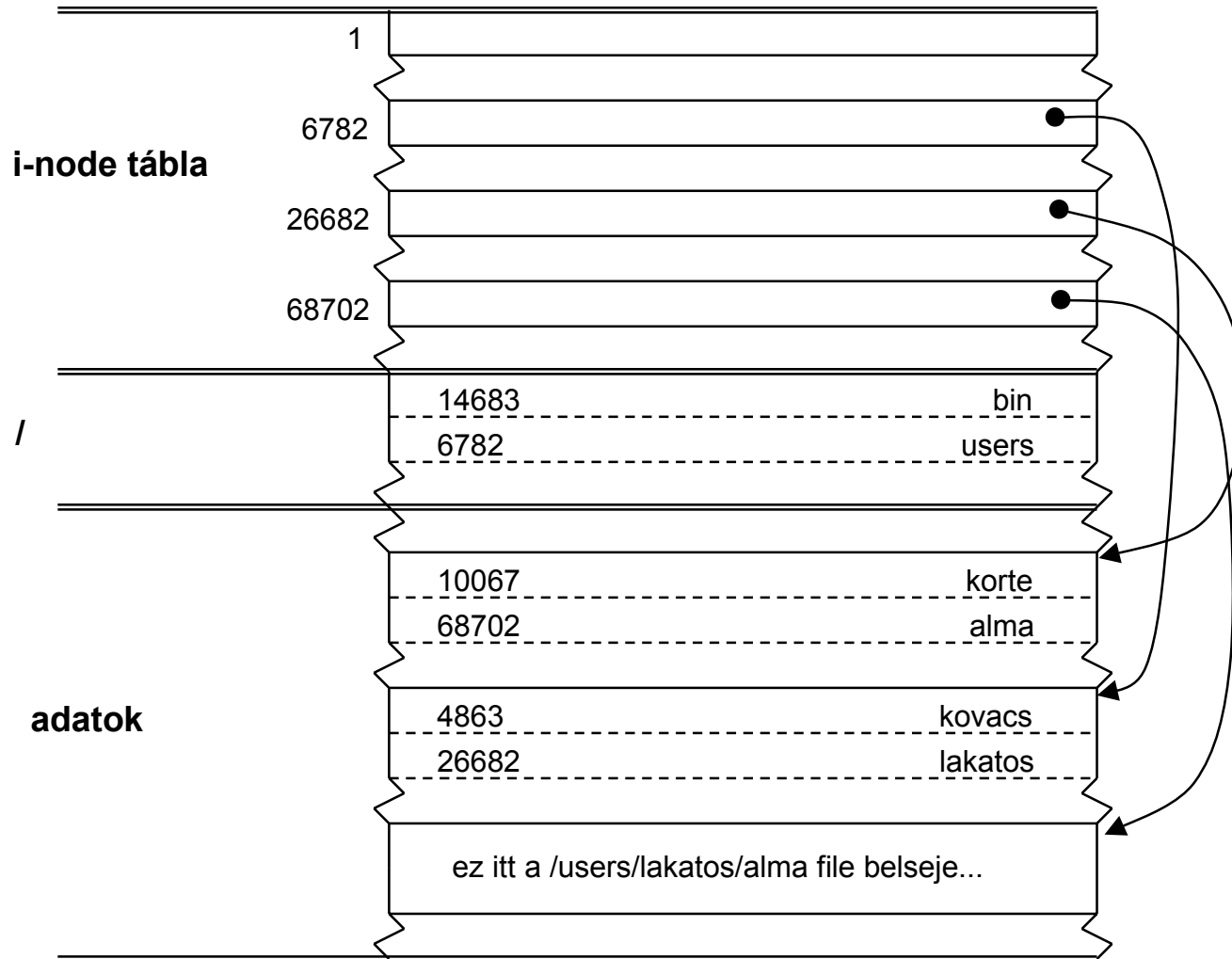
# Logikai file rendszer - fizikai eszközök

- ◆ **Kötet - volume (diszk)**
- ◆ **Perifériák (/dev)**
  - terminal
  - külső diszk drive
  - szalagos egység
  - hangkártya
  - CD drive
  - hálózat
- ◆ **Beillesztés a logikai rendszerbe**
  - device: device file
  - file rendszer: mount
- ◆ **Multiuser szolgáltatás**
  - védelem: jogok

# File név konvenciók

- ◆ **A könyvtárak (directory) speciális file-ok**
  - tartalmuk file bejegyzés
- ◆ **Gyöker könyvtár: /**
- ◆ **Általános file-név megadás**
  - abszolút
    - » `/users/lakatos/alma`, `/bin/ls`
  - relatív
    - `../lakatos/alma`, `../../bin/ls`
- ◆ **File név**
  - Kisbetű - Nagybetű
  - Maximális hossz 14 (gyakorlatban sokkal több)
  - Szinte minden karakter megengedett
  - a '.' kitüntetett szerepe:  
rejtett file-ok első karaktere

# Egy Unix file rendszer belső szerkezete





# Következmények

- ◆ **File bejegyzés**
  - név
  - i-node szám
- ◆ **egy file-nak több neve - link**
  - ugyanolyan néven, de különböző direktoryban
- ◆ **file információk i-node-ban**
  - típus: reguláris, directory, eszköz, szimbolikus link, pipe
  - linkek száma
  - tulajdonos (UID, GID)
  - méret
  - jogok
  - dátumok (létrehozás, módosítás, hozzáférés)
- ◆ **linkek file rendszerek között - szimbolikus link**
  - a szimbolikus link egy file, tartalma: elérési út
  - ki lehet húzni a lába alól a talajt

# Segédprogramok file és directory kezelés

## ◆ Általános forma

- parancs -opciók argumentumok
- input:: argumentum, hiányában standard input
- output: standard output, vagy file -o file megadással

## ◆ Filekezelés

ls	cat
mv	cp
rm	ln
touch	

## ◆ Directory kezelés

pwd	cd
mkdir	rmdir

# Directory tartalma: `ls`

## ◆ `ls -options dir/filename`

- » `-a` minden bejegyzést kilistáz (a rejtetteket is)
- » `-C` több oszlopban listáz
- » `-d` directory esetén csak a nevét írja ki
- » `-g` hosszú listában nem írja ki a tulajdonost
- » `-i` az i-node számot is kiírja
- » `-l` hosszú lista, részletezve a jogokat is
- » `-o` hosszú listában nem írja ki a csoportot
- » `-p` directory neve után ír egy / jelet
- » `-r` megfordítja a sorrendet
- » `-R` rekurzívan bejárja az alkönyvtárakat is
- » `-t` névsor szerinti lista helyett időrendben listáz
- » `-x` többoszlopos listában a rendezés vízszintesen
- » `-1` egyoszlopos lista

# File parancsok 1.

- ◆ file tartalmát írja ki a standard kimenetre
- ◆ **cat** *-options filename*
  - » -e mint -v, és minden sor végére \$
  - » -s nincs hibaüzenet
  - » -t mint -v, TAB helyett ^I
  - » -u a kimenetet nem buffereli
  - » -v a nem látható karaktereket is kiírja (kivéve TAB és NL) úgy, hogy azok láthatók legyenek (^X)
- ◆ file-ok másolása
- ◆ **cp** *[-iR] dir/filename dir/filename*
  - » -i interaktív mód
  - » -R rekurzív másolás
- ◆ file-ok átnevezése
- ◆ **mv** *[-if] dir/filename dir/filename*
  - » -i interaktív mód
  - » -f üzenet nélkül felülírja a meglévő file-t

## File parancsok 2.

- ◆ **file-ok törlése**
- ◆ **rm [-if] filename**
  - » mint mv
- ◆ **file-ok linkelése**
- ◆ **ln [-fs] filename linkname**
  - linkname nem létező file
- ◆ **ln [-fs] filename ... dirname**
  - több file egy directoryban eredeti nevén
    - » -f üzenet nélkül felülírja a meglévő file-t
    - » -s szimbolikus link
- ◆ **file módosítási és hozzáférési idejének frissítése**
- ◆ **touch [-amc] [-r rfile | -t time] filename**
  - » -a csak hozzáférési idő (access time)
  - » -m csak módosítási idő (modification time)
  - » -c nem jön létre a file, ha nincs
  - » -t adott idő, formátum: [ [CC]YY]MMDDhhmm[.ss]
  - » -r rfile idejével egyező

# Directory parancsok

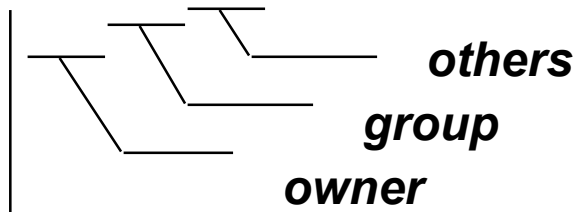
- ◆ **az aktuális directory kiírása**
  - `pwd`
- ◆ **directory váltás**
  - `cd dirname`
  - paraméter nélkül: `home`
- ◆ **directory létrehozás**
  - `mkdir [-p] dirname`
    - » `-p` a hiányzó alkönyvtárak is keletkeznek
- ◆ **directory törlés**
  - `rmdir dirname`

# File-lista

```
pyxis.hit.bme.hu:szandi />ls -l
drwxrwxr-x   3 root   system   2048 May  1 11:17 dev
drwxrwxr-x  14 root   system   6656 May  1 11:16 etc
drwxrwxr-x   8 root   system    512 Aug  4 1994 export
drwxr-xr-x  32 bin    bin     1024 Apr  1 15:36 home
lrwxrwxrwx   1 root   sys      8 Aug  4 1994 lib -> /usr/lib
drwx-----  2 root   system   512 Aug  4 1994 lost+found
drwxr-xr-x  14 bin    bin     512 Aug  4 1994 lpp
drwxr-sr-x   9 sys    sys     512 Feb 21 1994 math
drwxr-xr-x   2 bin    bin     512 Aug  4 1994 mnt
drwxr-xr-x   2 root   system   512 Aug  4 1994 save
drwxr-xr-x   3 bin    bin     512 May 21 1994 sbin
drwxr-sr-x   8 sys    sys     512 Jan 30 1994 simscript
-rw-r--r--   1 root   system 120111 Mar 14 12:41 smit.log
-rw-r--r--   1 root   system 12975  Mar 14 12:41 smit.script
drwxrwxr-x   2 root   system   512 Aug  4 1994 tftpboot
drwxrwxrwt   4 bin    bin    36352 May  8 14:34 tmp
lrwxrwxrwx   1 root   sys      5 Aug  4 1994 u -> /home
drwxr-xr-x  19 bin    bin     512 Feb 14 1994 usr
drwxr-xr-x  10 bin    bin     512 Aug  4 1994 var
pyxis.hit.bme.hu:szandi />
```

# Jogok 1.

**ls -l**  
**drwxrwxrwx**     **1 szandi staff**     **221**     **Sep 28 1994 mod.pas**



érvényesítés  
→

-	none
r	read
w	write
x	execute
s	set-uid,set-gid
t	sticky bit

-	regular file
d	directory
l	szimbolic link
p	named pipe
c	character device
b	block device



# Jogok 2.

## ◆ Olvasás (r)

- file: a file (tartalma) olvasható
- dir: a direktory tartalma megtekinhető, azaz a direktoryban levő összes file-bejegyzés olvasható, de maga a file nem nyitható meg, csak ha 'x' jog is van

## ◆ Írás (w)

- file: a file (tartalma) megváltoztatható
- dir: a direktory tartalma írható, azaz új file-ok hozhatók létre, illetve meglévők törölhetők

## ◆ Execute/search (x)

- file: a file végrehajtható (bináris vagy script)
- dir: a direktoryban lehet keresni, azaz konkrét file-ra lehet hivatkozni (ha arra van értelemszerű jog), de a direktory teljes tartalma nem olvasható

# Jogok 3.

- ◆ **real - effective UID (GID)**
  - új file létrehozása, program futása: effective UID, GID
  - effective group váltás: newgrp
- ◆ **listában az execute (x) flag pozíciójában más betű**
- ◆ **owner pozícióban 's' : set-uid**
  - file (bináris): a program futásakor effective UID=file UID
- ◆ **group pozícióban 's' : set-gid**
  - file (bináris) : a program futásakor effective GID=file GID
  - dir: új file GID=directory GID
- ◆ **others pozícióban 't' : sticky bit**
  - file: a program a futás után a memóriában marad
  - dir: csak a tulajdonos törölheti a file-okat (/tmp)

# Jogok módosítása

- ◆ **File-bejegyzés jogainak megváltoztatása**
  - A tulajdonos és root végezheti el
  - `chmod`
- ◆ **File-bejegyzés csoportjának megváltoztatása**
  - A tulajdonos akkor végezheti el, ha a megadott csoportba tartozik, root mindig
  - `chgrp`
- ◆ **File-bejegyzés tulajdonosának megváltoztatása**
  - Csak root végezheti el
  - `chown`
- ◆ **Új file jogai létrehozáskor**
  - adatfile: `rw-rw-rw-`
  - futtatható file: `rwxrwxrwx`
  - fenti maszkolása (szűkítése): `user mask`
  - `umask` (alapértelmezés: `022`, azaz `rwxr-xr-x`)

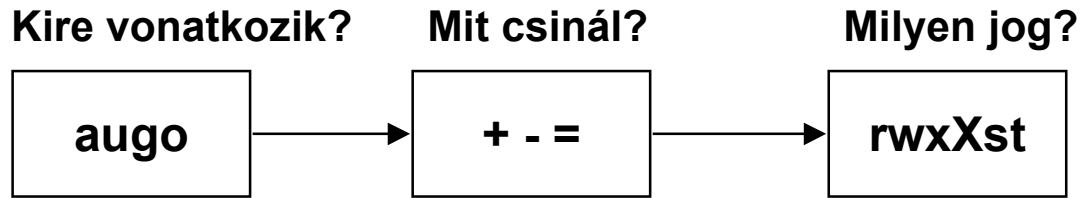
# Jogok megváltoztatása

- ◆ **chmod [-Rf] *mode dir/filename***
  - » -R rekurzív végrehajtás
  - » -f nincs hibaüzenet sikertelenség esetén
- ◆ ***mode* megadása**
  - abszolút módon
  - szimbolikusan
- ◆ **chgrp [-Rf] *newgroup dir/filename***
  - » opciók, mint chmod
- ◆ **chown [-Rf] *newowner dir/filename***
  - » opciók, mint chmod

# Jogok megadása abszolút módon (összeadandó, oktális)

» 4000	setuid
» 2000	setgid
» 1000	sticky bit (csak root)
» 0400	tulajdonos - olvasás
» 0200	tulajdonos - írás
» 0100	tulajdonos - végrehajtás, keresés
» 0040	csoport - olvasás
» 0020	csoport - írás
» 0010	csoport - végrehajtás, keresés
» 0004	mindenki - olvasás
» 0002	mindenki - írás
» 0001	mindenki - végrehajtás, keresés

# Jogok megadása szimbolikusan



**u** user  
(tulajdonos)  
**g** group  
(csoport)  
**o** others  
(többiek)  
**a** a fentiek  
együtt

**+** hozzáad  
**-** letilt  
**=** pontosan  
beállít

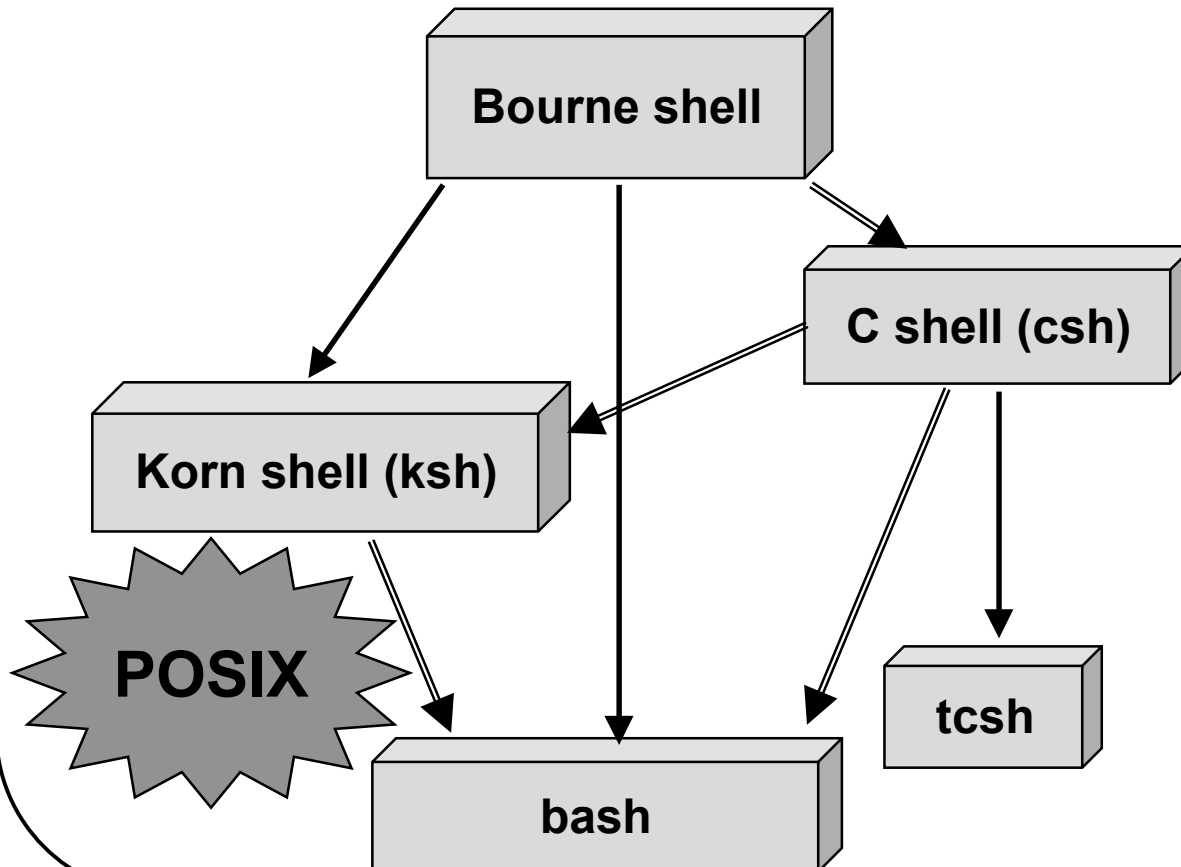
**r** olvasás  
**w** írás  
**x** végrehajtás/  
keresés  
**X** keresés/  
végrehajtás  
ha már van másik  
**s** setiud/setgid  
**t** sticky bit

# Felhasználók

- ◆ felhasználó (csoport) ID kiírása
- ◆ `id [[-G] | [-g] | [-u]] [-n] [user]`
  - » `-G` összes GID (real, effective, supplementary)
  - » `-g` csak effektív GID
  - » `-u` csak effektív UID
  - » `-n` ID helyett név
- ◆ aktív felhasználó
  - `id`
    - » összes ID (real, effektív is ha eltér) és nevük
  - `who am i`
    - » terminál és a bejelentkezés ideje is
- ◆ adott felhasználó összes ID-je és neve
  - `id user`
- ◆ bejelentkezett felhasználók
  - `who`
    - » terminál és a bejelentkezés ideje is

# Shell

- parancs interpreter: kapcsolat a felhasználó felé
  - » közös Unix program
- „programozási nyelv”: sciptek
  - » egy feladatot csak egyszer oldjunk meg





# Shell meghívása

- ◆ **login shell**

- `chsh shell`

- ◆ **végrehajtott scriptek**

	<u>cs</u> h	<u>ba</u> sh	<u>POSIX</u>
<b>login</b>	<code>/etc/login</code> <code>~/.login</code>	<code>/etc/profile</code> <code>~/.bash_profile</code> <b>vagy</b> <code>~/.bash_login</code> <b>vagy</b> <code>~/.profile</code>	<code>/etc/profile</code> <code>~/.profile</code>
<b>start</b>	<code>/etc/cshrc</code> <code>~/.cshrc</code>	<code>~/.bashrc</code>	<code>\$ENV</code>
<b>logout</b>	<code>/etc/logout</code> <code>~/.logout</code>	<code>~/.bash_logout</code>	

- ◆ **`sh -c script-string`**

# Folyamatok 1.

## ◆ futó program: folyamat

- fa-struktúra
- (effektív) UID, PID, PPID
- command-line: job (%n)
  - » shell kezeli, process groupok

## ◆ státusz

## ◆ ps [-arlu] [-t term] [pid]

- » -a minden processzt listáz
- » -r csak a futó processzeket listázza
- » -l hosszú listát készít
- » -u a lista tartalmazza a user nevet is
- » -t a term terminálhoz kapcsolódó processzek (opcionális)
- » pid csak a pid processz (opcionális)
- » egyéb nem említett opciók is vannak

## ◆ jobs [-l | -p] [job]

- » -l részletes információ
- » -p csak a PID

# Folyamatok 2.

- ◆ **job control (előtér - háttér)**
  - `fg [%jobnum]`
  - `bg [%jobnum ...]`
- ◆ **jelek (küldése)**
  - HUP, KILL, TERM, CONT
  - `kill -sig [PID | %jobnum]`
  - `kill -KILL 1234`
    - » összes szignál: `kill -l`
- ◆ **futtatás alacsonyabb prioritással**
  - `nice [-n] command-line`
    - » `-n` nice value: -20...20
    - » `n` prioritással alacsonyabban fut, mint alapértelmezésben
    - » tipikus használat: `nice parancs &`
- ◆ **prioritás állítása**
  - `renice [nv] pid`
    - » `nv` nice value: -20...20

# Folyamatok 3.

- ◆ **kijelentkezés után is fut**
  - `nohup command-line`
    - » tipikus használat: `nohup parancs &`
- ◆ **késleltetés n másodpercig**
  - `sleep n`
- ◆ **várakozás adott processz befejeződéséig**
  - `wait [pid | %job]`
  - ha nincs paraméter, az összes háttérben futó processz befejeződéséig vár
  - visszatérési értéke a processz visszatérési értéke
- ◆ **szinkronitás**
  - szinkron parancs-sorozat: `ls ; who`
    - » a második csak akkor fut, ha az első befejeződött
  - aszinkronitás: `ls &`
    - » azonnal visszkapjuk a command-promptot
  - csoportosítás: `(date; ls) > lista &`
    - » a zárójeles rész külön fut, outputjuk összegződik

# Folyamatok 4.

## ◆ futtatás adott időpontban

- **at [-f file] [-m] atime [adate] [+ increment]**
  - » mail-be kerül a job standard outputja és errorja, ha van
  - » -f file – elindítandó shell script (hiányában standard input)
  - » -m – mail-t kap a felhasználó, ha lefutott a job
  - » atime – indítási időpont
    - h[h][:][mm][am|pm] | noon | midnight | now
  - » adate – indítási dátum (alapértelmezés: today | tomorrow)
    - [today | tomorrow | day-of-week | [[CC]YY] MMM DD]
  - » increment – az adott időhöz képest később
    - n [minutes | hours | days | weeks | months | years]
- **at [-f file] [-m] -t time**
  - » time – indítási időpont (mint touch)
    - [[CC]YY]MMDDhhmm[.ss]
- **at -l [job...]**
  - » lista a felhasználó által elindítani kezdeményezett jobokról
- **at -r job...**
  - » job(ok) törlése a várakozási sorból
  - » péda – az összes job törlése: `at -r `at -l | cut -f1``

# Operátorok

- ◆ **logikai igaz - sikeres végrehajtás**

- exit status 0

- ◆ **pipeline**

`c1 | c2`

- c1 outptja c2 inputja lesz

- ◆ **logikai negálás (NOT)**

`! c`

- c exit státusza negálódik

- ◆ **logikai ÉS (AND)**

`c1 && c2`

- c2 csak akkor fut, ha c1 sikeres

- ◆ **logikai VAGY (OR)**

`c1 || c2`

- c2 csak akkor fut, ha c1 sikertelen

- ◆ **parancs vége**

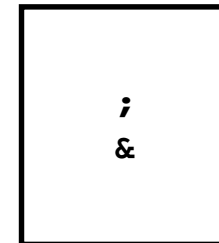
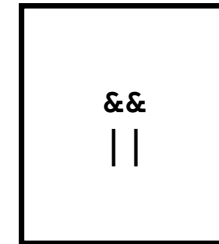
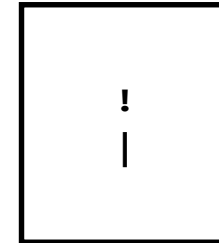
`c1 ; c2`

- c2 akkor fut, ha c1 befejeződött
- egyenértékű egy sorvégével

- ◆ **futtatás háttérben**

`c1 & c2`

- c1 elindul a háttérben
- c2 futása azonnal elkezdődik



→ precedencia →

# Átírányítások

- ◆ **standard output** >
- ◆ **standard input** <
- ◆ **standard output append** >>
- ◆ **output és error összekapcsolódik** 2>&1
- ◆ **output helyett n. filedescriptor** >&n
- ◆ **input helyett n. filedescriptor** <&n
- ◆ **n. filedescriptor helyett m.** n>&m
- ◆ **here documents word-ig** << word

- példa:

```
cat << ***END***  
Listing of $1 as `date`  
`ls -l $1`  
*** End of listing ***  
***END***
```

# Paraméterek, változók

- ◆ **pozicionális paraméterek**

\$1 \$2 \$3 \$4 \$5 ...

shift [n]

- ◆ **speciális paraméterek**

\$\* az összes paraméter

\$@ ugyanaz mint fent, de "-ben külön szavak

\$# paraméterek száma

\$? exit statusz

\$\$ processz ID

\$! háttér processz ID

\$0 a shell

- ◆ **előre definiált változók**

- HOME
- PATH
- PS1

- ◆ **felhasználó által definiált változók**

- a1ma=korte



# Helyettesítések

## ◆ file-név (az illeszkedő file neve helyettesítődik)

- \* 0 vagy tetszőleges karakter
- ? egy karakter
- [abc] [a-z] halmaz, vagy intervallum
  - » példa: összes file az adott direktoryban path-al együtt:
  - » `echo dir/*`

## ◆ paraméter (változó)

- `$var`
  - » elérési útvonal kiírása: `echo $PATH`

## ◆ parancs

- ```
  - » a parancs outputja helyettesítődik, pld: `i=`expr $i + 1``

## ◆ tilde (Bourne shellben nincs)

- `~[user]`
  - » aktuális (vagy user) felhasználó HOME könyvtára helyettesítődik
  - » példa: `echo ~ ~szandi`

# Idézet (quotation)

## ◆ Backslash quotation

- `\c`
- a sor végén: folytatósor
- `\\`

## ◆ Single quotation

- `'text'`
- a két egyszeres idézőjel között semmi sem speciális (csak ')

## ◆ Double quotation

- `"text"`
- `$ ` " ' & \` megtartják jelentésüket
- `\` elnyomja a fentiek jelentését

## ◆ Példa

- `echo "My home is $HOME but not \ $HOME"`
- `echo 'My home isn''''t $HOME'`
- `echo My home isn\'t \ $HOME`

## (basic) Reguláris kifejezések (BRE) 1.

- ◆ **c** a 'c' karakter, ha nem speciális (. [^\*\$\)
- ◆ **\c** a 'c' karakter
- ◆ **^** sor eleje
- ◆ **\$** sor vége
- ◆ **.** egy darab akármilyen karakter
- ◆ **[abc]** egy karakter az 'abc'-ből
- ◆ **[^abc]** egy karakter, de nem 'abc'
- ◆ **r\*** r kifejezés 0 vagy tetszőleges számszor
- ◆ **r<sub>1</sub>r<sub>2</sub>** r<sub>1</sub> és r<sub>2</sub> egymás után, r<sub>1</sub> a lehető leghosszabban illeszkedik
- ◆ **\(r\)** r kifejezésre később lehet hivatkozni
- ◆ **\i** az i-edik (1-9) \**(r\)** -al definiált kifejezés

## (extended) Reguláris kifejezések (ERE) 2.

- ◆ BRE és még ...
  - ◆ újabb speciális karakterek: + ? | ( )
  - ◆ ( ) kifejezések zárójelezhetők (csoportosíthatók)
  - ◆  $r^+$   $r$  kifejezés 1 vagy tetszőleges számszor
  - ◆  $r^?$   $r$  kifejezés 0 vagy 1 számú előfordulással
  - ◆  $r_1/r_2$   $r_1$  vagy  $r_2$
  - ◆ precedencia:
  - ◆ [ ]
  - ◆ ( )
  - ◆ \* + ?
  - ◆ |
- helyettesítő stringben:
- ◆ & az aktuális illeszkedő reguláris kifejezés
  - ◆  $\backslash i$  az  $i$ -edik (1-9)  $\backslash(r\backslash)$  -al definiált kifejezés

# Oldaltördelő (pager) segédprogramok

- ◆ feladatuk: file-ok kiírása a terminálra oldalanként
  - more less
- ◆ gyakori használat szűrőként
  - ls | less
- ◆ futás közben billentyűparancsok (a legfontosabbak)
  - kilépés: q
  - shell parancs: *!cmd*
  - mozgás: 

	<u>előre</u>	<u>vissza</u>
• egy sor	<i>enter</i>	k
• egy képernyő	<i>space</i>	b
• file vége	g	G
• minta	<i>/regexpr</i>	? <i>regexpr</i>
• következő minta	n	N
- ◆ csak less - nem POSIX
  - lehet visszafele is mozogni
  - nem olvassa be a teljes file-t (nagy file-ok kezelése is gyors)
  - a file végén nem lép ki ( a file futás közben is bővílhet)

# Keresés

## ◆ Szövegminta keresése

- alapértelmezésben kiírja az illeszkedő sorokat
- több file esetében minden sor elé kerül a file neve
- file argumentum nélkül a standard inputról olvas

## ◆ **grep [-ivnc]s *regexp* [*file...*]**

- » -i ignore case (kis-nagybetű nem különbözik)
- » -v azokat a sorokat írja ki, melyek nem illeszkednek
- » -n minden sort elején lesz filenév és sorszám
- » -c csak az illeszkedő sorok számát írja ki
- » -l csak a file(ok) nevét írja ki
- » -s olvashatatlan input file-ok esetén nincs hibaüzenet
- » egyéb nem említett opciók is vannak

## ◆ Kiterjesztett (extended) reguláris kifejezés használata

- egrep: `grep -E`

## ◆ Fix string használata

- fgrep: `grep -F`

# További szűrők

- ◆ **File eleje (első 10 sor)**
- ◆ **head [-n num] [*file...*]**
  - » -n az első num darab sort írja ki
- ◆ **File vége (utolsó 10 sor)**
- ◆ **tail [-f] [-c numbyte |-n numlin] [*file...*]**
  - » -c a kiírandó szöveget byteban mérjük
    - +numbyte a kiírás a numbyte-adik byte-nál kezdődik
    - -numbyte az utolsó numbyte darab byte kerül kiírásra
  - » -n az utolsó numlin darab sort írja ki
  - » -f a file végén nem lép ki
- ◆ **Csővezeték megcsapolása**
  - a standard inputot a standard outputra és file-ba másolja
- ◆ **tee [-ia] [*file...*]**
  - » -i figyelmen kívül hagyja az interruptot
  - » -a hozzáfűzi (append) az outputot a file-hoz
  - `ls | tee file`

# Rendezés (szűrő)

- ◆ **sort** [-bdfiMnr] [-tc] [+p1[s][-p2[s]]] [-o *file*] [*file...*]
  - Rendezi az inputot kulcsok (vagy teljes sorok) alapján
  - Hasonlítást befolyásoló opciók
    - » **b** kezdő szóközök tabulátorok figyelmen kívül hagyva
    - » **d** szótárrendezés: csak betűk és számjegyek
    - » **f** kis- és nagybetűk nincsenek megkülönböztetve
    - » **i** nem ASCII karakterek figyelmen kívül hagyva
    - » **M** hónapsorrend (angol, a nem jók JAN elé)
    - » **n** numerikus sorrend
    - » **r** megfordítja az aktuális rendezést
  - **tc** a mezők (kulcsok) közti szeparátor *c* lesz (whitespace)
  - Rendezési kulcs (key) a sor egy mezője (szava)
    - » mezők számozása 0-tól kezdődik
    - » **+p1** első kulcs
    - » **-p2** utolsó, már nem használt kulcs
    - » **s** hasonlítást befolyásoló opció (lásd fent)
  - az output a megadott file: -o file
  - példa: passwd file rendezése GID, azon belül UID szerint
    - » `sort -t: +3 -4n +2 -3n /etc/passwd`



# File keresése 1.

- ◆ ***find path feltétel***
- ◆ **megkeres minden file-t path-ban és alatta**
- ◆ **feltételeket kiértékeli, mely teljesül, ha a file ...**
  - » **numerikus értékek (n: pontosan n; ±n: nagyobb/kisebb, mint n)**
  - **-name *fname*** neve illeszkedik *fname*-ra
  - **-type *c*** típusa *c* (f-file, d-directory, l-szimbolikus link  
p-pipe, c-character device, b-block device)
  - **-links [±]*n*** linkjeinek száma *n*
  - **-perm *mode*** védelmi módja *mode* (mint `chmod`)
  - **-user *uname*** tulajdonosa *uname*
  - **-group *gname*** *gname* csoporthoz tartozik
  - **-size [±]*n*** *n* blokk (512 byte) méretű
  - **-size [±]*nc*** *n* karakter méretű
  - **-atime [±]*n*** hozzáférési (access) ideje *n* (nap)
  - **-mtime [±]*n*** módosítási (modification) ideje *n* (nap)
  - **-ctime [±]*n*** létrehozási (creation) ideje *n* (nap)
  - **-newer *file*** újabb, mint *file*
  - **-inum [±]*n*** i-node száma *n*

## File keresése 2.

- ◆ **parancsok alkalmazása**
- ◆ **feltételként teljesül, ha a visszatérési érték 0**
  - `-exec cmd`                      `cmd` parancs futtatása
  - `-ok cmd`                              `cmd` parancs futtatása csak nyugtázás után
    - » `cmd`-ben speciális jelentések:
      - » `{}`                              a keresett file teljes path-al behelyettesítődik
      - » `;`                                a `cmd` végét jelöli
- ◆ **egyéb feltételek mellékhattással - mindig igaz értékűek**
  - `-print`                              kiírja a file nevet path-al együtt
  - `-follow`                              követi a szimbolikus linkeket (`-type l`)
- ◆ **tesztek kombinálása logikai operátorokkal**
  - `(kif.)`                              csoportosítás, külön argumentum, idézni kell: `\(`
  - `!`                                      logikai nem (NOT)
  - `-a`                                    logikai és (AND)
  - `-o`                                    logikai vagy (OR)
- ◆ **példa: töröljük le a \*.bak file-okat**
  - `find . -name "*.bak" -type f -exec rm {} \;`

# További segédprogramok 1.

- ◆ **Sorok, szavak, karakterek számlálása**
- ◆ **wc [-c|w] [*file...*]**
  - » -c karakterek száma
  - » -l sorok száma
  - » -w szavak (whitespace között) száma
- ◆ **Kifejezések kiszámítása**
- ◆ **expr *kif.***
  - » *kif.* whitespace-ek között operátorok, argumentumok
  - » + - \* / % < > <= >= != | &
  - » kifejezés kiszámított eredménye standard outputra
- ◆ **File tartalmának kiírása**
- ◆ **od -f *file...***
  - » *f* formátum:
    - d-decimal, u-unsigned decimal, o-octal, x-hexadecimal
    - f-floating point number, c-character, a-ascii character

## További segédprogramok 2.

- ◆ **Karakterek cseréje**
- ◆ **tr [-c ds] str1 [str2]**
  - » **str1** a helyettesítendő vagy törlendő karakterek
  - » **str2** a helyettesített karakterek (str1-ben azonos sorszám)
  - » **-c** str1 komplementjét használja
  - » **-d** a karakterek törlése
  - » **-s** ismétlődéseket csak egyszer írja ki (squeeze)
  - » **stri** megadhatók tartományok: [a-z], ismétlődések [c\*n]
    - például kis-nagybetű csere: tr "[a-z]" "[A-Z]"
    - DOS textből UNIX text: tr -d "\r"
- ◆ **Karakterek, mezők kivágása (számozásuk 1-től)**
- ◆ **cut [-c|-f [-dc]] lista [file...]**
  - » **-c** karakterek kivágása
  - » **-f** mezők kivágása
  - » **-dc** c a mezőhatároló
  - » **lista** számok, tartományok, pld: 1,3-5,12-

## További segédprogramok 3.

- ◆ **Filenév/Direktorynév rész kiírása**
- ◆ **basename/dirname str1 [str2]**
  - » str1 a teljes fájlnev elérési úttal
  - » str2 a kiterjesztés (suffix) (basename esetében)
- ◆ **Fájlok típusának megállapítása**
- ◆ **file [-L] [-f file1] file...**
  - » -L szimbolikus link "követése"
  - » -f file1-ből is olvassa a fájlok listáját
- ◆ **Ismétlődő sorok törlése**
- ◆ **uniq [-cdu] [infile [outfile]]**
  - » -d az ismétlődő sorokból csak egyet ír ki
  - » -u csak a nem ismétlődő sorokat írja ki
  - » -c minden sor elé kiírja az ismétlődési számot
- ◆ **Fájl feldarabolása (xaa, xab, ... xaz, xba, ...)**
- ◆ **split [-l lines] [file [name]]**
  - » -l az egyes darabok sorainak száma (alapért. 1000)
  - » name a darab nevek előtagja (prefix)

# Fájlok összehasonlítása

- ◆ **Rendezett szövegfájlok összehasonlítása, kiírása**
- ◆ **comm [-123] file1 file2**
  - » -1 csak az első file-ban előforduló sorok
  - » -2 csak a második fájlban előforduló sorok
  - » -3 mindkét fájlban előforduló sorok
- ◆ **Pontos összehasonlítás (csak a különbség kiírása)**
- ◆ **cmp [-s] file1 file2**
  - » -s nincs semmilyen kiírása
  - » visszatérési érték 0,1,2: a két fájl azonos, különböző, hiba
- ◆ **Szövegfájlok összehasonlítása**
- ◆ **diff [-eb] file1 file2**
  - » -e kiír egy script-et, melyet az ed editorral futtatva, file1-re alkalmazva eredményül file2-t kapjuk
  - » -b összehasonlításkor egy szóköznek tekinti az ismétlődő whitespace karaktereket
  - » visszatérési érték, mint cmp
  - » egyéb, nem említett opciók is vannak

# (bash) shell programozás

## input feldolgozása

- ◆ 1. tokenekre bontás
- ◆ 2. tokenek parancsba szerkesztése

### ◆ Tokenek

- szavak

- » határolói: szeparátor vagy operátor

- » szeparátor: space, tab, newline

- operátorok

> >> >& >| < << <<- <& <>  
| & ; ( ) || && ;; (( )) |&

### ◆ newline értelmezése

- 1. ha a sor értelmezhető befejezett parancsként, akkor értelmeződik
- 2. ha a sorban nincsen szó, akkor törlődik
- 3. egyébként a newline érvénytelen, a parancs értelmezése folytatódik

# Parancsok értelmezése

1. Kulcsszó
2. Nem idézett alias
3. Belső parancs
4. Függvény definíció
5. Beépített segédprogram
6. Egyéb beépített segédprogram
7. Abszolút pathnév
8. Futtatható program a path-ban



# Kulcsszavak

<code>case</code>	<code>function</code>
<code>do</code>	<code>if</code>
<code>done</code>	<code>in</code>
<code>elif</code>	<code>select</code>
<code>else</code>	<code>then</code>
<code>esac</code>	<code>until</code>
<code>fi</code>	<code>while</code>
<code>for</code>	<code>{ }</code>

**nem POSIX: `select`**

# Belső parancsok

`break`

`:`

`continue`

`. file`

`eval`

`exec`

`exit`

`export`

`readonly`

`return`

`set`

`shift`

`trap`

`unset`

# Beépített segédprogramok

`alias`

`bg`

`cd`

`command`

`false`

`fc`

`fg`

`getopts`

`jobs`

`kill`

`newgrp`

`read`

`true`

`umask`

`unalias`

`wait`

# Összetett parancsok

## ◆ futtatás sub-shellben

- ( list )

## ◆ futtatás direct

- { list ; }
- { list & }

## ◆ feltételes végrehajtás

- if
- case
- select (bash)

## ◆ ciklusok

- for
- while
- until

## ◆ függvény definíció

- `function name { list; }`
- `name () { list; }`

# Logikai kifejezések

- ◆ `test expr`
- ◆ `[ expr ]`
- ◆ `true`
- ◆ `false`
- ◆ **string összehasonlítások**
  - `s1 = s2`
  - `s1 != s2`
- ◆ **numerikus összehasonlítások**
  - `n1 -eq n2`                    `n1 = n2`
  - `n1 -ne n2`                    `n1 ≠ n2`
  - `n1 -lt n2`                    `n1 < n2`
  - `n1 -le n2`                    `n1 ≤ n2`
  - `n1 -gt n2`                    `n1 > n2`
  - `n1 -ge n2`                    `n1 ≥ n2`

# File vizsgálatok

- ◆ minden teszt `-o file` alakú
- ◆ **s**: a file létezik és nem 0 méretű
- ◆ **f**: a file normál file
- ◆ **r**: a file létezik és olvasható
- ◆ **w**: a file létezik és írható
- ◆ **x**: a file létezik és futtatható
- ◆ **d**: a file létezik és directory
- ◆ **u**: a file létezik és set-uid bit be van állítva
- ◆ **g**: a file létezik és set-gid bit be van állítva
- ◆ **file1 -nt file2**
  - file1 újabb file2-nél
- ◆ **file1 -ot file2**
  - file1 régebbi file2-nél
- ◆ **file1 -ef file2**
  - file1 egy másik neve file2

# Logikai kifejezések kombinációja

- ◆ NOT  $! \textit{expr}$
- ◆ AND  $\textit{expr1} \textit{-a} \textit{expr2}$
- ◆ OR  $\textit{expr1} \textit{-o} \textit{expr2}$
- ◆ csoportosítás  $( \textit{expr} )$

# Feltételes végrehajtás

```
if list
    then list
[elif list
    then list]
...
[else list]
fi
```

```
if test $1 = 'alma'
then echo '$1=alma'
fi
```

```
case word in
pattern $\bar{i}$ )
    comm $\bar{i}$  ;;
...
esac
```

```
case $fn in
*.c | *.cpp ) cppcompile $fn;;
*.pas       ) pascompile $fn;;
*           ) echo "Nem definiált $fn";;
esac
```



# Ciklusok

```
for name [ in word; ]  
do  
  list ;  
done
```

```
while list  
do  
  list ;  
done
```

```
repeat list  
do  
  list ;  
done
```

# Parancsok futtatása

- ◆ **direct futtatás**
- ◆ **direct futtatás sub-shellben**
  - `()`
  - `&`
  - ```
  - `|`
- ◆ **indirect futtatás sub-shellben**
- ◆ **Példa**

```
cd ~/one
X=1; Y=2; Z=3
export X
(cd ../two; pwd; Y=7; echo $X $Y $Z)
pwd
echo $X $Y $Z
```

# Kifejtés (expansion)

- ◆ **Brace expansion**
  - `a{d,c,b}e` ----> `'ade ace abe'`
- ◆ **Tilde expansion**
  - `login név`
- ◆ **Parameter expansion**
  - `${parameter}`
- ◆ **Command substitution**
  - `$(command)`
  - ``command``
- ◆ **Arithmetic expansion**
  - `$(expression)`
  - `$(expression)`
- ◆ **Pathname expansion**
  - `*`
  - `?`
  - `[abc] [a-z]`

# Brace expansion

- ◆ **nem POSIX és nem Bourne kompatibilis**

- ◆ **lelitható**

- `bash -nobreaceexpansion`
- `set +o braceexpand`

- ◆ **szintaktika**

- `{}` zárójelpár
- `{}` között legalább egy `,`

- ◆ **egymásba ágyazható**

- ◆ **végrehajtás balról jobbra**

- ◆ **eredménye több szó**

- ◆ **példa**

```
mkdir {Varga{Tibi,Kati,Judit,Kriszti},TothJutka}
```

```
mkdir VargaTibi VargaKati VargaJudit VargaKriszti TothJutka
```

# Tilde expansion

- ◆ HOME directory

~

- ◆ adott felhasználó HOME directory-ja

~*user*

- ◆ HOME directory alatti directory-k

~/alma/korte

~lakatos/alma/korte

- ◆ aktuális directory (PWD változó értéke)

~+

- ◆ előző directory (OLDPWD változó értéke)

~-

- ◆ ha *user* nem létezik nincs ~ helyettesítés

# Parameter expansion 1.

## ◆ alap (basic expansion)

`$param`

`${param}`

## ◆ feltételes (conditional expansion)

`${param:-word}`

- ha param létezik és értéke nem zérus - `$param`
- ha param nem létezik - `word`

`${param:?word}`

- ha param létezik és értéke nem zérus - `$param`
- ha param nem létezik vagy zérus akkor
  - » hibaüzenet, ami `word`
  - » ha nem interaktív shell, akkor kilép

`${param:+word}`

- ha param létezik és értéke nem zérus - `semmi`
- ha param nem létezik - `word`

# Parameter expansion 2.

## ◆ opcionális értékadás (csak változókra)

`${param:=word}`

- ha param létezik és értéke nem zérus - \$param
- ha param nem létezik - word; mellékhatás: param értéke word

## ◆ hossz meghatározás

`${#param}`

- param hossza (karakterek száma)

## ◆ mintaillesztés (pat-re pathname expansion)

`${param#pat}`

- param eleje illeszkedik - (legrövidebb)

`${param##pat}`

- param eleje illeszkedik - (leghosszabb)

`${param%pat}`

- param vége illeszkedik - (legrövidebb)

`${param%%pat}`

- param vége illeszkedik - (leghosszabb)

# Command substitution parancshelyettesítés

- ◆ a parancs output helyettesítődik
- ◆ ``command``
  - `$` \` megtartják speciális jelentésüket
- ◆ `$(command)`
  - nincs semminek speciális jelentése
- ◆ a parancshelyettesítés egymásba ágyazható
- ◆ példák

```
echo `echo `echo korte` alma`  
bash: alma: command not found  
echo korte  
echo $(echo $(echo korte) alma)  
korte alma
```



# Arithmetic expansion

- ♦ aritmetikai értelmezés

  - \$[expression]

  - \$((expression))

- ♦ egymásba ágyazható

- ♦ oktális számok 0 prefix

- ♦ hexadecimális számok 0x vagy 0X prefix

- ♦ tetszőleges számrendszerű számok

  - [base#]number

- ♦ operátorok, mint C

- ♦ példa

  - a=b+9; b=-2; echo \$((a\*5))

  - 35

# Aritmetikai operátorok

- +	unary minus and plus
! ~	logical and bitwise negation
* / %	multiplication, division, remainder
+ -	addition, subtraction
<< >>	left and right bitwise shifts
<= >= < >	comparison
== !=	equality and inequality
&	bitwise AND
^	bitwise exclusive OR
	bitwise OR
&&	logical AND
	logical OR
= *= /= %=	assignment
+= -= <<= >>=	
&= ^=  =	

# Néhány előre definiált változó

- ◆ **POSIX**

`PATH, CDPATH, HOME, ENV, PS1, PS2, IFS, ...`

- ◆ **directoryk**

`PWD, OLDPWD`

- ◆ **prompt**

`PROMPT_COMMAND`

- ◆ **rejtett file-ok file-név helyettesítésnél**

`glob_dot_filenames`

- ◆ **változók értékadása**

`PROMPT_COMMAND=alma`

- ◆ **változók beállítása**

`glob_dot_filenames=`

- ◆ **változók törlése**

`unset glob_dot_filenames`

# Shell belső parancsok 1.

- ◆ **üres utasítás (az argumentumok kiértékelődnek!)**  
: [args]
- ◆ **shell script behelyezése**  
.  
*file* [args]  
source *file* [args]
- ◆ **parancs futtatása subshell nélkül**  
exec *command*
- ◆ **parancs képzés, majd végrehajtás**  
eval [args]
  - » alma=korte
  - » eval \$alma
  - » bash: korte: command not found
- ◆ **kilépés a shellből**  
exit [n]

## Shell belső parancsok 2.

- ◆ ciklusból kilépés (több szintet is)

`break [n]`

- ◆ ciklus következő (akár több) iterációja

`continue [n]`

- ◆ függvény visszatérési értéke

`return [n]`

- ◆ változók exportja (környezeti változók)

`export [name=[word]] ...`

- ◆ változók attribútumai (nem POSIX)

`declare [-frxi] [name=value]`

`typeset [-frxi] [name=value]`

`readonly [-f] [name...]`

- ◆ változók törlése

`unset [name...]`

## Shell belső parancsok 3.

- ◆ új változók automatikus/nem automatikus exportja

```
set -a
```

- ◆ pozicionális paraméterek megváltoztatása

```
set arg1 arg2 ...
```

- ◆ pozicionális paraméterek léptetése

```
shift [n]
```

- ◆ terminálódott háttérprocesszek jelzése  
azonnal/következő promptnál

```
set -b
```

- ◆ file-név helyettesítés letiltása/engedélyezése

```
set -f
```

- ◆ command line editor beállítása

```
set -o vi
```

- ◆ szignálok kezelése

```
trap [command] [sigspec]
```

# Shell beépített parancsok 1.

- ◆ **rövidítések (alias) megadása, kiírása**

```
alias [name=value]
```

- ◆ **alias megszüntetése**

```
unalias [name]
```

- ◆ **jobok kiírása**

```
jobs
```

- ◆ **job háttérbe helyezése**

```
bg [jobspec]
```

- ◆ **job előtérbe helyezése**

```
fg [jobspec]
```

- ◆ **processz befejezésének megvárása**

```
wait [pid]
```

- ◆ **szignál küldése egy job-nak**

```
kill [-signspec] [pid|jobspec]
```

```
kill -HUP 14745
```

## Shell beépített parancsok 2.

- ◆ **parancs futtatása**

`command command`

- ◆ **directory váltás**

`cd [dir]`

- ◆ **aktuális directory kiírása**

`pwd`

- ◆ **változó értékének interaktív olvasása**

`read [name...]`

- ◆ **új file létekezésének maszkja**

`umask [mode]`

- ◆ **kiírás a standard outputra**

`echo [-neE] [args...]`

- ◆ **kilépés a login shellből**

`logout`



## Shell beépített parancsok 3.

- ◆ **directory stack kiiratása**

`dirs`

- ◆ **directory stack növelése**

`pushd [dir]`

- ◆ **directory stack csökkentése**

`popd [+/-n]`

- ◆ **directory stack rotálása**

`pushd +/-n`

- ◆ **aritmetikai kifejezés kiértékelése**

`let expr ...`

- ◆ **rövid információ a beépített(!) parancsokról**

`help pattern`

- ◆ **command history kiiratása**

`history`

# Példák 1.

- ◆ Jelezzük, ha péntek, 13-a van:
- ◆ 1. változat

```
set `date`  
if test $1 = 'Fri'  
then  
if test $3 = '13'  
then echo Vigyazz!  
fi  
fi
```

- 2. változat

```
alma=`date '+%a%d'`  
if test $alma = 'Fri13'  
then echo Vigyazz  
fi
```

## Példák 2.

- ◆ **file-ok átnevezése \*.htm-ről \*.html-re**

```
for fn in *.htm
do
mv $fn $(basename $fn .htm).html
done
```

- **file-ok összecsomagolása**

```
echo '# To unbundle, sh this file'
for i
do
    echo "echo $i 1>&2"
    echo "cat >$i <<'End of $i'"
    cat $i
    echo "End of $i"
done
```