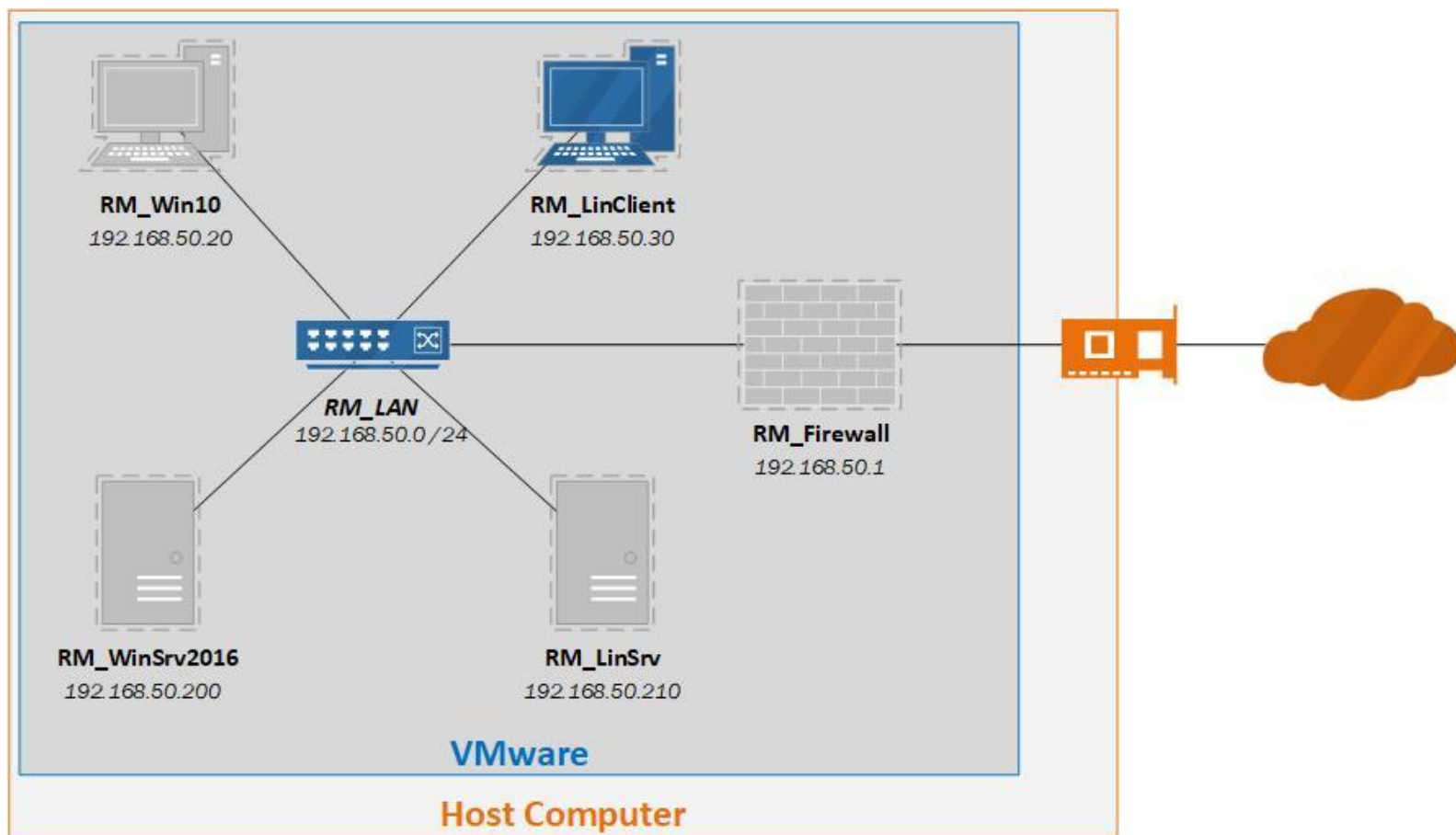


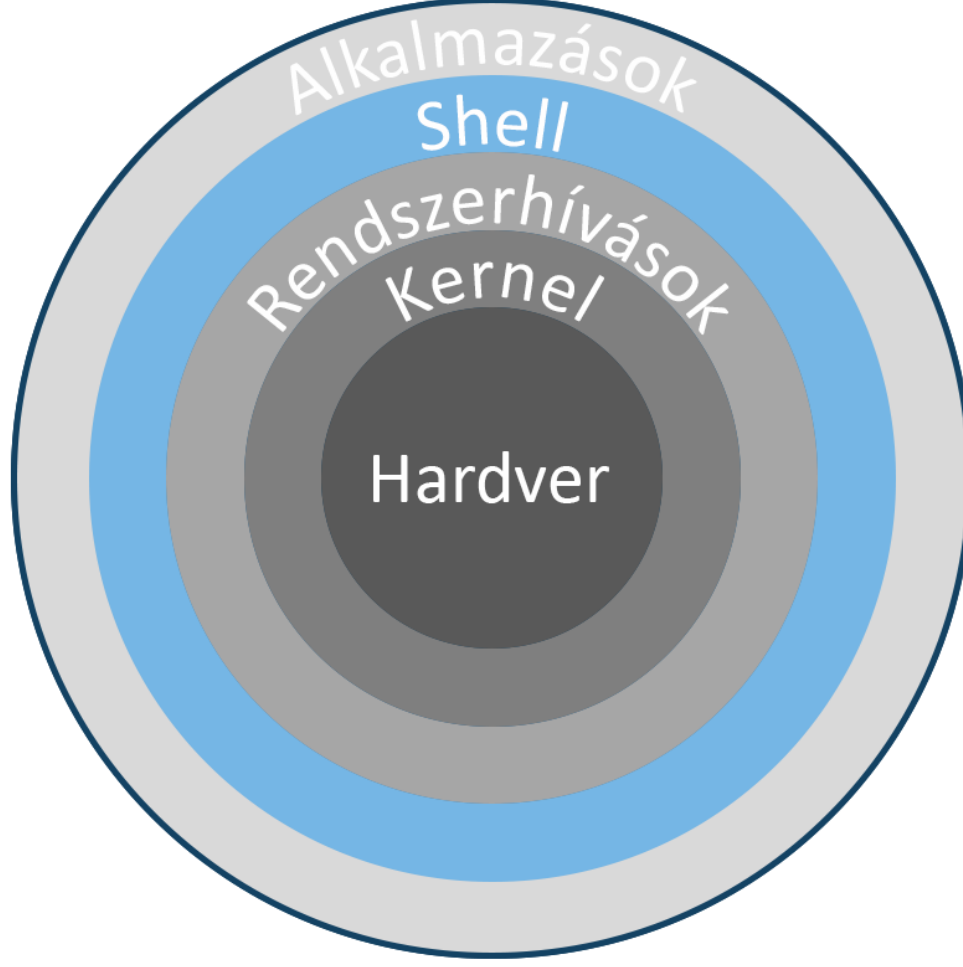
Linux script – szöveg és fájl feldolgozás

Operációs rendszerek

Óbudai Egyetem

A szükséges virtuális gépek





Szövegfeldolgozás

Keresés fájlban és szövegben – 1/2

- Keresni fájlban és szövegben a **grep**-el tudunk
- A keresett szöveget tartalmazó sort adja vissza
- Keresés fájlban:

grep keresendő_szöveg fájlnev/fájlnevek

pl.: **grep allatok.txt macska**

- Keresés szövegben vagy parancsok kimenetében:

parancs | grep keresendő_szöveg

pl.: **ls -l | grep '.mp3'**

Keresés fájlban és szövegben – 2/2

- Néhány hasznos **grep** kapcsoló:
 - **-v** – negált keresés (azokat a sorokat adja vissza, amiben nem található meg a keresett szöveg)
 - **-h** – több fájlban kereséskor azt is megadja, hogy melyik fájlban találta meg
 - **-i** – kis- és nagybetűk között nem tesz különbséget
 - **-w** – egész szóra keres
 - **-n** – a sorszámot is visszaadja

.sh-t tartalmazó fájlnevek listázása

Feladat: Listázzuk ki azokat a fájlokat, amik .sh-t tartalmaznak a nevükben! Nem szükséges erre script létrehozása!

.sh-t tartalmazó fájlnevek listázása (megoldása)

- A megoldás a következő parancs:

```
ls -l | grep '.sh'
```

- Futtatás után a következő az eredmény:

```
student@linclient:~$ ls -l | grep .sh
-rwxr-xr-x 1 student student 361 aug 28 12:53 ciklus.sh
-rwxr-xr-x 1 student student 53 aug 23 11:01 datum.sh
-rwxr-xr-x 1 student student 195 aug 23 15:28 hello.sh
-rwxr-xr-x 1 student student 59 aug 23 11:51 lista.sh
-rwxr-xr-x 1 student student 415 aug 28 13:04 menu.sh
-rwxr-xr-x 1 student student 19 aug 23 12:33 parancs.sh
-rwxr-xr-x 1 student student 105 aug 23 15:15 szamologep.sh
student@linclient:~$
```


Reguláris kifejezések – 1/5

- Sokszor van, hogy a keresett szöveg bizonyos részei változhatnak (például telefonszámok keresése: tudjuk a formátumot, de maga a telefonszám változhat)
- Reguláris kifejezések használata esetén az **egrep** vagy a **grep -E**-t használjuk
- Karakterek: az adott karakter keresése
 - Eddig ezt csináltuk
 - Speciális karakter keresése esetén a kifejezésbe `\` karaktert kell a keresendő jel elé tenni
 - Például: ha `*`-ra akarunk keresni, akkor **egrep** `'*'`
 - Speciális karakterek a következők: `* . , [] \ ^ $`
- `^` (önmagában): a keresési mintát a sor elejére teszi (a sor elejének kell megfelelnie a keresett kifejezések)
 - Példa: Linux scriptelést tanulunk
egrep `'^Lin'`

Reguláris kifejezések – 2/5

- **\$** (önmagában): a keresési mintát a sor végére teszi (a sor végének kell megfelelnie a keresett kifejezések)
 - Példa: Linux scriptelést tanulunk
`egrep '$ulunk'`
- Konkrét sor keresése:
`egrep '^Linux scriptelést tanulunk$'`
- **\<** : szó elejének keresése
 - Példa: Linux scriptelést tanulunk
`egrep '\<scr'`
- **\>** : szó végének keresése
 - Példa: Linux scriptelést tanulunk
`egrep 'lést\>'`

Mappák listázása

Feladat: Listázzuk ki a mappákat egrep segítségével!
Nem szükséges erre script létrehozása!

Mappák listázása (megoldás)

- A megoldás a következő parancs:

```
ls -l | egrep '^d'
```

- Futtatás után a következő az eredmény:

```
student@linclient:~$ ls -l | egrep '^d'
drwxr-xr-x 2 student student 4096 aug 11 15:59 Desktop
drwxr-xr-x 2 student student 4096 júl 16 16:06 Documents
drwxr-xr-x 2 student student 4096 júl 24 11:28 Downloads
drwxr-xr-x 2 student student 4096 júl 16 16:06 Music
drwxr-xr-x 2 student student 4096 aug 11 15:59 Pictures
drwxr-xr-x 2 student student 4096 júl 16 16:06 Public
drwxr-xr-x 2 student student 4096 júl 16 16:06 Templates
drwxr-xr-x 2 student student 4096 júl 16 16:06 Videos
student@linclient:~$
```

Kommentek kiszedése

Feladat: Szedjük ki a /etc/debconf.conf fájlból a kommentelt sorokat és az eredményt mentjük el egy fájlba! Nem szükséges erre script létrehozása!

Kommentek kiszedése (megoldás)

- A megoldás a következő parancs:

```
cat /etc/debconf.conf | egrep -v '^#' > ujfajl
```

Reguláris kifejezések – 3/5

- `.` : bármilyen karakterre keres (egy karakter van a helyén) (kivéve az új sort jelző karakterre)
 - Példa: Linux scriptelést tanulunk
`egrep '.....'` – olyan sort keres amiben van legalább 5 betűs szó
 - Ha olyan sort keresünk, amiben három pont van, akkor:
`egrep '\.\\.\\. '`
 - Ha olyan sort keresünk, amiben pontosan 5 karakteres szó van:
`egrep '\<.....\>'`
 - Ha olyan sort keresünk, amiben pontosan 5 karakterből áll:
`egrep '^.....$'`

Szavak keresése

Feladat: Listázzuk ki a /usr/share/dict/words fájlból azokat a szavakat, melyek:

- 5 karakteresek,
- c betűvel kezdődnek,
- f betűvel végződnek.

Nem szükséges erre script létrehozása!

Szavak keresése (megoldás)

- A megoldás a következő parancs:

```
cat /usr/share/dict/words | egrep '\<c...h\>'
```

- Futtatás után a következő az eredmény:

```
student@linclient:~$ cat /usr/share/dict/words | egrep '\<c...h\>'
catch
catch's
cinch
cinch's
clash
clash's
cloth
cloth's
coach
coach's
conch
conch's
couch
couch's
cough
cough's
crash
crash's
crush
crush's
student@linclient:~$
```

Reguláris kifejezések – 4/5

- **[]** : a két szögletes zárójel között levő karakterek valamelyikére keres
 - Példa: Linux scriptelést tanulunk
`egrep '^[HLZ]'` – olyan sort keres ami H, L vagy Z karakterrel kezdődik
 - Negálás:
`egrep '^[^HLZ]'` – olyan sort keres ami nem H, L vagy Z karakterrel kezdődik (a példamondatot nem adná vissza!)
- **[-]** : karakter tartományra illeszti rá
 - Példa: Linux scriptelést tanulunk
`egrep '^[A-Z]'` – nagybetűvel kezdődő sorok
 - Tartományok:
 - `[a-z]` – kisbetűk
 - `[A-Z]` – nagybetűk
 - `[a-zA-Z]` – kis- és nagybetűk
 - `[0-9]` – számok

Reguláris kifejezések – 5/5

- A `[]` és a `[-]` használható egyszerre is:
 - `[a-zAÉÍÓÖŐÜÚ]`
 - `[A-ZÁÉÍÓÖŐÜÚ]`

Nagybetűs fájlok

Feladat: Listázzuk ki azokat a fájlokat és mappákat, melyek neve nagy betűvel kezdődik! Nem szükséges erre script létrehozása!

Nagybetűs fájlok (megoldás)

- A megoldás a következő parancs:

```
ls -1 | egrep '^[A-ZÁÉÍÓÖŐÜŰ]'
```

- Futtatás után a következő az eredmény:

```
student@linclient:~$ ls -1 | egrep '^[A-ZÁÉÍÓÖŐÜŰ]'  
Desktop  
Documents  
Downloads  
Music  
Pictures  
Public  
ScriptekEgyben  
Templates  
Videos  
student@linclient:~$
```

Jelentésmódosító jelek – 1/2

- A jelentésmódosító jelek a már megismert reguláris kifejezésekkel együtt használhatók
- **?** : az előtte álló karakter opcionális
 - Példa: color és colour
egrep 'colou?r'
- ***** : az előtte álló karakterből tetszőleges számút keres (0 is lehet!)
 - Példa: 15618916 vagy 1 – olyan sorokat keresünk, ami 1-el kezdődő számokat tartalmaz
egrep '1[0-9]*'
- **+** : az előtte álló karakterből tetszőleges számút keres, de legalább egyet
 - Példa: 1111111111 – olyan számokat keresünk, melyekben csak 1-esek vannak
egrep '1+'

Jelentésmódosító jelek – 2/2

- **{}** : az előtte álló karakterből a kapcsoszárójelben megadott számút keres
 - Példa: Hello World – két l betűs szavakat tartalmazó sorokra keresünk
egrep 'l{2}'
 - Intervallum is megadható: **egrep 'l{2,5}'** – l betűk száma 2 és 5 között van
- **()** : karakterek csoportosítására szolgál, így a jelentésmódosító arra a csoportra, karakterláncra vonatkozik, nem egy karakterre
 - Példa: Linux scriptelést tanulunk – kétszer van benne „nu”
egrep '(nu){2}'
- **|** : vagy kapcsolat
 - Példa: asztalon – keressük azokat a sorokat, ahol az asztal valamilyen raggal szerepel
egrep 'asztal(ban|on|hoz|ra|nak)'

Szavak keresése 2.

Feladat: Listázzuk ki a /usr/share/dict/words fájlból azokat a szavakat, melyek pontosan 2 „e” betűt tartalmaznak!

Nem szükséges erre script létrehozása!

Szavak keresése 2. (megoldás)

- A megoldás a következő parancs:

```
cat /usr/share/dict/words | egrep 'e{2}'
```

Szavak keresése 3.

Feladat: Listázzuk ki a /usr/share/dict/words fájlból azokat a szavakat, melyek legalább egyszer tartalmazzák az „es” karakterláncot!

Nem szükséges erre script létrehozása!

Szavak keresése 2. (megoldás)

- A megoldás a következő parancs:

```
cat /usr/share/dict/words | egrep '(es)+'
```

Sed

Sed

- A sed egy parancssoros szövegszerkesztő program
- Az utasításokat a parancssorról vagy scriptből szedi, nem interaktív
- A bemeneti szöveget soronként dolgozza fel, az adott sort a mintatérbe olvas be és úgy hajtja végre rajta az utasítás(oka)t
- Feldolgozás után a mintatérben levő sort kiírja
- Parancs végrehajtása bemeneti fájlon:

```
sed 'utasítás' bemeneti_fájl > kimeneti_fájl
```

vagy

```
cat bemeneti_fájl | sed 'utasítás' > kimeneti_fájl
```

- A kimeneti fájl ne egyezzen meg a bemeneti fájlal!

Sed utasítások - mintatér kiírása – 1/2

- **p** : Mintatér aktuális tartalmának kiírása

`sed 'p' datum.sh`

```
student@linclient:~$ sed 'p' datum.sh
#!/bin/sh
#!/bin/sh

echo "Az aktuális dátum:" `date`
echo "Az aktuális dátum:" `date`
exit 0
exit 0
student@linclient:~$
```

- Minden sor kétszer jelenik meg, mert egyszer lefut a parancs (ami a kiírás), ezt követően pedig a parancs végrehajtása után a sed kiírja a mintatér tartalmát
- Ahhoz, hogy az utasítás végrehajtása után ne írja ki az aktuális mintatér tartalmát, használhatjuk az **-n** kapcsolót:

```
student@linclient:~$ sed -n 'p' datum.sh
#!/bin/sh

echo "Az aktuális dátum:" `date`
exit 0
student@linclient:~$
```

Sed utasítások - mintatér kiírása – 2/2

- Ezen felül megadható, hogy mely sorokon futtassa a parancsot

- **# p** : A #-edik soron hajtja végre az utasítást

`sed -n '3 p' datum.sh`

```
student@linclient:~$ sed -n '3 p' datum.sh
echo "Az aktuális dátum:" `date`
student@linclient:~$
```

- **#, # p** : A két # közötti sorokon hajtja végre az utasítást

`sed -n '2,4 p' datum.sh`

```
student@linclient:~$ sed -n '2,4 p' datum.sh
echo "Az aktuális dátum:" `date`
exit 0
student@linclient:~$
```

Sed utasítások - mintatér (sor) törlése

- **d** : Mintatér aktuális tartalmának törlése
 - Itt is használhatók a kiírás során megismert sorcímezési módok!
sed '4 d' datum.sh

```
student@linclient:~$ sed '4 d' datum.sh
#!/bin/sh

echo "Az aktuális dátum:" `date`
student@linclient:~$
```


Sed utasítások - szöveg cseréje a sorban – 1/2

- **s** : A megadott reguláris kifejezést cseréli a megadott szövegre, szintaktikája: **s/mit/mire/**
- Csak az első találatot cseréli ki!

sed 's/exit 0/exit 1/' datum.sh – kicseréljük az exit kódot

```
student@linclient:~$ cat datum.sh
#!/bin/sh

echo "Az aktuális dátum:" `date`
exit 0
student@linclient:~$ sed 's/exit 0/exit 1/' datum.sh
#!/bin/sh

echo "Az aktuális dátum:" `date`
exit 1
student@linclient:~$
```

- A szintaktikában zölddel jelölt elválasztó karakter tetszőleges, tehát lehetne akár ez is : **s.exit 0.exit 1.**
- Célszerű olyan elválasztó karaktert választani, ami nem szerepel a cserélendő szövegben

Sed utasítások - szöveg cseréje a sorban – 2/2

- Ha azt szeretnénk, hogy a sorban ne csak az első találatot cserélje ki, hanem az összeset, akkor azt a g utasítással tudjuk megtenni a következő szintaktika szerint:

s/mit/mire/g

sed 's/u/U/g' datum.sh – kicseréljük a kis „u” betűket nagy „U” betűkre

```
student@linclient:~$ cat datum.sh
#!/bin/sh

echo "Az aktuális dátum:" `date`
exit 0
student@linclient:~$ sed 's/u/U/g' datum.sh
#!/bin/sh

echo "Az aktUális dátUm:" `date`
exit 0
student@linclient:~$
```

Menü keretének módosítása

Feladat: Módosítsuk a menu.sh scriptben levő *-okból álló keretet +-okból álló keretre a sed segítségével! A kimeneti fájl neve legyen menu2.sh!

Nem szükséges erre script létrehozása!

Menü keretének módosítása (megoldás)

- A megoldás a következő parancs:

```
sed 's/*/+/g' menu.sh > menu2.sh
```

- Futtatás után a következő az eredmény:

```
student@linclient:~$ sed 's/*/+/g' menu.sh > menu2.sh
student@linclient:~$ cat menu2.sh
#!/bin/sh

valasz=""
until [ "$valasz" = "3" ];
do
echo '+++++++'
echo '+ Menü +'
echo '+ 1 - Mappa tartalma +'
echo '+ 2 - Aktuális mappa +'
echo '+ 3 - Kilépés +'
echo '+++++++'
echo 'Kérem válasszon:'
read valasz
case "$valasz" in
    "1") ls -l ;;
    "2") pwd ;;
    "3") echo "Viszlát!" ;;
    +) echo "Csak 1-3 értékek között válasszon!" ;;
esac
done
student@linclient:~$
```

IP cím „hamisítás”

Feladat: Az ifconfig parancs segítségével lekérhető a számítógép IP címe. Az ifconfig kimenetét módosítsuk úgy sed segítségével, mintha egy másik számítógép IPv4 címe lenne beállítva! Használjunk reguláris kifejezést az IP cím megtalálásához!

Nem szükséges erre script létrehozása!

- Az ifconfig az összes interfész beállításait kilistázza. Ha utána írjuk, hogy melyik interfész információira vagyunk kíváncsiak, akkor csak az adott interfészt írja ki!

IP cím „hamisítás” (nem jó megoldás)

- A megoldás a következő parancs:

```
ifconfig ens33 | sed 's/inet [0-9]{1,3}\.[0-9]{1,3}\.[0-9]{1,3}\.[0-9]{1,3}/inet 8.8.8.8/'
```

- Ez egy nem jó megoldás!
- A problémát az okozza, hogy a sed picit máshogy értelmezi a reguláris kifejezéseket, esetében a { és a } elé kell \ karakter

IP cím „hamisítás” (jó megoldás)

- A megoldás a következő parancs:

```
ifconfig ens33 | sed 's/inet [0-9]\{1,3\}\.[0-9]\{1,3\}\.[0-9]\{1,3\}/inet 8.8.8.8/'
```

```
student@linclient:~$ ifconfig ens33
ens33: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
    inet 192.168.50.30  netmask 255.255.255.0  broadcast 192.168.50.255
    inet6 fe80::20c:29ff:fe1f:659c  prefixlen 64  scopeid 0x20<link>
    ether 00:0c:29:1f:65:9c  txqueuelen 1000  (Ethernet)
    RX packets 318  bytes 19080 (19.0 KB)
    RX errors 0  dropped 0  overruns 0  frame 0
    TX packets 46220  bytes 2785788 (2.7 MB)
    TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

student@linclient:~$ ifconfig ens33 | sed 's/inet [0-9]\{1,3\}\.[0-9]\{1,3\}\.[0-9]\{1,3\}/inet 8.8.8.8/'
ens33: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
    inet 8.8.8.8  netmask 255.255.255.0  broadcast 192.168.50.255
    inet6 fe80::20c:29ff:fe1f:659c  prefixlen 64  scopeid 0x20<link>
    ether 00:0c:29:1f:65:9c  txqueuelen 1000  (Ethernet)
    RX packets 318  bytes 19080 (19.0 KB)
    RX errors 0  dropped 0  overruns 0  frame 0
    TX packets 46224  bytes 2786028 (2.7 MB)
    TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

student@linclient:~$
```

cut

- A **cut** program segítségével a sorokat oszlopokra tudjuk bontani

cut **kapcsolók** **bemeneti_fájl**

- Kapcsolói:
 - -d – az elválasztó karakter megadása
 - -f – a megjelenítendő oszlop száma
- Példa: csoportok listázása

cut -d : -f 1 /etc/group

```
whoopste  
scanner  
sane  
pulse  
pulse-access  
avahi  
colord  
geoclue  
gdm  
student  
smbshare  
student@linclient:~$
```

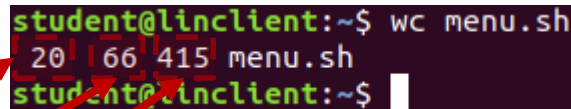

WC

- A **wc** megszámlolja a bemenetként adott fájlban levő sorokat, szavakat és karaktereket

wc bemeneti_fájl

- Példa: nézzük meg a menu.sh script sorainak számát

wc menu.sh



```
student@linclient:~$ wc menu.sh
20 66 415 menu.sh
student@linclient:~$
```

Sorok száma

Szavak száma

Karakterek száma

- Kapcsolókkal megadhatjuk, hogy csak a sorok vagy csak a szavak számát adja vissza

head és tail

- **head** : A bemenetként megadott fájl első 10 sorát írja ki

head bemeneti_fájl

- A **-n** kapcsolóval megadható, hogy az első 10 helyett az első X sort írja ki

head -n 5 menu.sh

- **tail** : A bemenetként megadott fájl utolsó 10 sorát írja ki

tail bemeneti_fájl

- A **-n** kapcsolóval megadható, hogy az utolsó 10 helyett az utolsó X sort írja ki

tail -n 5 menu.sh

tac

- A **tac** a bemenetként kapott fájl sorait fordított sorrendben írja ki (utolsó sor lesz az első)

tac bemeneti_fájl

- Példa: írassuk ki a menu.sh fájl tartalmát visszafelé

tac menu.sh

Felhasználók száma

Feladat: írassuk ki a rendszer felhasználóinak számát!

Nem szükséges erre script létrehozása!

A felhasználók listáját a `/etc/passwd` fájl tartalmazza.

Felhasználók száma (megoldás)

- A megoldás a következő parancs:

```
wc -l /etc/passwd
```

- Futtatás után a következő az eredmény:

```
student@linclient:~$ wc -l /etc/passwd  
41 /etc/passwd  
student@linclient:~$ █
```

- Ha ténylegesen csak a számra van szükség (pl.: script miatt):

```
wc -l /etc/passwd | cut -d ' ' -f 1
```

```
student@linclient:~$ wc -l /etc/passwd | cut -d ' ' -f 1  
41  
student@linclient:~$ █
```

Fájl tulajdonosa

Feladat: Írjunk egy olyan scriptet, amely kiírja a bemeneti paraméterben megadott fájl tulajdonosát!

- Mentsük el a felhasználó saját mappájába **tulaj.sh** néven

Fájl tulajdonosa (megoldás)

- A script a következő sorokból áll:

```
#!/bin/sh  
  
tulaj=`ls -l "$1" | cut -d ' ' -f 3`  
echo "A fájl tulajdonosa: $tulaj"
```

- Futtatás után a következő az eredmény:

```
student@linclient:~$ ./tulaj.sh 2018-08-23_11-41.txt  
A fájl tulajdonosa: student  
student@linclient:~$
```

További hasznos programok

Fájl tulajdonosának módosítása

Feladat: Módosítsuk a tulaj.sh scriptet úgy, hogy abban az esetben, ha a fájl tulajdonosa a „student” felhasználó, akkor módosítsuk a fájl tulajdonosát root felhasználóra!

- Tulajdonos módosítása: **chown**

Fájl tulajdonosának módosítása (megoldás)

- A script a következő sorokból áll:

```
#!/bin/sh

if [ "`ls -l $1 | cut -d ' ' -f 3`" = "student" ]
then
    chown root $1
fi
```

Fájl tulajdonosának módosítása 2.

Feladat: Módosítsuk a tulaj.sh scriptet úgy, hogy abban az esetben, ha a fájl tulajdonosa a bejelentkezett felhasználó, akkor módosítsuk a fájl tulajdonosát root felhasználóra!

Fájl tulajdonosának módosítása 2. (megoldás)

- A script a következő sorokból áll:

```
#!/bin/sh

if [ "`ls -l $1 | cut -d ' ' -f 3`" = „$USER” ]
then
    chown root $1
fi
```

Fájl tulajdonosának módosítása 3.

Feladat: Módosítsuk a tulaj.sh scriptet úgy, hogy a második paraméterként megadott felhasználó legyen a fájl tulajdonosa, de előtte ellenőrizze le, hogy a megadott felhasználó valóban létezik-e!

Fájl tulajdonosának módosítása 3. (megodás)

- A script a következő sorokból áll:

```
#!/bin/sh

if [ "`cat /etc/passwd | cut -d : -f 1 | egrep "^$2$" | wc -l`" -eq 1 ]
then
    if [ "`ls -l $1 | cut -d ' ' -f 3`" = "$USER" ]
    then
        chown $2 $1
    fi
else
    echo "Nem található ilyen felhasználó"
fi
```

Tömörítés

Tömörítés – 1/2

- tar – fájlok összecsomagolására szolgál
- gzip – fájlok tömörítésére szolgál
- Ha a fájl kiterjesztése tar.gz, akkor csomagolva és tömörítve is van
- tar segítségével összecsomagolhatjuk, majd átadhatjuk a gzip-nek tömörítésre
- A kicsomagolás és a becsomagolás a paramétereiktől függ

Tömörítés – 2/2

- Becsomagolás:

tar -cvzf cél_fájlnev forrás_fájlnevek

- c: compress
- v: verbose
- z: gzip
- A forrás fájlnevek lehetnek felsorolva vagy ha az adott mappa tartalmát szeretnénk tömöríteni, akkor használhatunk *-ot is

- Kicsomagolás:

tar -xvzf forrás_fájlnev

- x: extract

Fájlok és mappák keresése

find – 1/2

- A **find** segítségével fájlokat és mappákat kereshetünk
- Megadhatjuk a kiindulási mappát
- Továbbá azt is megadhatjuk, hogy hajtson végre bizonyos parancsokat a találatokon

find kezdő_könyvtár mit_keresünk mit_hajtson_végre

- Példa: jpg kiterjesztésű fájlok keresése abból a mappából indulva, ahol vagyunk

```
find . -name "*.jpg" -print
```

find – 2/2

- A **find** fontosabb kapcsolói:
 - **-name** – fájl vagy könyvtár neve
 - **-type** – típus (f: file, d: mappa)
 - **-exec** – parancs futtatása a fájlon
- Az **exec** kapcsoló használata:
 - **-exec `parancs {} \;`**
 - **{}** – ennek a helyére jön az aktuális fájlnev
 - **\;** – ezzel zárjuk le a parancsot

Fájl keresése és információk listázása

Feladat: Írjunk egy olyan scriptet, amely a gyökérből kiindulva keresi a paraméterként kapott fájlnevet és találat esetén listázza annak információit!

- Mentsük el a felhasználó saját mappájába **keres.sh** néven

Fájl keresése és információk listázása (megoldás)

- A script a következő sorokból áll:

```
#!/bin/sh
```

```
find / -type f -name "$1" -exec ls -l {} \;
```

Fájlok letöltése

wget

- A wget egy letöltőprogram, melyel a hálózatról tudunk letölteni fájlokat HTTP-n, HTTPS-en és FTP-n keresztül

wget **elérési_út**



MAGYARORSZÁG
KORMÁNYA

Európai Unió
Európai Szociális
Alap



BEFEKTETÉS A JÖVŐBE

SZÉCHENYI 2020

Köszönöm a figyelmet!

Operációs rendszerek

Óbudai Egyetem