

## Unix történet

- eredetileg két vonal: System V - BSD
  - jelenleg összemósódik
- 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

szandi@hit.bme.hu

1

## A Unix felépítése



szandi@hit.bme.hu

2

## 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

szandi@hit.bme.hu

3

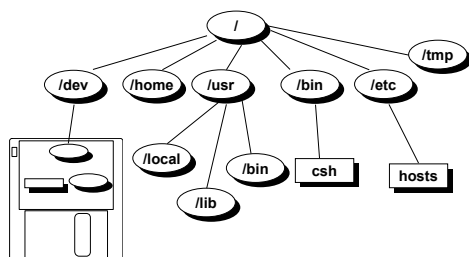
## Unix felhasználói felületek

- 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 (sort értelmez), raw (nincs értelmezés)
- X-Window rendszer
  - ablakozós, többtaskos felület (terminál ablakok)
  - window manager-ek
    - » beállítások a felhasználó izlése szerint
- man - online help
  - man *command*

szandi@hit.bme.hu

4

## Logikai file rendszer egységes directory struktúra



szandi@hit.bme.hu

5

## 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

szandi@hit.bme.hu

6

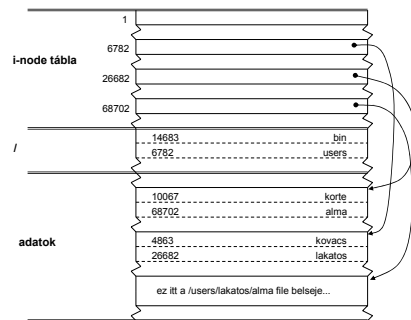
## File név konvenciók

- A könyvtárak (directory) speciális file-ok
  - tartalmuk file bejegyzés
- Gyökér 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

szandi@nit.bme.hu

7

## Egy Unix file rendszer belső szerkezete



szandi@nit.bme.hu

8

## 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

szandi@nit.bme.hu

9

## Segédprogramok file és directory kezelés

- Általános forma
  - parancs -optiók 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

szandi@nit.bme.hu

10

## 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 többszlopos listában a rendezés vízszintesen
  - » -1 egyszlopos lista

szandi@nit.bme.hu

11

## File parancsok 1.

- file tartalmát írja ki a standard kimenetre
- `cat -options filename`
  - » -e mint -v, és minden sor végére \$
  - » -s nincs hibüzenet
  - » -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

szandi@nit.bme.hu

12

## 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 névén
    - » `-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

szandi@hit.bme.hu

13

## 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`

szandi@hit.bme.hu

14

## 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 sbinscript
-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 tftboot
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 />

```

szandi@hit.bme.hu

15

## Jogok 1.

```

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

```

- owner  
 - group  
 - others  
 → érvényesítés

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

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

szandi@hit.bme.hu

16

## Jogok 2.

- **Olvadás (r)**
  - file: a file (tartalma) olvasható
  - dir: a directory tartalma megtekinthető, azaz a directoryban 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 directory tartalma írható, azaz új file-ok hozhatók létre, illetve meglévők törölhetők
- **Execute/search (x)**
  - file: a file végrehajtható (bináris vagy script)
  - dir: a directoryban lehet keresni, azaz konkrét file-ra lehet hivatkozni (ha arra van értelemszerű jog), de a directory teljes tartalma nem olvasható

szandi@hit.bme.hu

17

## 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)

szandi@hit.bme.hu

18

## 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 létrehozásakor**
  - `adattfile: rw-rw-rw-`
  - `futtatható file: rwxrwxrwx`
  - fenti maszkolása (szűkítése): `user mask`
  - `umask` (alapértelmezés: `022`, azaz `rwxr-xr-x`)

szandi@nit.bme.hu

19

## Jogok megváltoztatása

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

szandi@nit.bme.hu

20

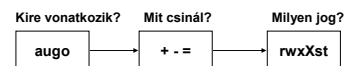
## Jogok megadása abszolút módon (összeadandó, oktális)

» 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

szandi@nit.bme.hu

21

## Jogok megadása szimbolikusan



<b>u</b> user (tulajdonos)	<b>+</b> hozzáad	<b>r</b> olvasás
<b>g</b> group (csoport)	<b>-</b> letilt	<b>w</b> írás
<b>o</b> others (többiek)	<b>=</b> pontosan beállít	<b>x</b> végrehajtás/ keresés
<b>a</b> a fentiek együtt		<b>X</b> keresés/ végrehajtás ha már van másik
		<b>s</b> setuid/setgid
		<b>t</b> sticky bit

szandi@nit.bme.hu

22

## Felhasználók

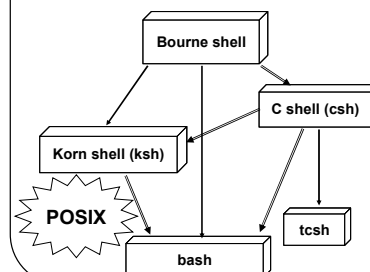
- **felhasználó (csoport) ID kiírása**
- `id [[-G] | [-g] | [-u]] [-n] [user]`
  - » `-G` összes GID (real, effective, supplementary)
  - » `-g` csak effektív GID
  - » `-u` csak effektív UID
  - » `-n` ID helyett név
- **aktív felhasználó**
  - `id`
    - » összes ID (real, effektív is ha eltér) és 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

szandi@nit.bme.hu

23

## Shell

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



szandi@nit.bme.hu

24

## Shell meghívása

- login shell
  - `chsh shell`
- végrehajtott scriptek

	<code>csh</code>	<code>bash</code>	<code>POSIX</code>
<b>login</b>	<code>/etc/login</code> <code>~/.login</code>	<code>/etc/profile</code> <code>~/.bash_profile</code> vagy <code>~/.bash_login</code> vagy <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`

szandi@ht.bme.hu

25

## 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

szandi@ht.bme.hu

26

## 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`

szandi@ht.bme.hu

27

## 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 visszakapjuk a `command-prompt`t
  - csoportosítás: `(date; ls) > lista &`
    - » a zárójeles rész külön fut, outputjuk összegződik

szandi@ht.bme.hu

28

## 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``

szandi@ht.bme.hu

29

## Operátorok

- logikai igaz - sikeres végrehajtás
  - exit status 0
- pipeline `c1 | c2`
  - `c1` outputja `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` indul a háttérben
  - `c2` futása azonnal elkezdődik

```
|
|
```

```
&&
||
```

```
;
&
```

→ precedencia →

szandi@ht.bme.hu

30

## 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***
```

szandi@ht.bme.hu

31

## 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**  
• alma=korte

szandi@ht.bme.hu

32

## 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

szandi@ht.bme.hu

33

## 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

szandi@ht.bme.hu

34

## (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 számú
- 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

szandi@ht.bme.hu

35

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

- BRE és még ...
- újabb speciális karakterek: + ? | ( )
- ( ) kifejezések zárójeljezhetők (csoportosíthatók)
- r+ r kifejezés 1 vagy tetszőleges számú
- r? r kifejezés 0 vagy 1 számú előfordulással
- r<sub>1</sub>/r<sub>2</sub> r<sub>1</sub> vagy r<sub>2</sub>
- **precedencia:**
  - [ ]
  - ( )
  - \* + ?
  - |
- **helyettesítő stringben:**
  - & az aktuális illeszkedő reguláris kifejezés
  - \ i az i-edik (1-9) \ (r\ ) -al definiált kifejezés

szandi@ht.bme.hu

36

## 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:

	<b>előre</b>	<b>vissza</b>
• egy sor	<code>enter</code>	<code>k</code>
• egy képernyő	<code>space</code>	<code>b</code>
• file vége	<code>g</code>	<code>G</code>
• minta	<code>/regexpr</code>	<code>?regexpr</code>
• következő minta	<code>n</code>	<code>N</code>
- 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 bővíthet)

szandi@hl.bme.hu

37

## 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 [-ivnc1s] regex [file...]`
  - » `-i` ignore case (kis-nagybetű nem különbözik)
  - » `-v` azokat a sorokat írja ki, melyek nem illeszkednek
  - » `-n` minden sort elején lesz file-név és sorszám
  - » `-c` csak az illeszkedő sorok számát írja ki
  - » `-l` csak a file(ok) nevét írja ki
  - » `-s` olvashatatlant 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`

szandi@hl.bme.hu

38

## 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`

szandi@hl.bme.hu

39

## Rendezés

- `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` szóárrendezés: csak betűk és számjegyek
    - » `f` kis- és nagybetűk nincsenek megkülönböztetve
    - » `i` nem ASCII karakterek figyelmen kívül hagyva
    - » `M` hónapsorrend (angol, a nem jök 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`

szandi@hl.bme.hu

40

## 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

szandi@hl.bme.hu

41

## 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 {} \;`

szandi@hl.bme.hu

42

### További segédprogramok 1.

- Sorok, szavak, karakterek számlálása
- **wc [-c|lw] [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

szandi@hit.bme.hu

43

### További segédprogramok 2.

- Karakterek cseréje
- **tr [-cds] 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 kiválogatása (számozásuk 1-től)
- **cut -c|-f [-dc] lista [file...]**
  - » -c karakterek kiválogatása
  - » -f mezők kiválogatása
  - » -dc c a mezőhatároló (alapértelmezés TAB)
  - » lista mező sorszámok, tartományok, pld: 1,3-5,12-

szandi@hit.bme.hu

44

### További segédprogramok 3.

- file-név/directory-név rész kiírása
- **basename/dirname str1 [str2]**
  - » str1 a teljes file-név elérési úttal
  - » str2 a kiterjesztés (suffix) (basename esetében)
- file-ok típusának megállapítása
- **file [-L] [-f file1] file...**
  - » -L szimbolikus link követése
  - » -f file1-ből is olvassa a file-ok listáját
- egymás utáni ismétlődő sorok
- **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
- file feldarabolása (xaa, xab, ... xaz, xba, ...)
- **split [-l lines] [file [name]]**
  - » -l az egyes darabok sorainak száma (alapért. 1000)
  - » name a darab nevek előtagja (prefix)

szandi@hit.bme.hu

45

### File-ok összehasonlítása

- rendezett szövegfile-ok összehasonlítása, kiírása
- **comm [-123] file1 file2**
  - » -1 csak az első file-ban előforduló sorok
  - » -2 csak a második file-ban előforduló sorok
  - » -3 mindkét file-ban előforduló sorok
- pontos összehasonlítás (csak a különbség kiírása)
- **cmp [-s] file1 file1**
  - » -s nincs semmilyen kiírás
  - » visszatérési érték: 0,1,2: a két file azonos, különböző, hiba
- összehasonlítás, mintha két verziója lenne egy file-nak
- **diff [-eb] file1 file2**
  - » -e kiír egy scriptet, melyet az ed editorral futtatva,
  - » file1-re alkalmazva file2-t kapjuk
  - » -b az ismétlődő white space-eket egy szóköznek tekintti
  - » visszatérési érték, mint cmp
  - » van sok más opció is

szandi@hit.bme.hu

46

### Stream editor (sed) 1.

- minden sorra végrehajtja a „szerkesztési” parancsokat
- **sed [-nfe] [command] [infile]**
  - » -n elnyomja az eredeti outputot
  - » -f file a parancsokat sfile tartalmazza
  - » -e a parancsok az utasításokban vannak
- parancsok: [c1m], [c1m2] művelet [argument]
- cím
  - sorszám (1-től kezdődik)
  - (basic) reguláris kifejezés
- műveletek
  - a\szoveg
    - » szoveg kiírása mielőtt a következő sort olvasná (append)
  - c\szoveg
    - » az input buffer cseréje szoveg-re
  - d
    - » az input buffer nem kerül az outputra (delete)

szandi@hit.bme.hu

47

### Stream editor (sed) 2.

- parancsok ...
  - i\szoveg
    - » szoveg kiírása az input buffer előtt (insert)
  - q a sed futása befejeződik (quit)
  - s/minta/cserere/flag
    - » minta-ra illeszkedő rész cseréje cserere-re
    - » flag (opcionális)
      - g global (az összes mintára illeszkedő rész cseréje)
      - p a mintapuffer kiírása
  - y/str1/str2/
    - » minden str1-ben előforduló karaktert a megfelelő str2-beli karakterre konvertál (str1 és str2 azonos hosszúságú)
  - w file
    - » a puffer tartalmát file-ba írja (append)
  - r file
    - » mielőtt a következő sort olvassa, file tartalmát kiírja
- Példák
  - ez első üres sorig írjuk ki az inputot: sed -n '1,/^\$/p'
  - első két szó cseréje: sed 's/\([^\ ]\*\) \*\(\([^\ ]\*\)/\2 \1/'

szandi@hit.bme.hu

48