

Jelen leírás [Debian GNU/Linux](#) alatt tesztelt és használható parancsokat tartalmaz, a teljesség igénye nélkül. A dokumentum tartalmát csak saját felelősségedre használd! Szabadon terjesztheted a forrás és szerző megjelölésével. Jó tanulást!

Utolsó frissítés.: 2016-04-14

[letix](#)

# Linux parancsok, kezdőknek

[Az első lépések](#)

[Könyvtárszerkezet](#)

[Állománykezelés](#)

[Rendszeradminisztráció](#)

[Folyamatok](#)

[Időzített parancsfeldolgozás](#)

[Kernel, modulok, fordítás](#)

[Csomagkezelés](#)

[Hálózat](#)

[Bash programozás](#)

[Érdekes/hasznos bash scriptek -experimental-](#)

[Egyéb programok, leírások](#)

[RAID1 készítése élő rendszeren](#)

[Adatmentés sw. RAID1-et alkotó HDD-ről](#)

[Local repository készítése apt-mirror segítségével](#)

[Távoli mappák felcsatolása SSHFS használatával](#)

[Hasznos linkek](#)

## Az első lépések [top](#)

Egy Debian GNU/Linux alaprendszer telepítésének menete részletesen, képekkel illusztrálva:  
[Debian GNU/Linux 4.0 rendszer telepítése - Expert módban](#)  
[Debian GNU/Linux 5.0 rendszer telepítése - GUI módban](#)

A telepítés és sikeres belépés után ajánlatos parancsok.:

```
dpkg-reconfigure debconf
```

```
# Ha a lehetőségek közül a low-t "alacsony" választva a későbbi programok reposi:  
# telepítésekor minden apró részletre, beállításra rákérdez a rendszer. Ajánlatos:
```

```
apt-get update
```

```
apt-get upgrade
```

```
# A rendszer frissítése. (a tárolók -repository- az /etc/apt/sources.list file-ba
```

**Sok keresgéléstől kímélheted meg magad, ha a következő parancsokat elsajátítod.:**

```

man parancs
    # A parancs manual oldalait nyitja meg, rövid, tömör, célratörő leírás

apropos szó
    # Minden olyan parancsot megad, mely manual oldalalaiban szerepel a "szó"

man -k szó
    # Az apropos program kimenete

parancs --help
    # Információ a "parancs" használatáról

info parancs
    # Információ a "parancs" használatáról

whatis parancs
    # Egysoros a parancsról.

whereis parancs
    # Hol is van a parancs?

which parancs
    # A program futtatható állományának elérési útvonalát adja meg (általában /usr/bin)

file:///usr/share/doc
    # böngészőbe írva, a legtöbb telepített program leírása megtalálható,
    # általában html formátumban is.

```

Fontos! Nem minden (a leírásban felsorolt) parancs tartozéka egy alaprendszernek, előfordulhat, hogy nem minden rendszeren van meg.

## Könyvtárszerkezet [top](#)

```

/      # A hierarchikus könyvtárfa kiindulópontja (gyökér könyvtár)
/boot  # A rendszer indításához szükséges állományok helye (grub, vmlinuz, stb)
/bin   # A futtatható parancsok könyvtára -binaries
/sbin  # A rendszergazda parancsai -superuser bin
/lib   # Az induláshoz szükséges osztott rendszerkönyvtárak -libraries
        Továbbá tartalmazza a rendszerhez csatlakozható modulokat, meghajtóprogramokat
/dev   # A rendszerhez csatlakozott, csatlakozható különleges állományok -devices
/etc   # Beállítófájlok, helyi indító parancsok, jelszavak, hálózati-beállítók, etc. helye
/home  # Minden felhasználó saját könyvtára itt foglal helyet. (Otthon, édes otthon)
/mnt   # A felcsatolt (mountolt) perifériák könyvtára. -mount
/proc  # Itt látható, ahogy a rendszer "él és lélegzik". (szellem a gépben) -process információk
        Érdemes tüzetesebben átnézni, hiszen érdekes dolgokat találhatunk itt.
        pl.: /proc/cpuinfo fájl kiíratásával információt kaphatsz processzorodról.
/root  # A rendszer gazdájának könyvtára.
/tmp   # Ideiglenes adatok tárolására használt könyvtár. -temp
/usr   # Alkalmazások, rendszereszközök tömkelege, a legforgalmasabb könyvtár. (pl X Window)
/var   # Változó adatokat tartalmazó állományok könyvtára. /pl.: nyomtatási munkák, levelek
        /var/log : napló fájlok, különös jelentőséggel bírnak a rendszer biztonságánál

```

## Állománykezelés [top](#)

```
alias
# A parancsok saját elképzelések szerint átnevezhetők.
# /home könyvtár .bashrc fájlja tartalmazza beállításokat.
alias rm='rm -i' : A törlés parancs kiadása után rákérdez, hogy valóban akarjuk e

unalias
# Az előzőekben beállított értékek visszaállítása. (törli a .bashrc-ből.)

cat
# Fájl tartalmát írja ki.
> fájl : várja a bemenetet, amely a "fájl" tartalma lesz. Ctrl + D kombinációval
-n fájl : beszámozza a fájl sorait
???.sh : Minden .sh kiterjesztésű, 2 betűs file tartalmát kiírja a képernyőre.
/dev/cdrom > /eleresi/utvonal/cd.iso : A CD tartalmának ISO-ban örténő mentése.
/etc/passwd |grep "/home" | cut -d: -f1 : A rendszerbe felvett felhasználók kiírása

cat file.txt 1> file2.txt 2>&1
# A hibacsatorna is a kimenetre keverhető, azaz a file1.txt tartalma ÉS a lehetséges
# is bekerülnek a file2.txt-be. A hibacsatornáról a bash programozás részben bővebben

cd
# Könyvtár váltás parancs.
cd : az aktuális felhasználó /home könyvtárába való belépéshez
cd .. : az aktuális könyvtárhoz képest egy szinttel feljebb lépés a könyvtár fájlrendszerben

pwd
# Az éppen aktuális könyvtár munkakönyvtár kiírása

mkdir
# Könyvtár létrehozása (make directory)
-p /home/user/1/2/3 : A teljes struktúra létrehozása, almappákkal együtt.

chmod
# Linuxon a fájlokra, könyvtárakra vonatkozó jogok állíthatóak be.
-R : Rekurzívan változtatja meg a jogosultságokat.

# DAC (háromszintű diszkrécionális maszk) szerinti beállítás.
# r-read (olvasás), w-write (írás), x-executable (futtatás) jogot jelent
#
# Általános jogosultság lista felépítése: (ls -la paranccsal lekérdezhető)
# tulajdonos (jele:U) | csoport felhasználó (jele:G) | mindenki más (jele:O)
#          rwX          |          rwX          |          rwX
#
# A sor elején található "d" a directory, "-" a file jele.
# Jogok nem csak szimbólikus jelekkel de számokkal is meghatározhatóak.
#
# Számokkal.: 4-read, 2-write, 1-executable jog, összeadva, külön U,G,O-nak
#
chmod 777 file : UGO-nak egyaránt minden jog. (4+2+1 4+2+1 4+2+1)
chmod 751 file : U-nak minden, G-nek írási és futtatási, O-nak futtatási jog.

# Betűkkel.: kinek+mit
#
chmod u+rw file : Tulajdonosnak (U) r,w,x jog adása az adott file-ra.
chmod g+rx file : Csoport felhasználónak (G) r,x jog beáll.
chmod a-rwx      : Mindenkitől (A=all) elveszünk minden jogot.

umask
# A file és könyvtár jogok beállítása kapcsán érdemes megemlíteni az umask-ot.
# Az umask meghatározza, hogy milyen jogosultságot kapjanak az újonnan létrehozott
# Értéke alapértelmezés szerint 022.
# Jelentése.: File-ok 644-et, Mappák 755 jogokat kapnak.
# File-ok esetén 666-ból, mappák esetében pedig 777-ből kell levonni a 022-t, így

chown
# Fájlok, könyvtárak tulajdonosának (létrehozójának változtatása)
```

```
-R          : Rekurzívan változtatja meg a tulajdonos(oka)t
-f          : Nem küld vissza hibaüzenetet a rendszer, ha valami nem sikerül
--no-dereference : Szimbólikus linkeknél a link jogosultságainak beállítása
--dereference   : Szimbólikus linkeknél a file (amire a link mutat) jogok változ
```

chgrp

```
# Fájlok tulajdonosi csoportjának megváltoztatása
-R : Rekurzívan változtatja meg a csoportokat
-f : Nem kapunk vissza hibaüzenetet, ha valami nem sikerült
-c : csak azokat a file-okat írja ki, amelyeknek valóban megváltozott a csoportja
```

lsattr

```
# Fájlok, könyvtárak attribútumát mutatja meg
-R : Rekurzívan mutatja meg az attribútumokat
-a : minden file-t kilistáz, beleértve a .-al kezdődőeket is
```

chattr +tulajdonság file

```
# Fájlok, könyvtárak attribútumát változtatja
#
# Tulajdonságok.:
A : Nem változtatja meg a fájl utolsó módosításának dátumát. (rendszergyorsító)
a : Csak hozzáfűzni tudunk a fájlhoz
c : Automatikusan tömörítve kerül a lemezre, és kitömörítve kerül beolvasásra
d : Ezekről az állományokról nem készül biztonsági másolat a dump parancs futtatásakor
s : Paranoia mód. Törléskor azonnal megsemmisül minden bit-je.
S : Minden változtatás azonnal lemezre íródik (sync hatás)
u : A Fájl törlésekor az adat megmarad, később visszaállítható
```

cfdisk

```
# Lemezkezelő, particionáló
```

cp

```
# Fájl, könyvtár másolásra használható program
file1 file2          : file1 állományból készít file2 nevű másolatot file1 megta
-R /honnan/mit /hova : rekurzívan mindent másol a /honnan/mit-ből a /hova mappába
```

cmp file1 file2

```
# Összehasonlítja a file1 és file2 fájl tartalmát
```

cut

```
# Bement (stdin), vagy paraméterként megadott fájl minden sorának egy megadott
# részét vágja ki
-c2 fájl          : második mező értéke
-c3,5             : harmadik, ötödik mező, sorrend nem számít
-c-4,6-          : negyedik mezőig és a hatodiktól

cut -d: -f1 :
# Kettősponttal elválasztott sorokban az első helyen lévő adatot adja vissza.
Pl.:echo ELSO:MASODIK:HARMADIK | cut -d: -f1
# Kimenete.: ELSO
```

diff -u file1 file2 > eredmény

```
# Összehasonlítja a fájl tartalmát, a különbséget pedig az eredmény-be írja
```

```
y -left-column file1 file2
```

```
# file1 és file2 összehasonlítása, az eredményt két egymás melletti oszlopba írja
# de az egyezőségeket csak a bal oszlopban tünteti fel.
```

du

```
# Az aktuális könyvtár fájljainak méretét adja meg
-H          : Olvashatóbb formátumban írja ki a méreteket (MByte, GByte, stb.)
--si        : A -h helyett már ezt a kapcsolót ajánlatos használni.
--max-depth=1 : 1 könyvtár mélységig vizsgál
```

```
df
    # Szabad terület számítása, partícióként
    -H      : Olvashatóbb formátumban írja ki a méreteket (MByte, GByte, stb.)
    --si    : A -H helyett már ezt a kapcsolót ajánlatos használni.

echo szoveg
    # Kiírja a képernyőre a szoveg-et
    echo szoveg > file      : a szoveg-et file-ba írja
    echo $HOME              : $HOME nevű változó értékét adja meg, ami az aktuális us

tr
    # karakterek lecserélése, változtatása adott karaktersorban
    echo vegyes | tr a-z A-Z : a vegyes szóban a kis betűket nagyra cseréli.
    echo egyesek | tr -d e   : az egyesek szóból kitörli az e betűket

    cat file.txt | tr -cs '[a-zA-Z0-9]' '[\n*]' > file2.txt
    # ha a file.txt több szóból álló szöveget tartalmaz, a szavak mögötti szóközt új
    # azaz minden szó új sorba kerül egymás alá, a file2.txt-be irányítva

    cat file.txt | tr -s '\n' > file2.txt
    # Ha a file.txt-ben több üres sor is van, az összes újsor karaktert összevonja, a

find
    # Keresés

    find / -name "*.jpg" -print : az összes kép keresése a gyökérben, majd az eredmér
    find / -iname ...          : kis és nagybetű különbség nincs!
    find -perm 777             : minden 777-es joggal rendelkező állomány keresése
    find -perm 4000            : Az összes SUID joggal rendelkező állományt keresi
    find /home -size +1024     : 500kb-nál nagyobb állományok keresése a /home-ban
    find -type "kapcsoló"      : különböző típusú fájlokat keres
        -type l : szimbólikus link
        -type d : könyvtár
        -type f : fájl

    find /etc -empty -maxdepth 1 -printf "%p-%m\n"
        # Az /etc könyvtárban lévő üres könyvtárakat írja ki, a jogosultságaival

    find /home -size +1024 \( -mtime +365 -o -atime +365 \) -ls -exec file{} \;
        # 512kb-nál nagyobb, maximum 365*24 órája módosított állományokat, valami
        # -exec file{} \; jelenti, hogy az exec után levő parancsnak adja át az e

    find -iname *.avi -a -size +1000M -o -iname *.ISO -a -size +500M
        # Keresési feltételek.: avi kiterjesztésű ÉS 1000MB fölötti, VAGY
        # ISO kiterjesztésű ÉS 500MB fölötti file-ok. (Kis-nagy betű nem számít a

file sajt
    # megvizsgálja a sajt fájl típusát
    -f filelista : Egy létező filelista állományban felsorolt file-okat vizsgálja meg
    -L : Követi a szimbólikus link kötést (nem a linket, hanem az arra mutató file-t
    --mime file: A file karakterkódolását mutatja meg.

fsck
    # lemezellenőrző

fgrep
    # Fájlokban, vagy stdin-ben keresek szöveget
    "abc" file.txt : Megkeresi az összes olyan sort a file.txt-ben, ami tartalmat "a

grep
    # Szövegrészleteket keres fájlokban, valamint a kimenetben.
    -i      : nem tesz különbséget kis és nagybetűk között
    -l      : nem az előfordulási sorokat, hanem csak a fájl neveket listázza
    -L      : azokat a fájl neveket adja meg, melyben nem szerepel a "minta".
    -v      : azokat a sorokat adja meg, amikben nem szerepel a keresett szó
    -e      : "-" -el kezdődő minta keresésekor hasznos kapcsoló. (nélküle érvénytelen
```

```

-x      : csak teljes sorokkal való illeszkedést vizsgál.
B.r     : azokat a sorokat adja meg, melyekben a "B" és az "r" között bármilyen ka
[ha]    : a kimenetben találhatóak meg azok a találatok, melyekben szerepel "h" va
[^sajt] : minden sor megtalálható a kimenetben, kivéve amelyben szerepel a "sajt"
^$      : Azokat a sorokat adja meg, melyek üresek.
^h      : A sor elején található kis "h" betűre illeszkedik.
# szóközt grep szó" "szó2 -vel ábrázolunk

head

# Szűrő eszköz. A fájl első 10 sorát írja ki
-n 100 fájl : A fájl első 100 sorát adja meg
-n 7 fájl   : utolsó 7 sort már nem írja ki
-n 4 fájl   : A fájl első 4 sorát írja ki. (megadható "-n 4"-el és "-n4"-el :
-c 4 fájl1 fájl2 : mindkét fájl első 4 karakterét írja ki

tail

# Szűrő eszköz. A fájl utolsó sorait írja ki.
-n 2 fájl : A fájl tartalmát a második sortól mutatja meg

tail -n 3 fájl | head -n 1
      vagy
head -n 3 fájl | tail -n 1
      # Egy fájl harmadik sorát így írhatjuk ki

mc

# Midnight Commander fájlkezelő

mcedit

# Az mc szövegszerkesztője

update-alternatives --config editor
# Alapértelmezett szövegszerkesztőnket a fenti paranccsal tudjuk módosítani (pl r

mkisofs (korábbi verziókban cdrtools csomag tartalmazza!)
# Hasznos segédprogram ISO-k készítéséhez.
-r -o cd.iso /cdrom/ : ISO készítése a CD lemezünkről

mkisofs -J -V "Label" adat/ | sudo cdrecord dev=0,0,0 speed=32 -data -v -eject d
      # adat mappa kiírása 32x-es sebességgel, Label nevű, verbose, az írás vég
cdrecord -scanbus
      # dev érték lekérdezése

mkswap

# Swap terület létrehozása

swapon

# Swap bekapcsolása

swapoff

# Swap kikapcsolása

install

# Fájlok másolása, attribútumok módosítása

less

# Szűrőprogram fájlok adatainak megjelenítéséhez.

ls

# A könyvtárstruktúrát jelenít meg
ls -lt      : méret szerint sorrendben
ls -ls     : utolsó módosítás szerint sorrendben
ls ??????? : minden 7 karakteres állományt jelenít meg
ls -a      : a rejtett fájlokat is kiírja
ls -r ?    : fordított sorrendben írja ki. pl.: -nr : ABC fordított sorrendjében
ls [aes]?  : azokat a 3 betűs fájlokat, melyek középső betűje a,e,s közül bármely
ls [nm]*   : azokat a fájlokat melyek n,m betűvel kezdődnek

```

```
ls *c      : amelyek c-betűre végződnek
ls [^s]*   : amely fájlok nem s-el kezdődnek
ls I szó    : kilistázza a könyvtár tartalmát, de a szó-t kihagyja
```

lsuf

```
# Nyitott fájlok kilistázása
-p pid      : Folyamat azonosító szerinti szűrés
| egrep ":port1|:port2" : A megadott portokat használó fájlokat listázza ki
```

lspci

```
# Hardver információk
-v : bőbeszédű
-vv : még bőbeszédűbb
```

lsusb

```
# Usb eszközök kiíratása
```

lshw

```
# Hardver információk, kicsit bővebben
-X      : Grafikus felületű eredmény
-html   : HTML formátumban menthetjük a kimenetet
-short  : Az eszközök fa-szerű megjelenítése
```

mv

```
# Fájlokat könyvtárakat mozgat, vagy nevez át
-i : A módosítások előtt rákérdez, hogy valóban akarjuk-e
-U : Nem mozgatja az újabb (módosítás dátuma szerint) célállománnyal rendelkező állományt
-f : force, a célállomány törlése, kérdés nélkül
```

mkfs

```
# Fájltrendszer létrehozása
-t filesystem eszköz [blokkméret]
-t ext2 -V /dev/eszköz : ext2-es fájlrendszer létrehozása
```

updatedb

```
# A PC-n található file-okról készít adatbázis, melyben az mlocate/locate parancsok
#
# /etc/updatedb.conf-ban meghatározható paraméterek:
# -----
#
# PRUNENAMES      : bizonyos kiterjesztésű állományokat ki lehet hagyni a vizsgálat
# PRUNEPATHS      : megadható, mely mappákban ne keressen. pl: /tmp
# PRUNEFS         : definiálható, mely fájlrendszereket hagyjon figyelmen kívül. pl:
#
# /var/lib/mlocate/mlocate.db : A default adatbázis elérési útvonala
```

updatedb -l 0 -o dbdir/dbname.db -U /mnt/server

```
# /mnt/server felcsatolt fájlrendszer tartalmáról adatbázis készítése dbdir/dbname.db
# lokális meghajtókról nem készül bejegyzés, csak a server-ről!!
```

mlocate -d dbdir/dbname.db macilaci

```
# a fent létrehozott dbname.db-ben olyan állományokat keres, melyeknek a nevében
```

mlocate fájl

```
# általánosan létrehozott updatedb adatbázisában (/var/lib/mlocate/mlocate.db) keres
```

ln -s

```
# Soft link létrehozása
-s fájl1 fájl2 : fájl1 ről készít fájl2 nevű soft-link-et.
```

mkdir -t ext2 /dev/ram0 4096

```

mount /dev/ram0 ramdisk/
    # 4Mbyte os ramdisk-et hozunk létre, és felcsatoljuk.

mount
    # Szimplán kiadva a rendszerbe felcsatolt eszközök listáját adja meg. (/etc/mtab

/dev/hda2 /mnt/usbhdd
    # hda2-es eszközt csatolja fel a usbhdd könyvtárba (létre kell előtte hozni)
    # IDE csatolófelületű eszközöknél volt használatos a Hda

-o noatime /dev/hdX
    # noatime attribútummal csatlakoztatja a lemezt. (I/O művelet csökkentés)

/eleresi/ut/cdimage.iso /mnt/cdrom -o loop
    # cdimage.iso felmountolása megtekintésre. (kell /mnt/cdrom mappa!)

-t cifs -ro username=NEV,password=JELSZO //server/megosztas /mnt/hova
    # CIFS fájlrendszert csatolunk read-only módban a server megosztásából, t

-t ntfs-3g /dev/sdb1 /mnt/ntfs
    # ntfs-3g csomag telepítését követően a /dev/sdb1 (NTFS fájlrendszerű) es

-t vfat /dev/sdc1 /mnt/usb
    # vfat rendszerű (általában a FAT/FAT32 pendrive-ok ilyenek) felcsatolása:

umount /dev/eszköz
    # Lecsatlakoztatás

pwd
    # Kiírja az aktuális könyvtárat

recode
    # file karakterkódolásának konvertálása
    iso-8859-15..utf8 modositando_file : Nyugat európai kódolás konvertálása UTF8-ra

rmdir
    # Könyvtár törlés

rm
    # Állományok eltávolítása
    -d : könyvtárat
    -i : rákérdez a törlés előtt (Ajánlott!)
    -rf : Könyvtárstruktúrát töröl (akkor is, ha nem üres)

rsync
    # Hasznos file másoló, szinkronizáló program
    -a : Archive módban másol. Tulajdonosok, attributumok, linkek megmaradnak.
    -v : Bőbeszédű a kimenetkor.
    -z : Tömörítve küldi el.
    -b : Biztonsági mentésekhez használatos. *.~ végű file keletkezik.
    --exclude '*' : kivéve a ~ végű file-ok.
    -avz /eleresi/ut/honnan/* 192.168.1.1:/eleresi/ut/hova
        # Relatív elérési úttal megadva, archive módban tömörítve menti a hálózati

sed
    # Stream editor, folyamatszerkesztő. A bemenetet a kimenetre másolja miközben megc
    echo "hablabda" | sed 's/a/K/g' : kimenete.: hKb1KbdK, azaz "a" betűket "K"-ra c
    '/' *#/d; /^ *$/d' file : a file-ból kiszűrjük a kommenteket, üres sorol

sort
    # Sorba rendezés
    -b fájl > kimenet : ABC sorrendbe rendezi a fájl-t, az eredményt a kimenetbe írja.
    -r fájl : fordított sorrendben rendez
    -n fájl : a sor elején levő számok szerint rendez

```



-u fájl : az azonos sorokat csak egyszer írja ki

sync

# A Ramból a merevlemezre még ki nem írt adatok szinkronizálása (Fontos)

touch fájl

# létrehoz egy üres állományt  
 -t MMDDHHmm fájl : A fájl időbélyegeinek dátumát változtatja meg. MM-Hónap DD-  
 -r file file2 : file időbélyegei alapján állítja be file2 időbélyegeit.  
 -a -t 03091315 file : a file létrehozási dátumát állítja Március 9., 13:15-re.  
 -m -t 03091315 file : a file módosítási dátumát állítja Március 9., 13:15-re.

tree

# Könyvtárstruktúrát írja ki  
 -d : csak a mappákat adja meg.  
 -f : teljes path-al kapjuk meg a file-ok elérési útvonalát a fa struktúrában.

tar

# Ki-be tömörítő

Listázás

-----

-tvf file.tar : megmutatja a mentes.tar tartalmát  
 -tvfz file.tar.gz : megmutatja a mentes.tar.gz tartalmát  
 -tvfj file.tar.bz2 : megmutatja a mentes.tar.bz2 tartalmát  
 -tvvf mentes.tar : részletes listázás, jogok, tulajdonos etc.

Betömörítés

-----

-cvf file.tar /eleresi/ut : az elérési útvonal alatt levő adat  
 -cvfz file.tar.gz /eleresi/ut/amit/mentunk : az elérési útvonal alatt levő adat  
 -cvfj file.tar.bz2 /eleresi/ut/amit/mentunk : az elérési útvonal alatt levő adat

Kitömörítés

-----

-xvf file.tar : Kitömöríti a tar-t és egy /file mappába teszi  
 -xvfz file.tar.gz : Kitömöríti a gz-t, majd a tar-t, és egy /file könyvtárba tes:  
 -xvfj file.bz2 : bzip2-t tömöríti ki

Exclude példa.:

-----

-cvfz file.tar.gz /home/mit/ --exclude={"/home/MIT\_NE,/home/MIT\_NE2}  
 # /home/mit mappa tartalmát betömörítjük gzip-be, kivéve /home/MIT\_NE, /l

uniq

# Több sorból álló szövegben az ismétlődő sorokkal kezd valamit  
 -c fájl : az egymás utáni azonos sorokból egyet hagy meg, és kiírja a sorok elejé  
 # ismétlődött az adott sor

vi

# Parancssoros szövegszerkesztő  
 # Kilépés mentéssel : ESC, majd :wq  
 # Kilépés mentés nélkül : ESC, majd :q!

vi -b +/mikulas fájl

b : Bináris állomány szerkeszthetőségét teszi lehetővé  
 +/mikulas : a fájl-ban a mikulas kifejezéshez ugrik

wc fájl

```
# sor, szó, karakter számítása
-m fájl : a fájlban lévő karakterek száma
-c fájl : a fájlban lévő bájtok száma
-w fájl : a fájlban lévő szavak száma
-l fájl : a fájlban lévő sorok száma
```

which program

```
# A program futtatható állományának elérési útvonalát adja meg (általában /usr/bin)
```

watch cat file

```
# A file változásait figyeli, 2 másodperces időközönként frissítve.
```

## Kvóták beállítása Linux alatt

Apt-vel telepítsük a quota csomagot.

quota

```
# Tároló kvótázás programja
```

quota -v

```
# A bejelentkezett felhasználó kvótabejegyzései
```

quotacheck -uagv

```
# Végignézi a fájlrendszert, a tárkorlátok adatainak begyűjtése érdekében
# Majd feltölti az adatbázist (aquota.user, aquota.grp)
-u : alapértelmezett
-a : minden fájlrendszer ellenőrzésre kerül
-g : csoportadatok keresése
-v : verbose, bőbeszédű
```

quotaon -augv

```
# kvóták bekapcsolása
```

quotaoff -av

```
# kvóták kikapcsolása
```

repquota -av

```
# megtekinthetjük felhasználónként a kvótabejegyzéseket
```

edquota -u user

edquota -g group

```
# Felhasználó, csoport kvóták beállítása
# Alapértelmezett szerkesztőnk átmeneti állományaként állíthatjuk be
# Ezért érdemes: export EDITOR=kedvenc_szerkesztőnk -et beállítani.
```

```
-p user user2: user2 kvótáit állítja be, user mintájára
```

```
/etc/fstab file-ban, az adott partícióhoz meg kell adnunk az options-ba a usrquot
```

## Rendszeradminisztráció [top](#)

login

```
# Bejelentkezés
```

```
logout
    # Kijelentkezés

who
    # Bejelentkezett felhasználók kiírása

w
    # Nemes egyszerűséggel megmutatja, ki van belépve, és mit csinál

whoami
    # Milyen néven is jelentkeztünk be?

users
    # Kiírja az rendszeren levő felhasználók nevét

adduser
    # Új felhasználó létrehozása (felhasználóbarát)

useradd
    # Új felhasználó hozzáadása
    # Ha opciókat használunk, a user-nevet a sor végére írjuk!
    -u szám                : felhasználói azonosító megadása (UID)
    -g csoport             : csoport tagság beállítása
    -d /home/user          : A felhasználó HOME mappáját állítjuk be /home/user -re.
    -s /bin/false          : Alapértelmezésként nem kap shell-t bejelentkezéskor.
    -G csoport1,csoport2   : vesszőkkel elválasztva, ha egyéb csoportnak is tagja a
    -m -k /home/letezo uj  : egy létező felhasználó mintájára építi fel az új user l
        # Minden fájl és könyvtár is belekerül a fiókba, + a beállítások is.

usermod
    # Felhasználói fiókok módosítása
    -u érték user          : a felhasználó azonosítóját változtatja (UID)
    -g csoport user        : a felhasználó csoportját változtatja
    -G csoport1,csoport2   : a jüzer-t több csoporthoz is rendeli
    -L user                : lock-olja a felhasználó hozzáférését (nem tud belépni)
    -U user                : unlockolja a hozzáférést

userdel
    # Felhasználó törlése
    -r user : home-al együtt törli

deluser
    # Felhasználó törlése

groupadd
    # Csoport létrehozása
    -g szám csoportnév : Egyénileg választott csoportazonosítóval (GUID)-el hozunk l
    # /etc/group : csoportlista

groups user
    # A user csoportjait adja meg.

id -nG user
    # Szintén.

groupmod
    # Csoport fiókját változtatja
    -n név újnév : csoport nevét változtatja

gpasswd
    # Csoport adminisztráció.
    groupka                : groupka nevű csoport létrehozása
    -a Peti groupka        : Peti nevű user-t hozzáadja groupka csoporthoz.
    -d Peti groupka        : Peti-t kitörli groupka csoportból
```

```

history
    # Sorszámozottan megadja az utolsó N parancsot, amit az aktuális felhasználó adott

cat /home/username/.bash_history
    # username nevű felhasználó saját history-jának kiíratása, root-ként engedélyezett

!n
    # n-edik sorszámú parancs újra futtatás

!-n
    # n-el korábbi parancs futtatása

!!
    # legutóbbi parancs újra futtatása

history -c
    # clear history

export HISTTIMEFORMAT='%F %T '
history
    # "EV-HONAP-NAP Ora:Perc:Masodperc Parancs" formátumban kapjuk meg a history-t

unset export HISTTIMEFORMAT
    # visszaállítás az eredeti állapotra/kimenetre

export HISTCONTROL=ignoredups
    # duplikálás kiszűrése. Kikapcsoláshoz szintén unset export..

export HISTSIZE=0
    # history kikapcsolása

Permanens beállításhoz
-----
/home/username/.bash_profile file-ba fel kell venni a fenti sorokat.

talk
    # Felhasználókkal való kommunikálás egyik eszköze

finger
    # Felhasználó információi

chfn
    # Felhasználói információk megváltoztatása (finger kimenete)

last
    # Ki jelentkezett be utoljára és honnan (felhasználó vagy terminál alapján)

lastlog
    # /var/log/lastlog kimenete

locale
    # Kiírja az aktuális lokalizációt
dpkg-reconfigure locales
update-locale
    # Újra konfiguráljuk a locales-t, ezáltal új nyelveket használhatunk a rendszeren
    # A második sor is kötelező.

passwd
    # A bejelentkezett felhasználó jelszavának módosítása
    # Userként kiadva saját jelszavunkat változtatjuk meg. A régit tudnunk kell hozzáadni
user      : user nevű felhasználó jelszavát változtatja meg. root-ként, nincs szükség
-l user  : lock user account. user jelszavát ideiglenesen megváltoztatja egy random
-u user  : unlock user account.: Az eredeti jelszót visszaállítja user számára.
-e user  : expire. A user jelszavának azonnali lejáratát. Következő belépésnél új jelszót

```

```

-S user : status. Státusz információk kiírása. 7 mezőből áll melynek értékei.:
    # 1.: Usernév,
    # 2.: L-locked, NP-No Password, P-Password
    # 3.: Utolsó jelszó váltás dátuma
    # 4.: min. év. (napokban)
    # 5.: max. év (napokban)
    # 6.: figyelmeztetési periódus (napokban)
    # 7.: inaktív periódusa a jelszó tekintetében.

stat file
    # File név, relatív elérési útvonal, méret, hozzáférési jogok, access, modify, ch
    # UID, GID, egyéb hasznos infók kiírása

su - user
    # Indít egy másik shell-t, user felhasználóként

smartctl
    # HDD vizsgálatára, SMART értékek figyelésére alkalmas program.
    # smartmontools csomag tartalma, telepítenünk kell repo-ból.

-i /dev/sda : Eszközinformációk kiírása.

    # Ha a SMART support nincs bekapcsolva az eszközön, az alábbi paranccsal
    -s on /dev/eszköz

-A /dev/sda : Attributum táblák megjelenítése. Legtöbbször ezekre az értékekre vagy:
-t long VAGY --test=long /dev/sda : sda eszköz teljes vizsgálata
-t short VAGY --test=short /dev/sda : sda eszköz gyors vizsgálata

    # Az eredményeket utóbbi két esetben az alábbi paranccsal nézhetjük meg.:
    -l selftest /dev/sda

-H /dev/sda : Az sda eszközünk jó-e vagy sem?
--all /dev/sda : Minden SMART információ kiírása.

write user tty
    # Egy Linux gépre bejelentkezett felhasználónak üzenet küldésre szolgáló parancs

mesg
    # A write üzenetküldő aktuális beállítását kérdezzük le. (engedélyezve vagy til:

mesg n
    # A write üzenetküldő tiltása

mesg y
    # A write üzenetküldő engedélyezése

```

## Folyamatok [top](#)

```

pidof folyamat
    # A folyamat azonosítóját adja meg (PID)

command &
    # command futtatása a háttérben
fg : felélesztés, majd előtérben futás
bg : felélesztés, majd háttérben futás

```

```

command1 && command2
# command1 sikeres futását követően command2 is lefut.
# Amennyiben command1 visszatérési értéke nem 0 (tehát sikertelen), úgy az utána

letix@microserver:~/test$ ls && echo masodik parancs sikeres
dir1 dir2
masodik parancs sikeres
letix@microserver:~/test$ ls dir3/ && echo masodik sikeres
ls: dir3/ nem érhető el: Nincs ilyen fájl vagy könyvtár

jobs
# Háttérben futó programok kiírása

ps
# Futó folyamatok kiírása
-u pisti : pisti felhasználó által futtatott folyamatok
aux      : minden folyamatot kiír, szinte minden információval
alxww    : minden folyamatot, még több infóval (pl.: PPID)
-tl      : ttyl-es terminál kilistázása
f        : erdő szerű megjelenítés
l        : kiírja a folyamatok PID-jét és PPID-jét is. (parent's process identifi:
ww       : a programok parancssori kapcsolóit írja ki.

pstree
# Folyamat struktúra fa-szerű ábrázolása.

nice
# Prioritása lekérdezése, beállítása
-n --20 program : a rendszer a legtöbbet ezzel a programmal fog foglalkozni. (+1)

renice -10 1124
# 1124-es folyamat -10-es prioritásra állítása

kill "pid"
# Folyamat leállítása PID szerint
-9 pid : erőszakos bezárás
-1     : jelentése SIGHUP, mely a folyamat bezárása, config fájl beolvasása, fo:
-9     : jelentése SIGKILL, folyamat erőltetett bezárása. (Csak végszükség eseté:
-15    : jelentése SIGTERM, szabályos programleállítás
-s "signal" : megadja, hogy milyen jelet küldünk a folyamatnak.
#signal-ok: TERM, QUIT, ABRT, KILL

killall parancs
# Ha több "parancs"-fut (pl.: iceweasel), mindet leállítja

nohup find / name t* > ~/nohup.log &
# A "t" betűvel kezdődő fájlok keresése a háttérben, az eredmény a nohup.log-ba l

init
# Futtatási szint beállítása (run levels)
# Az egész rendszerünk állapotjelzője, ami meghatározza, mely szolgáltatások
# működnek vagy épp indulnak el rendszerindításkor. A futási szinteket számokkal
# Definiálhatunk magunknak különböző szinteket annak függvényében, hogy milyen je
# szánjuk éppen a gépünket. Pl ha X-el, bluetooth-al és egyéb erőforrás igényes a
# szeretnénk használni a gépünket létrehozhatunk pl egy 4-es init szintet.
# Ezt a szintet az inittab-ban beállíthatjuk alapértelmezettként induláshoz, de a
# közben is ugrálhatunk a szintek között. -> Akár több szolgáltatást is indíthatu
# állíthatunk le egy paranccsal.
#
# Szintek:
# -----

0 : kikapcsolás
1 : single-user mód (speciális rendszeradminisztrációs funkciókra)

```

```

2-5 : multi-user mód, (normál működés)
6   : reboot

# /etc/inittab -ban állítható be a gép alapértelmezett indulási init szintje.
# configból.:
    The default runlevel.
    id:2:initdefault:

# /etc/rcX.d könyvtárak a megfelelő szinteken induló programok linkjeit tartalma:
#
# Új szolgáltatás hozzáadása init szintekhez : Program bemásolása /etc/init.d-be,
# egy link létrehozása a kiválasztott init szint könyvtárába (pl.: rc2.d)
# Csak akkor indulnak el, ha S betűvel kezdődnek.
# A kezdőbetű után levő számok az induló folyamatok sorrendjét befolyásolják.

```

Daemonok kezelése.:

-----

```

/etc/init.d/daemon_nev start      : daemon indítása
/etc/init.d/daemon_nev stop      : daemon leállítása
/etc/init.d/daemon_nev restart   : daemon újraindítása
/etc/init.d/daemon_nev status    : daemon status infók kiírása

```

runlevel

```

# Megadja, hanyas init szinten voltunk és vagyunk. (kimenet pl.: N 3 , vagy 3 2.)
# 3 N jelenti, hogy 3-ason voltunk és vagyunk, 3 2 pedig hogy 3-ason voltunk 2-es

```

shutdown

```

# Kikapcsolás
-h now      : Azonnali kikapcsolás (időt is megadhatunk)
-h 12:00 &  : A gép kikapcsolása 12:00-kor.
-c          : Az időzített kikapcsolási folyamat megszakítása
-r 0        : Azonnali újraindítás

```

## Időzített parancsfeldolgozás [top](#)

at

```

# Megadott időpontban futtathatunk programokat
-f todo 23.59 : előre megírt parancsainkat (todo fájlban) 23:59-kor lefuttatja a:
# Idő formátumok
    # 13.13 01.01.02 : 13 óra 13 perc, 2002, január 1
    # 2pm tomorrow : holnap délután 2
    # 1am Sun : hajnali 1 óra, vasárnap

```

atq

```

# Az at várakozási sorrendjét ismerteti

```

atrm pid

```

# Az at várakozási sorából való eltávolítás, Process ID alapján

```

cron

```

# A Linux parancs ütemezője
# /etc/cron.d ; /etc/cron.daily ; /etc/cron.weekly
crontab -l : kilistázza a belépett user beállított ütemezéseit
crontab -e : Editáljuk a belépett user ütemezéseit

```

Crontab paraméterezése.:

-----

Perc (0-59)	Óra (0-23)	Hónap napja (1-31)	Hónap (1-12 v. Jan-Dec)	Hét napja (0-6 v Vas.-Szo.)	Parancs Parancs
----------------	---------------	-----------------------	----------------------------	--------------------------------	--------------------

Példák  
-----

**0**                **2**                **12**                **\***                **0,6**                **ping 192.168.1.1**

Minden hónap minden szombatján és vasárnapján amelyek 12.-ére esnek, 2:00-kor me

**30**                **10**                **\***                **\***                **\***                **ping 192.168.1.1**

Minden nap, 10:30 kor megpingeli a címet.

**00**                **1-8,12-17**                **\***                **\***                **\***                **ping 192.168.1.1**

Minden nap, 1:00-től 8:00-ig és 12:00-zól 17:00-ig minden óra 0. percében fut

Alkalmazhatunk például vesszőt is, az első rublikába írva "0,30" jelenti minden óra 0. illetve 30. percét, illetve intervallum is megadható.

További példák.:  
-----

```
@reboot parancs : A következő indításnál lefutó parancs.
@weekly          : "0 0 * * 0" : Hetente egyszer fut le, vasárnap éjféltkor.
@daily           : "0 0 * * *" : Naponta egyszer fut le, éjféltkor.
@midnight        : "0 0 * * *" : éjféltkor, ekv. az előzővel.
@hourly          : "0 * * * *" : Minden egész órakor fut le.
```

date

```
# Dátum kiíratása vagy beállítására használatos program
+%F : 2011-11-04 formátumban írja ki az aktuális dátumot.
+%Y%m%d : 20111104 formátumban írja ki az aktuális dátumot.
-d-2day +%F : 2011-11-02 formátumban írja ki a 2 nappal korábbi dátumot.
11041153 : Beállítja az dátumot aktuális év, November 04., 11:53-ra.
```

sleep 5 parancs

```
# 5 Másodperc múlva indítja a "parancs"-ot.
```

schedutils

```
# Linux rendszer ütemező
```

screen

```
# Ablakkezelő, virtuális terminál emulátorral.
# Háttérbeli programok futtatásához alkalmazható program.
```

```
screen          : Egy új VT-t (virtuális terminált) kapunk.
Ctrl+a+d        : bill. kombinációval tehetjük háttérbe.
screen -ls       : screen-ek listázása
screen -d -m -S name : elindítja a háttérben a screen-t "name" névvel
screen -x name    : name nevű terminált hozzuk előtérbe
screen -R PID     : paranccsal tudjuk előtérbe hozni. (PID - processID, p
```

```
# SCREEN-ben kiadható billentyű kombinációk
-----
```



```

Ctrl+a+d          : bill. kombinációval tehetjük háttérbe.
Ctrl+a+?          : előhozhatjuk a legfontosabb bill. kombinációkat
Ctrl+a+c          : új ablakot hozhatunk létre.
Ctrl+a+p VAGY n   : előző vagy következő ablakra ugrás. (previous, next)
Ctrl+a :kill       : aktuális screen lelövése
Ctrl-a :acladd USER : USER nevű felhasználó engedélyezése becsatlakozásra

# MULTIUSER MODE
# -----
#
# A screen többfelhasználós üzemmódja
# Amennyiben kontrollált körülmények között szeretnénk beengedni valakit a gépünkre
# használva, úgy az alábbi lépéseket kell alkalmaznunk.:

screen
    # screen indítása a kiszemelt többfelhasználós gépen

Ctrl+a :multiuser on
    # Ezt begépelve aktiválhatjuk az adott session-on a többfelhasználós módot
    # Másik oldalnak szükséges ismernie az adott user-ünk nevét/jelszavát, ill.
    # csatlakozni a gépünkre. Ha mindez megvan, és SSH-n bejött.:

screen -x
    #Voila!

```

```

time command
    # A command lefutási idejét méri

command &
    # command háttérbeli futtatása

command1 && command2
    # command1 visszatérési értékének függvényében command2 is lefuthat. (ha command1

```

## Kernel,modulok,fordítás [top](#)

```

lsmod
    # Betöltött modulok kilistázása

rmmod modulnév
    # Betöltött modul lekapcsolása a kernelből

modprobe modulnév
    # Modult tölt be a kernelbe

modinfo modulnév
    # a modulról kapunk információt

ldd program
    # Kiírja a programhoz szükséges megosztott könyvtárakat, függőségeket

# Linux kernel fordítás menete, röviden. (ez csak az egyik módszer)

```

[bővebben](#)

```
# Megfelelő kernel verzió beszerzése (www.kernel.org, pl.: linux-2.6.20.3.tar.bz2)
# apt-get install wget bzip2 binutils kernel-package module-init-tools initrd-tools
# tar zxvf linux-2.6.20.3.tar.bz2 (érdekemes a /usr/src-be)
# Belépés a létrejött könyvtárba
# make mrproper (ha már próbálkoztunk beállításokkal)
# apt-get install libncurses5-dev
# make menuconfig (az érdemi munka)
# make-kpkg clean
# make-kpkg --initrd kernel-image kernel-headers
# cd ..
# dpkg -i *.deb (2db állomány)
```

## Csomagkezelés [top](#)

```
apt      /Advanced Package Tools, A Linux csomagkezelője/
-cache search      : keresés
-cache showpkg csomag : megmutatja a csomag függőségeit
-cache policy csomag : a csomag telepített és telepíthető verzióit adja meg
-get install csomag : telepítés
-get remove csomag  : törlés
-get autoclean      : nem használt csomagok törlése
-get update         : frissítés
-get upgrade        : frissítések telepítése
-get dist-upgrade   : distro frissítés
-get -f install     : törött, hiányzó csomagoknál a barátunk.
-get clean          : apt cache tisztítása
-get build-dep csomag : telepítés minden egyes csomagjával együtt
-get source csomag  : a csomag forrásának letöltése
-file list csomag   : kilistázza a csomag tartalmát

-s kapcsoló esetén csak szimulálja a megadott feladatokat.
```

```
echo "csomag hold" | dpkg --set-selections
# csomag nevű csomagot visszatartjuk a frissítések során.
# visszaállítása: hold helyett install-al adjuk meg.
```

```
dpkg --get-selections
# feltelepített csomagjaink listáját kapjuk meg, állapotuk feltüntetésével.
```

```
dpkg      /Debian packages/
# .deb csomagkezelő
-i csomag      : telepítés
-r vagy --purge : törlés
-reconfigure csomag : csomag újrakonfigurálása
-l            : csomagok listázása
-I csomag      : csomag információk megjelenítése
-S fájl        : megmutatja melyik csomaghoz tartozik a fájl
```

A feltelepített csomaglistánk mentése, majd a mentett listát felhasználva a csomagok állapota és a csomagok telepítése.

```
dpkg --get-selections > /eleresi/ut/csomagok.txt : TXT-be mentjük a feltelepített
dpkg --set-selections < /eleresi/ut/csomagok.txt : a TXT-ből kiolvassa a felsorolt
apt-get -y update
apt-get dselect-upgrade
# Első lépésben a feltelepített csomagok listáját a csomagok.txt-be mentjük
# Második, harmadik, negyedik lépésben pedig a telepített sw.-k listájába
```

```
# csomagokat telepítjük rendszerünkre.

dselect
    # A dpkg barátságosabb arca

./configure
make
make install
    # Csomag telepítése forrásból.
    # A parancsot abban a mappában szükséges kiadni, ahova lett másolva az adott prog

make clean
    # Ha esetleg valamilyen kapcsolót kihagytunk a fordítás közben,
    # ezzel a parancsal pucolhatunk az újrafordítás előtt.

make uninstall
    # Forrásból telepített program uninstall

alien
    # Csomag konverter program (pl.: .rpm .deb etc.)
```

## A Red Hat csomagkezelője

```
rpm
    # Red Hat Package Manager

rpm -ivh csomag.rpm
    # csomag telepítése

rpm -Uvh csomag.rpm
    # Frissítés vagy telepítés.
    # Ha nincs telepítve , telepíti, ha telepítve van, megpróbálja frissíteni

rpm -Fvh csomag.rpm
    # Csak frissítés. Ha nincs telepítve a csomag, kilép

rpm -e csomag.rpm
    # Törlés

rpm -q csomag.rpm
    # Megvizsgálja, hogy telepítve van e a csomag, és milyen verzió

rpm -qa
    # Az összes telepített csomag vizsgálata
```

## Hálózat [top](#)

```
arp
    # ARP cache manipulálása

ifconfig
    # Hálózati információk lekérése, konfigurálása

ifconfig eth0 192.168.10.1 netmask 255.255.255.0 up
    # 192.168.10.1-es IP-t osztunk az eth0-ra (0-s hálókártya)
```

```
# 255.255.255.0 a netmask.

ifdown eth0
# eth0 eszközt lekapcsolja

ifup eth0
# eth0 eszközt felkapcsolja ( dhcp esetén kér IP-t a kiszolgálótól)
```

## iptables

```
# A Netfilter (hálózati csomagok feldolgozására szolgáló eszköz) kezelője.
# Az iptables a hálózati csomagok feldolgozási szabályait szerepük alapján szerve
# táblák formájában kezeli.
#
# Szerepek (táblák - a feldolgozási folyamat típusát jelentik)::
#   Csomagszűrés (filter)
#   Hálózati címfordítás (nat)
#   Egyéb csomagmódosítás (mangle)
# Ezen táblák mindegyikére feldolgozási szabályok (rules) lánc (chain) vonatkozik.
# A szabályok illesztésekből (match - mely csomagokra vonatkozik,) és célokból (target - melyre illeszkedik az illeszkedő csomaggal) épülnek fel.
#
# Az iptables öt kapcsolódási pontot határoz meg a feldolgozás során::
#   PREROUTING : A hálózati csatolótol kapott csomag, még az INPUT lánc előtt
#   INPUT : Épp mielőtt megérkezne a csomag egy helyi folyamathoz. (itt illik a csomagot feldolgozni)
#   FORWARD : Egy átjáró gépen a továbbított csomagok itt áramlanak keresztül
#   OUTPUT : Miután egy helyi folyamat létrehozta a csomagot, itt távozik (kimeneti)
#   POSTROUTING : Mielőtt elhagyná a hálózati csatolót a csomagunk, az OUTPUT lánc után
#
# A táblák, és beépített láncok:
#   filter tábla: FORWARD, INPUT, OUTPUT
#   nat tábla: PREROUTING, OUTPUT, POSTROUTING
#   mangle tábla: PREROUTING, FORWARD, INPUT, OUTPUT, POSTROUTING
#
# A láncok házirendje (policy)::
#   Ezt a policy-t láncként (INPUT, OUTPUT, FORWARD) a tűzfal szkriptünk leírja.
#   Azok a csomagok, amelyek a tűzfal láncában felsorolt szabályokra nem illeszkednek, azaz nem
#   policy vonatkozik, azaz vagy ACCEPT, vagy DROP.
#   Pl.: Ha a tűzfalunk INPUT láncaiban csak a 80-as portot engedélyezzük, a port 80-ra
#   akkor minden más porton érkező csomagot (amely nem illeszkedik a 80-as portra)
#   DROP-oljuk.
#
# Lánc műveletek, szabály módosítások:
#   -N : új lánc létrehozása
#   -X : üres láncok törlése
#   -P : default policy beállítása
#   -L : a láncok szabályainak megjelenítése
#   -F : a lánc összes szabályának törlése
#   -A : új szabály hozzáadása egy lánchoz
#   -D : törlés az adott láncból
#
# Forrás (-s, --source, --src) és célcímek (-d, --destination, --dst) megadásánál
#   Teljes név használata.: localhost, vagy pl.: www.debian.org
#   IP cím alapján a meghatározás.: 127.0.0.1
#   IP cím tartomány(ok) alapján.: 192.168.1.0/24 vagy 192.168.1.0/255.255.255.0
#   (Mindkét tartomány 192.168.1.1-től 192.168.1.254-ig fedi le az IP-címeket)
#   Inverz módon is megadhatunk forrást és célt, "-s !" localhost" formájában
#
# Protokollok megadása.:
#   -p TCP vagy UDP vagy ICMP formájában történik.
#   Itt is alkalmazható az inverz, azaz "-p ! TCP", minden csomagra illeszkedő
#
# Interface meghatározása.:
#   Az interface maga a fizikai eszköz amelyen bejön, vagy távozik a csomag.
#   INPUT láncon "-i eth0" értelmezhető, nincs kimeneti interface-ük.
#   OUTPUT láncon "-o eth0" adható meg, "-i" kapcsolóra egy csomag sem fog illeszkedni
```

```
# FORWARD láncon alkalmazhatóak a "-i eth0" és "-o eth0" interface-ek.
# Alkalmazható joker karakter az interface-eknél, ez a "+" jel. "-i eth+" ->
#
# Portok, port tartományok megadása
# A portok megadhatóak számmal, vagy névvel. (utóbbi esetben.: /etc/service
# Port tartományokat "port:port" formában kell alkalmaznunk. Invertálás itt
# --source-port VAGY --sport : a csomagok forrásportját adjuk meg
# --destination-port VAGY --dport : a csomagok cél portját határozzuk meg.
```

Folytatás később. Addig emésszétek! :)

ipcalc

```
# TUI felületű IP számláló. (hasznos lehet hálózat tervezés/bővítéskor)
```

darkstat

```
# Hálózati statisztikák gyűjtésére szolgáló program.
```

host

```
# Lekérdezéseket kezdeményez a DNS kiszolgáló felé, tartományokról, zónákról.
-t ns valami.hu : valami.hu Name Server-ét adja meg
-t mx valami.hu : valami.hu levelező szervereiről ad információt
```

hostname

```
# Beállítja vagy megjeleníti a rendszer host-nevét.
```

iperf

```
# Sáv szélesség korlátozásra használatos program.
```

```
curlftpfs -o allow_other username:password@ftp.domain.dom /csatolasi/pont
```

```
# Felcsatolhatunk ftp tárhelyeket, lokális fájlrendszerünkbe. (fusermount segítsé
```

```
# Ha a hiba.:
```

```
# "If you don't use udev, you may get this error message:
```

```
# fusermount: failed to open /dev/fuse: No such fdevice or address
```

```
# Before loading the fuse kernel module, create the device node manually:
```

```
# A megoldás.:
```

```
mknode -m 666 /dev/fuse c 10 229
```

```
echo 0 > /proc/sys/net/ipv4/ip_forward
```

```
# Ha gépünket router-ként használjuk, érdemes beállítanunk a csomag továbbítást
```

```
# 1-es engedélyezi a tiltást, 0 tiltja a tiltást!
```

```
echo 0 > /proc/sys/net/ipv4/icmp_echo_ignore_broadcasts
```

```
# Beállítjuk, hogy broadcast ping-re ne válaszoljon gépünk
```

```
echo 0 > /proc/sys/net/ipv4/icmp_echo_ignore_all
```

```
# Beállítjuk, hogy ne válaszoljon gépünk a ping-re
```

```
echo 0 > /proc/sys/net/ipv4/conf/all/rp_filter
```

```
# Engedélyezzük a hamis csomagok szűrését
```

...etc

```
-----
```

```
# Ezek a beállítások rendszerleállításkor elvesznek, érdemes őket valamely indulá
```

```
# lefutó script-be tenni, a hálózati interfészeket konfiguráló script elé.
```

```
-----
```

etherwake

```
# Magic packet-et küld a wake-on-lan enabled hálókártyára
```

ethtool

```
# Ethernet kártya beállítások megjelenítése, változtatása

iptraf
# Hálózat terhelést figyelő program

iptstat
# Top-szerű hálózat elemző eszköz.

mii-tool
# Hálózati tesztelő

nbtscan
# NetBIOS információkat adja meg, a hálózaton

netstat
# Információkat ír ki a hálózati alrendszeréről
-r : route kimenete
-i : interfész tábla
-tP : TCP/IP -t használó processzek listája

netstat -tn|grep ':80'|tr -s ' '|cut -d ' ' -f5|cut -d ':' -f1
# kik csatlakoznak a gépünkre a 80-as porton?

netstat-nat
# Nat-olt kapcsolatok megjelenítésére szolgáló eszköz.
-n : Nem fordítja le a megjelenített IP-ket Hostname-ekre.

netcat
# TCP/IP swiss army knife

nload
# Network load - Hálózati terhelés vizsgálat

ngrep
# grep, hálózati forgalomhoz

netsed
# Hálózati csomagok valós idejű megváltoztatására alkalmas program

nslookup www.host.com
# host.com IP címének vizsgálata

nmap
# Elsősorban port scanner, de igen sokrétű alkalmazás
-O ip_cim : Op.rendszer detektálás
-p 1-10 ip_cim : Portok vizsgálata 1-10 ig.
-sP 192.168.10.0/24 : komplett 10.0-s hálózat ping scan.
-p U:53,111,137,T:21-25,80,139,8080 ip_cim : UDP-n 53,111,137, TCP-n 21-25,80,139

ping host
# Ping jelet küld a host-nak
-b ip_tartomány : broadcast üzenet küldés

route
# A Linux útválasztó táblájának megjelenítése (Routing table)

route add -net 192.168.10.0 netmask 255.255.255.0 eth0
# A 192.168.10.0-s hálózatra menő csomagokat az eth0-ra irányítja a rendszer
# A hálót így már látnunk kell

route add default gw 192.168.10.254
# 192.168.10.154-es IP-vel rendelkező router (gateway) hozzáadása a routing table
```

```
route del default gw
    # Alapértelmezett átjáró törlése a routing table-ből.

Static route hozzáadás
-----

up route add -net 192.168.2.0 netmask 255.255.255.0 gw 192.168.2.1
    # Static route hozzáadása az adott eth eszközhöz. ( /etc/network/interfaces ethX
    # Az ethX eszközünk 192.168.2.0 hálózat felé a 192.168.2.1 legyen a default gatev

down route del -net 192.168.2.0 netmask 255.255.255.0 gw 192.168.2.1
    # A Fenti kikapcsolása.

rsh
    # Távoli shell
    -l user gepnev parancs : a "parancs"-ot lefuttatja a távoli gépen, user nevében

smbclient
    # SMB hálózati erőforrások tallózása

smbclient -M gépnév
    # Üzenet küldése Windows-os gépekre (net-send szolgáltatás). Elküldés:Ctrl + D a:
    echo üzenet | smbclient -M gépnév : azonnal küldi, nem kell Ctrl+D

ssh host          /Secure SHell/
    # Titkosított hálózati bejelentkezés
    # config file-k.: /etc/ssh/sshd_config, /usr/local/etc
    ssh-keygen -t rsa : RSA kulcs létrehozása ssh-hoz.

scp
    # Biztonságos távoli fájl átvitel

sntop
    # Top-szerű hálózati forgalom figyelő szoftver.

tcpdump
    # Hálózaton átmenő forgalmat figyeli.
    -i ethX : Az ethX csatolón átmenő forgalmat figyeli.

traceroute host
    # A hálózati útvonal kiírása a célszerverig (host) /IP hálózaton/

trickle
    # Sáv szélesség korlátozásra használatos program.
    -u 100 -d 200 proftpd : A proftpd démon számára biztosít 100KB fel, 200KB letölté

tsclient
    # Terminal Services Client

proftpd
    # FTP kiszolgáló démon
    # config file.: /etc/proftpd

pure-ftp
    # FTP kiszolgáló

rdesktop
    # Remote desktop kliens

stunnel
    # SSL (Secure Socket Layer) titkosított kapcsolatba csomagolhatunk protokollokat.
    # pl.: POP3, IMAP, etc.
```

```

squid
    # Proxy kiszolgáló
    # config file.: /etc/squid/squid.conf

vconfig
    # virtuális eth eszközök létrehozására alkalmas program.
    # vlan csomag tartalmazza, tehát ezt telepítenünk kell.
    # vconfig add eth0 1 : az eth0:1 eszköz hozzáadása, majd fel kell vennünk a /etc/i

    # Példa config.:
    # -----

    ...

    auto eth0
    allow-hotplug eth0
    iface eth0 inet static
        address 192.168.1.42
        netmask 255.255.255.0
        gateway 192.168.1.254

    auto eth0:0
    allow-hotplug eth0:0
    iface eth0:0 inet static
        address 192.168.1.43
        netmask 255.255.255.0

    auto eth0:1
    allow-hotplug eth0:1
    iface eth0:1 inet static
        address 192.168.1.44
        netmask 255.255.255.0

vsftpd /Very Secure FTP daemon/
    # config file.: /etc/vsftpd
    # Néhány fontos kérdés.:
    -anonymus_enable : anonymus ftp engedélyezése. Szükség van egy ftp nevű user-re.
    -local_enable : normal felhasználókat engedjük be. (kell az anonymus hoz is)
    -write_enable : engedi az írást
    -local_umask : felhasználók automatikus jogai (default 022)
    -anon_upload_enable : anonymusként feltölthetünk
    -anon_mkdir_write_enable : anonymusként mkdir engedélyezés
    -xferlog_enable : xfer log-ot ír a daemon
    -connet_from_port_20 : 20-as portot használ
    -chown_uploads : anonymusként feltöltött fájlok tulajdonosának megváltoztatása
    -chown_username : Kinek a tulajdona legyen a file
    -idle_session_timeout : inaktív kapcsolatok bontási ideje

wireshark
    # Hálózati forgalom figyelő (GUI)

wondershaper
    # Egyszerű sávszélesség korlátozásra használatos program.
    ethX 1024 1024 : Az ethX eszköz részére biztosít 1024Kbit le és feltöltési sávszé
    ethX clear : Törli a beállításokat.

```



```
# A parancsértelmezőnk által ismert parancsokat fűzhetjük össze egy "programba".
# Programunk első sorában minden esetben meg kell adnunk a futtató programot, jelezzük, hogy a fájlunk kezdete legyen: #!/bin/bash
# Felkiáltójel nélkül más futtató programot (parancsértelmezőt) is megadhatunk a
# Futtatási jog nélkül nem tudjuk használni, ezért chmod +x programnev
# Programunk futtatásához ./programnev parancsot kell használnunk.
```

## Írásjelek

```
# Escape-karakter: \
    Bourne Again Shell escape-karaktere. Többek között ezzel a jellel lehet jelezni, hogy az őt követő speciális karakter elveszítse módosító hatását.

# Dollárjel: $
    Segítségével változó neve és értéke között tudunk különbséget tenni.
    Ha $változo-t írunk, az a változónk értékét fogja visszaadni.

# Kétszeres idézőjel: " "
    Az idézőjelek arra valók, hogy egyes karakterek vagy lefoglalt szavak speciális jellegét feloldják.

# Egyszeres idézőjel: ' '
    Megakadályozza a héjat (shell), hogy a $-t a változók jelzésének tekintse.

# Szögletes zárójel: []
    Intervallumok megadására szolgáló eszköz.

# Kérdőjel: ?
    Egy elem helyettesítésére szolgáló jel.

# Csillag: *
    Joker karakter, bármennyi karakter helyettesítésére.

# Felsővessző: ` (jobbra tartó - AltGr + 7)
    Operátor, mely a két ` ` között lévő jelsort végrehajtja, és a kimenetét úgy, hogy az új sor karakter helyére szőközt rak.

# Kettőskereszt: #
    Kommentezés eszköze. (megjegyzés)

# And jel: &
    Parancsok háttérbeli futtatására szolgáló eszköz.
    Pl.: updatedb &

# Kettős And jel, vagy pontosvessző: && = ;
    Parancsokat futtathatunk egymás után.
    Pl.: cp -R /var/www /home/backup && shutdown -h now
```

## Dedikált környezeti változók

```
EDITOR : Az alapértelmezett szövegszerkesztőnk
HOME : Felhasználói könyvtárunk elérési útvonala
MAIL : Beérkező email-jeink alapértelmezett könyvtára
PATH : Azok az elérési útvonalak, melyekben a shell futtatható állományokat keres
OSTYPE : Az operációs rendszerünk típusa
PAGER : Az alapértelmezett file nézegetőnk elérési útvonalát kell tartalmaznia
PWD : A könyvtár, ahol éppen vagyunk. (Szimbólikus linknél nem egyezik meg a pwd
SHELL : A jelenleg futó shell-ünk teljes elérési útvonala
TERM : A Terminálunk típusát tartalmazza
```

Ezek értékeinek kiíratására a következő parancs használható:

```
echo $VALTOZO_NEV
```

## Shell változók

Értékadó utasítással.:

```
valtozo=érték : a változónk értéke legyen "ertek"
valtozo=$(echo 2+3 | bc) : a változónk értéke legyen 2+3, bc-vel kiszámolva.
export valtozo : shell változónkat környezeti változóvá tesszük
    # Ha azt szeretnénk, hogy restart után is permanens maradjon változónk,
    # az /etc/profile -ba is fel kell vennünk.

unset valtozo : ha már nincs szükség rá, mint környezeti változó.
```

## Belső változók

```
echo $0
    # A futó program neve

echo $$
    # A futó program PID-je

echo $?
    # A legutolsó parancs visszatérési értéke
```

## Csővezetékek (pipe-ok)

```
# " | " segítségével tudjuk egy program kimenetét (stdout) egy másik program
bemenetére (stdin) csatolni. pl.:

ps aux | grep iceweasel
    # Az összes folyamat kimenetét a grep szűrő bemenetére irányítjuk, amely az iceweasel

# stdin - standard input, jele : fd0 (alapértelmezett bemenet, pl.: billentyűzet)
# stdout - standard output, jele : fd1 (alapértelmezett kimenet, pl.: monitor)
# stderr - standard error, jele : fd2 (alapértelmezett hibacsatorna)

./program 2> hiba.txt
    # Programunkat lefuttatjuk, az esetleges hibákat (stderr) a hiba.txt-be irányítjuk
    # az stdout helyett.

./program 1> kimenet.txt 2>&1
    # Programunkat lefuttatjuk, kimenetét (stdout) a kimenet.txt-be mentjük, valamint
    # az esetleges hibákat (stderr) hozzáfűzzük a kimenet.txt-hez.

cat file1 | sort | wc -w > file2
    # file1-et tartalmát a sort-nak adja át, az rendezi, majd átadja a wc-nek, majd a
    # kimenete a file2-be íródik
```

## Néhány egyszerű példa.:

```
#!/bin/bash
echo Hello World!
    # A szokásos Hello World kiírása, programunk futtatásakor.
```

```
#!/bin/bash
STR="Hello World"
echo $STR
    # Az STR változó deklarálása, majd értékének kiíratása.

#!/bin/bash
tar -czf /var/backup.tgz /home/user
    # /home/user tartalmát menti és tömöríti a /var/backup-ba.

#!/bin/bash
OF= /home/user/$(date +%Y%m%d).tgz
tar -czf $OF /home/user/sajt
    # Egyszerű backup script, mely a /hme/user/sajt tartalmát, napi dátummal
    # ellátva egy .tgz állományba tömörítve menti.

#!/bin/bash
if ["foo" = "fool" ]; then
    echo "a feltétel igaz"
else
    echo "a feltétel hamis"
fi
    # Egyszerű if-else szerkezet
```

## Egyszerű szerkezetek.: if, while, until, for

-if szerkezet.:  
-----

```
#!/bin/bash
T1="foo"
T2="bar"
if [ "$T1" = "$T2" ]; then
    echo "a feltétel igaz"
else
    echo "a feltétel hamis"
fi
    # eldöntjük T1 és T2 értékéről, hogy egyenlőek e.

#!/bin/bash
if [ -f /etc/passwd ]
then
    echo "letezik!"
else
    echo "nem letezik!"
fi
    # eldöntjük, hogy /etc/passwd file létezik-e vagy sem.
    # Blokkos eszköz (pl. pendrive) esetén -b kapcsolót kell használni!

#!/bin/bash
test -f /etc/passwd && echo "letezik" || echo "nem letezik"
    # A egyel feljebb levő probléma másik megoldása.
    # Blokkos eszköz (pl. pendrive) esetén -b kapcsolót kell használni!
```

-File szintű kapcsolók

```

if [ -a file ] then ...

    -a : igaz, ha létező file
    -e : igaz, ha létező file
    -b : igaz, ha blokkos eszköz (pl /dev/sda)
    -c : igaz, ha "karakterspeciális" file Pl.: (/dev/ttyACM0 - ez egy nokia telefon.
    -d : igaz, ha könyvtár
    -f : igaz, ha regular file
    -G : igaz, ha a futtató user csoportjái a file
    -N : igaz, ha módosították az utolsó olvasás óta
    -O : igaz, ha a futtató useré a file
    -r : igaz, ha olvasható?
    -L : igaz, ha symbolic link
    -s : igaz, hogy nem üres
    -z : igaz, hogy üres
    -w : igaz, ha írható a scriptet futtató user számára

    # Tagadás mindenhol használható, azaz.:
    ! -f : nem regular file?

```

#### -Változó szintű kapcsolók

```

if [ file == file2 ] then ...

    file1 == file2    egyenlőek?
    file1 != file2    igaz hogy nem egyenlőek?

    if [ "$(cat tibi)" != "$(cat peti)" ]; then
        echo "Tibi nem Peti! :)"
        exit 1 # Stop!
    fi

```

#### -Szám szintű kapcsolók

```

if [ $foo -eq 3 ] then ...

    $foo -eq 3 : igaz, ha foo értéke egyenlő 3-al (equal)
    $foo -ne 3 : igaz, ha foo értéke nem egyenlő 3-al (not-equal)
    $foo -gt 3 : igaz, ha foo értéke nagyobb 3-nál (greater than)
    $foo -ge 3 : igaz, ha foo értéke nagyobb egyenlő 3-al (Greater or equal)
    $foo -lt 3 : igaz, ha foo értéke kisebb 3-nál (Less than)
    $foo -le 3 : igaz, ha foo értéke kisebb egyenlő 3-al (less than or equal)

    "$foo" == "10"    foo értéke 10?

    # Dupla zárójellel használhatóak ==, !=, >=, >, <, <=..
    (($foo <= 3))      foo értéke nagyobb egyenlő 3-nál

```

#### -while szerkezet.:

```

#!/bin/bash
COUNTER=0
while [ $COUNTER -lt 10 ]; do
    echo a szamlalo erteke: $COUNTER
    let COUNTER=COUNTER+1
done
# COUNTER változónkat 1-el növeljük addig, amíg el nem éri a 10-et.

```

-until szerkezet.:

-----

```
#!/bin/bash
COUNT=20
until [ $COUNT -lt 10 ]; do
    echo a szamlalo erteke: $COUNT
    let COUNT-=1
done
# COUNTER változónkat 20-as kezdőértékről 1-el csökkentjük, amíg el nem éri a 10-
```

-for ciklus (C szerű)

-----

```
#!/bin/bash
for i in `seq 1 10`;do
    echo $i
done
# az i változónkat felveszi 1-től 10-ig az értékeket, majd ki is írjuk a képernyőre
```

-for ciklus (a könyvtár elemeire)

-----

```
#!/bin/bash
for i in $( ls );do
    echo a konyvtar elemei: $i
done
# kiíratjuk a képernyőre az aktuális könyvtár elemeit
```

## Egyszerű függvények.:

```
#!/bin/bash
function quit {
    exit
}
function hello {
    echo Hello!
}
hello
quit
echo foo
# Egyszerű fv. deklarálás, majd az értékek kiíratása ( a 10. sor sosem fut le!)
```

```
#!/bin/bash
function quit {
    exit
}
function e {
    echo $1
}
e Hello
e World
quit
echo foo
# Egyszerű fv. deklarálás, majd az értékek kiíratása ( a 11. sor sosem fut le!)
```

## Érdekes/hasznos bash scriptek -experimental- [top](#)

```
echo "Most `w -h | wc -l` felhasználó van bejelentkezve."
    # Kiíratjuk programunk segítségével, hogy hány felhasználó van bejelentkezve

test -d /home/user && echo "van" || echo "nincs"
    # Megvizsgáljuk, hogy van e /home/user könyvtár, két kimenetű elágazással (&&)
    # Ha van.: van
    # Ha nincs.: nincs
    # -d : van-e ilyen?

if [ -d /home/user ]; then echo "van"; else echo "nincs";
fi
    # Ugyanaz mint az előző példa, csak kicsit másként.
    # [-test parancs, ]-test zárás

if [ -s /etc/motd ]; then echo "nagyobb"; else echo "zerus";
fi
    # Megvizsgáljuk, hogy a /etc/motd nagyobb e nullánál, vagy zérus.
    # -s akkor ad IGAZ-at, ha a fájl létezik és nagyobb 0-nál.

for i in *.mp3; do mpg123 "$i"; done
    # Az aktuális könyvtárban az összes mp3 kiterjesztésű állományt, az mpg123
    # nevű programmal lejátszuk.

< /dev/urandom tr -dc A-Za-z0-9_ | head -c8
    # 8 karakteres random jelszó generátor

for i in $(ls); do cat $i >> 123.txt; done
    # aktuális könyvtárban található file-ok 123.txt file-ba összefűzése.

host -t mx mydomain.com
nslookup -q=mx mydomain.com
dig -t mx mydomain.com
    # MX record check-re használható parancsok

net rpc shutdown -I IP -U DOMAIN/user
    # shell-ből Windows klienst állíthatunk le távolról. (ha tudjuk a local admin jele)

net rpc shutdown -I IP -U DOMAIN/user -f -t 60
    # Előbbi parancs, időzítve 60sec-re.
```

```
echo tartalom | mailsend -to sendto@domain.dom -from from@domain.dom -starttls
-port 587 -auth -smtp SMTP.server.address -sub tárgy +cc +bc -v -user Username -pass P:
# 587-es porton (TLS) levélküldés. A fenti változók jelentései:
#
# tartalom           : az levél tartalma
# sendto@domain.dom  : címzett
# from@domain.dom    : feladó
# SMTP.server.address : milyen SMTP-n keresztül
# tárgy             : az üzenet tárgya
# user/pass          : SMTP user neve, jelszava.
```

## Egyéb programok [top](#)

```
abook
# Address book.

antiword
# Microsoft .DOC nézegető

amsn
# MSN kliens (GUI)

acpi
# ACPI eszközök információi

acpid
# acpi program démonja

audacious
# xmms utód, grafikus felületre mp3 lejátszó program

bitchx
# IRC kliens (TUI) -volt-

badblocks
# Lemez ellenőrző (Bad sectorokat keres)
-o file_nev : a hibás sectorokat a file_nev állományba menti

cal
# Szöveges naptár
-y : az aktuális év teljes naptárát mutatja meg

clamav
# víruskereső szoftver

clear
# Törli a képernyőt

dict
# Szótár program
opendict
dict-freedict-hun-eng : magyar-angol
dict-freedict-eng-hun : angol-magyar

elinks
# Konzolos böngésző.

false
# Nem csinál semmit, sikertelenül. (1-es visszatérési értékkel)
```

```
finch
    # Konzolos Instant messenger, pidgin, ..

firestarter
    # Tűzfal konfiguráló

fsck
    # Filesystem check.
    # Lemez-ellenőrző (helytelen leállítás után automatikusan lefut.)
    # A helyre nem hozott állományokat a lost+found könyvtárba teszi.

fwbuilder
    # Tűzfal beállító

gaim
    # MSN, irc, etc, kliens

gpg -c file
    # Fájl titkosítása (OpenPGP)

gpg file
    # Fájl kititkosítása

gparted
    # Lemez partícionáló

hdparm
    # Merevlemez információk
    -i /dev/hdx : Merevlemezünkről ad információt
    -Tt /dev/hdx : Sebességmérő eszköz

hddtemp /dev/hdx
    # Lemez hőmérséklet figyelő

irssi
    # IRC kliens program, konzolos felületre.

k3b
    # KDE alapú cd-dvd író program (GUI)

kopete
    # MSN kliens

lynx
    # Konzolos böngésző

mbmon
    # Motherboard monitoring

memtest86
    # Memória tesztelő alkalmazás

mp3blaster
    # Konzolos mp3 lejátszó, tracklist support-al.

mplayer
    # Média lejátszó

mutt
    # Konzolos levelező kliens

qemu
    # Processzor emulátor

setterm
```



```
# Konzol beállításai

tpconfig
# Touchpad konfiguráló

tpctl
# IBM ThinkPad hardver konfiguráló

true
# Nem csinál semmit, sikeresen. (0-ás visszatérési értékkel)

thosutils
# Toshiba laptop programok

reset
# Visszaállítja a konzol beállításait

vlc
# Média lejátszó

xchat
# IRC kliens (GUI)

xmms
# Audió lejátszó program

xmms-liveice
# Audió kimenetet küldi a southcast-nek

xdpyinfo
# X kiszolgáló információk

yes > file
# Csupa y-t ír a file-ba, elég gyorsan. :)

zgv
# Konzolos képnézegető
```

## RAID1 készítése élő rendszeren [top](#)

Linux software RAID 1 készítése, élő Debian/GNU Linux (Etch) rendszeren

Az alábbi leírást mindenki csak saját felelősségére használja!  
A folyamat elindítása előtt javasolt biztonsági mentést készíteni az adatokról,  
hiszen bármely elütés, figyelmetlenség következtében az adatok elveszhetnek!

RAID technológiáról bővebben.:

-----

[RAID Wikipedia](#)  
[Software RAID Linux alatt](#)  
[mdadm Wikipedia - eng](#)

/dev/sda - élő rendszerem lemeze  
/dev/sdb - a majdani RAID1 másik lába.

```
# Az én élő rendszerem partíciói.: (nálad ez valószínűleg egészen más)
# A lemez/partíció elnevezéseket saját rendszeredhez kell igazítani. Csak megfont
#
# Eszköz      csatolási pont    file rendszer
# -----
# /dev/sda1:  /boot          ext3
# /dev/sda2:                swap
# /dev/sda3:  /              ext3
```

```
# A procedura végén (az én esetemben) ilyen felállást szeretnék elérni.:
```

```
#
# Eszköz      csatolási pont    file rendszer    résztvevők
# -----
# /dev/md0:    /boot          ext3             (/dev/sda1 és /dev/sdb1-ből)
# /dev/md1:                swap             (/dev/sda2 és /dev/sdb2-ből)
# /dev/md2:    /              ext3             (/dev/sda3 és /dev/sdb3-ből)
```

```
apt-get install initramfs-tools mdadm
```

```
# Két szükséges csomag telepítése, mdadm kérdésénél érdemes ALL-t választani.
```

```
modprobe md
modprobe linear
modprobe multipath
modprobe raid0
modprobe raid1
modprobe raid5
modprobe raid10
# Töltsük be az összes modult.
```

```
sfdisk -d /dev/sda | sfdisk /dev/sdb
# /dev/sda lemezünk partíciós tábláját /dev/sdb lemezünkre másoljuk!
```

```
fdisk -l
# Ellenőrizzük le, hogy valóban sikerült-e.
```

```
fdisk /dev/sdb
# Készítsük el a file rendszereket a partíciókra.
# A gombok lenyomásának sorrendje a következő.:
m : manual
t : partíció rendszer id változtatása
1 : első partíció
L : listázás
fd : Linux Raid Autodetect filerendszer (ezt be kell gépelni!)
t : ..
2 : második partíció
fd : ..
t : ..
3 : harmadik partíció
fd : ..
w : kiírás.
```

```
# Ha már korábban próbálkoztunk RAID építésével a /dev/sdb lemezzel, az alábbi parancsok:
```

```
mdadm --zero-superblock /dev/sdb1
mdadm --zero-superblock /dev/sdb2
mdadm --zero-superblock /dev/sdb3
# Error-al elszállhat a dolog, de ez nem ad okot aggodalomra.
```

```
# Most pedig hozzuk létre a "fállábú" (degraded) tömbünket a /dev/sdb lemezre.
# Mivel a rendszer jelenleg a /dev/sda lemezzel fut, egyelőre azt nem tudjuk a tömbhöz adni.

mdadm --create /dev/md0 --level=1 --raid-disks=2 missing /dev/sdb1
mdadm --create /dev/md1 --level=1 --raid-disks=2 missing /dev/sdb2
mdadm --create /dev/md2 --level=1 --raid-disks=2 missing /dev/sdb3

cat /proc/mdstat
    # Ellenőrizzük le, hogy sikerült-e
    # [_U] jelzi hogy jelenleg fáállábú a tömbünk (sdb)

mkfs.ext3 /dev/md0
mkswap /dev/md1
mkfs.ext3 /dev/md2
    # Hozzuk létre a file rendszereket a tömbökre.

cp /etc/mdadm/mdadm.conf /etc/mdadm/mdadm.conf_orig
mdadm --examine --scan >> /etc/mdadm/mdadm.conf
    # Mentés készítése az mdadm.conf-ról
    # Az új "helyzet" elmentése az mdadm.conf-ba.

mkdir /mnt/md0
mkdir /mnt/md2
    # md0, md2 mappák létrehozása, ahova majd felmountoljuk a tömböket.

mount /dev/md0 /mnt/md0
mount /dev/md2 /mnt/md2
    # Mountoljunk!

mount
    # Az utolsó sorokban leellenőrizhetjük, hogy sikerült-e a csatolás

cp /etc/fstab /etc/fstab-orig
    # fstab-ról backup készítése

mcedit /etc/fstab
    # Az fstab-ban cseréljük ki a következőket.:
    /dev/sda1      /dev/md0 -ra
    /dev/sda2      /dev/md1 -ra
    /dev/sda3      /dev/md2 -ra
    # Majd mentünk rá a configra.

cp /etc/mtab /etc/mtab-orig
    # mtab-ról backup készítése

mcedit /etc/mtab
    # Az mtab-ban cseréljük ki a következőket.:
    /dev/sda1      /dev/md0 -ra
    /dev/sda3      /dev/md2 -ra
    # Majd mentünk rá a configra.
```

## Figyelem!

Amennyiben már **Grub2**-vel rendelkezünk, az alábbi linkről állítsuk be a bootloadert.: [Link](#)

Egyébiránt mehetünk tovább.

```
cp /boot/grub/menu.lst /boot/grub/menu.lst-orig
```

```

# menu.lst mentése.

mcedit /boot/grub/menu.lst
# Az alábbi változtatásokat állítsuk be.:

[...]
```

<i>default</i>	<i>0</i>
<i>fallback</i>	<i>1</i>

```

[...]
```

# A config végén készítsünk az első (title-savedefault közötti) részből egy új be-  
# valamint, a /dev/sda3 -at cseréljük /dev/md2 re  
# a root (hd0,0) -t (hd1,0)-ra. (Ha különbözőek a lemezeid elnevezése, itt is lé-  
# valahogy így.:

```

## ## End Default Options ##

title                Debian GNU/Linux, kernel 2.6.18-4-486 RAID (hd1)
root                 (hd1,0)
kernel               /vmlinuz-2.6.18-4-486 root=/dev/md2 ro
initrd               /initrd.img-2.6.18-4-486
savedefault
```

update-initramfs -u  
# Ramdisk frissítése.

cp -dpRx / /mnt/md2  
# Másoljuk a /dev/sda3 tartalmát /mnt/md2-re (Aminek egyelőre csak /dev/sdb3 rész:

cd /boot  
cp -dpRx . /mnt/md0  
# Másoljuk a /dev/sda1 tartalmát /mnt/md0-ra (Aminek egyelőre csak /dev/sdb1 rész:

```

grub
root (hd0,0)
setup (hd0)
root (hd1,0)
setup (hd1)
quit
```

# Telepítsük a grub-ot a második lemezünkre.  
# A következő indításkor, ha minden sikeres volt, már a második HDD-ről indul a  
# azaz neki állhatunk majd a /dev/sda lemez RAID-be állításához!

reboot  
# Restartoljunk.

df -h  
cat /proc/mdstat  
# Ellenőrizzük le ismét, hogy áll a tömbünk.

fdisk /dev/sda  
# hozzuk létre a Linux Raid Autodetect file rendszereket a partícióinkra az alábbi  
t 1 fd , t 2 fd, t 3 fd , w

```

mdadm --add /dev/md0 /dev/sda1
mdadm --add /dev/md1 /dev/sda2
mdadm --add /dev/md2 /dev/sda3
# Adjuk hozzá a féllábú (sdb-n álló) tömbünkhöz a /dev/sda partícióit.
```

```

cat /proc/mdstat
    # Ellenőrizzük le, hogy szinkronizálódnak-e a lemezeink.
    # Ilyesmit kellene látnunk.:

    Personalities : [raid1]
    md2 : active raid1 sda3[2] sdb3[1]
    4594496 blocks [2/1] [_U]
    [====>.....]    recovery = 29.7%

    # Várjuk meg míg befejezi az összes partíció szinkronizálását.

watch cat /proc/mdstat
    # 2 másodperces frissítéssel figyelemmel követhetjük a file tartalmát.

cp /etc/mdadm/mdadm.conf /etc/mdadm/mdadm.conf_orig
mdadm --examine --scan >> /etc/mdadm/mdadm.conf
    # Mentés készítése az mdadm.conf-ról
    # A legújabb "helyzet" elmentése az mdadm.conf-ba.

mcedit /boot/grub/menu.lst

    # Másoljuk le a korábban létrehozott kernel stanza-t és cseréljük
    # a (hd1,0) -t hd(0,0) -ra.
    # Ez a te esetedben ismét más lehet!

    ## ## End Default Options ##

    title                Debian GNU/Linux, kernel 2.6.18-4-486 RAID (hd0)
    root                  (hd0,0)
    kernel                /vmlinuz-2.6.18-4-486 root=/dev/md2 ro
    initrd                /initrd.img-2.6.18-4-486
    savedefault

    # valamint szerkesszük az alábbi sort eszerint, majd mentsük a file-t.:
    [...]
    # kopt=root=/dev/md2 ro
    [...]

update-initramfs -u
    # Ramdisk frissítése

reboot

    # ha mindent jól tettünk, a RAID1-ünk aktív, és használható.

```

Forrás .: [Howtoforge](http://howtoforge.com)

```

# Hamarosan:
#
# Elhasalt/elromlott HDD cseréje élő RAID1 alatt.

```

## Adatmentés sw. RAID1-et alkotó HDD-ről [top](#)

Előfordulhat, hogy Linux alatt létrehozott sw. RAID1 kötetet alkotó HDD-t egy másik géphez. Alábbi leírás ehhez a procedúrához nyújt segítséget. (a dev. elnevezések nálad egészen má

```
dmesg |grep Attached
# Mi az újonnan behelyezett disk neve?
# Pl.:
```

```
[ 2.628336] sd 1:0:0:0: [sdb] Attached SCSI disk
[ 2.629248] sd 0:0:0:0: [sda] Attached SCSI disk
[ 2.644424] sd 2:0:0:0: [sdc] Attached SCSI disk
```

```
fdisk -l /dev/sdc
# Milyen file rendszerrel rendelkezik a /dev/sdc?
# Valami ilyesmit kellene látnunk.:
```

```
Device Boot Start End Blocks Id System
/dev/sdc1 1 243202 1953513472 fd Linux raid autodetect
```

```
mdadm --examine /dev/sdc1
# újabb ellenőrzés, a HDD sdc1 partícióját illetően. A RAID partícióra vonatkozó
```

```
mkdir /mnt/raid1
# hozzunk létre egy csatolási pontot a RAID1-et alkotó HDD-nek.
```

```
mdadm -A -R /dev/md9 /dev/sdc1
# Hozzunk létre egy (még nem létező!) md9 nevű féllábú tömböt, melyet /dev/sdc1 a
# Amennyiben sikeres, ilyesmit kellene látnunk:
```

```
mdadm: /dev/md9 has been started with 1 drive (out of 2).
```

```
mount /dev/md9 /mnt/raid1
# csatoljuk a féllábú tömböt /mnt/raid1 mappába.
# a HDD-n található adatokat mostantól el tudjuk érni.
```

### Ha már nincs rá szükség.:

```
umount /dev/md9
# a tömb lecsatolása a file-rendszerből.
```

```
mdadm -S /dev/md9
# Az ideiglenes RAID tömb leállítás.
```

### Ha a RAID1-et alkotó HDD-t szeretnék visszahelyezni eredeti helyére a RAID tömbbe.:

(dev és md elnevezésekre kiemelt figyelmet fordítva)

```
mdadm -a /dev/md0 /dev/sdc1
# md0 tömbhöz /dev/sdc1 partíció hozzáadása
# Ha sikeres volt:
```

```
mdadm: re-added /dev/sdc1
```

```
cat /proc/mdstat
# tömb állapotának ellenőrzése
# Kimenet:
```

```
Personalities : [raid1]
md0 : active raid1 sdb1[1] sdc1[0]
```

```
1953512312 blocks super 1.2 [2/1] [U_]
[=====>.....] recovery = 36.9% (721315776/1953512312) finish=.
```

```
# A fenti kivastagított sor jelzi, hogy egyelőre még féllábú (degraded) a tömb.
# Ha a sync lefutott, [UU] lesz látható.
```

## Local repository készítése apt-mirror segítségével [top](#)

A cél egy saját repository készítése a lokális hálózaton figyelő gépek számára (http-n), forgalom csökkentésének érdekében. Magyarán a teljes tartalmat (2011.10. hóban: ~70Gb.) : a későbbiekben pedig csak frissítenünk kell, ezt a /etc/cron.d/apt-mirror kikommentelésével érhetjük el.

```
apt-get install apt-mirror apache2
# A majdani tároló-gépünkön kell telepítenünk.
# Mivel http-n fogjuk elérni a repo.-t, ezért egy web server alkalmazást is telepítenünk kell.
```

```
mcedit /etc/apt/mirror.list
# Állítsuk át az alábbi címekre.: (innen fogja a "server" összeszedni az adatot)

deb http://ftp.hu.debian.org/debian squeeze main contrib non-free
deb-src http://ftp.hu.debian.org/debian squeeze main contrib non-free
deb http://security.debian.org/debian squeeze/updates main contrib non-free
deb-src http://security.debian.org/debian squeeze/updates main contrib non-free

clean http://ftp.hu.debian.org/debian
clean http://security.debian.org
```

```
-----
Az alábbi parancsot érdemes éjszakára ütemezni.:
-----
```

```
su - apt-mirror -c apt-mirror
# Most pedig töltsük le a csomagokat, az alapértelmezésben beállított helyre (mirror)
# Ha kész a letöltés, a /var/spool/apt-mirror/mirror/ftp.hu.debian.org/debian alá
```

```
/bin/bash /var/spool/apt-mirror/var/clean.sh
# Fölösleg pucolása
```

```
ln -s /var/spool/apt-mirror/mirror/ftp.hu.debian.org/debian /var/www/debian
ln -s /var/spool/apt-mirror/mirror/security.debian.org/ /var/www/security
# Két symlinket kell létrehoznunk, hogy a kliensek be tudják frissíteni csomaglistáikat
```

Majd a kliens oldalon a sources.list-et kell editálni.:

```
deb http://SZERVERUNK_CIME/debian squeeze main contrib non-free
```

```
deb-src http://SZERVERUNK_CIME/debian squeeze main contrib non-free
deb http://SZERVERUNK_CIME/debian squeeze/updates main contrib non-free
deb-src http://SZERVERUNK_CIME/debian squeeze/updates main contrib non-free
```

```
apt-get update
apt-get upgrade
# Kliens oldali tesztelés.
```

## Távoli mappák felcsatolása SSHFS használatával [top](#)

Ezen leírás bemutatja távoli mappák biztonságos felcsatolásának módját a helyi fájlrendszer segítségével. A felcsatolás FUSE-on (Filesystem in Userspace) keresztül valósul meg.

```
apt-get install sshfs
# A kliensre szükséges telepíteni az SSHFS csomagot, ahova felcsatolásra kerül a
# A szerverre természetesen szükséges SSH szervert is telepíteni.

lsmod |grep fuse
# Kliens oldalon szükség lesz a fuse nevű kernel modulra is, szóval ellenőrizzük.

apt-get install fuse-utils
# Amennyiben a fenti modul hiányzik, a fuse-utils csomag telepítésével pótolni s:

useradd sshbackup
# A szerveren az sshbackup felhasználó létrehozása. Az ő nevében jelentkezünk be

passwd sshbackup
# Jelszó beállítása sshbackup felhasználónak.

gpasswd -a sshbackup dirgroup
# A távoli mappához szükséges a távoli sshbackup usernek hozzáférnie. Ezt szerveri
# Egy példa.:
# sshbackup user hozzáadása a dirgroup csoporthoz, melynek jogosultsága van (vagy)
# Ez természetesen kikerülhet, amennyiben a felcsatolandó mappa tulajdonosa már

sshfs -o idmap=user -p 22 sshbackup@192.168.1.1:/home/sshbackup /home/user/sshbackuplink

# sshbackup nevében felcsatoljuk a távoli szerver (192.168.1.1) /home/sshbackup r
# A csatolási pont a helyi fájlrendszerben /home/user/sshbackuplink mappa lesz.

mount
# Bizonyosodjunk meg a kliensen a mount-olás sikerességéről. Valami ilyesmit kell:

sshbackup@192.168.1.1:/home/sshbackup on /home/user/sshbackuplink type fuse.sshfs:

df --si
# Megbizonyosodás

sshbackup@192.168.1.1:/home/sshbackup
...

fusermount -u /home/user/sshbackuplink
# Leccsatolás
```

Forrás .: [Howtoforge](#)



## Hasznos linkek [top](#)

### HUN

[A nagy Linux topic - PH](#)

[Bash programozás](#)

[Debian - Felhasználói dokumentáció](#)

[Debian - Biztonság - PH! \(The DJ\)](#)

[DNS Howto](#)

[DNS Wiki](#)

[DRBD + Heartbeat](#)

[Hálózati programozás - Beej](#)

[Hungarian Unix Portal](#)

[IPv6 topic - HUP](#)

[Linux alkalmazások - prog.hu](#)

[Samba beállítása - hogyan.org](#)

[Shell-programozás](#)

[Squid - prog.hu](#)

[Szabilinux](#)

[TLDP - Magyar Linux Dokumentációs Project](#)

[Tűzfal tervezése, kezdőknek - budacsik - hogyan.org](#)

### ENG

[Advanced Bash-Scripting Guide](#)

[Bash programming](#)

[Debian Reference - debian.org](#)

[Explain Shell!](#)

[Swiss Army Knife Internet Tool](#)

© 2008-2016 Copyright letix

---

## Ha a Föld hozzánk szólna, biztos megkérne minket arra...

... hogy ne terheljük Őt feleslegesen. A háztartások elektromos energia fogyasztásának átlagosan 10 százaléka készenléti állapotban (standby mode) hagyott berendezésektől származik ([Energy Saving Electronics](#)), tehát feleslegesen megy el energia, feleslegesen szennyezzük a környezetet és feleslegesen adunk ki pénzt.

Kapcsold ki teljesen készülékeidet, így Te is tehetsz egy keveset a Földért - és pénzt is spórolhatsz a villanyszámlán...