

<http://explainshell.com/>

Mentsétek el

IRÜ Linux labor

Melyik feladatot kell megcsinálni?

https://github.com/thereallacac/iru_whats_my_task

Eddigi tapasztalatok szerint jó a kód

<https://repl.it/FpL6/0>

1. A rendszer indítása után...

Iptables: <http://www.howtogeek.com/177621/the-beginners-guide-to-iptables-the-linux-firewall/>

1.4 feladat: A céges policy-k szerint a távoli bejelentkezést biztosító szolgáltatásoknak (ssh) csak a lokális hálózatról kell elérhetőnek lenniük. Módosítsa a tűzfalszabályokat ennek megfelelően!

```
sudo iptables -I INPUT -p tcp --dport 22 -s 192.168.150.0/8 -j ACCEPT
```

```
sudo iptables -A INPUT -p tcp --dport 22 -j REJECT
```

```
sudo /sbin/iptables-save
```

(Nem tudom mennyire kőbevésett az ip cím)(internetről 192.168.0.0/16)

(forrás: <http://askubuntu.com/questions/115940/how-can-i-setup-ssh-so-that-it-is-restricted-to-my-local-network>)

Magyarázat: első helyre -I (insert) a szabályt ami elfogadja a lokális hálózatról érkező kéréseket (22-es port a destination port, ide jönnek a tcpk)

A lista végére meg az elutasító szabályt rakjuk be. Azért fog működni, mert fentről lefele halad végig a szabályok listáján, és ha egyezést talál, azt alkalmazza.

A végén el kell menteni a szabályokat!

1.5. feladat: A céges policy-k szerint a gépnek nem szabad válaszolnia a pingelésre. Módosítsa ennek megfelelően a tűzfal szabályait!

```
sudo iptables --policy INPUT DROP
```

Vagy (még jobb)

```
iptables -A INPUT -p ICMP -j DROP
```

```
sudo /sbin/iptables-save
```

1.6. feladat: Adjon hozzá a rendszerhez egy új felhasználót, akinek a neve legyen "mekkelek", jelszava pedig az Ön NEPTUN kódja! Nézze meg, milyen új bejegyzés született az /etc/passwd fájlban!

```
sudo adduser mekkelek
```

```
sudo cat /etc/passwd
```

1.7. feladat: Tegye lehetővé az új felhasználó számára, hogy rendszergazda jogokkal futtathasson minden programot!

```
sudo visudo
```

Hozzá kell fűzni a fájlhoz, hogy

```
mekkelek ALL=(ALL:ALL) ALL
```

Attól függően, milyen szövegszerkesztővel történt, menteni és kilépni

1.8. feladat: A biztonság érdekében tiltsa le a root felhasználó SSH-n történő bejelentkezésének jogát!

Be kell menni az ssh mappába

```
cd /etc/ssh/
```

Szerkeszteni kell az sshd_config file-t (nem összekeverni az ssh_configgal)

```
sudo nano sshd_config
```

Meg kell keresni a *#PermitRootLogin yes* sort (Keresni a ctrl+w-val lehet)

Általában ki van kommentelve, át kell írni *PermitRootLogin no*-ra.

Menteni kell, majd

```
sudo service sshd restart
```

vagy

```
systemctl restart sshd
```

su - mekkelek (ha nem lennénk ott)

ssh-keygen

Utána elvileg amit a feladat mond

<http://realprogrammers.com/how-to/set-up-an-ssh-tunnel-with-putty.html>

1.10. feladat: Telepítse a MySQL-t a guest gépre! Az adminisztrátor jelszava az egyszerűség kedvéért legyen "root"! Természetesen éles környezetben nehezen kitalálható jelszavakat kell használni. Nézze meg, hogy a szerver melyik verzióját sikerült installálnia!

```
sudo apt-get install mysql-server
```

```
sudo mysql_install_db
```

```
cat /etc/mysql/my.cnf | grep port
```

```
cat /etc/mysql/my.cnf | grep -i datadir
```

Láthatjuk, hogy a mysql-hez a 3306 os port tartozik

```
netstat -tln
```

Így láthatjuk a (-t) tcp re szűrt (-l listening) (-p program) (-n portokat numerikusan kiírva)

```
mysql -V
```

1.14. feladat: A /root/students.sql fájlban található exportált adatbázist töltse fel a MySQL szerverre.s

```
mysql -u root -p
```

```
SHOW DATABASES;
```

```
CREATE DATABASE mydb;
```

```
use mydb;
```

```
mysql> source /root/students.sql
```

Kívülről is lehet rögtön:

```
mysql -u root -p root mydb < /root/students.sql
```

```
SHOW DATABASES;
```

Ahhoz, hogy egy adatbázisra átváltsunk

```
use otherdb;
```

A szorgalmat csináljátok meg!

♠ **1.16. feladat (opcionális):** Töltsön fel táblánként legalább egy-egy új rekordot az adatbázisba, amelyek megfelelnek a sémáknak, és a táblák összefüggéseinek is. Az új hallgató neve legyen "Mekk Elek" az Ön NEPTUN kódjával. Mekk Elek 1974. április 1-én született.

Ide írom kb. Mit kell csinálni, de nem emlékszem pontosan a mezők/táblák neveire, így azt lehet, hogy módosítani kell. A dátum formátumát is csekkold, hogy ugyanolyan legyen, mint a többi rekordban.

```
INSERT INTO students (name, dateofbirth, neptun) VALUES ("Mekk Elek", "1974-04-01", "NEPTUN")
```

```
INSERT INTO courses (name, credit) VALUES ("találj ki egy tárgynevet", 3)
```

```
INSERT INTO results (studentid, courseid, result) VALUES (10, 1, 2)
```

Utolsónál bármilyen random id jó, de célszerű létezőt választani.

2. Az Apache2 webkiszolgáló

♠ **2.1. feladat:** Telepítse az apache2 csomagot a beépített csomagkezelőn keresztül!

```
sudo apt-cache search apache2
```

```
sudo apt-get install apache2
```

♠ **2.3. feladat:** Az Apache 2 dokumentációja és az alap sablon alapján állítson be egy virtuális kiszolgálót, mely az irulabor.vmware domén névre töltődik be. A kiszolgáló által visszaküldött html oldalak kódja megtalálható és letölthető a címen. Ne feledje el aktiválni az elkészült konfigurációt! Az /etc/hosts fájlban készítse el egy bejegyzést, hogy a virtuális gép ismerje a irulabor.vmware nevet

Segédlet 9-10 oldal !!!!

Vikwiki

a megadott oldalról letöltjük a zipet, kicsomagoljuk és az iru-master mappát bemásoljuk a /var/www mappába

etc/hosts fájlt módosítjuk:

```
127.0.0.1 localhost irulabor.vmware
```

```
127.0.1.1 irulabor.irulabor irulabor
```

apache 2-t újraindítjuk:

```
sudo /etc/init.d/apache2 restart
```

Otthon:

<http://askubuntu.com/questions/52147/how-can-i-access-apache-on-virtualbox-guest-from-host>

etc/apache2/sites-available conf fájlban a

DocumentRootot át kell írni html-ről a mappára, amiben az irus dolgok vannak pl.

```
DocumentRoot /var/www/irulabor
```

Minden config fájl módosítás után az apache-t reloadolni kell:

```
sudo service apache2 reload
```

♠ **2.4. feladat:** A fizikai gépen (tehát nem a virtuális linuxon!) tölts be a böngészőben a `http://irulabor.vmware/vedett` címet. Mint láthatja, a védett mappában található fájlok jelenleg elérhetőek a sudo-művelettel is. A feladat az, hogy a védett mappa tartalma csak a virtuális gépről legyen elérhető a webszerverbe épített IP korlátozás segítségével.

irulabor config fájlba: (sites-available mappa)

```
<Directory /var/www/irulabor/vedett>
    Options Indexes FollowSymLinks MultiViews
    AllowOverride None
    Order deny,allow
    deny from all
    Allow from 127.0.0.0/8
</Directory>
```

♠ **2.6. feladat:** A feladat az, hogy a nagyonvédett mappa tartalma csak felhasználónév és jelszó segítségével legyen elérhető a webszerverbe épített korlátozás segítségével. A felhasználók a rendszer beépített felhasználói. Kísérletet tegyen meg belépni a fenti címre.

Ha jól emlékszem telepíteni kellett a pwauth-ot (de az is lehet, hogy alapból fenn van):

```
sudo apt-get install pwauth
```

Engedélyezni kell az authnz_external apache modult:

```
sudo a2enmod authnz_external
```

Config fájlba

```
<IfModule mod_authnz_external.c>
    AddExternalAuth pwauth /usr/sbin/pwauth
```

```

        SetExternalAuthMethod pwauth pipe
    </IfModule>

    <Directory /var/www/irulabor/nagyonvedett>
        AuthType Basic
        AuthName "nagyonvedett"
        AuthBasicProvider external
        AuthExternal pwauth
        Require valid-user
    </Directory>

```

Ezután amikor böngészőben megnyitjuk a nagyonvedett-et, akkor a laboruser/laboruser kombóval is működnie kéne.

♠ **2.8. feladat: Az alapértelmezett webkiszolgáló konfigurációban ennek a fájlnek a használata tiltva van. Milyen konfigurációs direktívával tudjuk mégis engedélyezni, és ezt hol kell megadni? A kérdés a nyilvános mappára vonatkozik, csak ott akarjuk engedélyezni a .htaccess használatát.**

```

<Directory /var/www/irulabor/nyilvanos>
    Options Indexes FollowSymLinks MultiViews
    AllowOverride All
    Order allow,deny
    Allow from all
</Directory>

```

♠ **2.9. feladat: A nyilvános mappára szeretné bekapcsolni a webszerver automatikus listázó funkcióját. Mit és milyen formátumban kell ehhez beírni a .htaccess fájlba?**

```
Options +Indexes
```

3. Linux héjprogramozás

3.1.A es B feladat: Írassa ki a számítógép processzorának vendor id-jét! Az elkészített bash héjprogram (/home/laboruser/bin/3 1.sh) csak a vendor id-t írja ki a standard kimenetre, semmi mást!

```

#!/usr/bin/env bash
lscpu | grep -oP "Vendor ID: *\K\w*"
/////
#!/usr/bin/env bash
cat /proc/cpuinfo | grep vendor_id | cut -d ':' -f 2 | cut -d ' ' -f 2
2

```

For reference:

-o kapcsoló: only matching, csak az illeszkedő részt írja ki

-P: perl regex

\K Keep the stuff left of the \K, don't include it in \$&

\w Match a "word" character

3.1.C es D feladat: Írassa ki a számítógép processzorának frekvenciáját! Az elkészített bash héjprogram (/home/laboruser/bin/3 1.sh) csak a processzor névleges frekvenciáját (mértékegység nélkül) írja ki a standard kimenetre, semmi mást!

```
#!/usr/bin/env bash
asd=$(lscpu | grep "MHz");
IFS=":" read -a cpuinfo <<<"${asd}";
echo ${cpuinfo[1]};
//////////
#!/usr/bin/env bash
lscpu | grep -oP "CPU max MHz: +\K([0-9]|,)*"
```

♠ 3.2. feladat: írjon egy bash scriptet (/home/laboruser/bin/3 2.sh), amely megszámolja a bemenetként kapott fájlban az üres sorokat!

```
#!/usr/bin/env bash
LOL=$(grep -c "^[ ]");
echo "$LOL";
```

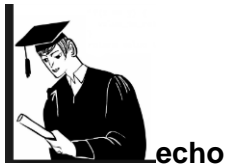
♠ 3.3.A es C feladat: írjon egy bash scriptet 3_3.sh néven, amely paraméterként egy pozitív egész számot kapva kilistázza a standard bemenete kapott öt oszlopból álló adatfolyam azon sorait, amelyek a parancssori paraméterként megadott számmal osztható számmal kezdődnek, és teszi mindezt úgy, hogy közben felcseréli a második és az ötödik oszlopok tartalmát.

Egyszerűen:

```
#!/usr/bin/env bash

while read -a line;
do
    if [ $((line[0] % $1)) -eq 0 ]
    then
        echo "${line[0]} ${line[4]} ${line[2]} ${line[3]}
${line[2]}"
    fi
done
```

//////////



```
#!/usr/bin/env bash

while read -a line;
do
    if [ $((line[0] % $1)) -eq 0 ]
    then
        SV=${line[4]};
        line[4]=${line[1]};
        line[1]=$SV;
        echo "${line[@]}"
    fi
done
```

♠ **3.3.B es D feladat: írjon egy bash scriptet 3 3.sh néven, amely paraméterként egy pozitív egész számot kapva kilistázza a standard bemenete kapott öt oszlopból álló adatfolyam azon sorait, amelyek a parancssori paraméterként megadott számmal osztható számmal kezdődnek, és teszi mindezt úgy, hogy közben összefűzi a negyedik és az ötödik oszlopok tartalmát.**

```
#!/usr/bin/env bash
while read -a line;
do
    if [ $((line[0] % $1)) -eq 0 ]
    then
        echo "${line[0]} ${line[1]} ${line[2]}
${line[3]}${line[4]}"
    fi
done

/////
#!/usr/bin/env bash
while IFS= read -r line;
do
    array=(${line// / })
    if [ $((array[0] % $1)) -eq 0 ]
    then
        echo "${array[0]} ${array[1]} ${array[2]}
${array[3]}${array[4]}"
    fi
done
```


- ♠ **3.4. feladat: írjon egy bash héjprogramot (/home/laboruser/bin/3_4.sh), amely futtatáskor soronként kiírja**
- 1. a futtató felhasználó nevét,**
 - 2. az aktuális dátumot (éééé. hh. nn. formátumban),**
 - 3. a bejelentkezett felhasználókat (mindegyiket egyszer és külön sorba) és**
 - 4. a felhasználó által épp futtatott 3 4.sh script pid-jét.**

```
#!/usr/bin/env bash
whoami
date +"%Y. %m. %d."
users | tr " " "\n" | sort -u
echo $$
```

A users szóközzel elválasztva írja ki a felhasználókat, a tr kicseréli a szóközőket newline-ra, a sort -u csak az egyedi felhasználókat tartja meg

- ♠ **3.6. feladat: írjon egy bash scriptet 3 6.sh néven, amely parancssori paraméterként megkapja két létező fájl nevét, és megállapítja, hogy hány különböző sor található bennük. A script futási eredményeként két sor jelenik meg a szabványos kimeneten: az első sor tartalmazza azoknak a soroknak a számát, amelyek benne vannak az elsőként megadott fájlban, de nincsenek benne a másodikban, a második sor pedig ugyanezt az adatot tartalmazza vica-versa. Diff**

```
#!/usr/bin/env bash
VAR1=$(diff --changed-group-format='%<' --unchanged-group-format=''
$1 $2 | wc -l)
VAR2=$(diff --changed-group-format='%<' --unchanged-group-format=''
$2 $1 | wc -l)
echo $VAR1;
echo $VAR2;
```

- ♠ **3.7.A feladat: írjon egy bash scriptet 3 7.sh néven, amely összegzi az egyes csoportokba tartozó számok szorzatának az összegét.**

```
#!/usr/bin/env bash
lables=()
sums=()
check=0

IFS=', ' read -a line
```

```

lables+=${line[0]}
sums+=((${line[1]}*${line[2]}))

while IFS=',' read -a line
do
i=0
for c in "${lables[@]}"
do
if [ "$c" = "${line[0]}" ]
then

check=1

sums[i]=$((${sums[i]}+${line[1]}*${line[2]}))

fi
i=$((i+1))
done

if [ "$check" -eq 0 ]
then
lables+=(${line[0]})
sums+=((${line[1]}*${line[2]}))

fi

check=0

done

j=0
for c in "${lables[@]}"
do
echo "${c}: ${sums[j]} "
j=$((j+1))
done

D-feladat:
#!/usr/bin/env bash
lables=()
sums=()
check=0
ok=1
IFS=',' read -a line
lables+=${line[0]}
sums+=(($ok))

```

```

while IFS=',' read -a line
do
i=0
for c in "${lables[@]}"
do
if [ "$c" = "${line[0]}" ]
then

check=1

sums[i]=$((${sums[i]}+$ok))
fi
i=$((i+1))
done

if [ "$check" -eq 0 ]
then
lables+=("${line[0]})
sums+=(($ok))

fi

check=0

done

j=0
for c in "${lables[@]}"
do
echo "${c}: ${sums[j]} "
j=$((j+1))
done

```

♠ **3.7.B feladat:** Írjon egy bash scriptet 3_7.sh néven, amely kiírja minden csoport esetében a harmadik oszlopban található legkisebb elemet. A kimenet formátuma kövesse az alábbi sémát:

```

#!/usr/bin/env bash
lables=()
min=()

db=0

IFS=',' read -a line
lables[$db]="${line[0]}"
min[$db]="${line[2]}"
db=$((db+1))

```

```

while IFS=',' read -a line;
do
    benn=0
    hanyadik=0
    i=0
    for c in "${labels[@]}"
    do
        if [ "$c" == "${line[0]}" ]
        then
            benn=1
hanyadik=$i
            fi
            i=$((i+1))
        done
        if [ $benn -eq 1 ]
        then
            if [ ${min[$hanyadik]} -gt ${line[2]} ];
            then
                min[$hanyadik]=${line[2]}
            fi
        else
            labels[$db]=${line[0]}
            min[$db]=${line[2]}
            db=$((db+1))
        fi
    done
    j=0
    for c in "${labels[@]}"
    do
        for d in "${c[@]}"
        do
            echo "${d}: ${min[$j]}"
            j=$((j+1))
        done
    done
done

```

MONIT

<http://unix.stackexchange.com/questions/128953/how-to-display-top-results-sorted-by-memory-usage-in-real-time>

♠ 4.2. feladat: Töltse le a virtuális gépen található operációs rendszer verziójának megfelelő monit programot a /root/bin könyvtárba (A labor során használja a tantárgy honlapján megtalálható verziót!). A program elérési útvonala legyen /root/bin/monit/bin! Indítsa el, majd jelentkezzen be a program webes felületén a admin/monit alapértelmezett felhasználónév/jelszó kombinációval. Ezen adatokat a feladatok során ne változtassa meg! Ügyeljen arra, hogy a monit parancssori paraméterek megadása nélkül is indítható legyen!
<https://mmonit.com/monit/>

A monitot magát és egyéb dolgait innen kell majd letölteni:

<http://www.medialab.bme.hu/course/VITMA314::VITMAC02>

A 4.2 feladat szerint kell telepíteni, aztán a monit/conf/monitrc fájlt be kell másolni olyan helyre, ahol a monit számíthat rá, pl /etc/-be:

```
cp /root/bin/monit/conf/monitrc /etc/monitrc
```

Ezután a /etc/monitrc fájlt kell szerkeszteni, a módosítás után a monitot reloadolni kell, hogy érvényesüljenek a változtatások.

```
sudo /root/bin/monit/bin/monit reload
```

♠ 4.3. feladat: Módosítsa a monit konfigurációs állományát, hogy jelezzon, ha a memóriahasználat túllépi a 25%-ot. A problémáról a program küldjön e-mailt a laboruser@irulabor.irulabor címre. Az e-mail tartalmazza az alapértelmezett mezőkön kívül az Ön NEPTUN kódját a Neptun: NEPTUN formátumban rögtön a hostot megadó sor után! Ellenőrizze, hogy az e-mail megérkezett!

A monitrc fájlban módosítani kell pár sort (meg a komment hashtaget kieszedni):

```
set mailserver localhost
```

```
set mail-format {
  from:      monit@$HOST
  subject:   monit alert -- $EVENT $SERVICE
  message:   $EVENT Service $SERVICE
              Date:          $DATE
              Action:         $ACTION
              Host:           $HOST
              Neptun:         Ide kell a neptunod
              Description:    $DESCRIPTION
}
```

Your faithful employee,

```

        Monit
    }

set alert laboruser@irulabor.irulabor

check system $HOST
#   if loadavg (1min) > 4 then alert
#   if loadavg (5min) > 2 then alert
#   if cpu usage > 95% for 10 cycles then alert
    if memory usage > 25% then alert
#   if swap usage > 25% then alert

```

Ha minden jól ment, akkor reload után a megadott paranccsal látni kéne az e-mailt.

4.5. feladat: Módosítsa a monit konfigurációs állományát úgy, hogy a monitorozó program jelezzen, ha olyan futó alkalmazást észlel, amely legalább 33%-ig leterheli a CPU-t. A problémáról a laboruser@irulabor.irulabor e-mail címre küldött levél tartalmazza, az alapértelmezett mezőkön kívül, az Ön NEPTUN kódját a 1 Neptun: NEPTUN formátumban rögtön a munkaállomást megadó sor után! Ellenőrizze, hogy az email megérkezett!

```

check process any matching `.*`
    if cpu > 33% then alert

```

♠ 4.6. feladat: Módosítsa a monit konfigurációs állományát, hogy a monitorozó program termináljon minden olyan futó alkalmazást, amely legalább 33%-ig leterheli a CPU-t. Tesztelje le, hogy működik-e az Ön által írt feltétel!

Forrás: <http://serverfault.com/questions/608094/using-monit-to-kill-the-right-process-without-knowing-its-pid>

Kell írunk egy scriptet hozzá, legyen mondjuk killprocess.sh, rakhatjuk bárhova pl. a /root/bin-be. A script:

```

#!/bin/bash
kill -9 $MONIT_PROCESS_PID

```

Monitrc-be meg be kell írni, hogy ezt a scriptet futassa:

```

check process any matching `.*`
    if cpu > 33% then exec "/root/bin/killprocess.sh"

```

A 4.4 feladatban megadott paranccsal lehet tesztelni (persze 33-nál nagyobb cpulimittel), ha minden jól megy, akkor szinte rögtön le kell állnia.

4.4. feladat: Töltse le a virtuális gép Linux disztribúciójának és verziójának megfelelő stress és cpulimit programokat. A leírt programok segítségével írjon egy bash scriptet, amely 25%-ig leterheli a CPU-t.

stress -c 1 & cpulimit -p \$(pidof -o \$! stress) -l 25

LOL nem lehet letölteni ezeket (a virtuális gépen)

Telepítés:

```
sudo dpkg -i cpulimit_1_1-13_i386.deb
```

```
sudo dpkg -i stress_1_0_1-1_i386.deb
```

<http://www.medialab.bme.hu/course/VITMA314::VITMAC02>