

Gyakorlati jegyzet - Szegedi Tudományegyetem - Természettudományi és Informatikai Kar:
Programozás alapjai gyakorlat

Készítette: Lékó Gábor, 2014 (észlelt hibákat a Leko.Gabor [at] stud.u-szeged.hu címre lehet küldeni)

Cél: Linux operációs rendszer kezelésének felhasználói szintű elsajátítása

Linux parancsok leírása a teljesség igénye nélkül (példákkal):

pwd
(print working directory)
Aktuális könyvtár elérési útvonala. pl.:
lekogabi@lekogabi:~/Letöltések/eclipse\$ pwd
/home/lekogabi/Letöltések/eclipse

history
Kiírja a terminal előzményeket.

cd <none,könyvtár,könyvtárszerkezet>
(change directory)
A könyvtár rendszerben való mozgást teszi lehetővé. Paraméterként a megcélzott könyvtár nevét kell megadni, vagy abszolút, vagy relatív elérési útvonal használatával. Az abszolút elérési útvonal megadásánál a keresett könyvtár teljes elérési útvonalát kell megadni. Tehát rootból indulva pl.: /home/hxxxxxx/mappa/file1 . Relatív címezés esetén azt mondjuk csak meg, hogy az aktuális könyvtárhoz képest hol helyezkedik el a megcélzott könyvtár. Tehát aktuális könyvtárból indulva pl.: ha éppen a /home/hxxxxxx-ben vagyunk: ./mappa/file1 . További példák:
aktuális könyvtár: .
szülőkönyvtár: ..

Abszolút címezés:
lekogabi@lekogabi:~\$
lekogabi@lekogabi:~\$ cd home/lekogabi/Letöltések/eclipse
lekogabi@lekogabi:~/Letöltések/eclipse\$

Relatív címezés:
lekogabi@lekogabi:~\$
lekogabi@lekogabi:~\$ cd ./Letöltések/eclipse/
lekogabi@lekogabi:~/Letöltések/eclipse\$

Visszalépés:
lekogabi@lekogabi:~/Letöltések/eclipse\$ cd ..
lekogabi@lekogabi:~/Letöltések\$ cd ..
lekogabi@lekogabi:~\$

lekogabi@lekogabi:~/Letöltések/eclipse\$ cd
lekogabi@lekogabi:~\$

ls
(list)
Ez a parancs megmutatja az aktuális könyvtárban található fájlok neveit. A lista alapértelmezés szerint abc sorrendben sorolja fel a neveket, de alkalmas kapcsolók segítségével más sorrendet is előírhatunk. pl.:
lekogabi@lekogabi:~/Letöltések/eclipse\$ ls

Eredményként kiíratódik a terminálban az 'eclipse' mappa tartalma.

Kapcsolók:

- l: oszlopokban mutatja az adatokat.
- d <könyvtár>: csak az adott könyvtár adatait írja ki.
- a: a rejtett fájlokat is mutatja.
- R: könyvtáron belüli könyvtárak tartalmát is listázza.
- r: fordított sorrendben listáz.
- h: "barátságosabb" formában listázza ki a számokat.

A kapcsolókat lehet kombinálva is használni, így pl.:

```
lekogabi@lekogabi:~/Letöltések/eclipse$ ls -l -a -r
```

```
lekogabi@lekogabi:~/Letöltések/eclipse$ ls -lar
```

Mindkét parancs kifogja listázni oszlopokba rendezve, részletes adatokkal a rejtett fájlokat is, fordított sorrendben.

`mkdir [kapcsoló]<új mappa>`

(make directory)

Új könyvtár létrehozására (csak az adott könyvtárban). pl.:

```
lekogabi@lekogabi:~/Dokumentumok$ mkdir hello
```

Létrejön egy "hello" nevű mappa a 'Dokumentumok'-on belül.

Kapcsolók:

-p: Létrehozza a kívánt könyvtár eléréséig az összes szükséges könyvtárat. pl.:

```
lekogabi@lekogabi:~/Dokumentumok$ mkdir -p hello/hello2/hello3
```

Végig létrehozza a könyvtárszerkezetet.

-m: Megadhatjuk az új katalógus hozzáférési jogát oktálisan a mode értékének beállításával. pl.:

```
lekogabi@lekogabi:~/Dokumentumok/hello$ mkdir -m 777 hello2
```

```
lekogabi@lekogabi:~/Dokumentumok/hello$ ls -l
```

```
drwxrwxrwx 2 lekogabi lekogabi 4096 aug 5 17:05 hello2
```

A '777' megadásával teljeskörű jogokat biztosítunk mindhárom felhasználói csoportnak (user, group, other). A 3 db 7-es számjegy egyenkénti csökkentésével (0-7) lehet egyre kisebb jogokkal felruházni a felhasználói csoportokat.

`rmdir [kapcsoló]<törlendő mappa>`

(remove directory)

Könyvtárak törlésére szolgáló parancs. Az rmdir parancs csak üres könyvtárat vagy könyvtárakat töröl. pl.:

```
lekogabi@lekogabi:~/Dokumentumok$ rmdir hello
```

Eredményül kitörlődik a 'hello' nevű mappa.

Kapcsolók:

-p: A könyvtár törlése után, a szülő könyvtárat is törli, ha üres, rekurzívan.

A Dokumentumok mappán belül található egy hello nevű mappa és azon belül egy hello2 nevű mappa. Ha az aktuális könyvtár a Dokumentumok, akkor a -p kapcsoló segítségével, ha kitöröljük a hello2 mappát, akkor a hello mappa is törlődni fog.

```
lekogabi@lekogabi:~/Dokumentumok$ rmdir -p hello/hello2
```

`mv [kapcsoló]<forrás><cél>`

(move)

A fájlok állományrendszeren belüli mozgatására szolgáló parancs. Ha nem adunk meg fájlnevet a célnál, akkor nem változik meg a neve, különben a megadott fájlnev lesz a célkönyvtárban. pl.:

```
lekogabi@lekogabi:~/Dokumentumok$ mv hello /home/lekogabi/Letöltések
```

Eredményül a 'hello' nevű fájl átkerül a Dokumentumok könyvtárból a Letöltések könyvtárba.

Kapcsolók:

- b: biztonsági másolatot készít a forrásfájlról.
- f: sosem kérdez felülírásnál.
- i: kétes esetben kérdez (pl. névütközés...).

-u: csak régebbit ír felül.

cp [kapcsoló]<forrás><cél>

(copy)

Átmásolja a forrás fájlt a megadott helyre. pl.:

```
lekogabi@lekogabi:~/Letöltések$ cp image.png /home/lekogabi/Dokumentumok
```

Eredményül átmásolja a 'image.png' fájlt a Letöltések könyvtárból a Dokumentumok könyvtárba.

Kapcsolók:

-r: könyvtár egész tartalmának másolása.

-R: -//-

-b: minden célfájlról mentés.

-f: sosem kérdez.

-i: kétes esetben kérdez.

-u: csak régebbit írja felül.

-l: linkelés másolás helyett.

-s: szimbólikus linket készít.

rm [kapcsoló]<törlendő fájl(ok)>

(remove)

Kitörli az rm parancs után található fájlokat (könyvtárakat nem, arra az rmdir használandó, kivéve kapcsoló hozzáadásával - lásd lentebb). pl.:

```
lekogabi@lekogabi:~/Dokumentumok$ rm file1.txt file2.txt
```

Eredményül a file1.txt és a file2.txt törlésre kerül.

Kapcsolók:

-f: sosem kérdez.

-i: kétes esetben kérdez.

-r: könyvtár törlésére, akkor is törli ha nem üres.

-R: -//-

ln [kapcsoló]<forrás><cél>

(link)

Az első paraméter a fájl, amihez szeretnénk linket készíteni, a második paraméter a link neve. Kapcsoló nélkül hard link készítése.

Kapcsolók:

-s: Szimbólikus (soft) link készítése.

Hard link pl.:

```
lekogabi@lekogabi:~/Dokumentumok$ cat > file1.txt
```

hello

```
lekogabi@lekogabi:~/Dokumentumok$ ln file1.txt file2.txt
```

```
lekogabi@lekogabi:~/Dokumentumok$ ls -l
```

```
-rw-r--r-- 2 lekogabi lekogabi      6 aug   7 01:15 file1.txt
```

```
-rw-r--r-- 2 lekogabi lekogabi      6 aug   7 01:15 file2.txt
```

```
lekogabi@lekogabi:~/Dokumentumok$ cat >> file1.txt
```

world!

Létrehozrunk egy .txt fájlt 'file1' névvel, melybe a "hello" szöveget írtuk. Majd készítettünk egy hard linket a fájlunkról 'file2' névvel. Majd módosítottuk a file1-et, amelynek eredményeképp a file2 is módosult. Így mindkét .txt fájlban a "hello world!" szöveg fog szerepelni. Listázásnál a második mezo jelenti a fájlra mutató hard linkek számát.

Szimbólikus link:

```
lekogabi@lekogabi:~/Dokumentumok$ ln -s file1.txt file3.txt
```

```
lekogabi@lekogabi:~/Dokumentumok$ ls -l
```

```
-rw-r--r-- 2 lekogabi lekogabi     13 aug   7 01:16 file1.txt
```

```
-rw-r--r-- 2 lekogabi lekogabi     13 aug   7 01:16 file2.txt
```

```
lrwxrwxrwx 1 lekogabi lekogabi      9 aug   7 01:29 file3.txt -> file1.txt
lekogabi@lekogabi:~/Dokumentumok$ cat >> file1.txt
lekogabi@lekogabi:~/Dokumentumok$ rm file1.txt
lekogabi@lekogabi:~/Dokumentumok$ ls -l
-rw-r--r-- 1 lekogabi lekogabi     29 aug   7 01:34 file2.txt
lrwxrwxrwx 1 lekogabi lekogabi      9 aug   7 01:29 file3.txt -> file1.txt
```

A meglévő file1.txt fájlunkhoz készítünk egy szimbólikus linket. Listázás után ez jól megfigyelhető az elkészült file3.txt után látható nyíl alapján, amely a file1.txt-re mutat. A hard linkkel ellentétben, amennyiben töröljük a file1.txt-t, akkor a file3.txt használhatatlan lesz, mivel töröltük a fájlt, amire mutatott.

du

A fájljaink és könyvtáraink által elfoglalt lemezterületet lehet vele megtekinteni. pl.:

```
lekogabi@lekogabi:~/Dokumentumok$ du file2.txt
4 file2.txt
lekogabi@lekogabi:~/Dokumentumok$ du file2.txt hello
4 file2.txt
4 hello
```

Kapcsolók:

- a: minden fájl adatait kiírja a könyvtár(ak)on belül, nem csak a könyvtár(ak)ét.
- h: "barátibb" alakban a számok.
- m: megabájtban írja ki a méretet.
- s: csak összméret.

chmod

Az elérési jogok megváltoztatása a chmod paranccsal lehetséges a tulajdonos számára.

Szintaxisa pedig:

```
#1:
  chmod [ugoa][+][rwx] fájlnev
```

Az [ugoa] kapcsolókkal írjuk elo, hogy kinek adjuk a jogot. Adhatunk jogokat a tulajdonosnak (u - user), a csoportnak (g - group), másoknak (o - others) vagy pedig mindenkinek egyszerre (a - all). A [+][rwx] azt jelenti, hogy adhatunk (+) vagy elvehetünk (-) jogokat. Végül pedig, hogy olvasási (r - Read), írási (w - Write) vagy futtatási (eXecute) jogot adunk. pl.:

```
lekogabi@lekogabi:~/Dokumentumok$ ls -l
drwxrwxrwx 2 lekogabi lekogabi    4096 aug   9 15:07 hello
lekogabi@lekogabi:~/Dokumentumok$ chmod g-w hello
lekogabi@lekogabi:~/Dokumentumok$ ls -l
drwxr-xrwx 2 lekogabi lekogabi    4096 aug   9 15:07 hello
#2:
```

```
  chmod <xxx> fájlnev
```

Hasonló az előzőhöz, viszont itt minden x helyén 0-7-ig való osztályozással adható meg a különböző felhasználói csoportok jogosultágai.

Rendre egy darab x helyettesítésének jelentése:

0:	000	---	semmi jog
1:	001	--x	csak futtatási
2:	010	-w-	csak írási
3:	011	-wx	írási és futtatási
4:	100	r--	csak olvasási
5:	101	r-x	olvasási és futtatási
6:	110	rw-	olvasási és írási
7:	111	rwx	olvasási, írási és futtatási jog

pl.:

```
lekogabi@lekogabi:~/Dokumentumok$ ls -l
drwx---wx 2 lekogabi lekogabi    4096 aug   9 15:07 hello
lekogabi@lekogabi:~/Dokumentumok$ chmod 777 hello
lekogabi@lekogabi:~/Dokumentumok$ ls -l
```

```
drwxrwxrwx 2 lekogabi lekogabi 4096 aug 9 15:07 hello
```

Kapcsolók:

-R: rekurzívan az összes fájlra és alkönyvtárra.

finger <argumentum>

Kiírja, hogy kik vannak bejelentkezve az aktuális gépre.

felhasználó: a megadott felhasználóról ír ki adatokat.

who

Kiírja, hogy kik vannak bejelentkezve az aktuális gépre, plusz adatokat ír ki a finger-hez képest.

w

Ugyanaz, mint a who, csak kiírja, hogy min dolgozik.

ps

Kiírja a képernyőre az aktuális process-eket (futó vagy várakozó programokat, amik az adott terminálhoz kapcsolódnak).

-e: az összes futó process-t kiírja (más terminálhoz/terminálhoz nem kapcsolódtak is).

-f: több információ.

-u user: csak a megadott user által "birtokolt" processzek listázása.

jobs

Kiírja az aktuális jobokat, amik az adott terminálhoz kapcsolódnak.

^C

Ctrl+c paranccsal leállíthatunk előtérben futó folyamatokat.

^Z

Ctrl+z paranccsal szüneteltethetünk előtérben futó folyamatokat.

bg %<szám>

Várakozó job-ok elindítását teszi lehetővé, a "szám" helyére a job azonosítója kerül, háttérben indítja el (konzolt nem "használja").

fg %<szám>

Várakozó job-ok elindítását teszi lehetővé, a "szám" helyére a job azonosítója kerül, előtérben indítja el (konzolt "használja").

kill %<szám>/PID

Process-eket és job-okat tudunk vele leállítani.

%<szám> formátumnál a szám helyére a job azonosítóját kell írni.

PID a process azonosítójának felel meg, és ezt a process-t szakítja meg (csak akkor, ha fut).

Kapcsolók:

-9: a leállított process-t is megszakítja.

-s: úgy állítja le a job-ot, hogy még újra lehet indítani.

példa a job-ok kezelésére:

```
lekogabi@lekogabi:~$ cat > file.txt
```

```
hello
```

```
^C
```

```
lekogabi@lekogabi:~$ cat > file.txt
```

```
hello
```

```
^Z
```

```
[1]+ Megállítva          cat > file.txt
```

```
lekogabi@lekogabi:~$ jobs
```

```
[1]+ Megállítva          cat > file.txt
lekogabi@lekogabi:~$ fg %1
cat > file.txt
world
^Z
[1]+ Megállítva          cat > file.txt
lekogabi@lekogabi:~$ jobs
[1]+ Megállítva          cat > file.txt
lekogabi@lekogabi:~$ kill %1
```

```
[1]+ Megállítva          cat > file.txt
lekogabi@lekogabi:~$ jobs
[1]+ Befejeződött        cat > file.txt
lekogabi@lekogabi:~$ jobs
lekogabi@lekogabi:~$
```

Beleírtunk egy sort a file.txt-be, majd leállítottuk (Ctrl+c) a folyamatot. Aztán ismét írtunk bele egy sort, viszont ekkor csak szüneteltettük (Ctrl+z) a folyamatot. jobs paranccsal lekértük az aktuális job-okat, ahol látható az előbbi fájlba írás folyamatunk. fg paranccsal "vissza hívtuk" a job-ot. Ismét írtunk bele egy sort, majd várakozni küldük és a következő lépésben a kill paranccsal "kilőttük" a folyamatot. Ezután a jobs paranccsal lekérdezve már látható, hogy a fájlba írás teljes egészében leállt.

&

Ha egy progamot a háttérben akarunk futtatni, akkor a futtatandó program neve után egy & jelet kell tenni.

pl.:

```
lekogabi@lekogabi:~$ gedit&
```

A fenti parancs hatására megnyílik a szövegszerkesztő a háttérben.

echo

Az utána lévő szöveget írja ki a képernyőre. A szóközöket a "\<szóköz>" segítségével lehet beszúrni. Vagy ha az egész szöveget "" közé tesszük, akkor az kerül a képernyőre ami az "" között van. pl.:

```
lekogabi@lekogabi:~$ echo hello
hello
lekogabi@lekogabi:~$ echo "hello Kitty!"
hello Kitty!
```

Változók is előállíthatóak az echo segítségével egy plussz '\$' jel használatával. pl.:

```
lekogabi@lekogabi:~$ echo $asd
(semmi)
lekogabi@lekogabi:~$ asd="hello"
lekogabi@lekogabi:~$ echo $asd
hello
lekogabi@lekogabi:~$ unset asd
lekogabi@lekogabi:~$ echo $asd
(semmi)
```

Létrehoztunk egy 'asd' nevű változót, amely kezdetben üres. Majd bele tettük a "hello" szöveget, majd az 'unset' parancs segítségével töröltük.

set

Kiírja az összes környezeti változót és az értéküket.

cat <fájl>

Paraméter nélkül írhatunk a képernyőre és Enter leütése után kiírja az addig beírt sort.

Fájl paraméter esetén kiírja a fájl tartalmát a képernyőre. Ha "cat > szövegfájl" kombinációt használjuk, akkor a konzolra írt sorokat a szövegfájlba menti, ha >>-t írunk > helyett, akkor pedig a sorokat hozzáfűzi a fájl végéhez. pl.:

```
lekogabi@lekogabi:~/Dokumentumok$ cat
hello
hello
^C
lekogabi@lekogabi:~/Dokumentumok$ cat > file.txt
first
^C
lekogabi@lekogabi:~/Dokumentumok$ cat file.txt
first
lekogabi@lekogabi:~/Dokumentumok$ cat >> file.txt
second
^C
lekogabi@lekogabi:~/Dokumentumok$ cat file.txt
first
second
```

Először kiírtuk a képernyőre, hogy "hello", majd a 'file.txt'-be bele írtuk a "first" sort. Aztán ezt kiírtattuk a képernyőre, majd hozzáfűztük ugyanehhez a fájlhoz a "second" sort és kiírtattuk.

Cat és a job műveletek használata, pl.:

```
lekogabi@lekogabi:~$ cat
hello
hello
^Z
[1]+  Megállítva          cat
lekogabi@lekogabi:~$ jobs
[1]+  Megállítva          cat
lekogabi@lekogabi:~$ fg %1
cat
hi
hi
^Z
[1]+  Megállítva          cat
lekogabi@lekogabi:~$ jobs
[1]+  Megállítva          cat
lekogabi@lekogabi:~$ kill %1

[1]+  Megállítva          cat
lekogabi@lekogabi:~$ jobs
[1]+  Befejeződött        cat
lekogabi@lekogabi:~$ jobs
lekogabi@lekogabi:~$
```

more

A fájl tartalmát oldalanként írja ki a képernyőre. ('Space' leütésével lehet léptetni.)

head [kapcsoló]<fájl>

Kiírja a fájl a bizonyos sorait a képernyőre.

Kapcsolók:

-n <szám>: az első "szám" számú sort írja ki.

-szám: az első "szám" számú sort írja ki

tail [kapcsoló]<fájl>

Kiírja a fájl a bizonyos sorait a képernyőre.

-n <szám>: az utolsó "szám" számú sort írja ki.

-szám: az utolsó "szám" számú sort írja ki
-f: folyamatosan bővülő fájl tartalmát írja ki.

head, tail és head|tail pl.:

```
lekogabi@lekogabi:~/Dokumentumok$ cat file2.txt
első
második
harmadik
negyedik
ötödik
hatodik
hetedik
nyolcadik
kilencedik
tizedik
lekogabi@lekogabi:~/Dokumentumok$ head -4 file2.txt
első
második
harmadik
negyedik
lekogabi@lekogabi:~/Dokumentumok$ tail -2 file2.txt
kilencedik
tizedik
lekogabi@lekogabi:~/Dokumentumok$ head -9 file2.txt | tail -3
hetedik
nyolcadik
kilencedik
```

wc [kapcsoló]<fájl>

Kiírja a fájl sorainak a számát, szavainak a számát, betűinek a számát és a fájl nevét.

Kapcsolók:

-l: csak a nevet és a sorok számát írja ki.
-w: csak a nevet és a szavak számát írja ki.
-c: csak a nevet és a byte-ok számát írja ki.
-m: csak a nevet és a betűk számát írja ki.

pl.:

```
lekogabi@lekogabi:~/Dokumentumok$ wc file2.txt
10 10 83 file2.txt
lekogabi@lekogabi:~/Dokumentumok$ wc -l file2.txt
10 file2.txt
```

^D

Ctrl+d parancs megszünteti a bemenetet vagy kilép a terminálból (vagy shell programból)

/dev/null

Egy speciális fájl, amely minden beleírt adatot töröl, miközben az írási művelet sikeres. Az eszközből való olvasás nem ad vissza semmilyen adatot, eredménye azonnali EOF, azaz fájl vége. Felfogható adatnyelő eszközként ("fekete lyuk"). A null eszközt tipikusan folyamatok kimeneti stream-jeinek eltüntetésére használják, vagy az üres bemenet biztosítására, általában átirányítás segítségével. pl.: OpenGL: ./LekoGabor > /dev/null

Beépített környezeti változók:

\$PWD: Tárolja az aktuális elérési útvonalat.
\$HOME: Tárolja a home könyvtár abszolút elérési útvonalát.
\$PS1: A prompt kinézetét leíró kifejezést tárolja.
\$PATH: A keresési könyvtárak elérési útvonalát tárolja, itt keresi a parancsokat.

~: Nem környezeti változó, de a home könyvtár elérési útvonalát tárolja.

Mintaillesztés

Ha nem tudjuk egy szöveg pontos alakját, csak egy részét, vagy több szövegrészlettel szeretnénk egyszerre dolgozni, akkor jön jól a mintaillesztés. Használható pl. egyszerre több fájl törlésénél, vagy keresésnél (fájlban, vagy fájlrendszerben).

Speciális karakterek:

? - pontosan egy karaktert helyettesít: pl. ?lma lehet alma vagy álma is így.

* - bármennyi (akár 0) karaktert helyettesít: pl. *gép lehet gép, mosógép, számítógép, stb.

[...] - a [] között felsorolt karakterekből pontosan egyet helyettesít: pl. [aó]lom lehet ólom vagy alom, fajl[0123456789] pedig fajl0, fajl1, ... fajl9.

Az összes .txt végződésű fájl kilistázása:

```
lekogabi@lekogabi:~$ ls *.txt
```

Minden .png fájl átmásolása a Dokumentumok/hello mappába:

```
lekogabi@lekogabi:~$ cp *.png /home/lekogabi/Dokumentumok/hello
```

Az összes .png kép törlése a Dokumentumok/hello mappából:

```
lekogabi@lekogabi:~/Dokumentumok/hello$ rm *.png
```

grep [kapcsoló][minta]<fájl>

Kiírja egy fájl azon sorait, amelyekben szerepel a minta szövegrészlet.

Kapcsolók:

-A # : # db sor kiírása az illeszkedő sorok után.

-B # : # db sor kiírása az illeszkedő sorok előtt.

-C # : # db sor kiírása az illeszkedő sorok előtt és után.

-e minta : minta megadása; segítségével több minta is megadható, illetve akkor is jó, ha a minta a - karakterrel kezdődik.

-r, -R: könyvtárak feldolgozása rekurzívan.

-v: azon sorait írja ki, amik nem tartalmazzák a minta szövegrészletet.

pl.:

```
lekogabi@lekogabi:~/Dokumentumok$ grep hat file2.txt
```

hatodik

```
lekogabi@lekogabi:~/Dokumentumok$ grep ed file2.txt
```

negyedik

hetedik

kilencedik

tizedik

```
lekogabi@lekogabi:~/Dokumentumok$ grep -v ed file2.txt
```

elso

masodik

harmadik

otodik

hatodik

nyolcadik

```
lekogabi@lekogabi:~/Dokumentumok$
```

parancs > szövegfájl

A parancs kimenete a konzol helyett a fájlba fog íródni, a fájl addigi tartalma elvész.

pl.:

```
lekogabi@lekogabi:~/Dokumentumok$ ls -l > file.txt
```

```
lekogabi@lekogabi:~/Dokumentumok$ cat file.txt
```

```
-rw-r--r-- 1 lekogabi lekogabi      83 aug   9 21:01 file2.txt
```

```
-rw-r--r-- 1 lekogabi lekogabi       0 aug  13 19:10 file.txt
```

```
drwxr-xrwx 2 lekogabi lekogabi   4096 aug   9 15:07 hello
```

```
-rw-r--r-- 1 lekogabi lekogabi  15796 aug  13 19:09 progalap_linux
```

parancs >> szövegfájl

A parancs kimenete a konzol helyett a szövegfájl végéhez fog hozzáfűződni. pl.:

```
lekogabi@lekogabi:~/Dokumentumok$ ls > file.txt
```

```
lekogabi@lekogabi:~/Dokumentumok$ cat file.txt
```

```
-rw-r--r-- 1 lekogabi lekogabi      83 aug   9 21:01 file2.txt
-rw-r--r-- 1 lekogabi lekogabi       0 aug  13 19:10 file.txt
drwxr-xrwx 2 lekogabi lekogabi    4096 aug   9 15:07 hello
-rw-r--r-- 1 lekogabi lekogabi   15796 aug  13 19:09 progalap_linux
file2.txt
file.txt
hello
progalap_linux
```

parancs < szövegfájl

A parancs bemeneteként kapja a szövegfájl tartalmát. pl.:

parancs1 | parancs2

A parancs1 kimenete konzolra kiírás helyett a parancs2 bemenete lesz. pl.:

```
lekogabi@lekogabi:~/Dokumentumok$ cat file2.txt
```

```
első
```

```
második
```

```
harmadik
```

```
negyedik
```

```
ötödik
```

```
hatodik
```

```
hetedik
```

```
nyolcadik
```

```
kilencedik
```

```
tizedik
```

```
lekogabi@lekogabi:~/Dokumentumok$ head -9 file2.txt | tail -3
```

```
hetedik
```

```
nyolcadik
```

```
kilencedik
```

parancs1 && parancs2

A parancs1 végrehajtása után a parancs2 csak akkor hajtódik végre, ha a parancs1 hiba nélkül futott le. pl.:

```
lekogabi@lekogabi:~/Dokumentumok$ mkdir ujmappa && rmdir ujmappa
```

Ha létre tudta hozni az 'ujmappa' nevű könyvtárat, akkor ki is törli a második paranccsal.

parancs1 || parancs2

A parancs1 végrehajtása után a parancs2 csak akkor hajtódik végre, ha a parancs1 futása közben hiba történt. pl.:

```
lekogabi@lekogabi:~/Dokumentumok$ rm -r ujmappa || mkdir ujmappa
```

Megpróbáljuk törölni az 'ujmappa' nevű könyvtárat. Mondjuk még nincs ilyen könyvtár, akkor hiba üzenetet kapunk, majd létrejön az 'ujmappa' nevű könyvtár.

parancs1;parancs2;parancs3

A parancsok ";"-vel elválasztva egymás után hajtódnak végre balról jobbra.

```
lekogabi@lekogabi:~/Dokumentumok$ mkdir ujmappa; cd ujmappa; ls -l
```

```
összesen 0
```

```
lekogabi@lekogabi:~/Dokumentumok/ujmappa$
```

scp <felhasználónév>@<szerver>:<távoli útvonal><helyi útvonal>

Átmásol egy bizonyos fájlt vagy könyvtárat a "távoli útvonal"-ról a "helyi útvonal"-ra.

```
pl.:
lekogabi@lekogabi:~$ scp
h165057@linux.inf.u-szeged.hu: /n/pub/ProgramozasAlapjai/tematika-2013.txt ./
h165057@linux.inf.u-szeged.hu's password:
tematika-2013.txt                100% 3659      3.6KB/s   00:01
```

ssh [kapcsoló]<felhasználónév>@<szerver>

Csatlakozni lehet a szerverre, futtatni konzolos programokat.

-X: Ezzel a kapcsolóval grafikus programot is indíthatunk.

pl.:

```
lekogabi@lekogabi:~$ ssh h165057@linux.inf.u-szeged.hu
h165057@linux.inf.u-szeged.hu's password:
Linux linux1.inf.u-szeged.hu 3.2.0-4-686-pae #1 SMP Debian 3.2.54-2 i686
```

```
*****
*           Udvozoljuk az Informatikai Tanszekcsoport linux szerveren!           *
*                                                                                   *
* A szerver azt a celt szolgálja, hogy tavoli bejelentkezes eseten ugyanazt *
* a kornyezetet biztositsa, amit egy tanteremben levo munkaallomas biztosit. *
*                                                                                   *
* A szerver mukodesevel, a bejelentkezessel kapcsolatos problemaikat kerjuk *
*           a kabinet@inf.u-szeged.hu cimem jelezzek.                             *
*****
```

Last login: Sun May 4 21:38:49 2014 from 79.101.5.52

```
h165057@linux1:~$ cd LekoGabor
```

```
h165057@linux1:~/LekoGabor$ make
```

```
make: Nothing to be done for `application'.
```

```
h165057@linux1:~/LekoGabor$ ./LekoGabor
```

```
freelut (./LekoGabor): failed to open display ''
```

```
h165057@linux1:~/LekoGabor$ exit
```

kijelentkezés

Connection to linux.inf.u-szeged.hu closed.

```
lekogabi@lekogabi:~$ ssh -X h165057@linux.inf.u-szeged.hu
```

```
h165057@linux.inf.u-szeged.hu's password:
```

```
Linux linux1.inf.u-szeged.hu 3.2.0-4-686-pae #1 SMP Debian 3.2.54-2 i686
```

```
*****
*           Udvozoljuk az Informatikai Tanszekcsoport linux szerveren!           *
*                                                                                   *
* A szerver azt a celt szolgálja, hogy tavoli bejelentkezes eseten ugyanazt *
* a kornyezetet biztositsa, amit egy tanteremben levo munkaallomas biztosit. *
*                                                                                   *
* A szerver mukodesevel, a bejelentkezessel kapcsolatos problemaikat kerjuk *
*           a kabinet@inf.u-szeged.hu cimem jelezzek.                             *
*****
```

Last login: Wed Aug 13 21:12:17 2014 from 79.101.5.234

```
h165057@linux1:~$ cd LekoGabor/
```

```
h165057@linux1:~/LekoGabor$ make
```

```
make: Nothing to be done for `application'.
```

```
h165057@linux1:~/LekoGabor$ ./LekoGabor > /dev/null
```

sftp <felhasználónév>@<szerver>

Csatlakozik a szerverre és lehetőségünk van lépkedni a szerver könyvtáraiban, illetve a lokális könyvtárunkban, majd leszedni illetve feltölteni bizonyos adatokat.

Egy 'l' betű hozzáadásával tudjuk közölni a géppel, hogy nem a szerveren szeretnék, hogy végrehajtsd az adott parancs, hanem a lokális gépen. pl.:

```
ls - lls, pwd - lpwd, cd - lcd, mkdir - lmkdir.  
put - fájl átmásolása a lokális gépről a szerver gépre.  
get - fájl átmásolása a szerver gépről a lokális gépre.  
exit - kilép a szerverről.  
pl.:
```

```
lekogabi@lekogabi:~$ sftp hl65057@linux.inf.u-szeged.hu  
hl65057@linux.inf.u-szeged.hu's password:  
Connected to linux.inf.u-szeged.hu.  
sftp> cd ..  
sftp> cd ..  
sftp> cd pub  
sftp> cd ProgramozasAlapjai/  
sftp> get tematika-2012.txt  
Fetching /n/pub/ProgramozasAlapjai/tematika-2012.txt to tematika-2013.txt  
/n/pub/ProgramozasAlapjai/tematika-2012.txt 100% 3659 3.6KB/s 00:00  
sftp> exit  
lekogabi@lekogabi:~$
```

források:

- #1: <http://szabilinux.hu/unix/index.html>
- #2: <http://www.letix.hu/>
- #3: 2011-es Programozás alapjai gyakorlati jegyzetem + Csiszi jegyzete.