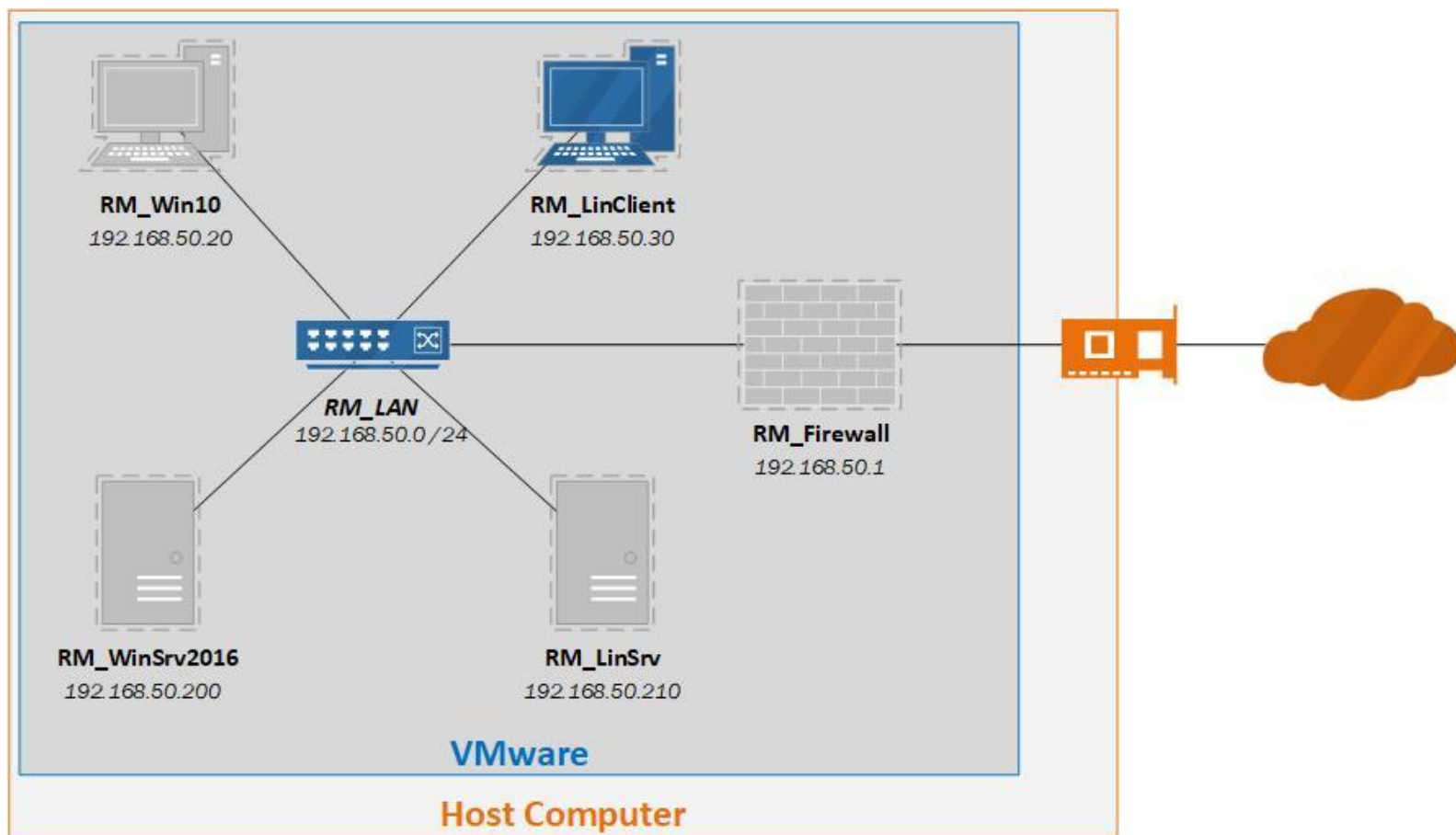


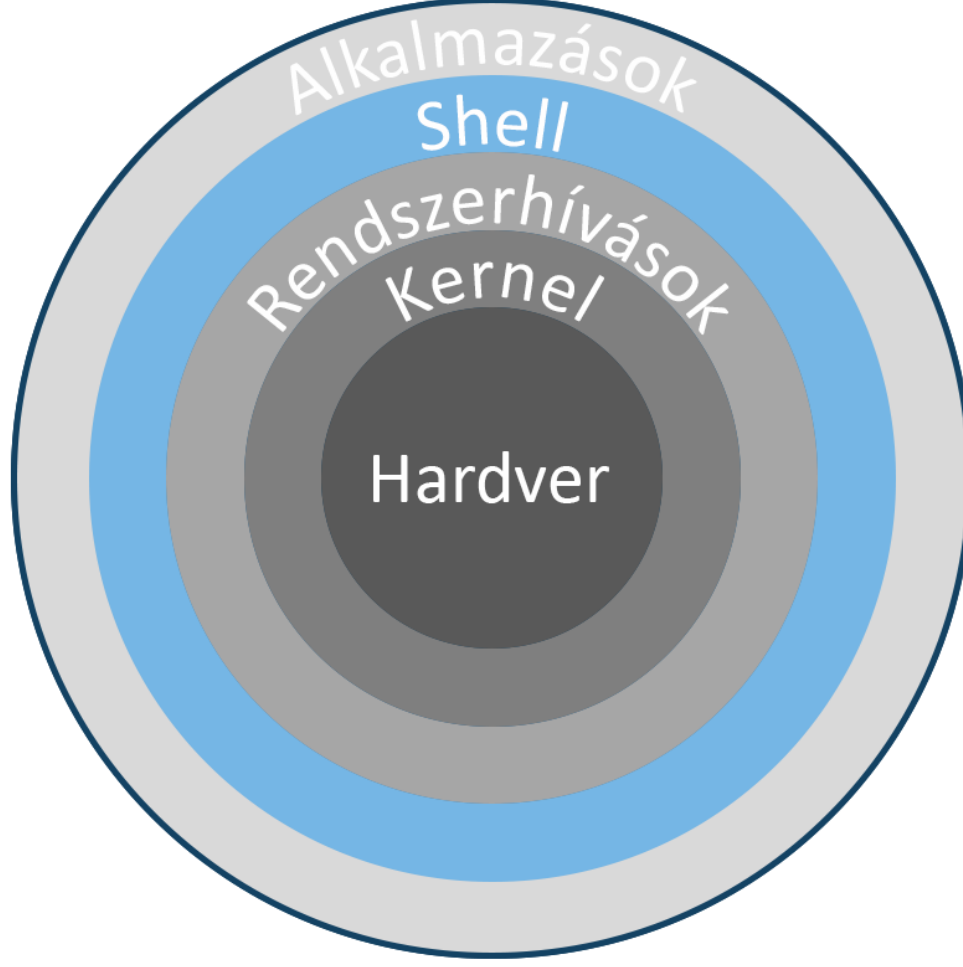
Linux script – vezérlési szerkezetek

Operációs rendszerek

Óbudai Egyetem

A szükséges virtuális gépek





Elágazások

IF feltételes elágazás

- IF feltételes elágazás esetén a belső rész akkor fut le, ha a logikai kifejezés igaz
- Szintaktikája:

```
if [ logikai_kifejezés1 ]
then
    # parancsok, ha a logikai_kifejezés1 igaz
elif [ logikai_kifejezés2 ]
then
    # parancsok, ha a logikai_kifejezés2 igaz
else
    # parancsok, ha sem a logikai_kifejezés1, sem a
    # logikai_kifejezés2 nem volt igaz
fi
```

Logikai kifejezések

- Fájl létezésének vizsgálata
 - Igaz, ha a fájl létezik
`if [-e fájlnev]`
 - Igaz, ha a fájl létezik és olvasható
`if [-r fájlnev]`
 - Igaz, ha a fájl létezik és írható
`if [-w fájlnev]`
- Egész számok vizsgálata
 - Igaz, ha két szám egyenlő
`if [$sz1 -eq $sz2]`
 - Igaz, ha két szám nem egyenlő
`if [$sz1 -ne $sz2]`
 - További operátorok: `-gt`, `-ge`, `-lt`, `-le`
- Stringek vizsgálata
 - Igaz, ha a két string megegyezik
`if ["$s1" = "$s2"]`

Több feltétel

- Legalább az egyik feltételnek teljesülni kell (OR):

```
if [ logikai_kifejezés1 ] || [ logikai_kifejezés2 ]  
then  
    # logikai_kifejezés2 nem volt igaz  
fi
```

- Mindkét feltételnek teljesülnie kell (AND):

```
if [ logikai_kifejezés1 ] && [ logikai_kifejezés2 ]  
then  
    # mindkét feltétel igaz volt  
fi
```

Case feltételes elágazás

- Case feltételes elágazás esetén a belső rész akkor fut le, ha a case után álló értéke megegyezik a megadott értékkel
- Szintaktikája:

```
case "$valtozo" in
    "ertek1") # parancs ertek1 esetén ;;
    "ertek2") # parancs1 ertek2 esetén
               # parancs2 ertek2 esetén
               # parancs3 ertek2 esetén ;;
    *) # parancsok bármilyen más érték esetén ;;
esac
```

- Példa:

```
szam=1
case "$szam" in
    "1") echo "Egy" ;;
    *) echo "Nem egy";;
esac
```


Két műveletes számológép (IF-el)

Feladat: Módosítsuk a `szamologep.sh` scriptet úgy, hogy az első paraméter:

- „+” esetén adja össze a második és harmadik paraméternek megadott számot
- „-” esetén vonja ki egymásból a második és harmadik paraméternek megadott számot
- amennyiben bármi más karakter, írja ki, hogy hibás operátor!

IF feltételes elágazást használjunk!

Két műveletes számológép (IF-el) (megoldás)

- A script a következő sorokból áll:

```
#!/bin/sh

if [ "$1" = "+" ]
then
    expr $2 + $3
elif [ "$1" = "-" ]
then
    expr $2 - $3
else
    echo "Hibás operátor!"
fi
```

- Futtatás után a következő az eredmény:

```
student@linclient:~$ ./szamologep.sh + 5 3
8
student@linclient:~$ ./szamologep.sh - 5 3
2
student@linclient:~$ ./szamologep.sh / 5 3
Hibás operátor!
student@linclient:~$
```

Két műveletes számológép (CASE-el)

Feladat: Módosítsuk a számologep.sh scriptben az IF elágazást Case-re!

Két műveletes számológép (CASE-el) (megoldás)

- A script a következő sorokból áll:

```
#!/bin/sh

case "$1" in
    "+") expr $2 + $3 ;;
    "-") expr $2 - $3 ;;
    *) echo "Hibás operátor!" ;;
esac
```

- Futtatás után a következő az eredmény:

```
student@linclient:~$ ./szamologep.sh + 5 3
8
student@linclient:~$ ./szamologep.sh - 5 3
2
student@linclient:~$ ./szamologep.sh / 5 3
Hibás operátor!
student@linclient:~$
```

Feltételes köszönés

Feladat: Módosítsuk a hello.sh scriptet úgy, hogy napszaknak megfelelően köszönjön!

- Jó reggelttel 9 óráig
- Jó nappal 18 óráig
- Jó estéttel 18 órától

Feltételes köszönés (megoldás 1/2)

- A script a következő sorokból áll:

```
#!/bin/sh

echo "Mi az ön neve? "
read nev

ora=`date +%H`
if [ $ora -lt 9 ]
then
    echo "Jó reggelt $nev !"
elif [ $ora -lt 18 ]
then
    echo "Jó napot $nev !"
else
    echo "Jó estét $nev !"
fi
```

Feltételes köszönés (megoldás 2/2)

- Futtatás után a következő az eredmény:

```
student@linclient:~$ date
2018. aug. 23., csütörtök, 15:28:52 CEST
student@linclient:~$ ./hello.sh
Mi az ön neve?
Gergely Dániel
Jó napot Gergely Dániel !
student@linclient:~$
```

Menü

Feladat: Írjunk egy olyan scriptet, amely futtatáskor kiír egy menüt. A menüben a következő lehetőségek legyenek:

- 1-esre listázza ki a mappa tartalmát
 - 2-esre írja ki, hogy a felhasználó éppen melyik mappában van
 - 3-asra kilép
 - Bármilyen más esetben írja ki, hogy a felhasználó csak 1 és 3 értékek között válasszon és lépjen ki
- Mentsük el a felhasználó saját mappájába **menu.sh** néven

Menü (megoldás 1/2)

- A script a következő sorokból áll:

```
#!/bin/sh

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
```

Menü (megoldás 2/2)

- Futtatás után a következő az eredmény:

```
student@linclient:~$ ./menu.sh
*****
*   Menü                               *
* 1 - Mappa tartalma                   *
* 2 - Aktuális mappa                   *
* 3 - Kilépés                          *
*****
Kérem válasszon:
2
/home/student
student@linclient:~$ ./menu.sh
*****
*   Menü                               *
* 1 - Mappa tartalma                   *
* 2 - Aktuális mappa                   *
* 3 - Kilépés                          *
*****
Kérem válasszon:
3
Viszlát!
student@linclient:~$
```

Ciklusok

For ciklus

- For ciklussal egy listán tudunk „végig” menni
- Ilyen lista lehet egy számsorozat (1-től 10-ig), fájlok listája, stb.
- Szintaktikája:

```
for ciklusváltozó in lista
do
    # parancsok
done
```

Listák for ciklushoz

- Kézzel beírt listák

```
for ciklusváltozó in 1 2 3 4 5 6
```

- Generált számlista

```
for ciklusváltozó in `seq 6`
```

- Számokból álló listát generálni a seq paranccsal tudunk

`seq utolsó` – 1-től a megadott paraméterig generál listát, 1-el inkrementálva

`seq első utolsó` – a megadott paraméterek között generál listát, 1-el inkrementálva

`seq első inkrementum utolsó` – a megadott paraméterek között generál listát, a megadott inkrementummal

- Fájlok listája

```
for ciklusváltozó in `ls`
```

Számsorozat egy sorban 1.

Feladat: Írjunk egy olyan scriptet, amely futtatáskor kiír 1-től a paraméterként megadott számig egy számsorozatot egyetlen sorban!

- Mentsük el a felhasználó saját mappájába **ciklus.sh** néven
- Változó értékadása összefűzéssel
(C#-ban stringeknél +=):

```
valtozo1="Hello"  
valtozo2="World!"  
valtozo1="$valtozo1 $valtozo2"
```

Számsorozat egy sorban 1. (megoldás)

- A script a következő sorokból áll:

```
#!/bin/sh

sor=""
for i in `seq $1`
do
    sor="$sor $i"
done
echo $sor
```

- Futtatás után a következő az eredmény:

```
student@linclient:~$ ./ciklus.sh 10
1 2 3 4 5 6 7 8 9 10
student@linclient:~$
```

Számsorozat egy sorban 2.

Feladat: Módosítsuk a ciklus.sh scriptet úgy, hogy a számsorozat két paramétert kérjen be és csökkenő sorozatot adjon vissza! Az első paraméter a kisebb szám legyen!

Számsorozat egy sorban 2. (megoldás)

- A script a következő sorokból áll:

```
#!/bin/sh

sor=""
for i in `seq $2 -1 $1`
do
    sor="$sor $i"
done
echo $sor
```

- Futtatás után a következő az eredmény:

```
student@linclient:~$ ./ciklus.sh 5 20
20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5
student@linclient:~$
```

Számsorozat egy sorban 3.

Feladat: Módosítsuk a ciklus.sh scriptet úgy, hogy a számsorozat két paramétert kérjen be és csökkenő sorozatot adjon vissza, illetve akkor is működjön, ha a felhasználó fordítva adja meg a paramétereket!

Számsorozat egy sorban 3. (megoldás 1/2)

- A script a következő sorokból áll:

```
#!/bin/sh

sor=""
if [ "$1" -lt "$2" ]
then
    for i in `seq $2 -1 $1`
    do
        sor="$sor $i"
    done
else
    for i in `seq $1 -1 $2`
    do
        sor="$sor $i"
    done
fi
echo $sor
```

Számsorozat egy sorban 3. (megoldás 2/2)

- Futtatás után a következő az eredmény:

```
student@linclient:~$ ./ciklus.sh 20 5
20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5
student@linclient:~$ ./ciklus.sh 5 20
20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5
student@linclient:~$
```

Fájlok tartalmának kiírása

Feladat: Módosítsuk a ciklus.sh scriptet úgy, hogy a mappában található fájlok tartalmát kiírja!

- Fájlok tartalmát a **cat** paranccsal írjuk ki!

Fájlok tartalmának kiírása (megoldás)

- A script a következő sorokból áll:

```
#!/bin/sh

for i in `ls`
do
    cat $i
done
```

- Korábban találkoztunk már a *-al, ami a mappa tartalmát helyettesíti be, így ez is jó megoldás:

```
#!/bin/sh

for i in *
do
    cat $i
done
```

Scriptek egybe másolása

Feladat: Módosítsuk a ciklus.sh scriptet úgy, hogy a mappában található fájlok tartalmát egy fájlba másolja át!

- Figyeljünk arra, hogy a fájl valójában script legyen! Erre használjuk a **file** parancsot.

file **fájlnév** – visszaadja, hogy a fájl milyen fájl

Scriptek egybe másolása (megoldás)

- A script a következő sorokból áll:

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

```
osszesscript="ScriptekEgyben"
```

```
echo "" > $osszesscript
```

```
for i in `ls`
```

```
do
```

```
    fajltipus=`file $i`
```

```
    vartfajltipus="$i: POSIX shell script, ASCII text executable"
```

```
    vartfajltipus2="$i: POSIX shell script, UTF-8 Unicode text executable"
```

```
    if [ "$fajltipus" = "$vartfajltipus" ]
```

```
        || [ "$fajltipus" = "$vartfajltipus2" ]
```

```
    then
```

```
        cat $i >> $osszesscript
```

```
    fi
```

```
done
```

Erre azért van szükség, hogy
a célfájl ne scriptként lássa
(shebang miatt)

Egy sorban!

While ciklus

- A ciklusmagban foglalt utasítások addig futnak, míg a feltétel igaz (kilép, amint a feltétel már nem igaz)

```
while [ logikai_kifejezés ];  
do  
    # parancsok  
done
```

- A logikai kifejezések az elágazásoknál megismertekkel egyeznek

Until ciklus

- A ciklusmagban foglalt utasítások addig futnak, míg a feltétel nem teljesül (kilép, amint a feltétel igaz lesz)

```
until [ logikai_kifejezés ];  
do  
    # parancsok  
done
```

- A logikai kifejezések az elágazásoknál megismertekkel egyeznek

Menü ciklussal

Feladat: Módosítsuk a menu.sh scriptet úgy, hogy addig ne lépjen ki, amíg nem válassza a felhasználó a kilépést!

Menü ciklussal (megoldás 1/2)

- A script a következő sorokból áll:

```
#!/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
```

Menü ciklussal (megoldás 2/2)

- Futtatás után a következő az eredmény:

```
student@linclient:~$ ./menu.sh
*****
* Menü *
* 1 - Mappa tartalma *
* 2 - Aktuális mappa *
* 3 - Kilépés *
*****
Kérem válasszon:
5
Csak 1-3 értékek között válasszon!
*****
* Menü *
* 1 - Mappa tartalma *
* 2 - Aktuális mappa *
* 3 - Kilépés *
*****
Kérem válasszon:
3
Viszlát!
student@linclient:~$
```

Köszönöm a figyelmet!

Operációs rendszerek

Óbudai Egyetem