

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

Utolsó frissítés.: 2017-10-08
[Changelog](#)

[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](#)
 Saját leírások:
[mdadm - RAID alapok](#)
[RAID1 készítése élő rendszeren](#)
[Adatmentés sw. RAID1-et alkotó HDD-ről](#)
[Local repository készítése apt-mirror segítségével](#)
[Távoli mappák felcsatolása SSHFS használatával](#)
[Távoli FTP felcsatolása lokális fájlrendszerbe](#)
[iSCSI Target létrehozása, Initiator beállítása \(IET\)](#)
[Rsync használata](#)
[iptables alapok](#)
[Fájlátvitel SCP segítségével](#)
[Hasznos linkek](#)

Az első lépések [top](#)

Egy Debian GNU/Linux alaprendszer telepítésének menete részletesen, képekkel illusztrálva az alábbi linkeken található.:
[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
# A lehetőségek közül a low-t "alacsony" választva a későbbi programok repository-ból történő
# telepítéskor 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-ban vannak megadva.)
```

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

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

man -L en parancs
# parancs manual oldalainak angol nyelvű megjelenítése

apropos szó
# Minden olyan parancsot megad, mely manual oldalaiiban 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 telepítened kell.

Könyvtárszerkezet [top](#)

```
/      # A hierarchikus könyvtárfa kiindulópontja (gyöker 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 csatolható modulokat, meghajtóprogramokat
/dev    # A rendszerhez csatlakozott, csatolható 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 information
        Érdemes tüzetesebben átnézni, hiszen érdekes dolgokat találhatunk itt.
        pl.: /proc/cpuinfo fájl kiírásával információt kaphatsz processzorodról.
/root   # A rendszer gazdájának könyvtára.
/tmp    # Ideiglenes adatok tárolására használt könyvtár. -temp
/usr    # Alkalmazások, rendszerezszközők tömkelege, a legforgalmasabb könyvtár. (pl X Window)
/var    # Változó adatokat tartalmazó állományok könyvtára. /pl.: nyomtatási munkák, levelek, etc)
        /var/log : napló fájlok, különös jelentőséggel bírnak a rendszer biztonságának szempontjából
```

Állománykezelés [top](#)

```
pwd      # Az éppen aktuális könyvtár munkakönyvtár kiírása

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

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

rmdir     # Könyvtár törlés

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

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 bármelyik
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

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

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 vizsgálja)
--mime file: A file karakterkódolását mutatja meg.

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

cat       # Fájlok tartalmát írja ki.
> file : várja a bemenetet, amely a "file" tartalma lesz. Ctrl + D kombinációval menthető.
-n filel: beszámozza a filel 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 < bemenet.txt > kimenet.txt
# a cat beolvassa a bemenet.txt tartalmát és a kimenet.txt-be irányítja.

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

```

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 user home-ja. pl /home/letix

touch file
# létrehoz egy file nevű üres állományt
-t MMDDHHmm fájl      : A fájl időbélyegeinek dátumát változtatja meg. MM-Hónap DD-Nap HH-Óra mm-Perc
-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.

find dir/ -name "*" -exec touch {} \;
# dir nevű mappa összes állományának módosítási dátumát megváltoztatja az aktuális dátumra.

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, particiónként
-H              : Olvashatóbb formátumban írja ki a méreteket (MByte, GByte, stb.)
--si           : A -H helyett már ezt a kapcsolót ajánlatos használni.

find
# Keresés

find / -name "*.jpg" -print : az összes kép keresése a gyökérben, majd az eredmény kiírása
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 együtt.

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

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 kiterjesztésben.)

find . -name "*.txt" -print | xargs grep "tcp80"
# a gyökérben olyan txt állományokat keres, melyek tartalmában szerepel a "tcp80" kifejezés

chmod
# Linuxon a fájlokra, könyvtárakra vonatkozó jogok állíthatók be ezen paranccsal
-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+rwx 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 file-ok, mappák.
# 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 kapjuk meg a jogokat.

chown
# Fájlok, könyvtárak tulajdonosának (létrehozójának) változtatása
-R              : Rekurzívan változtatja meg a tulajdonos(oka)t
-f             : Nem küld vissza hibaüzenetet a rendszer, ha valami nem sikerült
--no-dereference : Szimbólikus linkeknél a link jogosultságainak beállítása
--dereference   : Szimbólikus linkeknél a file (amire a link mutat) jogok változtathatók meg.

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 csoportjuk

```

```

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 fájlok utolsó módosításának dátumát. (rendszergyorsító hatás)
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

cmp file1 file2
# Összehasonlítja a file1 és file2 fájlok 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
-d: -f1       : Kettősponttal elválasztott sorokban az első helyen lévő adatot adja vissza.

        echo ELSO:MASODIK:HARMADIK | cut -d: -f1
        ELSO

colrm
# Fájlból oszlopok távolíthatóak el
1 5 file: adott bemeneti állomány első oszloptól az ötödikig töröl minden sorból

        Letix@microserver:~$ cat colrmtest
        1234567890
        2345678910
        3456789101
        Letix@microserver:~$ cat colrmtest | colrm 1 5
        67890
        78910
        89101

diff -u file1 file2 > eredmény
# Összehasonlítja a fájlok 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 egyezéseket csak a bal oszlopban tünteti fel.

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 újsor karakterre cseréli,
# 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, azaz üres sorokat töröl.

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

grep
# Szövegrészleteket keres fájlokban, valamint a kimenetben. A kapcsolók után kell megadni a file-t.
-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 kapcsoló hibát dob.)
-x      : csak teljes sorokkal való illeszkedést vizsgál.
B.r     : azokat a sorokat adja meg, melyekben a "B" és az "r" között bármilyen karakter szerepel.
[ha]    : a kimenetben találhatóak meg azok a találatok, melyekben szerepel "h" vagy "a" betű.
1[5678]: azon sorok megadása, melyben szerepel 15,16,17,18
1[5-8]  : azon sorok megadása, melyben szerepel 15,16,17,18
[^sajt]: minden sor megtalálható a kimenetben, kivéve amelyben szerepel a "sajt" kifejezés.
^$      : Azokat a sorokat adja meg, melyek üresek.
^h      : A sor elején található kis "h" betűre illeszkedik.
A[-]4   : olyan sorokat ad vissza, melyben A-4 karaktersor szerepel

-i -E '(ertek1|ertek2)' file : érték1 vagy érték2 -re keresése a file-ban, kis és nagybetű különbség nélkül.
-i -E 'ertek1|ertek2' file   : érték1 vagy érték2 -re keresése a file-ban, kis és nagybetű különbség nélkül.
-i -e érték1 -e érték2 file  : érték1 vagy érték2 -re keresése a file-ban, kis és nagybetű különbség nélkül.

-A1 B1 ertek file : a file-ban az ertek-et tartalmazó sorokat adja meg úgy, hogy az egyel előtte és
                  : utána levő sorokat is kiírja

-r minta /etc      : azon fájlok elérését és illeszkedő sorait adja meg a /etc-n belül, melyben szerepel
                  : a minta.

grep '^[^#;]' /etc/samba/smb.conf
# smb.conf tartalmának kiírása úgy, hogy a # és ; jelekkel kezdődő sorokat nem írja ki.

# szóközt grep szó "szó" -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 is. Az előjel mindig pozitív.)
-c4 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 -n1
vagy
head -n3 fájl | tail -n1
# Egy fájl harmadik sorát így lehet kiíratni

paste
# adatoszlopok vagy adatfájlok oszlopainak összemácsolása egymás mellé tabulátorral
# file1 tartalma: Peti
#                  Jenő
#                  Anna
# file2 tartalma: 1980
#                  1950
#                  2007
# file3 tartalma: December
#                  Február
#                  Július

paste file[123]
# file1,2,3 állományok oszlopainak összemácsolása
# Kimenet:

      Peti    1980    December
      Jenő    1950    Február
      Anna    2007    Július

paste -d ':' file[123]
# ugyanaz mint a fenti, de tabulátor helyett ":" az elválasztó.

sed
# Stream editor, folyamatszerkesztő. A bemenetet a kimenetre másolja miközben megszerkeszti.
echo "hablalda" | sed 's/a/K/g' : kimenete.: hKb1KbdK, azaz "a" betűket "K"-ra cseréli
'/ *#/d; /^ *$/d' file : a file-ből kiszűrjük a kommenteket, üres sorokat.

cat file | sed 's:/mnt/test:\\\\server\\share:g' > out
# file ban található /mnt/test elérési útvonalakat cseréli \\server\share -re az out file-ba
# irányítva.

sort
# Sorba rendezés
-b fájl > kimenet : ABC sorrendbe rendezi a fájlt, 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
-k 2 fájl : 2 oszlopos file-ban a második oszlop alapján rendezi sorba
-k 2,2 fájl : a fájl 2. oszlopának második karaktere alapján rendez
-k 3,3,3,5 fájl : a fájl 3. oszlopának 3,4 és 5. karaktere alapján rendez

rev (reverse lines)
# adott állományban a karakterek sorrendjének megfordítása

cat file4
ABCDEF GH
12345678
rev file4
HGFEDCBA
87654321

nl (number lines of files)

# file-ok soronkénti beszámozása.
-n ln : sorszámozás balra zárt, nullák nélkül
-n rn : sorszámozás jobbra zárt, nullák nélkül
-n rz : sorszámozás jobbra zárt, nullázva
-w4 : rz kapcsolóval együtt a nullák számát lehet megadni
-s: : separator ":"

nl -n rz -w4 -s: file1
0001:Peti
0002:Jeno
0003:Anna

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

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

mc
# Midnight Commander fájlkezelő

```

```

mcedit
# Az mc szövegszerkesztője

update-alternatives --config editor
# Alapértelmezett szövegszerkesztőnket a fenti paranccsal tudjuk módosítani (pl mcedit-re)

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

mkisofs -J -V "Label" adat/ | sudo crecord dev=0,0,0 speed=32 -data -v -eject driveropts=burnfree -
# adat mappa kiírása 32x-es sebességgel, Label nevű, verbose, az írás végén CD kiadással.
cdrecord -scanbus
# dev érték lekérdezése

install
# Fájlok másolása, attribútumok módosítása

less
# Szűrőprogram fájlok adatainak megjelenítéséhez.

lsof
# 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 kiírata

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) célfájlal rendelkező állományokat.
-f : force, a célfájl törlése, kérdés nélkül

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

ln -s
# Szimbolikus link létrehozása
-s fájl1 fájl2 : fájl1 ről készít fájl2 nevű soft-link-et.

readlink
# szimbolikus link tartalmának kiírata

Letix@debian-ssd:~$ ln -s /home/Letix/raidtest/testLink Link
Letix@debian-ssd:~$ readlink Link
/home/Letix/raidtest/testLink

mkdir -t ext2 /dev/ram0 4096
mount /dev/ram0 ramdisk/
# 4Mbyte os ramdisk-et hozunk létre, és felcsatoljuk.

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

false
# nem csinál semmit, sikertelenül. Visszatérési értéke 1

true
# nem csinál semmit, sikeresen. Visszatérési értéke 0

Letix@debian-ssd:~$ false
Letix@debian-ssd:~$ echo $?
1
Letix@debian-ssd:~$ true
Letix@debian-ssd:~$ echo $?
0

```

updatedb

Adott gépről, vagy felcsatolt eszközökön található adatokról adatbázis készítése, melyben az mlocate/locate parancsokkal könnyen és gyorsan lehet keresni.

Adatbázisa : /var/lib/mlocate/mlocate.db
 Paraméterei : /etc/updatedb.conf

pl.:
 PRUNENAMES : bizonyos kiterjesztésű állományokat ki lehet hagyni a vizsgálatból. pl: .svn
 PRUNEPATHS : megadható, mely mappákban ne keressen. pl: /tmp
 PRUNEFS : definiálható, mely fájlrendszereket hagyjon figyelmen kívül. pl: CIFS

```

locate fájl
# általánosan létrehozott updatedb adatbázisában (/var/lib/mlocate/mlocate.db) keres fájl-t

```

```
updatedb -l 0 -o dbdir/dbname.db -U /mnt/server
# /mnt/server felcsatolt fájlrendszer tartalmáról adatbázis készítése dbdir/dbname.db állományba
# Ezen esetben lokális meghajtókról nem készül bejegyzés, csak a server-ről!

locate -d dbdir/dbname.db macilaci
# a fent létrehozott dbname.db-ben olyan állományokat keres, melyeknek a nevében szerepel macilaci

locate -d dbdir/dbname.db --statistics
# dbname.db nevű adatbázis statisztikáinak megjelenítése

Database /var/lib/mlocate/mlocate.db:
6146 directories
57770 files
3564748 bytes in file names
1612746 bytes used to store database

locate / >/tmp/locatedb.txt
# a teljes fájlrendszerről készít listát a megadott text állományba.
```

Fstab állomány

Az /etc/fstab fájl definiálja, hogy a diszkek partícióit, más blokkos eszközöket, vagy távoli fájlrendszereket hogyan csatolunk a fájlrendszerbe, Pl a mount parancs is ezen beállító állományt használja. Ha az fstab helyesen van kitöltve, a mount-al elég csak az eszközt megadni, pl mount /dev/sdb7

Felépítése

file system	mount point	type	options	dump	pass
/dev/sda1	/	ext4	defaults,noatime	0	1
/dev/sda2	none	swap	defaults	0	0
/dev/sda3	/home	ext4	defaults,noatime	0	2

file system : csatolandó partíció vagy tároló eszköz
 mount point : csatolási pont
 type : fájlrendszer típusa (pl: ext2-3-4, reiserfs, xfs, jfs, smbfs, iso9660, vfat, ntfs, swap)
 options : csatolási opciók, melyek csak az adott csatolásra érvényesek
 auto : automatikus csatolás indításkor. inverze: noauto
 exec : bináris futtatható fájlok végrehajtásának engedélyezése. inverze: noexec
 ro : read-only, azaz csak olvasható lesz a felcsatolt fs.
 rw : írás és olvasás engedélyezett
 user : bármely felhasználó felcsatolhatja az adott eszközt. inv: nouser, azaz csak root.
 users : a users csoport tagjai felcsatolhatják az eszközt.
 owner : csak az eszköz tulajdonosa csatolhat
 sync : I/O műveletek szinkronban zajlanak le. inv: async
 dev : speciális blokkeszközök használata. inv: nodev
 noatime, nodiratime : file-ok, mappák elérési, módosítási adatait nem menti.
 discard : TRIM parancsok futtatása az adott blokkeszközön. SSD lemezeknél ajánlatos.
 nofail : amennyiben az eszköz nem elérhető, úgy nem vesz róla tudomást a rendszer. (pl külső HDD)
 defaults: alaeértelmezett beállítás. Tartalma: rw, suid, dev, exec, auto, nouser, async
 dump : Értéke: 0 vagy 1. Készüljön-e az fs-ről biztonsági másolat vagy sem. Általában 0 -> nem.
 pass : Értéke: 0,1,2. Lemezellenőrzés (fsck) prioritásának beállítása. Rootfs 1, minden mást érdemes 2-re.

Lemezek azonosítása

```
-Kernel leíró alapján : pl /dev/sda
-UUID alapján : ls /dev/disk/by-uuid vagy lsblk -f
-label alapján : ls /dev/disk/by-label
```

```
lsblk
# List block devices - diszkek és a rajtuk lévő partíciók alapvető információi (name, fs, size, type, mount point...)
-f : UUID információkat is megadja
/dev/sda : sda eszköz partíciói
-no UUID /dev/sda : az eszköz partícióinak felsorolása UUID alapján

    pl:NAME      MAJ:MIN RM   SIZE RO TYPE MOUNTPOINT
       fd0       2:0    1    4K 0 disk
       sda       8:0    0   16G 0 disk
       \sda1     8:1    0  487M 0 part [SWAP]
       \sda2     8:2    0   7.5G 0 part /
```

Szóköz kezelése fstab-ban

Szóköz használata esetén a „\040” karaktert szükséges az elérés útvonalba szűrni a szóköz helyére.
 External Storage esetén az fstab-ba írandó:

```
/mnt/External\040Storage
```

```
lsblk -io KNAME,TYPE,SIZE,MODEL,VENDOR
# beépített eszközzel a HDD-k adatainak lekérdezése
```

```
udevadm info --query=all --name=/dev/sda
# ugyanez.
```

Forrás.: Archlinux.wiki

mount

```
# Szimplán kiadva a rendszerbe felcsatolt eszközök listáját adja meg. (/etc/mtab file is)

/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 ntfs-3g /dev/sdb1 /mnt/ntfs
# ntfs-3g csomag telepítését követően a /dev/sdb1 (NTFS fájlrendszerű) eszköz felcsatolása /mnt/ntfs-be.

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

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

-t cifs //192.168.1.1/D$ /mnt/cifs -o username=admin,password=password
# CIFS fájlrendszer felcsatolása a 192.168.1.1 D$ adminisztratív megosztásából a megadott helyre a megadott névben.
#
# FONTOS:
# Windows 7 esetében csak azt követően működik a felcsatolás, hogy az alábbi registry kulcs fel lett véve!

Hely      : HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\Policies\System
Típus     : Key DWORD (32-bit)
Kulcs     : LocalAccountTokenFilterPolicy
Érték     : 1

e2label /dev/sda1
# sda1 partíció LABEL értékének kiírása, amennyiben van.

e2label /dev/sda1 NEWLABEL
# sda1 partícióra új label beállítása

umount /dev/eszköz
# Lecsatlakoztatás

eject
# leválasztja a hordozható eszközt
sda      : első SCSI (sata) eszköz leválasztása

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

tar
# Ki-be tömörítő

Listázás
-----
-tvf file.tar      : megmutatja a mentes.tar tartalmát
-ztvf file.tar.gz  : megmutatja a mentes.tar.gz tartalmát
-jtvf file.tar.bz2 : megmutatja a mentes.tar.bz2 tartalmát
-tvf mentes.tar    : részletes listázás, jogok, tulajdonos etc.

Betömörítés
-----
-cvf file.tar /eleresi/ut      : az elérési útvonal alatt levő adat tar-ba tömörítése.
-zcvf file.tar.gz /eleresi/ut/amit/mentunk : az elérési útvonal alatt levő adat gzip-be tömörítése.
-jcvf file.tar.bz2 /eleresi/ut/amit/mentunk : az elérési útvonal alatt levő adat bz2-be tömörítése.

Kitömörítés
-----
-xvf file.tar      : Kitömöríti a tar-t és egy /file mappába teszi
-zxvf file.tar.gz : Kitömöríti a gz-t, majd a tar-t, és egy /file könyvtárba teszi
-jxvf file.bz2     : bz2-t tömöríti ki

Exclude példa
-----
-zcvf file.tar.gz /home/mit/ --exclude={„/home/MIT_NE”,„/home/MIT_NE2”}
# /home/mit mappa tartalmát betömörítjük gzip-be, kivéve /home/MIT_NE, /home/MIT_NE2-t.

tar jcvf - sourcedir/ | ssh user@IPADDRESS "cat > /home/user/destdir/test.tar.bz2"
# tar over ssh, azaz lokális állományok betömörítése és ssh-n történő átmozgatása távoli gépre

# jcvf      : verbose kimenettel tar.bz2-t készít
# sourcedir : ezen mappa tartalmának betömörítése
# ssh user1@.. : user nevében bejelentkezik az IPADDRESS című gépre
# cat > ..   : user felhasználó adott mappájába menti a tar.bz2 állományt.

mccrypt file.txt
# mdecrypt-el történő jelszóval védett file titkosítás. (mccrypt csomag telepítése szükséges)
# a kimeneti állomány file.txt.nc, tartalma visszafejtés nélkül olvashatatlan.

mdmccrypt file.txt.nc
# fentebb titkosított file.txt.nc állomány visszafejtése
# érdemes az eredeti file.txt-vel nem azonos mappában kiadni visszafejtési parancsot, ugyanis
# felülírhatja az eredetit.
```



```

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

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

watch cat file
    # A file változásait figyeli, 2 másodperces időközönként frissítve.

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

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

```

Kvóták beállítása Linux alatt

APT-vel szükséges telepíteni 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
    # kvótabejegyzések listázása felhasználónként.

edquota -u user
edquota -g group
    # Felhasználó, csoport kvóták beállítása
    # Alapértelmezett szerkesztő átmeneti állományaként állítható 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 szükséges az options-be megadni: usrquota, grpquota kapcsolókat.

```

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)

adduser --home /dev/null --shell /bin/false --no-create-home --uid 65533 --disabled-password guest
    # home, shell és jelszó nélküli guest account létrehozása (pl samba guest userhez..)

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 felhasználó
-m -k /home/letezo uj : egy létező felhasználó mintájára építi fel az új user home-ját.
                        Minden fájl és könyvtár is belekerül a fiókba, + a beállítások is.
```

```
usermod
# Felhasználói fiókok módosítása
-u 1110 user        : a felhasználó azonosítóját változtatja 1110-re (UID)
-g csoport user     : a felhasználó csoportját változtatja
-g user1 group1     : user1 elsődleges csoportja legyen group1
-G grp1,grp2 user   : user felhasználót több (másodlagos) csoporthoz is rendeli
-L user            : lock-olja a felhasználó hozzáférését (nem tud belépni)
-U user            : unlockolja a hozzáférést
-s /bin/false user  : adott user shell tulajdonságának módosítása /bin/false-ra.
-m -d /home/def user1 : user1 home mappájának módosítása def-re.
```

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

```
deluser
# Felhasználó törlése
```

```
groupadd
# Csoport létrehozása
-g szám csoportnév : Egyénileg választott csoportazonosítóval (GUID)-el hozunk létre csoportot
# /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 ki. (default: 500)
```

```
cat /home/username/.bash_history
# username nevű felhasználó saját history-jának kiírítása, 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 kimenetét
```

```
unset export HISTTIMEFORMAT
# visszaállítás az eredeti állapotra/kimenetre
```

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

```
export HISTSIZE=0
# history kikapcsolása
```

```
Perzisztens 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 alapértelmezettként.
# A második sor is kötelező.

passwd
# A bejelentkezett felhasználó jelszavának módosítása
# Userként kiadva saját jelszavunkat változtatjuk meg. A régit tudnunk kell hozzá.
user : user nevű felhasználó jelszavát változtatja meg. root-ként, nincs szükség a régi jelszó megadására.
-l user : lock user account. user jelszavát ideiglenesen megváltoztatja egy random karaktersorra.
-u user : unlock user account.: Az eredeti jelszót visszaállítja user számára.
-e user : expire. A user jelszavának azonnali lejáratát. Következő belépésnél új jelszót kell adnia.
-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, change,
# UID, GID, egyéb hasznos infók kiírása

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

cfdisk
# Lemezkezelő, particionáló

fsck
# lemezzellenőrző

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 kapcsolhatjuk be.:
    -s on /dev/eszköz

-A /dev/sda : Attributum táblák megjelenítése. Legtöbben ezekre az értékekre vagyunk kíváncsiak.
-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.

tty
# Bejelentkezett felhasználó terminál információi

stty
# A terminál beállításainak kiírása
-a : verbose mód

uname # Rendszerinformációk megadása.
-a : minden információ kiírása. Linux, kernel verzió, node name (hosztnév), stb.
-r : csak a kernel verziójának megadása, pl.: 3.2.0-4-686-pae

lsb_release -a
# Adott distribúció információinak kiírása

cat /etc/debian_version
# Debian verziójának kiírása

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

mesg
# A write üzenetküldő aktuális beállítását kérdezhetjük le. (engedélyezve vagy tiltva van)

mesg n
# A write üzenetküldő tiltása

mesg y
# A write üzenetküldő engedélyezése

mkswap
# Swap terület létrehozása

swapon
# Swap bekapcsolása

swapoff
# Swap kikapcsolása

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

```

Folyamatok [top](#)

```

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

Ctrl+Z
# Előtérben futó process (pl mcedit) háttérbe helyezése.

Ctrl+C
# Félresikerült/megakadt process bezárására használható billentyű kombináció

jobs
# Háttérben futó programok kiírása
# Az itt megkapott értékek használhatóak az fg, bg parancsoknál.

command1 && command2
# command1 sikeres futását követően command2 is lefut.
# Amennyiben command1 visszatérési értéke nem 0 (tehát sikertelen), úgy az utána
# soron következő parancs(ok) nem fut(nak) le

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

top
# Futó folyamatok kiírása

h      : help
u      : felhasználókra szűrhető lista
z,b    : a táblázat színezése, illetve kivastagítása
Z,B    : a táblázat színeinek, kivastagíthatóságának módosítása
l,t,m  : a lista fejléce, terhelési adatok, egyebek jeleníthetők meg/kapcsolhatóak ki

pidof folyamat
# A folyamat azonosítóját adja meg (PID - process ID)

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)
-tt      : tty-es terminál kilistázása
f        : erdő szerű megjelenítés
l        : kiírja a folyamatok PID-jét és PPID-jét is. (parent's process identifier)
ww       : a programok parancssori kapcsolóit írja ki.
fax      : fa-szerű struktúrában jeleníti meg a folyamatlistát

-o user,pid,ppid,start_time,uid,%cpu,%mem,cmd
# felhasználónév, processid, parent processid, kezdési idő, CPU, MEM, kapcsolók
# formában adja meg a folyamatokat.

USER      PID  PPID  START   UID %CPU %MEM CMD
Letix    19284 19283 10:39   1000  0.0  0.1 -bash
Letix    22095 19284 14:33   1000  0.0  0.0 ps -o user,pid,ppid,start_time,uid,%cpu,%mem,cmd

pstree
# Folyamat struktúra fa-szerű ábrázolása.

pgrep
# Folyamatlista szűrése adott minta alapján
-u user      : megadja user nevében futtatott folyamatok PID-jét
-u user screen : megadja user nevében futtatott screen folyamat PID-jét
-lu 0        : megadja a 0 UID-el rendelkező (root) felhasználó folyamatainak nevét és PID-jét

pkill
-9 -u user screen : user nevében futtatott screen folyamat erőltetett leállítása
                  hasznos lehet, ha többen futtatnak screen-t

kill "pid"
# Folyamat leállítása PID szerint
-1      : jelentése SIGHUP. A folyamat bezárása, config fájl beolvasása, folyamat újraindítása
-9      : jelentése SIGKILL, folyamat erőltetett bezárása. (Csak végszükség esetén)
-15     : jelentése SIGTERM, szabályos programleállítás
-1      : teljes signal táblázat kiírása. Ezek a jelek küldhetőek folyamatoknak
-s "signal" : -l által megadott táblázatban található signal-ok küldhetőek adott processnek.

Letix@microserver:~$ yes > /dev/null &
[1] 22268
Letix@microserver:~$ kill -s 6 22268
[1]+  Félbeszakítva      yes > /dev/null

killall command
# az összes felhasználó által futtatott "command" nevű folyamat leállítása

nice
# Prioritása lekérdezése, beállítása
-n --20 program : a rendszer a legtöbbet ezzel a programmal fog foglalkozni. (+19-től -20 ig)

```

```

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

init
# Futtatási szint beállítása (run levels)
# Az egész rendszer állapotjelzője, ami meghatározza, mely szolgáltatások működnek, vagy épp indulnak el
# rendszerindításkor. A futási szintek számokkal kerülnek azonosításra.
# Definiálhatóak különböző szintek annak függvényében, hogy milyen jellegű munkára lesz használva a gép.
# Például ha X-el, bluetooth-al és egyéb erőforrás igényes alkalmazásokkal lesz használva, úgy létrehozható
# egy 5-6-os init szint. Ezt a szintet az inittab-ban szükséges beállítani alapértelmezettként induláshoz,
# de akár a rendszer futása közben is módosítható az aktuális szint. -> Akár több szolgáltatás is indítható vagy
# leállítható egy paranccsal.
#
# Init szintek
# -----

0   : kikapcsolás
1   : single-user mód (speciális rendszeradminisztrációs funkciókra)
2-5 : multi-user mód, (normál működés)
6   : reboot

# Új szolgáltatás hozzáadása
# -----
# Program bemásolása /etc/init.d-be, majd erről 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.
#
# /etc/inittab -ban állítható be a gép alapértelmezett indulási init szintje.

    The default runlevel.
    id:2:initdefault:

# 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.) típusú.
# 3 N jelenti, hogy 3-ason voltunk és vagyunk, 3 2 pedig hogy 3-ason voltunk 2-esen vagyunk

fuser
# Folyamatok azonosítása nyitott file-ok vagy process-ek alapján (érdemes root-ként futtatni)
-v .           : aktuális felhasználó folyamatai
-v -n tcp 80    : mely folyamat használja a TCP/80-at?
-vm /mnt/test  : megadja azon folyamatot, mely fogja /mnt/test mappát.
-vmk /mnt/test : kilövi azon folyamatot, mely fogja /mnt/test-et
-v /var/run/mysqld/mysqld.sock : mely folyamat használja a mysqld.sock socketet?

    root@microserver:/home/Letix# fuser -v -n tcp 10000
                                FELHASZNÁLÓ  PID  HOZZÁFÉRÉS  PARANCS
    10000/tcp:                  root  1764  F....  miniserv.pl
    root@microserver:/home/Letix# mlocate miniserv.pl
    /usr/share/webmin/miniserv.pl

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

-h `date --date "now + 30 seconds" "+%H:%M"` : aktuális dátumhoz képest 30 másodperccel későbbi leállítás kezdeményezése

```

Időzített parancsfeldolgozás [top](#)

```

at
# Megadott időpontban futtathatunk programokat
-f todo 23:59 : előre megírt parancsainkat (todo fájlban) 23:59-kor lefuttatja az at.
# Idő formátumok
# 13.13 01.01.02 : 13 óra 13 perc, 2002, január 1
# 2pm tomorrow : honlap 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 feladatütemezője
# /etc/cron.d ; /etc/cron.daily ; /etc/cron.weekly
crontab -l : kilistázza a belépett user beállított ütemezéseit
crontab -e : Editáljuk a belépett user ütemezéseit

```

Cron job 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	<i>ping 192.168.1.1</i>
---	---	----	---	-----	-------------------------

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

30	10	*	*	*	<i>ping 192.168.1.1</i>
----	----	---	---	---	-------------------------

Minden nap, 10:30 kor megpingeli a címet.

00	1-8,12-17	*	*	*	<i>ping 192.168.1.1</i>
----	-----------	---	---	---	-------------------------

Minden nap, 1:00-től 8:00-ig és 12:00-től 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 éjjélkor.
@daily           : "0 0 * * *" : Naponta egyszer fut le, éjjélkor.
@midnight        : "0 0 * * *" : éjjélkor, ekv. az előzővel.
@hourly         : "0 * * * *" : Minden egész órakor fut le.
```

Cron job hibacsatorna /dev/null-ba irányítása

```
0 1 5 10 * /path/script.sh >/dev/null 2>&1
```

```
# amennyiben szükséges, hogy az adott job hibacsatornája ne árássza el a /var/log-ot, úgy a fenti
# kivastagított eljárást szükséges alkalmazni
```

```
date
# Dátum kiírása 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.
110411532011 : Beállítja a dátumot November 04., 11:53-ra, 2011-ben. (Hónap Nap Ora Perc Ev)

-d '+3 hour' +%Y.%m.%d "%H:%M.%S" : az aktuális dátumhoz képest 3 órával későbbi dátumot adja meg 2017.01.05 14:07.12 formátumban.
```

```
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) indítható
Ctrl+a+d       : bill. kombinációval tehetjő háttérbe.
screen -ls      : screen-ek listázása
screen -d -m -S name : elindítja a háttérben a screen-t "name" névvel
screen -x name  : name nevű terminál hozható előtérbe
screen -R PID   : paranccsal lehet előtérbe hozni. (PID - processID, ps aux-al megnézhető)

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

Ctrl+a+d       : bill. kombinációval tehető háttérbe.
Ctrl+a+?       : előhozható a legfontosabb bill. kombinációkat
Ctrl+a+c       : új ablak létrehozása
Ctrl+a+p VAGY n : előző vagy következő ablakra ugrás. (previous, next)
Ctrl+a :kill    : aktuális screen lelvérese
Ctrl-a :acladd USER : USER nevű felhasználó engedélyezése becsatlakozásra (lásd MULTIUSER MODE)
```

```
# MULTIUSER MODE
# -----
#
# A screen többfelhasználós üzemmódja
# Amennyiben kontrollált körülmények között szükséges beengedni valakit a gépre közös shellt
# használva, úgy az alábbi lépéseket kell alkalmazni

screen
# screen indítása a kiszemelt többfelhasználós gépen

Ctrl+a :multiuser on
# Ezt begépelve aktiválható az adott session-ön a többfelhasználós mód.
# Másik oldalnak szükséges ismernie az adott user nevét/jelszavát, illetve SSH-n be kell tudnia
# csatlakozni a gépre. 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 sikeres volt)
```

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 yaird
# tar zxvf linux-2.6.20.3.tar.bz2 (érdekes 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, az aptitude front-endje/
-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     : törölt, hiányzó csomagoknál jóbarát!
-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.
```

```
aptitude
install program      : program telepítése
search program       : program keresése a tárolókban
remove csomag        : csomag eltávolítása, a configok megtartásával
purge csomag         : csomag és configok eltávolítása
clean                : gyorsítótár ürítése
show csomag          : telepíthető "csomag"-ról információk
reinstall program    : adott program újratelepítése
update              : csomaglista frissítése
safe-upgrade         : a legújabb verzióra frissíti a telepített csomagokat.
full-upgrade         : a legújabb verzióra frissíti a csomagokat. és eltávolít ha szükséges
dist-upgrade        : újabb verzióra való teljes rendszerfrissítés
```

```
echo "csomag hold" | dpkg --set-selections
# csomag nevű csomag visszatartása a frissítések során.
# visszaállítás: hold helyett install-t szükséges megadni
```

```
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 csomaglista mentése, visszaállítása

```
dpkg --get-selections > /eleresi/ut/csomagok.txt : feltelepített csomagok neveinek mentése TXT-be.
dpkg --set-selections < /eleresi/ut/csomagok.txt : a TXT-ből kiolvassa a felsorolt csomagokat.
apt-get -y update
apt-get dselect-upgrade
# Első lépésben a feltelepített csomagok listája a csomagok.txt-be kerül mentésre.
# Második, harmadik, negyedik lépésben pedig a telepített sw.-k listájában felsorolt
# csomagok újratelepítése
```

```
dselect
# A dpkg barátságosabb arca
```

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

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
-d cím : arp táblából adott bejegyzés törlése

cat /proc/net/arp : aktuális arp tábla kiírása

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
# Netfilter csomagszűrő, leírás lentebb található, itt:
iptables alapok

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.

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 induláskor
# 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.

ifstat
# interface statisztikák készítése
# ifstat programot szükséges telepíteni

ifstatus
# interface státusz kiírása (van-e link adott kártyán/kábelén)
# ifplugd program részét képezi
# újabb verzióban már az ifplugstatus -t szükséges használni.

mii-tool
# Hálózati tesztelő

nbtscan
# NetBIOS információkat adja meg, a hálózaton

netstat
# Információkat ír ki a hálózati alrendszeréről
-r : route kimenete
-i : interfész tábla
-t : 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.

netstat -an | egrep 'Proto|LISTEN'
# mely protokollon/portokon figyel a gép?

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

domainname
nisdomainname
ypdomainname
# Megjeleníti vagy beállítja a rendszer NIS/YP-tartománynevét.

dnsdomainname
#Megjeleníti vagy beállítja a rendszer DNS-tartománynevét.

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 21-25,80,139,8080-at vizsgál.
```

```
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-hez.

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 szekszíójához hozzáadandó)
# Az ethX eszközünk 192.168.2.0 hálózat felé a 192.168.2.1 legyen a default gateway.

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 gépnev 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 az üzenet végén
echo üzenet | smbclient -M gépnév : azonnal küldi a 192.168.2.1 legyen a default gateway.
```

```
smbclient -L 192.168.1.1 -U HOSTNAME/user -W workgroup
# kilistázza a 192.168.1.1 gépen található megosztásokat HOSNAME\user nevében, workgroup tartományban.
```

ssh

```
/Secure SHell/
# Titkosított hálózati bejelentkezés
# config: /etc/ssh/sshd_config
```

```
ssh user@host -t 'sudo sed -i "$a text to insert" /path/to/file'
# adott user nevében a host-on lévő file-ba adott "text to .." szöveg hozzáfűzése
```

```
ssh user@host 'bash -s' < script.sh
# távoli hoston adott user nevében script.sh futtatása, mely a helyi gépen van.
```

```
ssh -t user@host "sudo sh -c 'ps aux | grep ssh > /root/1.txt'"
# távoli gépen adott sudo parancs futtatása, kimeneti állomány is a távoli gép adott mappájába kerül.
```

```
scp
# Biztonságos távoli fájl átvitel lokális gép és távoli gép, illetve távoli gépek között.
Bővebben itt olvasható róla az oldalon:
Fájlátvitel SCP segítségével
```

```
sntop
# Top-szerű hálózati forgalom figyelő szoftver.
```

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

```
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 letöltési sáv szélt.
```

```
tsclient
# Terminal Services Client
```

```
proftpd
# FTP server
```

```
pure-ftp
# FTP server
```

```
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
```

Virtuális hálókártya létrehozása

```
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 venni a /etc/interfaces-be!
```

```
...
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 192.168.10.42
netmask 255.255.255.0
```

```
auto eth0:1
allow-hotplug eth0:1
iface eth0:1 inet static
address 192.168.100.42
netmask 255.255.255.0
```

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

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

```
vsftpd /Very Secure FTP daemon/
# config file.: /etc/vsftpd
# Néhány fontos kérdés.:
-anonymous_enable : anonymous ftp engedélyezése. Szükség van egy ftp nevű user-re.
-local_enable : normal felhasználókat engedjük be. (kell az anonymous 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 sávszélt.
ethX clear : Törli a beállításokat.
```

Bash programozás [top](#)

```
# A parancsértelmező által ismert parancsok fűzhetőek össze egy "programba" vagy más néven shell script-be. (lásd .bat)
# A program első sorában minden esetben meg kell adni a futtató programot (shell-t), jelen esetben a bash-t.
# Tehát a fájl kezdete legyen bash esetén:
#!/bin/bash
# Felkiáltójel nélkül más futtató program (parancsértelmező) is megadható a program első sorában.
# Futtatási jog nélkül nem futtatható, ezért
chmod +x programnev
# Program futtatásához a lehetséges parancs(ok)
./programnev
sh programnev
bash programnev
```

Írásjelek

```
\      Escape-karakter:
      # Bourne Again Shell escape-karakter. Többek között ezzel a jellel lehet elérni,
      # 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 adja vissza
        # úgy, hogy az új sor karakter helyére szóközt rak.

#        Kettőskereszt
        # Kommentezés eszköze. (megjegyzés)

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

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

Dedikált környezeti változók

```
EDITOR   : Az alapértelmezett szövegszerkesztő
HOME     : Aktuális felhasználói könyvtár elérési útvonala
MAIL     : Beérkező emailek alapértelmezett könyvtára
PATH     : Azok az elérési útvonalak, melyekben a shell futtatható állományokat keres
OSTYPE   : Az operációs rendszer típusa
PAGER    : Az alapértelmezett file nézegető elérési útvonalát kell tartalmaznia
PWD      : Az éppen aktuális munkakönyvtár (Szimbólikus linknél nem egyezik meg a pwd kimenetével!)
SHELL    : A jelenleg futó shell teljes elérési útvonala
TERM     : A terminál típusát tartalmazza
```

Ezek értékeinek kiírására 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 változó környezeti változóvá alakítása
        # Amennyiben szükséges permanens beállítás, úgy:
        # az /etc/profile -ba is szükséges felvenni
```

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 lehet egy program kimenetét (stdout) egy másik program
bemenetére (stdin) csatolni. pl.:
```

```
ps aux | grep iceweasel
        # Az összes folyamat listázásának kimenetét a grep szűrő bemenetére irányítja, amely az iceweasel-t keresi.
```

```
# 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
        # program lefuttatása és az esetleges hibacsatorna (stderr) hiba.txt-be irányítása
        # az stdout helyett.
```

```
./program 1> kimenet.txt 2>&1
        # program lefuttatását követően a kimenetet (stdout) a kimenet.txt be menti, valamint
        # az esetleges hibák (stderr) hozzáfűzésre kerülnek a 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 wc
        # kimenete a file2-be íródik
```

Néhány egyszerű példa.:

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

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

#!/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" = "foo1" ]; 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
# eldönti T1 és T2 értékéről, hogy egyenlőek e.

#!/bin/bash
if [ -f /etc/passwd ]
then
    echo "letezik!"
else
    echo "nem letezik!"
fi
# eldönti, 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 változót 1-el növeli addig, amíg el nem éri a 10-et.
```

-until szerkezet.:

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

-for ciklus (C szerű)

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

-for ciklus (a könyvtár elemeire)

```
#!/bin/bash
for i in $( ls );do
    echo a konyvtar elemei: $i
done
# kiírja 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 kiírása ( 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 kiírása ( a 11. sor sosem fut le!)
```

```

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

test -d /home/user && echo "van" || echo "nincs"
# Megvizsgálja, 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
# Megvizsgálja, 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 jelszavát..)

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 Password
# 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.

while read; do
rm „$REPLY”
done < delete.txt > log.txt
# delete.txt file-ban található elérési útvonalakon lévő fájlokat törli, a kimenet
# pedig a log.txt-be kerül.

cat smb.conf | egrep -v "^\s*(#|\$)"
# smb.conf file tartalmának kiírása a #-al vagy szóközzel kezdődő sorok nélkül.

```

File-ban egymás alá felsorolt számok összegének kiszámítása

```

-----
awk '{ sum += $1 } END { print sum }' file

paste -sd+ file | bc

sum=0;for i in $(cat file);do sum=$((sum+$i));done;echo $sum

s=0;while read l; do s=$((s+$1));done

```

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 particionáló

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

zgv
# Konzolos képnézegető

```

mdadm - RAID alapok [top](#)

Az mdadm (multiple devices) a Linux eszköze a tömbök (RAID) létrehozására, kezelésére, monitorozására. Ezen eszközzel software-es RAID alakítható ki, hardware-es RAID vezérlővel rendelkező rendszeren az alábbi leírás nem alkalmazható. Ebben a fejezetben tárgyalt parancsokat virtuális környezetben teszteltem, éles adatokat és rendszer lemezt/partíciót mellőzve.

A dokumentum tartalma

```

Linkek
Gyakorláshoz szükséges feltételek
Az mdadm 7 üzemmódja
Lemezek előkészítése
RAID tömbök létrehozása
  Linear - JBOD
  RAID0 - Stripe
  RAID1 - Mirror
  RAID5 - Stripe set with parity
  RAID10 - Striped mirrors

RAID config mentés, tömbök formázása, fs beállítása, mount point
Növelés (Grow), csökkentés, leállítás, törlés, vizsgálat, Pending
Monitorozás, email küldés
RAID1 meghibásodás szimulálása hot-spare lemezzel
RAID10 tesztelés 2 hibás lemezzel

```

Linkek:

[RAID szintek - Wikipedia](#)
[mdadm - unixlinux.tmit.bme.hu](#)
[mdadm - Wikipedia - eng](#)

Gyakorláshoz szükséges feltételek

```

Fenti RAID szintek link tartalmának ismerete
Linux-ot futtató PC, 1db rendszer és minimum 2db RAID gyakorláshoz használható lemezzel.
VAGY
Virtuális környezetben futtatott Linux VM, 1 + min.2db virtuális lemezzel. pl: Windows-on VMware Player free
Türelem, Pizza, koffeines ital (copyright by raid.wiki.kernel.org)

```

Az mdadm 7 üzemmódja

```

Assemble      : Egy korábban létrehozott tömböt alakít át aktívvá (láthatóvá)
Build         : Szuperblokkmentes tömb összeállítása. Ilyenkor az első létrehozás és a későbbi újbóli
               elindítás ugyanaz a művelet
Create        : Új (szuperblokkos) tömb létrehozása

```

```
Follow/Monitor : A megadott tömbök állapotát figyeli, reagál a változásokra . Csak redundáns
                tömbökre van értelme , pl RAID1, RAID5
Grow           : Újracsíkozás. pl növelés, csökkentés, diszk szám növelés, chunk méret változtatás,
                áttérés RAID1->RAID5
Manage        : Kezelés. pl eszköz hozzáadása tömbhöz, kiszedése, hibásnak jelölése, spare hozzáadása...
Misc          : Egyéb. pl RAID szuperblokk törlése, információkérés. Adott tömbből egy lemez végleges
                eltávolításához szükséges a szuperblokkot törölni.
```

Lemezek előkészítése

Ahhoz, hogy a lemezek RAID tömbbe helyezhetőek legyenek, szükséges előkészíteni az eszközök partíciós tábláit. Ez többek között az alábbi két módszerrel valósítható meg:

```
fdisk /dev/sdb
# /dev/sdb eszköz 1.partíciójának felkészítése
# 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!)
w          : kiírás.

# Fenti parancsot az aktuális rendszerhez kell igazítani, és minden érintett eszközön le kell futtatni.
# Lemez elnevezések és partíció számok sdb1 !
```

VAGY

```
cfdisk /dev/sdb
# formázatlan HDD esetén:

Label type  : dos
New         : méretezés igény szerint
Primary/Ext. : Elsődleges vagy Kiterjesztett. Logikai partíció csak kiterjesztett alá készíthető.
Type        : fd - Raid autodetect
Write       : változások kiírása.

# Fenti folyamatot szintén minden érintett lemezen szükséges elvégezni.
```

RAID tömbök létrehozása

Linear vagy JBOD (just a bunch of disks/drives)

- Több különböző (vagy akár azonos) lemez (vagy partíció) 1db nagy dinamikus lemezzé konvertálása
- Sebességben nincs különbség egy szimpla lemezhez képest.
- Elérhető maximális méret: a résztvevő hdd-k vagy partíciók összege
- Hibatűrés nincs! Bármely lemez kiesése esetén szétesik a tömb

```
mdadm --create --verbose /dev/md0 --level=linear --raid-devices=2 /dev/sdb3 /dev/sdc4
# create      :létrehozás
# verbose     :bővesztű
# /dev/md0    :/dev/md0 lesz a tömb device
# --level=    :RAID típusa linear
# --raid-devices= :tömbbe kerülő eszközök száma
# /dev/sdb3, sdc4 :a résztvevő lemezek partíciói
```

RAID0 - Stripe

- Stripe, vagyis csíkozás. A résztvevők között az adat annyi egyenlő részre lesz szétosztva, ahány résztvevő van, így minden lemezre az adat egy darabja kerül.
- Sebesség: Az írás/olvasás párhuzamosan történik -> **Gyors**
- Elérhető maximális méret: a résztvevő legkisebb hdd vagy partíció mérete.
- erősen ajánlott azonos típusú lemezeket használni.
- Hibatűrés nincs! Bármely lemez kiesése esetén szétesik a tömb.

```
mdadm --create --verbose /dev/md0 --level=stripe --raid-devices=2 /dev/sdb1 /dev/sdc1
# 2 eszközökből álló RAID0 tömb létrehozása
```

RAID1 - Mirror

- Tükrözés 2 vagy több lemez között.
- a résztvevő hdd-kre vagy partíciókra az adat 1-1 példányban kerül írásra. **Pazarló helykihasználtság!**
- Sebesség: Olvasás ideális esetben kétszerese egy szimpla HDD-nek. Írásban minimális a növekedés.
- Elérhető maximális méret: a résztvevő legkisebb hdd vagy partíció mérete.
- ajánlott azonos típusú lemezeket használni.
- Hibatűrés van! 1db lemez kiesése esetén a tömb ép marad, adat írható/olvasható.

```
mdadm --create --verbose /dev/md0 --level=mirror --raid-devices=2 /dev/sdb1 missing
# /dev/sdb1 lemezzel Degraded (félháibú) tömb létrehozása. A hiányzó eszköz helyére missing kell.
```

```
mdadm --create --verbose /dev/md0 --level=mirror --raid-devices=2 /dev/sdb1 /dev/sdc1
# /dev/sdb1 és sdc1 résztvevőkkel RAID1 létrehozása
```

```
mdadm --create --verbose /dev/md0 --level=mirror --raid-devices=2 /dev/sdb1 /dev/sdc1 --spare-devices=1 /dev/sdd1
# md0 RAID1 tömb létrehozása 1db Spare disk-el (melegtartalék)
# Amennyiben bármely lemez kiesik, úgy a rendszer a spare disk-et automatikusan sync-eli és átveszi
# a hibás lemez helyét. Redundancia növekedés!
```

RAID5 - Stripe set with parity

- Minimum 3db, lehetőleg azonos méretű lemez szükséges RAID5 kialakításhoz. Egy adatblokk 3db lemez esetén két részre osztva egy-egy lemezre íródik, (disk1-data1, disk2-data2) ezen adatok paritás értéke pedig (disk3-dataP) a harmadik lemezre kerül.

a paritás egy ellenőrző összeg, melyből a CPU (vagy vezérlő) data1 VAGY data2 elvesztése esetén pótolni tudja a kiesést. a RAID5 körbeforgó paritási (rotating parity) eljárást alkalmaz, nincs kitüntetett paritás lemez (pl RAID4 melyben szűk keresztmetszete lenne a rendszernek) a paritást az összes meghajtón elosztva tárolja.

- Sebesség: párhuzamos olvasásnak köszönhetően gyors olvasás, az írás sw RAID esetén fokozottan CPU igényes! kiszámítása $(N-1)*S$ ahol N:részrtvevő darabszám, S:leglassabb résztvevő sebessége
- Elérhető maximális méret kiszámítása: $(N-1)*S$ ahol N:részrtvevő darabszám, S:legkisebb résztvevő mérete, tehát 3db 500GB-os lemeznél 1TByte.
- Hibatűrés van! 1db lemez kiesése esetén a tömb ép marad, adat írható/olvasható.

```
mdadm --create --verbose /dev/md0 --level=5 --raid-devices=3 /dev/sdb1 /dev/sdc1 /dev/sdd1
# md0 RAID5 tömb létrehozása 3 lemez 1.partíciójából.
```

```
mdadm --create --verbose /dev/md0 --level=5 --raid-devices=3 /dev/sdb1 /dev/sdc1 /dev/sdd1 --spare-devices=1 /dev/sde1
# md0 RAID5 tömb létrehozása, 1db spare disk-el.
```

RAID10 - Striped mirrors

- Minimum 4db, lehetőleg azonos méretű lemez szükséges RAID10 (1+0)kialakításhoz. 2db (2 lemezből álló) előre létrehozott RAID1 tömb kerül összefűzésre RAID0-ba. A RAID10 technológia ötvözi a RAID0 gyorsaságát a RAID1 megbízhatóságával.
- Sebesség: párhuzamos olvasásnak és írásnak köszönhetően szimpla HDD-hez képest legalább kétszer gyorsabb.
- Elérhető maximális méret kiszámítása: $(N*S)/2$ ahol N:részrtvevő darabszám, S:legkisebb résztvevő mérete, tehát 4db 500GB-os lemeznél 1TByte.
- Hibatűrés van! Amennyiben a RAID1 tömbökből esik ki 1-1 lemez, úgy a rendszer továbbra is üzemképes marad.

```
mdadm --create --verbose /dev/md0 --level=10 --raid-devices=4 /dev/sdb1 /dev/sdc1 /dev/sdd1 /dev/sde1
# md0 RAID10 tömb létrehozása 4db lemez 1.partíciójából.
# Tesztelés lentebb.
```

RAID config mentés, tömbök formázása, fs beállítása, mount point

Amennyiben a fentiek alapján sikeresen létre lett hozva a tömb, úgy szükséges az alábbi lépéseket elvégezni a config véglegesítéséhez illetve a tömb használatba vételéhez.

Példa RAID1 esetén

```
cat /proc/mdstat
# az összeállított RAID tömb állapotának lekérése
# Alábbi kimenet RAID1 config sdb1 és sdc1-el.

    Personalities : [raid1]
    md0 : active raid1 sdb1[0] sdc1[1]
    7810036 blocks super 1.2 [2/2] [UU]

cp /etc/mdadm/mdadm.conf /etc/mdadm/mdadm.conf-orig
# aktuális mdadm.conf mentése mdadm.conf-orig néven

mdadm --detail --scan >> /etc/mdadm/mdadm.conf
# a kialakított állapot, azaz a config mentése

mkfs.ext3 /dev/md0
# /dev/md0 tömb ext3 típusú fájlrendszerre formázása

mkdir /mnt/md0
# csatolási pont - mount point létrehozása ahol el lehet majd érni a tömböt
# helye szabadon választott.

mount /dev/md0 /mnt/md0
# a tömb /mnt/md0 csatolási pontra mount-olás

mount
# az előbbi mount tesztelése. Valami hasonló lesz a kimenet:
..
/dev/md0 on /mnt/md0 type ext3 (rw,errors=remount-ro)

cp /etc/fstab /etc/fstab-orig
# /etc/fstab mentése.

blkid /mnt/md0
# Ahhoz, hogy minden újraindítás után fel legyen csatolva a tömb, bele kell írni az fstab-ba UUID alapján.
# Az adott tömb UUID értékének kiírása. Az fstab-ba ezen értéket szükséges felvenni, valahogy így:
..
    UUID=ede160d0-9504-4a21-8460-add12df04b77 /mnt/md0 ext3 errors=remount-ro 0 1

mdadm --detail /dev/md0
# md0 tömb tulajdonságok.
# RAID1, 1db spare disk-el valahogy így néz ki:
/dev/md0:
    Version : 1.2
    Creation Time : Thu Apr 19 14:37:01 2016
    Raid Level : raid1
    Array Size : 103296 (100.89 MiB 105.78 MB)
    Used Dev Size : 103296 (100.89 MiB 105.78 MB)
    Raid Devices : 2
    Total Devices : 3
    Persistence : Superblock is persistent
    Update Time : Thu Apr 19 19:04:01 2016
    State : clean
    Active Devices : 2
    Working Devices : 3
    Failed Devices : 0
    .....
```

Ha minden sikerült, kész a felcsatolt RAID1 tömb.

Növelés (Grow), csökkentés, leállítás, törlés, vizsgálat, PendingRövidítések

```
-a      : --add
-c      : --create
-l1     : --level=mirror   (raid1)
-n3     : --raid-devices=3
-x1     : --spare-devices=1
```

```
mdadm --grow --raid-devices=3 /dev/md0
mdadm --add /dev/md0 /dev/sde1
# Adott /dev/md0 tömbben a diszkek számának növelése 3-ra, majd /dev/sde1 hozzáadása.
```

```
mdadm --fail /dev/md0 /dev/sdc1
mdadm --remove /dev/md0 /dev/sdc1
mdadm --grow --raid-devices=2 /dev/md0
# /dev/md0 tömbben az sdc1 résztvevő hibásnak beállítása és eltávolítása
# ezt követően md0 tömb lemezek számának csökkentése 2-re.
```

```
mdadm --stop /dev/md0
# md0 tömb leállítása
```

```
mdadm --stop --scan
# az mdadm.conf -ban az összes felsorolt tömb leállítása
```

```
mdadm --remove /dev/md0
# md0 végleges törlése, stop-ot követően
```

```
mdadm --zero-superblock /dev/sdb1
# superblock ürítés aktuális eszközön.
# akkor lehet szükséges, ha sdb1 korábban már RAID tömb tagja volt, de most egy új tömbbe kell felvenni.
```

```
mdadm --examine /dev/sda1
# /dev/sda1 eszközön md superblock keresése/vizsgálata - examine
# sda1 része vagy része volt-e aktív RAID tömbnek?
```

```
Ha nem:
mdadm : No md superblock detected on /dev/sda1
```

```
Ha igen:
/dev/sdb1:
  Magic : a92b4efc
  Version : 1.2
  Feature Map : 0x0
  Array UUID : 29473177:dcda05f4:7b158228:7c7326f1
  Name : debian830:0 (local to host debian830)
  Creation Time : Wed Apr 26 14:58:06 2016
  Raid Level : raid10
  Raid Devices : 4
  .....
```

```
mdadm --readwrite /dev/mdX
# Amennyiben /dev/mdX állapota valamilyen okból kifolyólag Pending: resync=PENDING
# úgy a fenti paranccsal indítható ismét a sync.
```

Monitorozás, email küldés

Az mdadm rávehető folyamatos monitorozásra is, ami egy daemont futtat a háttérben és hiba esetén riasztást küld, illetve ha beállításra kerül, be is tud avatkozni.

Levélküldéshez használható kliens például **ssmtp**. A configjában a mailhub-hoz fel kell venni az SMTP címet

```
/etc/mdadm/mdadm.conf
# -ba fel kell venni a címzettet, ellenkező esetben a helyi root kapja az értesítést.
```

```
MAILADDR recipient@mail.com
```

```
mdadm --monitor --scan --test -1
# test mail kiküldése mdadm-on keresztül.
```

```
mdadm --monitor --daemonise --mail=recipient@mail.com --delay=30 /dev/md0
# daemon módban futó scan 30másodpercenként, címzettet megadva.
# delay default értéke 1800sec, én csak a teszt erejéig húztam le 30s-re
```

```
mdadm --fail /dev/md0 /dev/sdd1
# teszt jelleggel sdd1 kivétele md0-ból:
```

```
mdadm: set /dev/sdd1 faulty in /dev/md0
```

```
# Alábbi levél érkezett másodpercekkel később:
# a resync DELAYED nem releváns, mire megnéztem az mdstat-ot, addigra már kész is volt a (spare) sync, automatikusan! :)
```

```
This is an automatically generated mail message from mdadm running on debian830
A Fail event had been detected on md device /dev/md0.
It could be related to component device /dev/sdd1.
Faithfully yours, etc.
P.S. The /proc/mdstat file currently contains the following:
```

```
Personalities : [raid1]
md0 : active raid1 sdc1[3] sdb1[4] sdd1[2](F)
103296 blocks super 1.2 [2/1] [U_]
resync=DELAYED
unused devices:
```

RAID1 meghibásodás szimulálása hot-spare lemezzel

Alábbi tesztben egy működő RAID1 tömb egyik aktív lemeze hibásodik meg úgy, hogy a tömb részét képezi 1db hot-spare lemez is (melegtartalék). Ilyen esetben az mdadm a hibásnak ítélt lemez eltávolítását követően automatikusan becsatolja a spare lemezt, és el is kezdi rászinkronizálni az adatokat.

```
cat /proc/mdstat
# aktuális helyzet felmérése (látható, hogy sdc1 a spare!)
```

```
Personalities : [raid1]
md0 : active raid1 sdc1[3](S) sdb1[0] sdd1[2]
103296 blocks super 1.2 [2/2] [UU] unused devices:
```

```
mdadm --fail /dev/md0 /dev/sdb1
# sdc1 hibásnak ítéltése
```

```
mdadm: set /dev/sdb1 faulty in dev/md0
```

```
cat /proc/mdstat
# látható, hogy sdb1 Faulty, és sdc1 spare átvette a helyét.
```

```
Personalities : [raid1]
md0 : active raid1 sdc1[3] sdb1[0](F) sdd1[2]
103296 blocks super 1.2 [2/2] [UU] unused devices:
```

```
mdadm --remove /dev/md0 /dev/sdb1
# sdb1 kivétele a tömbből
```

```
mdadm: hot removed /dev/sdb1 from /dev/md0
```

```
cat /proc/mdstat
# sdb1 el is tűnt.
```

```
Personalities : [raid1]
md0 : active raid1 sdc1[0] sdd1[2]
103296 blocks super 1.2 [2/2] [UU] unused devices:
```

```
mdadm --add /dev/md0 /sdb1
# sdb1 visszahelyezése a tömbbe. Ekkor a frissen behelyezett eszköz lesz a spare.
```

```
Personalities : [raid1]
md0 : active raid1 sdc1[3] sdb1[4](S) sdd1[2]
103296 blocks super 1.2 [2/2] [UU] unused devices:
```

RAID10 tesztelés 2 hibás lemezzel

Alábbi tesztben egy működő RAID10 tömbből 2db lemez esik ki, RAID1 tömböknént 1-1, melyet elméleti síkon még túl kellene élnie.

```
cat /proc/mdstat
# Jelenlegi helyzet így néz ki.: RAID10, sdb,sdc,sdd,sde lemezek első partícióiból
```

```
Personalities : [raid10]
md0 : active raid10 sde1[3] sdd1[2] sdc1[1] sdb1[0]
204800 blocks super 1.2 512K chunks 2 near-copies [4/4] [UUUU]
```

```
mdadm --detail /dev/md0
# detail információk ide vágó része:
# Látható set-A set-B infók alapján, hogy sdb1 és sdd1 képezi az egyik RAID1-et, sdc1 sde1 a másikat.
# A két RAID1 tömbre pedig szét van osztva egy adatblokk (A+B) -> RAID0
```

Number	Major	Minor	RaidDevice	State		
0	8	17	0	active sync	set-A	/dev/sdb1
1	8	33	1	active sync	set-B	/dev/sdc1
2	8	49	2	active sync	set-A	/dev/sdd1
3	8	65	3	active sync	set-B	/dev/sde1

```
ls -la /mnt/md0
# md0 tömb becsatolva /mnt/md0-ba, teszt adattal feltöltve:
```

```
total 124528
drwxr-xr-x 4 root root    1024 Jan 17 16:04 .
drwxr-xr-x 3 root root    4096 Jan 15 01:48 ..
-rw-r--r-- 1 root root 126998528 Jan 17 16:05 123
drwx----- 2 root root   12288 Jan 17 16:04 lost+found
drwxr-xr-x 2 root root    1024 Jan 17 16:04 test-data
```

```
mdadm --fail /dev/md0 /dev/sdb1
mdadm --fail /dev/md0 /dev/sde1
mdadm --remove /dev/md0 /dev/sdb1
mdadm --remove /dev/md0 /dev/sde1
# sdb1 és sde1 résztvevők hibásnak ítéltése és eltávolítása a tömbből.
```

```
cat /proc/mdstat
# Jelenlegi helyzet:
```

```
Personalities : [raid10]
md0 : active raid10 sdd1[2] sdc1[1]
204800 blocks super 1.2 512K chunks 2 near-copies [4/2] [_UU_]

```

```
mdadm --detail /dev/md0
# detail információk ide vágó része:
```

Number	Major	Minor	RaidDevice	State	
0	0	0	0	removed	
1	8	33	1	active sync	set-B /dev/sdc1
2	8	49	2	active sync	set-A /dev/sdd1
6	0	0	6	removed	

```
ls -la /mnt/md0
# Voial! /mnt/md0-ba felcsatolt degraded tömb még mindig él, 2 hibás lemezzel.
```

```
total 124528
drwxr-xr-x 4 root root    1024 Jan 17 16:04 .
drwxr-xr-x 3 root root    4096 Jan 15 01:48 ..
-rw-r--r-- 1 root root 126998528 Jan 17 16:05 123
drwx----- 2 root root   12288 Jan 17 16:04 lost+found
drwxr-xr-x 2 root root    1024 Jan 17 16:04 test-data
```

A fenti RAID10 tömb képes túlélni két lemez kiesést abban az esetben, ha set-A és set-B lemezekből legalább 1-1 darab épen marad. Amit nem képes elviselni, ha 2db set-A vagy 2db set-B esik ki. Egész jó nem? Ha a tömb részét képezte volna 2db hot spare lemez, úgy a faulty-remove művelet után azonnal munkába is álltak volna, beavatkozás nélkül.

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 megfontoltan.
#
# 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 parancsokat adjuk ki.

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 "féllábú" (degraded) tömbünket a /dev/sdb lemezre.
# Mivel a rendszer jelenleg a /dev/sda lemezzel fut, egyelőre azt nem tudjuk a tömbhöz adni, csak utólag.

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 féllábú 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 mentsünk 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 mentsünk 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.:

[...]  
default 0  
fallback 1  
[...]  
  
# A config végén készítsünk az első (title-savedefault közötti) részből egy új bekezdést.  
# valamint, a /dev/sda3 -at cseréljük /dev/md2 re  
# a root (hd0,0) -t (hd1,0)-ra. (Ha különbözőek a lemezeid elnevezése, itt is légy körültekintő!)
```

```
# 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 része.)

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

grub
root (hd0,0)
setup (hd0)
root (hd1,0)
setup (hd1)
quit
# Telepítjük a grub-ot a második lemezünkre.
# A következő indításkor, ha minden sikeres volt, már a második HDD-ről indul a rendszerünk!
# 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 billentyűkkel.:
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://www.howtoforge.com/linux-raid1-on-2-disks)


```
# 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 gépben szeretnénk menteni. Alábbi leírás ehhez a procedúrához nyújt segítséget. (a dev. elnevezések nálad egészen mások lehetnek!)

```
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 vonatkozik!

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 alkot.
# 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 szükséges 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=154.1min speed=133211K/sec

# 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), az internetes forgalom csökkentésének érdekében. Magyarán a teljes tartalmat (2011.10. hóban: ~70Gb.) letöltjük egyszer, a későbbiekben pedig csak frissíteniük 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 telepítünk.

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.list ben van a config.)
# Ha kész a letöltés, a /var/spool/apt-mirror/mirror/ftp.hu.debian.org/debian alatt ott lesznek a fájlok.

/bin/bash /var/spool/apt-mirror/var/clean.sh
# Főlöslég 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 létrehozunk, hogy a kliensek be tudják frissíteni csomaglistájukat

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 fájlrendszerbe SSHFS (Secure Shell FileSystem) 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 távoli mappa.
# A szerverre természetesen szükséges SSH szervert is telepíteni.

lsmod |grep fuse
# Kliens oldalon szükség lesz a fuse nevű kernel modulra is, szóval ellenőrizzük. Üres kimenet esetén nincs betöltődve!

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

useradd sshbackup
# A szerveren az sshbackup felhasználó létrehozása. Az ő nevében jelentkezünk be SSH-n keresztül a kliensről.

passwd sshbackup
# Jelszó beállítása sshbackup felhasználónak.

gpsswd -a sshbackup dirgroup
# A távoli mappához szükséges a távoli sshbackup usernek hozzáférnie. Ezt szerver oldalon tisztázni kell.
# Egy példa.:
# sshbackup user hozzáadása a dirgroup csoporthoz, melynek jogosultsága van (vagy lesz) a távoli mappához.
# Ez természetesen kikerülhető, amennyiben a felcsatolandó mappa tulajdonosa már az sshbackup user.

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 mappáját, tcp22-es portot használva.
# A csatolási pont a helyi fájlrendszerben /home/user/sshbackuplink mappa lesz.

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

sshbackup@192.168.1.1:/home/sshbackup on /home/user/sshbackuplink type fuse.sshfs (rw,nosuid,nodev,max_read=65536)

df --si
# Megbizonyosodás2

sshbackup@192.168.1.1:/home/sshbackup
...

fusermount -u /home/user/sshbackuplink
# Lecsatolás

```

Forrás:
[Howtoforge](#)

Távoli FTP felcsatolása lokális fájlrendszerbe [top](#)

Alábbi rövid leírás ismerteti távoli FTP(s) szerver felcsatolásának lehetőségét a lokális fájlrendszerbe curlftpfs segítségével.

```
apt-get install curlftpfs
# curlftpfs csomag telepítése

mkdir /mnt/ftp
# csatolási pont létrehozása, ahova csatolásra kerül a távoli FTP tárhely

curlftpfs felhasználó:jelszo@ftp.domain.com /mnt/ftp
# ftp.domain.com tárhely felcsatolása felhasználó/jelszo autentikációval.
# FONTOS!
# Ezen felcsatolási módszerrel a folyamat listában látszódni fog a felhasználónév és jelszó! (ps aux)
# Ennek elkerülése érdekében a lenti .netrc-vel kell megoldani az auth.-ot

touch /root/.netrc
# .netrc állomány létrehozása, az alábbi tartalmat szükséges beleírni:

machine ftp.domain.com
login felhasználó
password jelszo

# Amennyiben a távoli kiszolgáló tárhely ftps-en is elérhető, úgy a machine ftps.. javasolt!

chown root:root /root/.netrc
chmod 600 /root/.netrc
# .netrc tulajdonos és jogosultság tulajdonságok korlátozása.

curlftpfs -o ssl ftp.domain.com /mnt/ftp
# ftp tárhely felcsatolása /mnt/ftp mappába, opcionális a -o ssl, ha erre a szolgáltató lehetőséget ad.
# Ezen módszerrel a folyamatlistában már nem látszik az auth.

umount /mnt/ftp
VAGY:
fusermount -u /mnt/ftp
# fenti két paranccsok használhatóak a tárhely lecsatolására
```

iSCSI Target létrehozása, Initiator beállítása (IET) [top](#)

Az iSCSI (Internet Small Computer Systems Interface) egy olyan, internetprotokollon alapuló hálózati tárolószabvány, amely segít az adattároló eszközök összekapcsolásában. Az iSCSI révén a tárolókiszolgálón (target) található tárhelyet a kliens (initiator) operációs rendszere helyi lemezként fogja kezelni.
 Bővebben: [iSCSI Wiki](#)

Alábbi leírásban 1db iSCSI target gépen (Debian Jessie) két tároló kerül kiajánlásra, LUN1.img a Windows, míg a LUN2.img a linux kliensnek. A target tárolóját egy már korábban létrehozott RAID1 kötet alkotja, mely /dev/md1.
 Ennek létrehozásáról itt lehet bővebben olvasni. [mdadm RAID alapok](#)
 A target kiszolgálásáról az IET (iSCSI Enterprise Target - [IET official](#)) gondoskodik.

Dokumentum tartalma

```
iSCSI Target telepítés
Tároló előkészítése
Target-ek felvétele
Initiator engedélyezése, teszt

iSCSI Initiator telepítés
Windows kliens
Linux Kliens
Target felderítése
Hitelesítés beállítása
Login, tesztelés, formázás, mount
Target session ellenőrzése
Logout, lekapcsolás
```

iSCSI Target telepítés

```
apt-get install iscsitarget iscsitarget-dkms
# szükséges programok telepítése

mcedit /etc/default/iscsitarget
# alább látható érték beállítása az adott file-ban

ISCSITARGET_ENABLE=true
```

Tároló előkészítése

```
mkdir /mnt/md1/disk1
mkdir /mnt/md1/disk2
dd if=/dev/zero of=/mnt/md1/disk1/lun1.img bs=1024k count=250
dd if=/dev/zero of=/mnt/md1/disk2/lun2.img bs=1024k count=250
# szeparált mappák illetve a 2db 250MB-os img létrehozása a két kliens számára (initiator)
```

Target-ek felvétele

```
mcedit /etc/iet/ietd.conf
# alább látható két target felvétele a két kliens számára.
```

```
Target iqn.2016-09:Local.home:disk1
IncomingUser windows-IQN-neve secretcomplexpass
OutgoingUser
Lun 0 Path=/mnt/md1/disk1/lun1.img, Type=fileio
Alias LUN1

Target iqn.2016-10:Local.home:disk2
IncomingUser linux-IQN-neve secret2complexpass
OutgoingUser
Lun 0 Path=/mnt/md1/disk2/lun2.img, Type=fileio
Alias LUN1
```

```
Target      : Egyedi IQN (Iscsi Qualified Name) azonosító + reverse domain + storage azonosító
Incoming user : Szükséges a kliensen beállított initiator nevet felvenni, és CHAP kompatibilis (komplex) jelszót beállítani.
               pl iqn.2016-09.local.home:win / ABCDEFGHIjkl123
               A user neveket egyelőre nem szükséges most fixálni, később windows és linux oldalon is létrehozuk,
               és azokat a neveket kell majd a fenti configba beírni
Outgoing user : Mutual (kölcsönös) CHAP auth-hoz kellene, itt nem kerül beállításra
Lun          : Logical Unit Number, azaz logikai kötet. Minden targethez az első ilyen a 0 sorszámú legyen!
               A Path értéke természetesen a már létrehozott .img lesz, de kiosztható teljes partíció/lemez is.
```

Initiator engedélyezése, teszt

```
vi /etc/iet/initiators.allow
# fentebb létrehozott targetekre engedélyezésre kerül a csatlakozás a 192.168.78.0/24-es hálózathoz.
# amennyiben nem szükséges a korlátozás, úgy ALL ALL, azaz mindenhol.
```

```
iqn.2016-09.local.home:disk1 192.168.78.0/24
iqn.2016-10.local.home:disk2 192.168.78.0/24
#ALL ALL
```

```
systemctl restart iscsitarget
/etc/init.d/iscsitarget start
# service újraindítása
```

```
cat /proc/net/iet/volume
# státusz ellenőrzés volume-ra
```

```
tid:2 name:iqn.2016-10.local.home:disk2
     lun:0 state:0 iotype:fileio iomode:wt blocks:245760 blocksize:512 path:/mnt/md1/disk2/lun2.img
tid:1 name:iqn.2016-09.local.home:disk1
     lun:0 state:0 iotype:fileio iomode:wt blocks:245760 blocksize:512 path:/mnt/md1/disk1/lun1.img
```

```
cat /proc/net/iet/session
# státusz ellenőrzés session-re
```

```
tid:2 name:iqn.2016-10.local.home:disk2
tid:1 name:iqn.2016-09.local.home:disk1
```

iSCSI Initiator telepítés

Windows kliens

Ezt itt nem részletezném, alábbi linken található hozzá segítség: [Windowsnetworking.com](http://windowsnetworking.com)

Linux kliens

```
apt-get install open-iscsi
# szükséges program telepítése
```

```
mcedit /etc/iscsi/iscsid.conf
# alább látható érték beállítása az adott file-ban

node.startaup = automatic
```

```
systemctl restart open-iscsi
# service újraindítása
```

Target felderítése

```
iscsiadm -m discovery -t st -p 192.168.78.133
# adott IP feltérképezése, felvehető storage-ok listázása
```

```
192.168.78.133:3260,1 iqn.2016-09.local.home:disk1
192.168.78.133:3260,1 iqn.2016-10.local.home:disk2
```

```
iscsiadm -m node
# látható, hogy az alábbi file-ba bekerültek a bejegyzések a felcsatolható targetekről

/etc/iscsi/nodes/
```

```
cat /etc/iscsi/initiatorname.iscsi
# itt található a kliens iqn azonosítója, ezt a nevet kell beírni a fenti target
# IncomingUser sorába, plusz a jelszót!!!!
# pl.: iqn.2016-10.local.home:linux
```

Hitelesítés beállítása

```
cat/etc/iscsi/nodes/
# ezen könyvtár tartalma alapján kell kiadni a következő mcedit parancsot

mcedit /etc/iscsi/nodes/iqn.2016-10.local.home:disk2/192.168.78.133,3260,1/default
# ezen file-ba kell felvenni a korábban kialakított targethez a user/pass-t, illetve az autentikációt
# közvetlenül a default config file-ba :

node.session.auth.authmethod = CHAP
node.session.auth.username = linux-iqn-neve # ami a fenti targetbe lett felvéve
node.session.auth.password = secret2complexpass # a jelszó...

# vagy kézzel.:
iscsiadm -m node --targetname "iqn.2016-10.local.home:disk2" --portal "192.168.78.133:3260" --op=update --name node.session.auth.authmethod --
iscsiadm -m node --targetname "iqn.2016-10.local.home:disk2" --portal "192.168.78.133:3260" --op=update --name node.session.auth.username --vc
iscsiadm -m node --targetname "iqn.2016-10.local.home:disk2" --portal "192.168.78.133:3260" --op=update --name node.session.auth.password --vc

# Fontos, hogy ebben a könyvtárban csak 1db default állomány foglaljon helyet!
```

Login, tesztelés, formázás, mount

```
iscsiadm -m node --targetname "iqn.2016-10.local.home:disk2" --portal "192.168.78.133:3260" --login
# Login !

Logging in to [iface: default, target: iqn.2016-10.local.home:disk2, portal: 192.168.78.133,3260] (multiple)
Login to [iface: default, target: iqn.2016-10.local.home:disk2, portal: 192.168.78.133,3260] successful.

dmesg
# dmesg kimenetében látható, hogy létrejött egy sdb eszköz

[ 1911.360273] sd 9:0:0:0: Attached scsi generic sg2 type 0
[ 1911.364627] sd 9:0:0:0: [sdb] 921600 512-byte logical blocks: (261 MB/250 MiB)
[ 1911.365153] sd 9:0:0:0: [sdb] Write Protect is off
[ 1911.365158] sd 9:0:0:0: [sdb] Mode Sense: 77 00 00 08
[ 1911.366381] sd 9:0:0:0: [sdb] Write cache: disabled, read cache: enabled, doesn't support DPO or FUA
[ 1911.376621] sdb: unknown partition table
[ 1911.380552] sd 9:0:0:0: [sdb] Attached SCSI disk

cfdisk /dev/sdb
# partíció létrehozása. RAID leírásban található a pontos lépéssor

mkfs.ext4 /dev/sdb1
# ext4 file-rendszer létrehozása

mkdir /mnt/iscsi
mount /dev/sdb1 /mnt/iscsi
# mount point létrehozása, és az eszköz felcsatolása

mount
# íme:

/dev/sdb1 on /mnt/iscsi type ext4 (rw,relatime,user_xattr,barrier=1,data=ordered)
```

Target session ellenőrzése

```
cat /proc/net/iet/session
# Fenti parancsot a targeten futtatva látható, hogy mindkét kiajánlott disk használatban van.

tid:2 name:iqn.2016-10.local.home:disk2
sid:281475047817728 initiator:iqn.2016-10.local.home:linux
cid:0 ip:192.168.78.134 state:active hd:none dd:none
tid:1 name:iqn.2016-09.local.home:disk1
sid:564050387861568 initiator:iqn.2016-09.local.home:win
cid:1 ip:192.168.78.135 state:active hd:none dd:none
```

Logout, lekapcsolás

Fontos hogy a fenti lépés előtt az adott kliensen le kell csatolni a távoli fájlrendszert!
Windows esetén a fenti link segít, Linux esetén pedig umount!

```
iscsiadm -m node --targetname "iqn.2016-10.local.home:disk2" --portal "192.168.78.133:3260" --logout
# Logout! azaz a Windows-nak kiajánlott lemez lecsatolása.
# Linuxon umountot követően logout-olható:

Logging out of session [sid: 8, target: iqn.2016-10.local.home:disk2, portal: 192.168.78.133,3260]
Logout of [sid: 10, target: iqn.2016-10.local.home:disk2, portal: 192.168.78.133,3260] successful.
```

```
iscsiadm -m discovery --portal "192.168.78.133:3260" --op=delete
# felderítés kikapcsolása az adott portál felé
```

Forrás:
[Howtoforge](http://Howtoforge.com/server-world.info)
server-world.info

Rsync használata [top](#)

Gyors, hatékony, sokoldalú, távoli (és helyi) backup eszköz.

Egy alaprendszernek nem tartozéka, úgyhogy apt-get install rsync minden gépre, amivel (és amire) szinkronizálnál.

Kapcsolók

```
-a      : Arhív mód. Rekurzívan másol, Jogosultságok, Módosítási dátumok, Tulajdonos és csoport adatok megtartásával.
-u      : frissítés
-r      : rekurzív
-n      : dry-run azaz csak teszt. Tanulási folyamat idején ajánlott!
-l      : symlink másolása symlink-ként.
-L      : symlink-ből file/dir készítése
-p      : jogosultságok megőrzése
-t      : módosítási dátumok megőrzése
-g      : csoport tulajdonság megőrzése
-o      : tulajdonos megőrzése
-z      : tömörítve küldi az adatokat.
-P      : progress, azaz a futás tulajdonságainak (file, sebesség) megjelenítése
-v      : bőbeszédű
-q      : csendes üzemmód
-H      : hard link-ek megtartása
-v4     : IPv4-et előnyben részesítve
-v6     : IPv6-ot előnyben részesítve
--stats : művelet végén rövid összegzés

--existing : csak azon file-ok frissítése melyek már léteznek a célnál.
--delete  : minden olyan adatot töröl a cél mappából, mely a forrás mappában (már) nem szerepel.
--force   : erőltetett törlés, akkor is ha a mappa nem üres.
--delete-excluded : also delete excluded files on the receiving side
--address=ADDRESS : adott cím előnyben részesítése
--password-file=FILE : a futtató jelszavának FILE-ből felolvasása
--bwlimit=KBPS : I/O korlát, kByte-ban megadva
```

Példák

```
rsync -av source/ dest/
# archive, azaz mindent másol, (verbose kimenettel) az aktuális mappa source mappájából a dest-be.
```

```
rsync -avn source/ dest/
-n azaz dry-run, csak tesztel, kiírja mi fog történni.
# magyarul ha a sourcesrv mappa tartalma nőtt, és a fenti parancsot lefuttatjuk, meg fogja adni azon
# állományok/mappák nevét melyeket élesben másolni a /dest alá.
```

```
rsync -av --delete source/ dest/
--delete : minden olyan adatot töröl a cél mappából, mely a forrás mappában (már) nem szerepel.
# Tegyük fel, hogy a source és dest is tartalmaz 1-1 korábban szinkronizált temp mappát, mely a source-ról törlésre kerül.
# A fenti parancs törölni fogja sync során a dest helyről a temp mappát, hiszen az már nincs meg a source-ban,
# ezáltal lesz a két mappa szinkronban.
```

```
rsync -av user@192.168.86.141:/home/user/backup/source/ /mnt/backup/dest/
# user nevében a 192.168.86.141-es IP mögötti /home/user/backup/source/ mappa tartalmának szinkronizálása a helyi
# filerendszer /mnt/backup/dest/ mappájával.
# távoli rsync-nél a túloldalra is kell rsync-et telepíteni!
```

```
rsync -av /mnt/backup/source/ user@192.168.86.141:/home/user/backup/dest/
# a helyi file-rendszer source mappáját szinkronizálja user nevében a 192.168.86.141-es gép /home/user/backup/dest mappájával
# távoli rsync-nél a túloldalra is kell rsync-et telepíteni!
```

```
rsync -av --exclude=dir1 source/ dest/
# source szinkronizálása dest mappába, kivéve a dir1 mappát és tartalmát
```

```
rsync -av --exclude-from=FILE source/ dest/
# source szinkronizálása dest mappába, kivéve a FILE file-ban felsorolt mappákat/állományokat.
```

Kulcs alapú hitelesítés beállítása

Azaz ha jelszó megadása nélkül szeretnél szinkronizálni távoli gépre ssh kulcs segítségével.

```
ssh-keygen
# Azon a gépen, ahol futtatod az rsync parancsot, a szükséges user nevében!
# Passphrase elérési útvonala maradthat default (Enter)
# Passphrase azaz a jelszó maradjon üres. (Enter)
```

```
ssh-copy-id -i ~/.ssh/id_rsa.pub 192.168.86.141
# azaz a frissen létrehozott publikus kulcs elhelyezése a távoli gépre.
# Ezt követően már az ssh és az rsync is menni fog jelszó nélkül.
```

iptables alapok [top](#)

A Netfilter (hálózati csomagok feldolgozására szolgáló eszköz) kezelője. Az iptables a hálózati csomagok feldolgozási szabályait szerepük alapján szervezett táblák formájában kezeli.

Szerepek - táblák

A feldolgozási folyamat típusát jelentik.

filter	: Csomagszűrés bejövő, kimenő és átmenő forgalomra. A legsűrűbben használt tábla.
nat	: Hálózati címfordítás, általában átmenő forgalomra.
mangle	: Egyéb csomagszűrések, például csomagok megjelölése.
raw	: Kivételek kezelése
security	: Jogosultsági hozzáférések szabályozása

Ezen táblák mindegyikére feldolgozási szabályok (rules) lánc (chain) vonatkozik. A szabályok illesztésekből (match - mely csomagokra vonatkozik,) és célokból (target - mit kell tenni az illeszkedő csomaggal) épülnek fel. Egy csomagra csak akkor illeszkedik szabály, ha annak minden feltétele teljesül rá.

A táblák beépített láncai

Az adott táblákban milyen láncokon lehet szűrni a hálózati forgalmat. Azt hogy melyikben milyen forgalmat érdemes, a csomagáramlás szekció taglalja.

filter tábla	: INPUT, FORWARD, OUTPUT
nat tábla	: PREROUTING, INPUT, OUTPUT, POSTROUTING
mangle tábla	: PREROUTING, INPUT, FORWARD, OUTPUT, POSTROUTING
raw tábla	: PREROUTING, OUTPUT
security tábla	: INPUT, FORWARD, OUTPUT

Kapcsolódási pontok a feldolgozás során

PREROUTING	: A hálózati csatolótól kapott csomag, még az INPUT lánc előtt ide kerül.
INPUT	: Épp mielőtt megérkezne a csomag egy helyi folyamathoz. (itt illik szűrni a bejövő forgalmat)
FORWARD	: Egy átjáró gépen a továbbított csomagok itt áramlanak keresztül. Routing.
OUTPUT	: Miután egy helyi folyamat létrehozta a csomagot, itt távozik (kimenő csomag szűrés!)
POSTROUTING	: Mielőtt elhagyná a hálózati csatolót a csomag, az OUTPUT lánc után. NAT-olás!

A láncok házirendje

Default Policy

Ezt a policy-t láncként (INPUT, OUTPUT, FORWARD) a tűzfal script legelején fixálni kell. Azok a csomagok, amelyek a tűzfal láncában felsorolt szabályokra nem illeszkednek, azokra a Default Policy vonatkozik, azaz vagy ACCEPT, vagy DROP. Pl.: Ha a tűzfal INPUT láncában csak a tcp 80-as kerül engedélyezésre, a Default Policy pedig DROP, akkor minden más porton és protokollon érkező csomagot (amely nem illeszkedik a tcp 80-as portot definiáló szabályra) eldob.

Csomagáramlás

A csomagok feldolgozása a szabályok felsorolásának sorrendjében történik. (pl DROP után nincs logolás!) Amennyiben egy szabályra az aktuális csomag nem illeszkedik, úgy a következő szabállyal kerül kiértékelésre, és így tovább. Ha egy szabályra sem illeszkedik, úgy az adott lánc alapértelmezett házirendje (default policy) érvényesül a csomagra. Amennyiben egy szabályban nincs illesztési feltétel, pl nincs megadva bejövő interface, úgy az adott szabály minden interface-re illeszkedik, ha viszont cél (target/művelet!) nincs, akkor a rá illeszkedő csomaggal semmi sem fog történni.

Alábbi listában látható a sorrend ahogyan megjelennek a csomagok a beépített táblákban és láncokban:

Két hálózati csatoló között - Forwarding

mangle	: PREROUTING
nat	: PREROUTING
mangle	: FORWARD
filter	: FORWARD
mangle	: POSTROUTING
nat	: POSTROUTING

tipp: -Port forwardnál először a nat tábla PREROUTING láncában kell engedélyezni a forgalmat, majd a filter tábla FORWARD láncában is, hiszen ez a következő szűrési pont.

Hálózati csatolótól helyi folyamathoz - Input

mangle	: PREROUTING
nat	: PREROUTING
mangle	: INPUT
filter	: INPUT

tipp: -Egyszerű bejövő szűrésnél (módosítás és átirányítás nélkül) a filter tábla INPUT lánc érvényesül, tehát itt kell szűrni.

Helyi folyamattól hálózati csatolóhoz - Output

mangle	: OUTPUT
nat	: OUTPUT
filter	: OUTPUT
mangle	: POSTROUTING
nat	: POSTROUTING

*tipp: -Egyszerű kimenő szűrésnél (módosítás és átirányítás nélkül) a filter tábla OUTPUT lánc érvényesül, tehát itt kell szűrni.
-NAT-olásnál (MASQUERADE vagy SNAT) a nat tábla POSTROUTING láncában kell a beállításokat elvégezni.*

Helyi folyamattól helyi folyamathoz - local

mangle	: OUTPUT
--------	----------

```

nat           : OUTPUT
filter        : OUTPUT
filter        : INPUT
mangle        : INPUT

```

Lánc műveletek, szabály módosítások

```

-L : a láncok szabályainak megjelenítése      (--list)
-S : aktuális lánc szabályok listázása röviden
-P : default policy beállítása
-N : új lánc létrehozása
-F : a lánc összes szabályának törlése        (--flush)
-X : üres láncok törlése                      (--delete-chain)
-Z : csomag és byte számlálók nullázása
-A : új szabály hozzáadása egy lánchoz        (--append)
-D : törlés az adott láncból
-I : szabály injektálás, beszúrás
-R : szabály csere, sorszám alapján

```

Csomag és bájt számlálók

Amikor egy csomag egy illeszkedő szabályhoz ér (azaz minden feltétel teljesült rá) akkor a rendszer a csomag számlálót egyel növeli, míg a bájt számláléhoz hozzáadja a csomag méretét.

iptables -nvxl kimenetében látható pkts és bytes oszlopok:

pkts	bytes	target	prot	opt	in	out	source	destination	
0	0	ACCEPT	tcp	--	eth0	*	191.85.11.60	192.168.1.254	tcp dpt:80
11	660	ACCEPT	tcp	--	eth0	*	191.85.11.60	192.168.1.254	tcp dpt:443

Interface-ek meghatározása

Az interface maga a (fizikai) eszköz amelyen bejön, vagy távozik a csomag.

```

PREROUTING láncon      : "-i eth0" értelmezhető, nincs kimeneti interface!
INPUT láncon           : "-i eth0" értelmezhető      -II-
OUTPUT láncon          : "-o eth0" adható meg, "-i" kapcsolóra egy csomag sem fog illeszkedni.
FORWARD láncon         : "-i eth0" és "-o eth0" interface-ek is alkalmazhatóak.
POSTROUTING láncon     : "-o eth0" értelmezhető, nincs bemeneti interface
Alkalmazható joker karakter az interface-eknél, ez a "+" jel. "-i eth+" -> pl eth0,eth1..

```

Forrás és célcímek megadása

Forrás (-s, --source) és célcímek (-d, --destination)

```

Hostname alapján      : localhost, vagy pl.: www.debian.org
IP alapján            : 127.0.0.1
IP tartomány alapján  : 192.168.1.0/24 vagy 192.168.1.0/255.255.255.0

```

Inverz módon is megadható forrás és cél, "-s !" localhost" formájában.

Protokollok megadása

-p tcp VAGY udp VAGY icmp formájában történik. Itt is alkalmazható negálás, azaz "-p ! TCP", minden csomagra illeszkedik a szabály amely nem TCP.

Portok, port tartományok megadása

A portok megadhatóak számmal, vagy névvel. (utóbbi esetben.: /etc/services)

```

ssh,smtp,http          : ssh , smtp és http portok beállítása
22,25,80               : u.a. mint az előző, számmal
--source-port VAGY --sport : a csomagok forrásportja
--destination-port VAGY --dport : a csomagok cél portjának meghatározása
22:80                  : port tartomány megadása, a két port közötti összes port
:22                     : 22 és az összes alacsonyabb port megadása
80:                     : 80 és az összes magasabb portszám beállítása
-m multiport --dports 22,25,80 : egy szabályban több CÉLport definiálása
-m multiport --sports 22,25,80 : egy szabályban több FORRÁSport definiálása

```

Célok (target) avagy műveletek

Az adott szabályban a rá illeszkedő csomagra vonatkozó művelet meghatározása azaz -j "target"

```

ACCEPT                : engedélyezés
DROP                  : eldobás válasz nélkül
REJECT                : eldobás, válasz küldéssel (pl. Destination Port Unreachable)
LOG                   : logolás

DNAT                  : destination NAT, célcímet módosítja. Port forwardhoz szokás alkalmazni
                      : a PREROUTING láncon a nat táblában.
                      --to-destination 192.168.1.253:80
                      : az adott IP-re és portra dobja át a forgalmat.

MASQUERADE            : maszkolás dinamikusan. Pl PPPoE kapcsolathoz ahol nincs fix publikus IP cím.
                      : POSTROUTING láncon a nat táblában szokás használni

SNAT                  : source NAT, a forráscímet módosítja a megadottra, maszkoláshoz használatos
                      : Általában több publikus IP-vel rendelkező hálózatban van értelme,

```



```
      : POSTROUTING láncon a nat táblában használható
--to-source 195.34.16.70
      : vagyis az adott kimenő forráscímmel látja el a csomagot.

REDIRECT      : lokális portra irányít át. Transzparens szűrésnél lehet értelme
      : PREROUTING láncon, pl 110-re jövő forgalmat átirányítja a 8110-re.

-j -vel akár saját láncba is irányítható a forgalom, amennyiben a lánc létezik és a csomag a szabályra illeszkedik.

-----
Folytatás később. Addig emésszék! :)
```

Fájltvitel SCP segítségével [top](#)

Az scp program segítségével a lokális gépről távoli gépre másolható adat (fájl vagy könyvtár), vagy távoli gépről a lokális fájlrendszerbe, illetve lehetőséget biztosít két távoli gép közötti adat átvitelére is. Az adatátvitel titkosított - ssh csatornán át történik, az autentikáció(k)hoz ismerni kell a távoli gép(ek)re történő bejutáshoz szükséges információkat.

```
scp user1@remotehost:/home/user1/test.txt /home/user2/
# távoli -> helyi
# user1 nevében a remotehost gép /home/user1 mappájából a test.txt másolása a helyi fájlrendszer adott mappájába.

scp /home/user2/test.txt user1@remotehost:/home/user1/
# helyi -> távoli
# lokális fájlrendszer user2/test.txt másolása a távoli rendszer adott mappájába user1 nevében.

scp /home/user2/dir user1@remotehost:/home/user1/dir
# helyi -> távoli (mappa)
# lokális fájlrendszer user2/dir mappájának másolása a távoli rendszer adott mappájába user1 nevében.

scp user1@remotehost1:/home/user1/test.txt user2@remotehost2:/home/user2/test.txt
# távoli -> távoli
# user1 nevében a remotehost1 adott mappájából a test.txt másolása a remotehost2 adott
# mappájába user2 nevében.

scp /home/user1/test1.txt /home/user1/test2.txt user2@remotehost2:/home/user2/
# helyi -> távoli (több fájl)
# helyi fájlrendszerből adott két db txt másolása a távoli rendszer adott mappájába user2 nevében.

scp -P 2264 /home/user1/test1.txt user2@remotehost2:/home/user2/
# helyi -> távoli (specifikus porton)
# adott helyi txt másolása user2 nevében a távoli gépre, 2264-es portot használva.
```

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](#)
[TLPD - 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](#)

Ha a Föld hozzánk szólna, biztos megkérné 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...