



`cmp /etc/passwd /etc/passwd` : összehasonlít v. mit v. mind  
(csak annyit tájékoztat, hogy hol különbözik)

`diff`: részletesebb összehasonlítás (részletes kiírást ad)

[SHIFT+PAGE UP - pal minden visszatekinthető]

`file, mi` : leírja a jellemzőket

pl.: `file passwd` : leírja a passwd jellemzőit

`locate, mi` : file-ok keresése (az összes olyan file-t felsorolja, ami tartalmazza a karakterstringet)

`find, mit` : keresés

`find -name test.txt` : a gyökérből keresse a test.txt nevű fájlt

`find -type f` : az összes file-t leírja

`find -type d` : könyvtárakat sorol

`find -type d -name test` : keresem azokat, amelyek a test

(jótudás: A legtöbb paraméterezhető)

`man find` : leírja a paramétereket

`info, man` : paraméterekről nyújt segítséget

tömítés: (rar, zip : megtalálható és használható)

↳ gzip

[ha kiválaszt (ls -l) : felhasználó, user csoport, byte, dátum]

GZIP : 1 file tömítésére alkalmas

pl.: `gzip passwd` ⇒ `passwd.gz`  
          ↓                          ↓  
          14908 byte              4723



`rm *`: mindent töröl, ha van hozz jog

Nem lehet a letörölt dolgokat visszaköni!!!

### Linkek használata:

link: olyan bejegyzés a tartalomjegyzékben, amely egy másikra mutat  
1 file-t tudunk esáltal más néven eleni → csak a tartalomjegyzékben  
van.

ln: linkelési parancsa (`ln file új hivatkozási név`),

pl.: `ln passwd hardlink` → minden ugyanaz, még a relációs dátu-  
ma is

`cat passwd`: írja a passwd-t

`cat hardlink`: -"- hardlink-et

`rm passwd`: törli a passwd-t

`cat passwd`: írja, de nincs passwd

`cat hardlink`: írja a hardlink-et

A link elkészése nem war 1 plusz bejegyzés. A tartalomjegyzékben  
van egy linkszám, ami megmutatja, hogy hányan hivatko-  
zár ana a bejegyzésre.

Korlátja: nem tud másik partícióban (disken) lévő adatra mutatni

A "hardlink" megkülönböztethetetlen az eredetivel.

Az új bejegyzés mutatja, hogy "softlink". létrehozásakor létrejön egy  
file. Van mérete (pl.: 6 byte → mert a passwd 6 betű). Minden sorlá-  
tot levethet. Lehet másik disken mutatni ill. könyvtárra mutatni.

softlink létrehozása: `ln -s file mutat, ami rámutat,`

pl.: `ln -s passwd softlink`

`cat softlink`: írja a softlink tartalmát

No such file or directory: Ha létező a passwd-t, akkor nem létezik, így hibát jelez.

Átirányítás: Amikor a parancsok kimenetét nem a képernyőre, hanem másra akarjuk irányítani.

pl.: Ha a passwd-t a nyomatatra akarjuk tenni:

```
cat passwd > /dev/lp0
```

szálgos  
egyigben  
van

nyomatás

pl.: `ls -l > tartalom.txt`

a képernyőn semmi nincs, de elrotás után látható, hogy keletkezett egy file (tartalom.txt)

- ha még egy tartalmat átirányítunk, akkor az új tartalommal felülírja.

- ha még akarjuk tartani a már létező tartalmat is:

```
ls -l >> tartalom.txt
```

pl.: `tar cvf test.tar > tarout.txt`

az eddigi file-é lesz  
összeírva a szalagra

minden, ami a kimenetre került  
vagy, ide kerül bele

tar file is the archive →

→ a hibázzon a taroutban jelent meg.

standard hiba: `tar cvf test.tar . 2 > tarout.txt`

```
cat . tarout.txt
```

Időközben feltársa van ilyen, error e-mailt küld a net, ha est nem akarjuk, 0-ba kell irányítani.

`ls > /dev/null` → mindent, ami ide kerül, a unix eldobja.

`cat /dev/zero` : ha olvasás mindig  $\emptyset$ -t kapunk (azért használják, amikor egy disket  $\emptyset$ -val töltünk fel  $\rightarrow$  szétpermettje)

Ha a `cat` használatakor nem adunk meg file nevet, a file-t a standard inputból veszi.

`cat < /etc/passwd` : előveszi a file-t, és olyan, mintha a tartalmából beépeltek volna

### Környezeti változók:

`T=alma`  $\Rightarrow$  létrejött egy "T" nevű változó, tartalma "alma"

ha hivatkozom az értékre:

`echo $T`  $\Rightarrow$  alma

`echo T`  $\Rightarrow$  T

pl.: `AL=6` és `MA=8`

`echo $AL$MA`  $\Rightarrow$  68

`echo $ALMA`  $\Rightarrow$  6MA

`echo ALMA`  $\Rightarrow$  semmit nem ír ki, mert nem volt ilyen változó

### sámság:

`#`: írja a hivatkozás hosszát

pl.: `T=ALMA`

`echo ${#T}`  $\Rightarrow$  4 (alma 4 betű)

`echo ${T:2:1}`  $\Rightarrow$  (ALMA) "M"  
                  /        |  
                  innen  egyet  1 2 3 4

T változó 2. karakterétől iratunk ki egyet

`echo ${T:0:1}`  $\Rightarrow$  "A"

pl.: T = date  
echo \$T ⇒ "DATE"-t ír ki az aktuális dátum helyett

T = `date`  
echo \$T ⇒ kiírja az akt. dátumot

(Mj.: ` ⇒ altgr #)  
Előbb végrehajtja a parancsot, majd beírja a változóba

pl.: T = `ls`  
T = `ls -l` : eldobja a sorvégetet, nem tudja kezelni

pl.: T = 6  
echo \$T ⇒ "6"  
echo "Ez a szám a \$T" ⇒ Ez a szám a 6.  
echo `Ez a szám a \$T` ⇒ Ez a szám a \$T

\* ` - jel söréggé kényszeríti azt, amit követog.

pl.: ls -la /\* \*.txt  
↳ elnyomja a \* hatását ⇒ azt írja ki, ami a\*-gal kezdődik

## Szűrők:

~: több parancs összekapcsolódása (A parancsok előtt „|”-t teszünk)

pl.: `ls | more` (oldalra törni) → új sorra új oldal jelenik meg,  
↳ alt W ha vége, a more is kilép

Az egyes parancs kimenetele nem megy a standard output-ra, hanem a másik program kimenetelére érkezik.

pl.: `ls | sort` → ABC-rendben jelenik meg.

pl.: `cat /etc/passwd | grep bubu` → azok jöttek, amik tartalmazzák a „bubu”-t.

`grep`: a tartalomban szóként ellenőriz

pl.: `cat /etc/passwd | grep .false` → amikben szerepel a „false”. (vhol a sorban)

`false`: nem létezik be a rendszerbe

`grep -v`: tagadás (amelyek nem tartalmazzák)

pl.: `cat /etc/passwd | grep -v bash` → amelyekben nincs „bash”  
`cat /etc/passwd | grep ^to` → ezzel kezdődő sorokat keresünk.  
- ” - | `grep to$` → ezzel végződő - ” -

`grep -i`: ne vegye figyelembe a kis-és nagybetűt közti különbséget.

pl.: `cat /etc/passwd | grep -i account`

`wc`: számlálás

pl.: `cat /etc/passwd | wc`

253  
↓  
sor

477  
↓  
szó

15188  
↓  
karakter



<code>wc -l</code> (line) sorok száma adja meg	<code>cat/etc/passwd   wc -l</code>
<code>wc -w</code> (word) szavak	<code>"   wc -w</code>
<code>wc -c</code> (character) karakterek	<code>"   wc -c</code>

pl.: Hány olyan felhasználó van, akik nem tudnak bejelentkezni?

`cat/etc/passwd | grep false $ | wc -l`  $\Rightarrow$  válasz: "10"

Ha beírjuk 1 változóba:

`T='cat/etc/passwd | grep false $ | wc -l'`  
`echo $T`  $\Rightarrow$  "10"

A mesőelválasztó karakter a ":", ezt fejezi ki a "-d"

pl.: Felcímek meg az első meső!

`cat/etc/passwd | cut -d: -f1`

Felcímek meg 1,5 meső!

`... | cut -d: -f1,5`

Felcímek meg 1-5 meső!

`... | cut -d: -f1-5`

Ha védeni akarjuk: ' ' kőre kell tenni, mert különben átíranyítja

pl.: `... | cut -d ' ' -f1-2`

pl.: `... | cut -c3`  $\Rightarrow$  összes sorból a 3. karaktert vágja ki

`... | cut -c1,2,3`  $\Rightarrow$  " " " " " "

`... | cut -c1-3`  $\Rightarrow$  " " " "

`... | cut -c1-8,12-14`  $\Rightarrow$  " " " " " " " " " "

pl.: `... | head`  $\Rightarrow$  az első 10 sort adja vissza

`... | tail`  $\Rightarrow$  az utolsó " " " "

`... | tail -n 2`  $\Rightarrow$  " " " " " "

`... | head -n 1`  $\Rightarrow$  az első " " " "

tr: karaktérket csere

pl.: cat/etc/passwd | tr 'a' 'x' → admin ⇒ xadmin

pl.: cat/etc/passwd | tr 'admin' '12345' → xrest ⇒ xlest

pl.: " " | tr -d 'a' → az összes "a" betűt törli

pl.: " " " "

pl.: " " | cut -d: -f1 | grep f

Azokat adja vissza, akiknek a bejelentkezési névben van "f".

pl.: " " | cut -d: -f1 | grep ^f

Azok a jöl, amelyek "f"-fel kezdődnek

pl.: " " | cut -d: -f1 | grep ^f\$

Azok az uszervek felelnek meg, amelyek "f"-fel kezdődnek és végződnek

pl.: " " | cut -d: -f1,5 | grep ^fatman: | cut -d: -f2

→ fatman: Kiss Bence → Kiss Bence

A "fatman" uszerveket keressük és a teljes nevet hatjuk ki.

pl.: " " | cut -d: -f1 | grep man

Azok jelennek meg, ahol az uszervekben benne van a "man".

grep mint alapján: valami alapján keres

last: azok jelennek meg, akik utoljára jelentkezték be

cut: visszadja az adott mezőt

pl.: last | cut -d: -f1 | sort: megvárja az egészet, majd rendszel

pl.: last | cut -d: -f1 | sort | uniq

↳ ha 1 elem többször fordul elő, csak egyszer írja ki

Csár rendezett adathalmazként működik jól, így rendezni KELL!!!

pl.: cat/etc/passwd | cut -d: -f3 | sort | more

↳ kijelöl a 3. mezőt ↓ olddalakra bontja

rendszel

karaktérként

pl.: cat /etc/passwd | cut -d: -f3 | sort -n | more  
↓  
számszerű rendezés

pl.: cat /etc/passwd | cut -d: -f3 | sort -n | head -n 1  
↓  
csak az első elem

pl.: cat /etc/passwd | cut -d: -f3 | sort -n -r | head -n 1 >  
/tmp/a.txt

bármilyen file  
megnevezéssel: cat /tmp/aa.txt  
65534

expr.: művelet végrehajtása

pl.: expr 3 + 2 ⇒ 5

expr 3 / 2 → 1

csak egész számokat használ  
(korlátozott művelet)

pl.: expr '3\*2' → nem jó

expr 3 \* 2 → nem jó

expr 3 \\* 2 → a jel a karakter hatását megszünteti így \* lesz.