

## Tartalomjegyzék

- 1 A Linux fájlrendszere
  - ◆ 1.1 Speciális könyvtárak
  - ◆ 1.2 Fájrendszer-böngészők
- 2 Jogok
- 3 Bash shell
  - ◆ 3.1 Néhány hasznos gyorsbillentyű
  - ◆ 3.2 Kimenet átirányítása
  - ◆ 3.3 Hasznos parancsok
- 4 Szövegszerkesztők (editor)
  - ◆ 4.1 Billentyűkombinációk

### A Linux fájlrendszere

Mindenkinek van egy saját mappája a `/home/student/felhasznaloi_nev` alatt. Ebben 50 MB-ot tárolhattok kedvetekre.

- Fontosabb parancsok
  - ◆ **pwd** : aktuális könyvtár kiírása (**print working directory**)
  - ◆ **cd** könyvtár : könyvtárváltás (**change directory**). Ha paraméter nélkül hívjuk meg, akkor a saját mappánkba visz.
  - ◆ **ls**, **ls -a**, **ls -l**, **ls -la** : listázás, a paraméterek hatására a rejtett fájlokat (amelyek `.-`-tal kezdődnek) is listázza, illetve részletesebb adatokat is kiír.
  - ◆ **cat**, fájl tartalmának kiírása, pl: `cat valami.txt`. Ha nem adunk meg fájlnevet, akkor a billentyűzet vár bevitelt. Kilépés: Ctrl-D-vel. Több fájl összefűzésére és kiírására is használható. (**catenate** vagy **concatenate**)
  - ◆ **mkdir** : könyvtár létrehozása, pl: `mkdir alma` (**make directory**)
  - ◆ **cp** : másolás, pl: `cp mit.txt hova.txt` (**copy**)
  - ◆ **mv** : mozgatás (az eredeti nem marad meg), pl: `mv mit.txt hova.txt` (**move**)
  - ◆ **rm** : törlés, pl: `rm alma.txt`. Könyvtár esetén további kapcsolókat kell megadni: `rm -fr alma`. *Óvatosan használjuk!* (**remove**)
  - ◆ **quota** : az 50 MB-ból felhasznált/megmaradt helyet írja ki. Ha a kvótánk betelik, akkor nem kapjuk meg a nekünk címzett leveleket és a grafikus terminálra sem tudunk belépni; ilyenkor a parancssoros terminálra lépve tudjuk kitakarítani a home-unkat.
  - ◆ **man** parancs : az adott parancsról leírást ad, pl: `man ls`. A leírásban navigálni a 'fel', illetve a 'le' billentyűkkel, továbbá a 'PageUp', 'PageDown'-nal lehet. Kereséshez '/' jel után kell beírni a kifejezést. Kilépés 'q'-val. Bocs, angol. (**manual**)
  - ◆ **history** : parancs történet lekérdezése
  - ◆ **su** : felhasználó váltás, pl: `su tothagi` (substitute **user** vagy switch **user**)

### Speciális könyvtárak

jel	jelentés	környezeti változó
.	aktuális könyvtár	PWD

..	szülőkönyvtár	
~	felhasználó könyvtára	HOME
~<név>	<név> felhasználó könyvtára	

## Fájlrendszer-böngészők

Próbáljátok ki a következő böngészőket:

- nautilus
- konqueror
- mc - Midnight Commander

## Jogok

Az `ls -la` parancs részletes listát készít, amelynek első oszlopa egy 10 karakterből álló sztring.

(A következő mező az ún. link számláló, tulajdonképpen megmondja, hogy hány néven hivatkozhatunk a fájlra, számunkra nem fontos. Ezt követi a fájl tulajdonosának és annak csoportjának a neve, majd a fájl mérete és utolsó módosításának az ideje következik, végül a fájl nevét olvashatjuk.)

```
-rw-r--r-- 1 morap student 2388 2007-07-30 03:51 .bashrc
-rw-r--r-- 1 morap student 117660 2007-08-19 17:36 .B.blend
drwxr-xr-x 5 morap student 4096 2007-08-19 17:35 .blender
drwx----- 5 morap student 4096 2007-08-21 20:50 .cache
drwx----- 2 morap student 4096 2007-08-28 00:23 .chewing
```

Az első betű utal a típusra, '-' esetén fájlról, 'd' esetén könyvtárról van szó. Az azt követő hármast a user (felhasználó, továbbiakban *u*), a következő hármast a group (felhasználó csoportja, továbbiakban *g*), majd az others (mindenki más, továbbiakban *o*) jogairól szól. Szokás még

"<http://wiki.math.bme.hu>" "<http://wiki.math.bme.hu>-val jelölni az all-t, vagyis mindenkit. Az *r* olvasás, a *w* írás, az *x* pedig végrehajthatóságot/futtathatóságot (execute) jelent. Fájlok esetében a végrehajtási jognak csak a futtatható fájlknál van jelentősége (bináris állományok, scriptek). Könyvtárak esetén az olvasási jog azt jelenti, hogy elolvashatja a fájl neveit az adott könyvtárban, az írási jog jelenti, hogy a könyvtárban állományt, könyvtárat hozhatunk létre, míg a futtatási jog megengedi a belépést a könyvtárba.

Változtatás `chmod` paranccsal.

Jogok állítására példa:

```
chmod "http://wiki.math.bme.hu"og-w"http://wiki.math.bme.hu file # leveszi az írásjogot a csoporttól
chmod "http://wiki.math.bme.hu"+x"http://wiki.math.bme.hu file # mindenki számára futtathatóvá teszi
chmod g=rx,o=r file # a csoport olvasási és futtatási jogot kap, mindenki más olvasásit
```

A hármast 0 és 7 közötti számokkal is kódolhatjuk, kettes számrendszerben elképzelve őket (a jog megléte jelenti az 1-et, hiánya a 0-t; az olvasás a 4-es, az írás a 2-es, az végrehajtás az 1-es helyiérték). Így a következő parancs hatására a felhasználó minden jogot megkap a file-hoz ( $7 = 111$ ), míg a csoport és mindenki más csak olvashatja és végrehajthatja azt ( $5 = 101$ ).

```
chmod 755 file # 755 = 111101101
# rwxr-xr-x
```

## Bash shell

A *bash* tulajdonképpen a legelterjedtebb *Linux shell*. A begépett parancsokból egy fájlt hoz létre ami elősegíti a parancsok szerkesztését. A shell legfontosabb funkciói közé tartoznak a *bash gyorsbillentyűk*. Ezek több

billentyű egyidejű lenyomásával azonnal végrehajtható, egyszeri utasítások, melyek segítik a bash parancsok kezelését karakteres felületen és terminálablakban.

**Jelölések:** gyakran jelölik a *Control* billentyűt C-vel, az *Alt* billentyűt M-mel (Meta), az *Enter* E-vel, a *Shift* billentyűt S-sel (én az egyszeriség kedvéért a szokásos jelöléseket használom).

### Néhány hasznos gyorsbillentyű

- *Ctrl-c* : megállítja a futó programot
- *Ctrl-a* : sor elejére ugrik
- *Ctrl-e* : sor végére ugrik
- *Ctrl-l* : az utolsó parancs kivételével törli a képernyűt
- *Ctrl-r* : keresés a már beírt parancsok között
- "[http://wiki.math.bme.hu/fel/le\\_nyilak](http://wiki.math.bme.hu/fel/le_nyilak)" "<http://wiki.math.bme.hu> : sorban visszahívja a régebbi parancsokat/újabb parancsokat
- *TAB* : automatikus parancskiegészítés: ha nem emlékszünk egy parancs utolsó betűire, vagy túl hosszú ideig tart begépelni, csak írjuk be az elejét és nyomjuk meg a *TAB* billentyűt. Ez automatikusan kiegészíti a parancsot (esetleg könyvtár- vagy programnevet). Ha több kiegészítési lehetőség is van, akkor először nem történik semmi, a *TAB* második lenyomására az összes azonosan kezdődő parancsnév megjelenik. Gépeljünk be még annyi betűt a parancsból, hogy az egyértelmű legyen, majd használjuk újra a *TAB* billentyűt.
- *Alt- /* : filenév-kiegészítés
- *Alt-~* : felhasználónév-kiegészítés
- *Shift-PgUp / PgDn* : az oldal tetejére/aljára ugrik
- *Ctrl-z* : háttérbe teszi az elfelejtett "<http://wiki.math.bme.hu>&" "<http://wiki.math.bme.hu> miatt otthagyt programot
- *Ctrl-d* : kilép a shell-ből (logout)

---

A *bash* gyorsbillentyűk használata rendkívül hasznos, meggyorsítja és megkönnyíti a számítógépes munkát, egyszeri, jól áttekinthető parancskezelést tesz lehetővé.

Bővebben a bash shellről: [www.szabilinux.hu/mandrake/007.html](http://www.szabilinux.hu/mandrake/007.html)

### Kimenet átirányítása

Gyakran előfordul, hogy el akarjuk menteni a program kimenetét, hogy valamit kezdjünk vele. Ezt hívjuk kimenet átirányításnak.

Például, ha rögzíteni szeretnénk az `ls /bin` parancs eredményét az `lsbin` nevű fájlba, gépeljük be:

```
ls /bin > /lsbin
```

Ha az `lsbin` fájl már létezik, a tartalma felülíródik. Ennek elkerülésére használjuk a `>>`-t az egyszeres `>` helyett (ez teszi lehetővé, hogy az `ls` kimenetét hozzáfűzzük az `lsbin` fájl végéhez).

A Linux lehetőséget biztosít egy elegánsabb megoldásra is. Ehhez úgynevezett csöveket fogunk használni (pipes). Ezzel a megoldással az egyik parancs a kimentét a másik parancs bemenetére adja át. Például:

```
ls /bin | less
```

Itt az `ls` kimenete a `less` (szöveges tartalmat megjelenítő) program bemenete lesz.

Különböz? csöveket kombinálhatunk, hogy összetett eljárásokat dolgozzunk ki.

### Hasznos parancsok

```
cat      kiírja egy file tartalmát
head     kiírja egy file elejét (-n kapcsolóval megadható a sorok száma, default: 10)
tail     kiírja egy file végét (hasonlóan mint a head)
sort     sorbarendezi a file sorait, a -n kapcsolóval számokként értelmezve,
        a -r kapcsoló megadásakor visszafelé rendez
wc       Word Count, megszámlolja a karaktereket, szavakat, sorokat a fájlban
uniq     ha több egyforma sor jön egymás után, akkor a uniq csak egyet hagy meg ezek közül;
        a uniq -c parancs megszámlolja az egymás utáni egyforma sorokat
diff     kiírja a megadott két file közötti különbséget
cut      oszloponként képes kivágni egy file-ból adatokat.
        Általában cut -d' ' -fx formában használjuk, itt a -d utáni karakter a mez?ket
        elválasztó karaktert jelenti (itt szóköz);
        a -f kapcsoló után pedig a mez?k számait kell megadni amiket szeretnénk megtartani.
        Példák:
        cut -d' ' -f1,3 az els? és harmadik mez?t írja ki,
        cut -d' ' -f1-4 pedig az els?t?l a negyedikig az összes mez?t.
```

### Szövegszerkeszt?k (editor)

Próbáljátok ki a 'gedit' és 'kate' szövegszerkeszt?ket!

Mindkett?ben van *syntax highlighting*, vagyis ismernek sokféle programnyelvet, és ezek kulcsszavait ki tudják emelni, hogy segítsék a programozók munkáját.

### Billenty?kombinációk

- *Ctrl-n* : új file
- *Ctrl-o* : file megnyitása
- *Ctrl-s* : a file mentése
- *Ctrl-c* : a vágólapra másolja a kijelölt szövegrészt
- *Ctrl-x* : az eredeti helyr?l kivágja, és a vágólapra teszi a kijelölt szövegrészt
- *Ctrl-v* : beilleszti a vágólap tartalmát