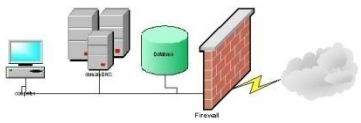
Linux Rendszerek adminisztrációja

- áttekintés és példák -

Varga Pál

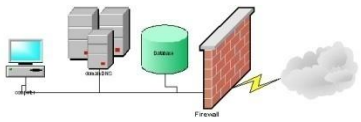
pvarga@tmit.bme.hu



Áttekintés

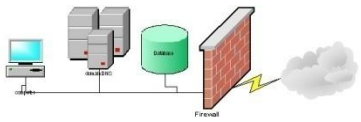
- Linux kernel általános felépítése
- File System
- Alapvető műveletek

- Shell
- Scripting
- Általános ismeretek a laborgyakorlatokhoz
- Rendszer-konfiguráció



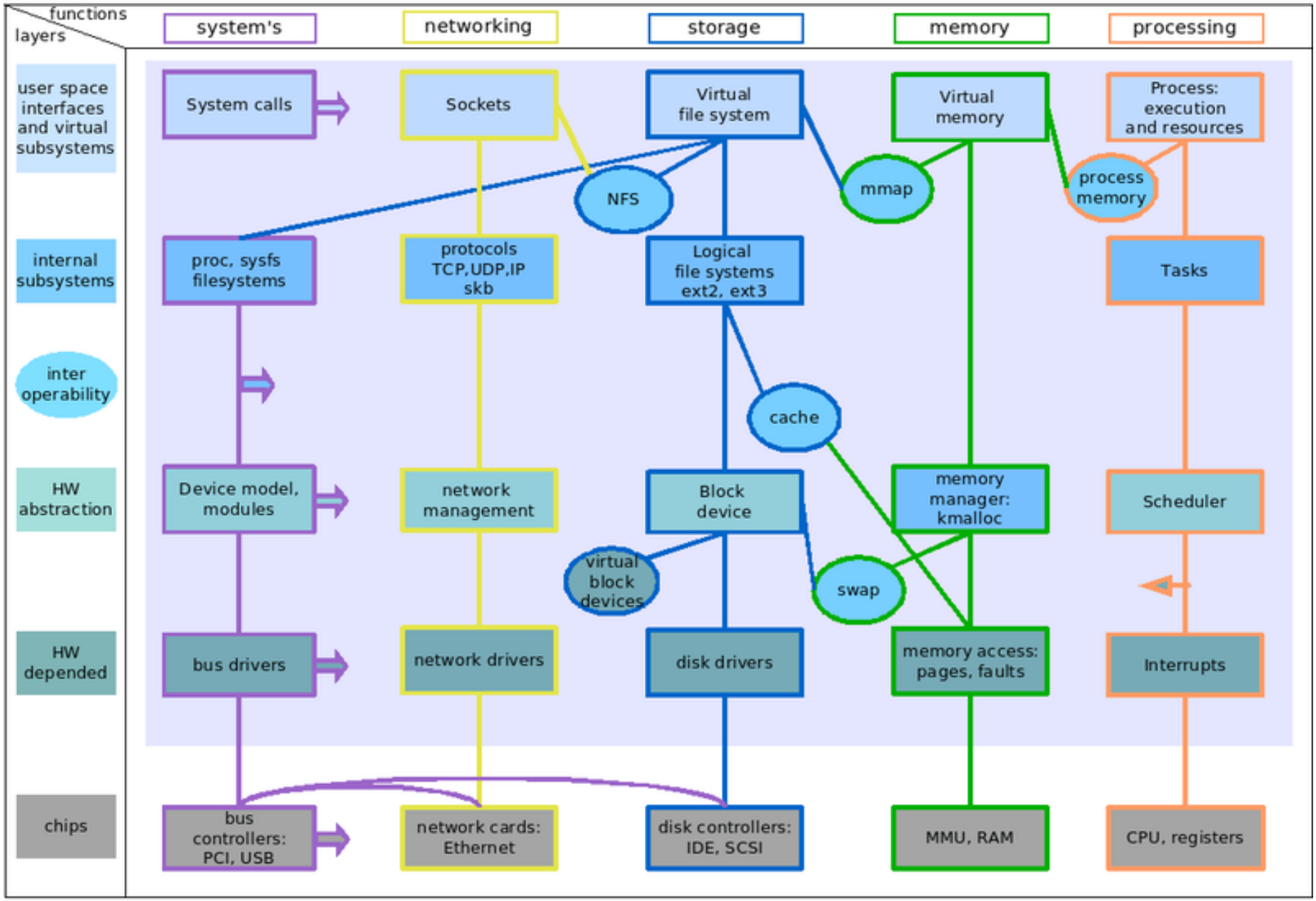
LINUX RENDSZEREK ADMINISZTRÁCIÓJA

I. Kernel
II. File System
III. Alapvető műveletek



By TechCityInc.com

Simplified Linux kernel diagram in form of a matrix map



Designed with OpenOffice.org by (cc) (by-nc-sa) Constantine Shulyupin, www.linuxdriver.co.il

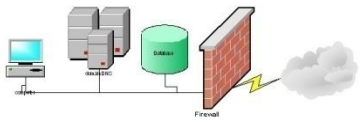
File System

- „device” vagy „partition” ami file-ok tárolására van formázva
- Sok formátum – a felhasználó előtt ugyanolyan megjelenés: könyvtárszerkezetben
- /proc/filesystems - pl. ext2 ext3 vfat NFS ntfs

- A hozzáféréshez *mount*-olni kell – így könyvtárként látszik/elérhető

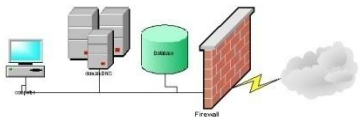
  mount -t vfat /dev/hda1 /windows/cdrive

- mount / umount /eject



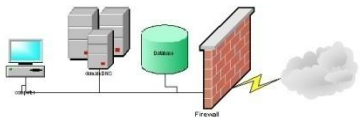
File System (cont)

- Applikáció is
  - Az adathozzáférés vezérlésére
  - A könyvtárak kezelésére és hozzáféréséhez
  - Más applikációk is használják a szolgáltatásait
    - állományokat és könyvtárakat készít
    - megnyit és módosít létezőket
    - törlés
    - hozzáférés-szabályozás



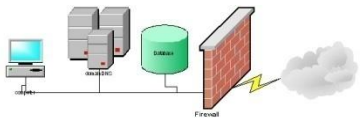
Néhány főbb alkönyvtár

/bin	common programs, shared by admins, users
/boot	startup files and kernel. Also GRUB data
/dev	references to all hardware as special files
/etc	important system configuration files. Sort of like Control Panel
/home	containes home directories for most users
/initrd	information contained for booting.. Don't mess with it!
/lib	library files, common files needed by many programs
/lost+found	files that were recovered after a system failure
/mnt	mount point for all external devices
/net	mount point for remote filesystems
/opt	extra third party software
/proc	info about system resources, man proc
/root	home dir. for the the admin
/tmp	temporary space
/usr	program, libraries, docs for most user programs
/var	other temporary files, such as logs, downloaded files, mail queue



Alapvető műveletek

- cd könyvtárak közötti navigációhoz
- mkdir könyvtár létrehozása
- rmdir könyvtár törlése
- ls könyvtár tartalma
- echo $PATH végrehajtható parancsok helyei
  pl.: /usr/local/bin/:/usr/bin:/bin/
- export környezeti változó beállítása
  pl. PATH=$PATH:/new/directory/path
- man help a parancsok használatához
- history eddig használt parancsok listája
- & parancs után írva: fusson a háttérben *ls & -> [1] 23142*
- fg futó job visszahozása előtérbe *fg 23142*

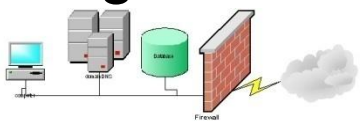


Szöveges manipuláció

- cat teljes file szövegének összeállítása (kiírás)
- tac utolsó sor legelöl (hasznos pl. log file esetén)
- head a file kezdő sorai (default:10)
- tail a file záró sorai (default:10)
- more szöveg kiírása oldalanként a terminálra
- less szöveg kiírása: a felhasználó mászkálhat benne

regxp – regular expressions

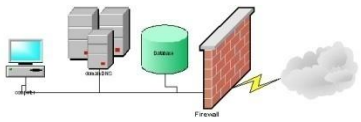
- grep szövegben keresés szabályok szerint
- (g)awk szövegben keresés/manipuláció
- sed szöveg egyszeri futás alatti szerkesztése
- vi klasszik szövegszerkesztő
- emacs legendás szövegszerkesztő



Accounting

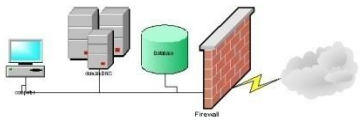
- Multiuser operációs rendszer
- Account:
  - felhasználónév azonosítja,
  - jelszó védi
- Password file:

  username:password:uid:gid:gecos:homedir:shell
- Kétféle account:
  - Root
  - User



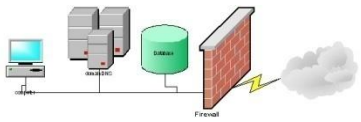
Accounting (cont)

- su — root átvált az adott userre/ré
- adduser — felhasználó hozzáadása
- passwd — jelszó megváltoztatása
- userdel — felhasználói account törlése
- /etc/passwd — felhasználói account információk
- /etc/shadow — felhasználói jelszó titkosítva; valamint account- és jelszó-lejárati infók



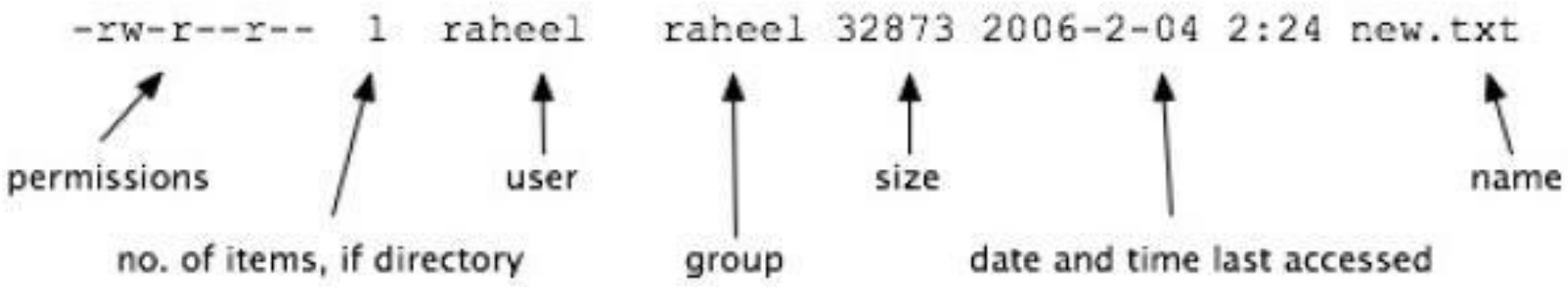
Authorization: ownership & access

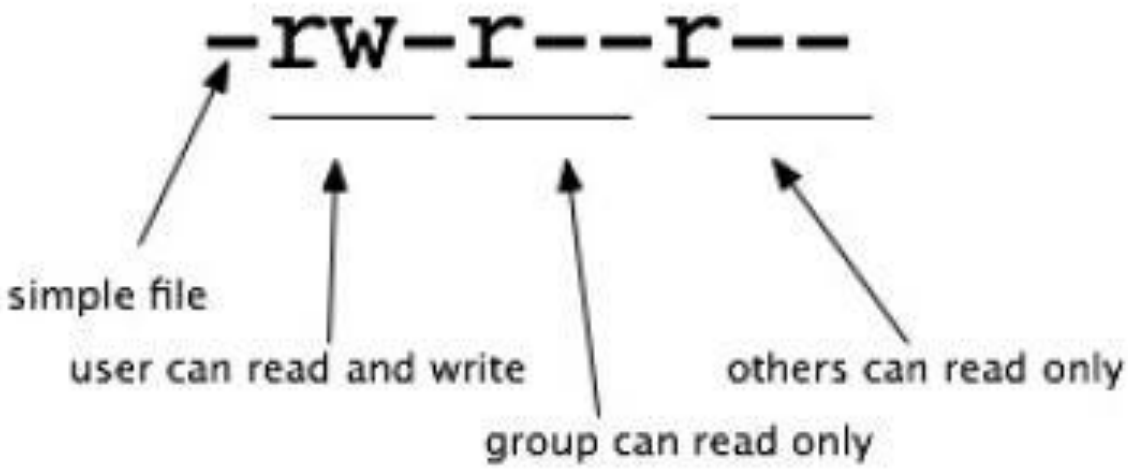
- Hozzáférés
  - Multi-user környezet!
  - Felhasználók nem férnek hozzá egymás file-jaihoz
  - Speciális file-okhoz csak a root fér hozzá
- Csoportok – groups
  - a felhasználók több csoporthoz tartozhatnak
  - könnyebb/rugalmasabb hozzáférés-kezelés
- Hozzáférés: felhasználók / csoportok / egyéb

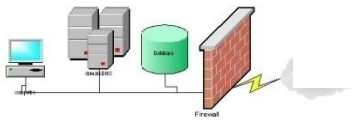


Hozzáférés - példa

- ls – l - parancs példa kimenete





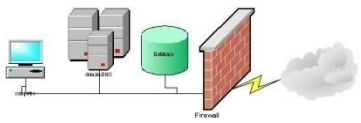


Ownership változtatás

- Ownership (birtoklás) változtatás: chown
  chown username file_or_dir

- Csoport-birtoklás változtatás: chgrp
  chgrp groupname file_or_dir

- Kombinált csoport és felhasználónév:
  chgrp name.name file_or_dir

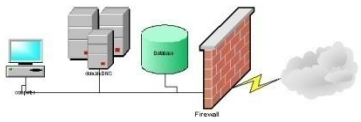


Hozzáférés változtatás

- *chmod* parancs

- *r, w, x: read, write, execute*
- *Root: megváltoztathatja bármely file/directory hozzáférés-szabályait*
- *A root mellett csak a birtokos (user) tudja változtatni*

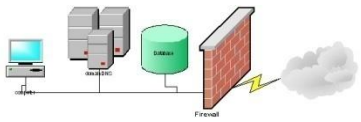
pl. chmod 755 myfile



SCRIPTING IN A (NUT)SHELL

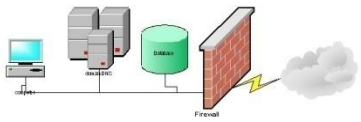
Bash

Ajánlott gyakorlatok:
http://linuxconfig.org/bash-scripting-tutorial



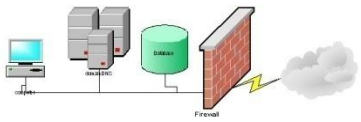
bash - „Bourne-again shell”

- a shell mindig elérhető (sh, bash, ksh,..)
- rendszergazdai script-ek: shell-ben
- összetettebb script-nyelvek (pl.)
  - Perl
  - Python
  - Ruby
- a script (v. interpreter-)nyelvek jellegzetességei
  - programkódja akár soronként is értelmezhető
  - minden futási időben értékelődik ki
  - string paramétert is képes értelmezni parancsként
  - típustalan változókezelés



processzek kommunikációs csatornái

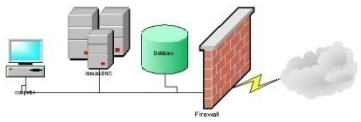
- Standard I/O, minden programnak
  - 0 – STDIN
  - 1 – STDOUT
  - 2 – STDERR
- Átirányítás
  - `tail file` #file→STDOUT
  - `tail file 2>&1` #STDERR→STDOUT
  - `tail file.o > file.n` #file.o→STDOUT→file.n
  - `tail file.o 2> file.n` #file.o→STDOUT, STDERR→file.n
  - grep ‘From’ < /etc/mail/laboruser #grep input: /etc/mail/laboruser



Csövezés: pipes

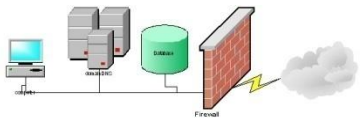
- `cat /etc/mail/laboruser | grep 'From'`
  `#cat STDOUT-ját a grep STDIN-be`
- `cat /etc/passwd | grep ‘:x:’ | sort`
  `#cat STDUT-ját a grep-en átszűrve a sort-ba`

- Láncolás pro és kontra:
  - Formázatlan bináris adatátadás!
  - Gyors, de strukturált adatot nem kezel
  - Strukturált adatot szöveges formába kell alakítani (valamilyen módon), majd a fogadó oldalon sorokra, majd azon belül mezőkre bontva feldolgozni
  - Erre használható programok: cut, awk, sed



bash: változók és nevek

- A változónevekre nincs külön jelölés
  - etcdir='/etc' # de ne legyenek space-ek a = körül!! (command)
- $-előtét, ha a változó értékét hivatkozzuk
  - echo $etcdir
- Egyéb szövegtől elválasztható {} jelekkel:
  - echo ”Saved ${rev}th version of file”

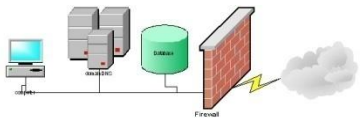


bash: tömbök

```
ARRAYVAR=(ERTEK1 Ertek2 "Ertek3");        #értékadás #felsorolással
OTHERARRAY[5]= "Ertek4";        #értékadás konkrét indexre, ritka tömb

echo "elemszam: ${#ARRAYVAR[@]} ";
echo "elso elem $ARRAYVAR";

   #for ciklus az összes elemre, figyeljünk a ;-k helyére!
   #ritka tömbnél így nem működik!
for ((i=0;i<${#ARRAYVAR[@]};i++));
do
   echo "${ARRAYVAR[$i]}";
done;
```

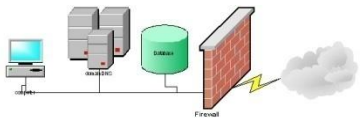



Az 'aposztrófok'

(legyen: mysport=chess)

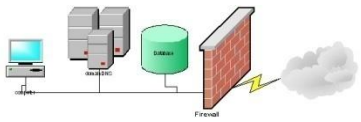
- A dupla-idézőjel kifejtő funkciójú
  - echo ”I play ${mysport}.” # I play chess.
- a szimpla nem fejti ki a változót
  - echo 'I play ${mysport}.' # I play ${mysport}.
- a `backtickek`
  - közé írt parancssor standard outputját a shell végrehajtja és behelyettesíti
  - echo ”There are `wc –l < /etc/passwd` lines in the passwd file.”

    # There are 28 lines in the passwd file.



Speciális változók, paraméterek

- Bemenő paraméterek
  - `$1, $2` … - bemenő paraméterek
  - `$#` - paraméterek száma
  - `$@` - paraméterek tömbje
- Előző parancs
  - `$?` – legutóbbi futtatott parancs visszatérési értéke
  - `$!` – legutóbbi háttérben indított folyamat PID-je
  - `$$` - éppen futó shell PID-je
- Mezőelválasztó karakter
  - `$IFS` – ennek az értéke alapján automatikusan darabolja a stringeket külön paraméterekké
  - Alapból az IFS értéke minden whitespace `" " "\n" "\t"`



Néhány általános szűrő

- cut - sorokat mezőnként szétválasztja
- sort - sorbarakja a sorokat

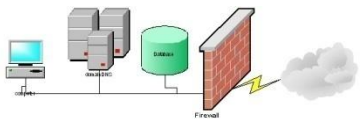
  -f : case sensitive sort

  -k : key column for sorting

  -n : fields are integer numbers

  -r : reverse sort order

  -u: otput unique records only
- unique - az egyedi sorokat írja ki
- wc - word count (-l, -w, -c vajon mi…)
- tee - kétfelé küldi a bemenetét



bash scripting

((a helloworld file tartalma:))

```
#!/bin/bash
echo ”Hello, world!”
```

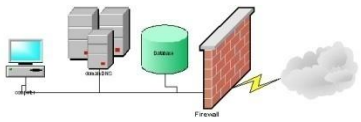
((beírjuk a shell-be:))

chmod +x helloworld

./helloworld

((kiírja a shell:))

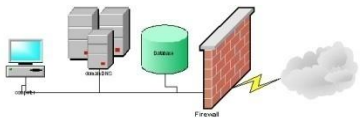
Hello, world!



Elágazások

```
#!/bin/bash
if [ "foo" = "foo" ];
then
    echo expression evaluated as true
else
    echo expression evaluated as false
fi
```

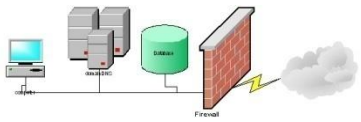
http://tldp.org/HOWTO/Bash-Prog-Intro-HOWTO.html



Ciklusok

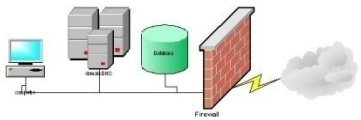
```
#!/bin/bash
for i in $( ls ); do
    echo item: $i
done
```

```
#!/bin/bash
COUNTER=0
while [ $COUNTER -lt 10 ]; do
    echo The counter is $COUNTER
    let COUNTER=COUNTER+1
done
```

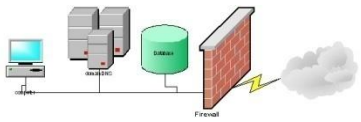



bash scripting hozzáállás: egy hasznos módszer

1. Fejlődjön a script lépésről lépésre, mint egy **pipeline**, végig a **command line**-ban
2. Mindent küldjünk a STDOUT-ra, hogy lássuk, minden rendben megy-e
3. A lépéseknél használjuk a shell command history-t és így szerkesszük tovább a parancsokat
4. Mindaddig, míg a kimenet nem tökéletes, lehet finomítani, semmi undo-ra nincs szükség
5. Amint megfelelő a kimenet, hajtsuk végre a konkrét parancsokat, így ellenőrízve, hogy tényleg rendben
6. Használjuk az **fc**-t a munka befejezésekor, letisztázásra, mentésre



ÁLTALÁNOS LABORISMERETEK

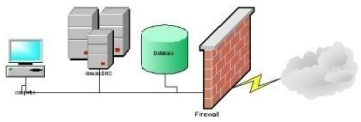


Általános laborismeretek

- VMWARE - Váltás a gazdagépre: **Ctrl+Alt**

- A feladatok mellett: hint
  - Milyen parancsot használjunk:
    man it!
    Google it!
  - Parancsok opciói/kapcsolói/szintaxisa: helyben kitalálandó!

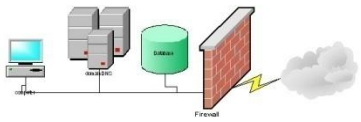
- Hasznos felkészülés
  - Alap Linux a HSZK-ban / otthon



Labor bemelegítés

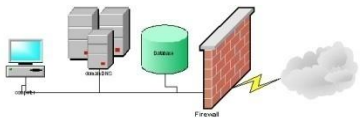
- Körbenézés: **mc**

- Hálózati kapcsolatok: **netstat**
  - Kapcsolók..!
- Tűzfalszabályok: **iptables**
  - Kapcsolók..!



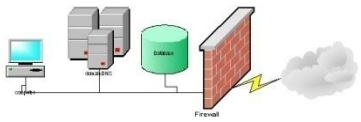
IRÜ Linux labor átnézendő témakörök

- Ismételni:
  - TCP/UDP,
  - Alap portok és szolgáltatások (ssh, http, https, ftp),
  - hálózati alapok (IP számítás, subnetek)
- Új:
  - sudo, netstat, iptables, ssh, programok indítása, linux alap könyvtárak (etc, etc/init.d, var/log, usr, bin, sbin), jogosultságok, chmod, futtatás, telepítés, csomagkezelő, mysql, alap sql parancsok, mysql belépés, mysql adatbázis betöltés, http, https, netstat, apache mappák, htpasswd, htaccess, fájl és mappaműveletek (létrehozás, átnevezés, másolás, törlés), nano, vim, vi, who, wc, cat, ls , who, whoami, uniq, echo, cut, grep, sort, df, date, man, passwd, ifconfig, ip addr, ip route, mc



Labor: felhasználókezelés

- visudo – rendszergazda jogokhoz
- felhasználói jogok:
  - /etc/passwd
- file-hozzáférés
  - chmod
  - chown
  - chgrp



Átnézendő anyagok

- Alap linuxos parancsokról egy jó összefoglaló:

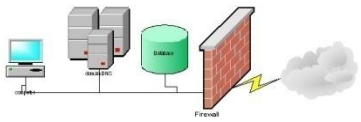
http://www.eotvos.u-szeged.hu/~sgeolte/linux.pdf

- TCP/UDP portok összefoglalója:

http://hu.wikipedia.org/wiki/TCP_%C3%A9s_UDP_portsz%C3%A1mok_list%C3%A1ja

- IP számítás, subnetting:

http://www.fuvesi.com/halozatok/alhalozat.html



Linux fájlműveletek és hálózati beállítások (kiegészítés)

- Az alap parancsokról és műveletekről egy rövidebb írás:

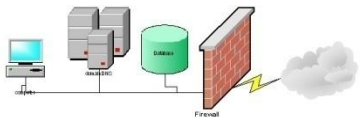
http://hup.hu/old/loptk/x330.htm

- Hálózati interfész beállítások listázása:

  ipconfig –a

- Jó általános leírás Linuxhoz és hálózatokhoz:

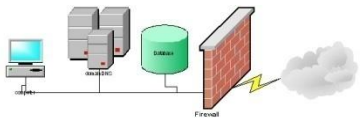
http://www.szabilinux.hu/ip/



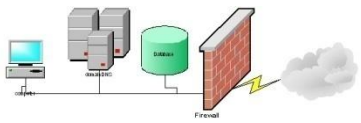
Apache2 (kiegészítés)

- Az Apache HTTP Server (röviden Apache) nyílt forráskódú webkiszolgáló alkalmazás, szabad szoftver, mely kulcsfontosságú szerepet játszott a World Wide Web elterjedésében.

- A projekt célja olyan webszerver program létrehozása, karbantartása, és fejlesztése, amely megfelel a gyorsan változó Internet követelményeinek, biztonságos, üzleti, vállalati felhasználásra is megfelelő és szabadon használható.



SCRIPTING BEST PRACTICES



Mitől lesz még jobb a script?

- Ha rosszul használják: usage v. –help hint!
- A bemeneti változókat ellenőrzi (sanity check)
- Használ visszatérési értéket: 0 – siker; egyékként más
- Változók, rutinok elnevezése következetes;
  - nem hosszú, és szeparált (_) nevek
- Felismerhető a stílus projekten v. csapaton belül
- Comment-block-kal kezdődik, alapadatok itt vannak (név, utolsó módosítás, milyen bővítményt használ, stb.)
- Comment for programmers, not dummies
- Figyelmes futtatás: bash –x ; perl –w …