

Minden parancs használatánál ajánlatos megnézni a

```
man parancsnév
```

segítségével a lehetséges opciókat. Van pár parancs (például a for), amelyhez nincs külön leírás. Ezeket a

```
man bash
```

leírásban lehet megtalálni. Ez több száz oldalas, és még kevésbé érthető kezdők számára. (Keresés: /keresendő_szó ENTER, következő keresése: n)

Használjuk a bal egérgombbal történő kijelölést, középső gombbal történő beillesztést. Sokat gyorsít.

További linkek:

<http://www.ibiblio.org/pub/Linux/docs/HOWTO/translations/hu/Bash-Prog-Intro-HOWTO-hu.txt>

http://www.szabilinux.hu/linux_doksi/node432.htm

<http://www.grymoire.com/Unix/Sed.html>

Tartalomjegyzék

- [1 Háttérképek letöltése](#)
- [2 Letöltött képek pdf formátumba konvertálása](#)
- [3 Szorozzuk össze az első 50 pozitív egészét](#)
- [4 Hozzunk létre egy olyan txt fájlt, amelyben felsoroljuk az összes tex fájlnkat, továbbá azt is, hogy hány sorból állnak](#)
- [5 sg.hu portálon lévő címek lementése](#)
- [6 A wget parancs pár opciója](#)
- [7 Hőmérséklet Londonban](#)
- [8 Nagy fájlok keresése a home mappánkban](#)
- [9 Friss fájlok keresése adott mappában](#)

Háttérképek letöltése

Szeretnénk szép háttérképeket letölteni az internetről. Nem szeretnénk, ha egyesével kellene lementeni a képeket, jobb lenne letölteni sokat, majd gyorsan végigszaladni rajtuk.

Nyissunk egy könyvtárat a képeknek:

```
mkdir természet  
cd természet
```

A google "http://wiki.math.bme.hu/nature backgrounds" http://wiki.math.bme.hu kulcsszavakra több millió oldalt kiad, az első pár között találtam a következő honlapot:

<http://www.batkhela.com/wallpapers/9071/nature-wallpapers.php>

Amikor én néztem a honlapot, akkor a második képen egy pad volt, szép színekkel. Rakattintottam, és megnyílt nagyban.

A képre jobb egérgombbal kattintva válasszuk a Properties menüpontot. Itt megtalálhatjuk a kép tulajdonságait.

Informatika1-2008/Gyakorlat13

Location: <http://www.batkhela.com/wallpapers/9071/displaypics/pics/pic-7452.jpg>

Ha ezt a képet menteni szeretnénk, akkor megtehetjük a következő paranccsal:

```
wget -c http://www.batkhela.com/wallpapers/9071/displaypics/pics/pic-7452.jpg
```

A wget -c parancsa a continue szóra utal, ha a fájl letöltése valami miatt megszakadna, akkor ezzel folytathatjuk. Ez az opció később még jól jöhet.

Ha van 7452-es kép, akkor valószínűleg lesz 7453-as is, és 7451-es is. Próbáljunk meg letölteni ezek közül 10-et.

```
for ((i=7450;i<7460;i++)); do echo ${i}; done
```

parancs eredménye:

```
7450
7451
7452
7453
7454
7455
7456
7457
7458
7459
```

Tehát: az i egy változó, a \${i} pedig visszaadja az értékét. A for ciklus a következőképpen működik: (inicializálás, feltétel, léptetés). A do és a done parancsok közé több parancsot is el lehet helyezni pontosvesszővel elválasztva. A 10 képet letöltő parancs így:

```
for ((i=7450;i<7460;i++)); do wget -c http://www.batkhela.com/wallpapers/9071/displaypics/pics/pic-7452.jpg
```

A képek megjelenítésére a

```
gqview
```

programot ajánlom. f billentyűvel lehet váltani teljes képernyős üzemmódba (f mint fullscreen), és egy nem túl szép képnél a del gomb után megkérdezi, hogy tényleg törölni akarjuk-e a képet.

Letöltött képek pdf formátumba konvertálása

A következőkben a letöltött képeket pdf formátumba konvertáljuk. Az a lényeg, hogy egy paranccsal elvégezzük ugyanazt a dolgot több fájlra.

Ha lemaradt valaki, akkor a

```
cp /STORE/infogyak/* .
```

paranccsal átmásolódnak hozzá a képek.

Megoldás:

```
for i in *.jpg; do echo ${i}; done
```

eredménye:

```
pic-7450.jpg
pic-7451.jpg
pic-7452.jpg
pic-7453.jpg
pic-7454.jpg
pic-7455.jpg
pic-7456.jpg
pic-7457.jpg
pic-7458.jpg
pic-7459.jpg
```

Gyakorlaton volt már szó a convert programról. Használata:

```
convert kép.jpg kép.pdf
```

Probléma: Meg kellene szabadulnunk a jpg kiterjesztéstől:

```
for i in *.jpg; do echo ${i%.jpg}; done
```

A fenti módon a % jel után megadott részt a végéig törli. Használhatunk *-ot is a keresésnél. (Ha több illeszkedés is van, pl: aaa.txt változóból vágjuk le a "http://wiki.math.bme.hua*"http://wiki.math.bme.hu vég? részt, akkor % hatására a legrövidebb illeszked? részt vágja le, %% hatására a leghosszabbat.

B?vebben: man bash, majd 1390 sortól) Így a megoldás:

```
for i in *.jpg; do convert ${i} ${i%.jpg}.pdf; done
```

Ha már nem kellene a jpg fájlok, akkor törölhetjük ?ket az

```
rm *.jpg
```

paranccsal.

Szorozzuk össze az els? 50 pozitív egészet

Ehhez a feladathoz szükségünk lesz a **bc** nev? programra. A programot **bc** paranccsal indítsuk el, majd üssük be a

```
2+2*2
```

sort. Ismeri a m?veleti sorrendet (még kisebb programokat is lehet benne írni, amire most nem térünk ki). Van *scale* változója, amelyet felülírva megadhatjuk a tizedes jegyek számának pontosságát:

```
scale=6
```

Ezután a

```
4/3
```

eredménye 8 jegy pontosságú lesz. Kilépés: Ctrl-d gombbal.

A következ?kben az els? 50 számot fogjuk összeszorozni. Az els? 50 számot már ki tudjuk listázni:

```
for ((i=1;i<=50;i++)); do echo ${i}; done
```

Vegyük észre, hogy a -n hatására az echo nem tesz újsort a végére (lásd: man echo).

```
for ((i=1;i<=50;i++)); do echo -n ${i}; done
```

A képlethez már csak a közbüls? szorzásjelek hiányoznak. Írjuk a kimenetet egy fájlba:

```
rm szorzas.txt
for ((i=1;i<=50;i++)); do echo -n ${i} >> szorzas.txt; done
```

Magyarázat: a > és a >> jel a parancs kimenetét beírja a fájlba. A különbség az, hogy a > hatására törlődik a fájl eddigi tartalma, míg a >> jelre a végére f?zi. Megnézhetjük, hogy mi a szorzas.txt fájl tartalma:

```
cat szorzas.txt
```

És most a * jel:

```
rm szorzas.txt
for ((i=1;i<=49;i++)); do echo -n "http://wiki.math.bme.hu${i}*"http://wiki.math.bme.hu >> szorzas.txt; done
```

Fontos, hogy az 50-es után legyen a file-ban newline karakter (az utolsó echo után nem kell -n opció)!
Enélkül a bc hibát ad.

A szorzás eredménye pedig:

```
cat szorzas.txt | bc
```

A bc a standard bemeneten kapott képletet örömmel végrehajtja.

Hozzunk létre egy olyan txt fájlt, amelyben felsoroljuk az összes tex fájlunkat, továbbá azt is, hogy hány sorból állnak

Lépjünk át a saját mappánkba:

```
cd
```

Keressük meg az összes tex fájlt, ami valaha készítettünk. A find parancs kilistázza az összes fájlt.

```
find
```

Keressünk a fájlnevek között úgy, hogy átírányítjuk a kimenetét, és egrep-pel csak azokat írjuk ki, amelyek .tex-re végződnek:

```
find | egrep '\.tex'
```

Ez megtalálja a tex_backup és tex~ kiterjesztés? file-okat is, úgyhogy helyette használjuk az alábbi (emlékezvén, hogy regexp-ben \$ karakterrel hivatkozhatunk a sor végére):

```
find | egrep '\.tex$'
```

Próbaképp számoljuk meg hány tex fájlunk van. Próbáljuk ki a wc parancsot:

```
find | egrep '\.tex$' | wc
```

Szorozzuk össze az els? 50 pozitív egészet

Informatika1-2008/Gyakorlat13

Alapértelmezésként 3 egész számot ad vissza: A sorok, a szók és a karakterek számát. Ha csak a sorok számát szeretnénk megkapni, akkor használjuk a -l paramétert (további paraméterekért lásd: man wc).

```
find | egrep '\.tex$' | wc -l
```

Valójában nem erre lesz szükségünk, hanem a fájlokban lévő sorok számára. A `` (fordított irányban d?!?) idézőjelek közötti rész eredménye beillesztésre kerül (olyan, mintha az eredményt az `` jelek közötti rész helyére beírtuk volna). Listázzuk ki az összes tex tartalmát:

```
cat `find | egrep '\.tex$`
```

A megoldás így:

```
rm tex_fajlok.txt
for f in `find | egrep '\.tex$`; do echo "http://wiki.math.bme.hu`cat ${f} | wc -l` sorbol all a
```

Kicsit szépíthetünk, ha sorszám szerinti növekvő sorba rendezzük a sorokat:

```
cat tex_fajlok.txt | sort
```

Ez a megoldás még nem jó. Az a baj, hogy az abc szerint a 98 soros nagyobb, mint a 101. Ezért adjunk meg a sort parancsnak egy -n paramétert (numeric-sort, lásd: man sort).

```
cat tex_fajlok.txt | sort -n
```

Végül mentjük el egy új fájlba:

```
cat tex_fajlok.txt | sort -n > rendezett.txt
```

sg.hu portálon lévő címek lementése

Mentsük el a <http://www.sg.hu> oldalt sg.html néven:

```
wget http://www.sg.hu -O sg.html
```

Az a baj, hogy alapból latin2-es kódolású. Konvertáljuk át utf8-ba az iconv paranccsal.

```
iconv sg.html -f latin2 -t utf8 > sg_utf8.html
```

Az iconv parancs használata: -f milyen_kódolásról -t milyen_kódolásra. Az ismert kódolások kilistázhatók az

```
iconv -l
```

paranccsal. Keressük meg azon részeket (az egrep -o opciója hatására nem adja ki a teljes sort), amelyek cikkek/ kezdet? linkekhez tartoznak.

```
cat sg_utf8.html | egrep -o '<a href="http://wiki.math.bme.hucikkek/[^"http://wiki.math.bme.hu]*"h
```

Tiltanunk kell a < jelet a végén ([^<]+ résznél), mert csak [^>]+ kifejezést használva kapunk fals találatokat is.

És most a címek:

```
cat sg_utf8.html | egrep -o '<a href="http://wiki.math.bme.hucikkek/[^"http://wiki.math.bme.hu]*"h
```

Hozzunk létre egy olyan txt fájlt, amelyben felsoroljuk az összes textfájlnkat, továbbá azt is, hogy **5**ány sorból a

A kiemelt cikk alá írt szöveget is megtalálja, de egyébként megfelel a célnak. Illetve van egy > jel minden sor elején, de ezt most egyszerűség kedvéért bent hagyjuk. Az egész parancs egyben, ideiglenes fájlok nélkül

```
wget http://www.sg.hu -q -O - | iconv -f latin2 -t utf8 | egrep -o '<a href="http://wiki.math.bme.
```

A wget parancs pár opciója

A

```
wget --help
```

parancs hatására felsorolja a fontosabb paramétereket (shift pgup/pgdown -nal lehet lapozni közöttük). A rekurzív letöltés gyakran hasznos. Az összes Párizs látványosság letöltése az adott könyvtárba úgy, hogy ne ugorjon át Londonra:

```
wget -c -m -p -np -nd http://utazas-nyaralas.info/parizs/altalanos.html
```

Opciók:

-c: ezt mindig odaírom, ennek hatására folytatja a letöltést, ha félbeszakadt

-m: mirror, végtelen rekurzió, linkek konvertálása (ezentúl nem a honlapra mutatnak, hanem a helyi fájlokra)

-p: a képek is kerüljenek mentésre (page-requisites)

-np: no parent directory, csak olyan oldal kerül lementésre, amely a <http://utazas-nyaralas.info/parizs/> könyvtárból nyílik.

-nd: ne hozzon létre könyvtárakat, hanem mindent ömlesszen az adott helyre.

H?mérséklet Londonban

```
wget http://www.idojaras.hu/vilag.php -q -O - | iconv -f latin2 -t utf8 | grep -Al London | tail -
```

Nagy fájlok keresése a home mappánkban

Erre a feladatra a **du** parancsot fogjuk használni. Alapértelmezés szerint a

```
du
```

parancs kiírja az aktuális könyvtárban lévő könyvtárakra (és azok alkönyvtáira) a bennük tárolt fájlok összméretét. A

```
man du
```

hatására elolvashatjuk a parancs opcióit. Az egyik hasznos opció a -h, amely hatására a mértékegységeket emberibb formába írja ki.

```
du -h
```

Végül hasznos lehet, ha az alkönyvtárakat nem listázzuk feleslegesen, hanem csak a legfelső könyvtárakra számoljuk ki a méteret:

```
du -h --max-depth=1
```

Friss fájlok keresése adott mappában

Ezt a feladatot legegyszerűbben a **find** paranccsal oldhatjuk meg.

```
find ~/alma -mtime 2
```

Ezzel megkapjuk az alma könyvtár összes olyan fájlát, melyek ez elmúlt két napban módosítottunk.