



UNIVERSITETI “UKSHIN HOTI” - PRIZREN
FAKULTETI I SHKENCAVE KOMPJUTERIKE
TEKNOLOGJITË E INFORMACIONIT DHE TELEKOMUNIKIMI (TIT)

Ushtrimi laboratorik 6

Lënda: Arkitektura e Kompjuterëve dhe Sistemet Operative

Ph. D. c. Arbër Beshiri

Objektivat e ushtrimit laboratorik: të sqarohen komandat themelore dhe sintaksa e tyre për Linux (Ubuntu), të zhvillohen dhe testohen detyra përmes Shell Script-ës së Linux (Ubuntu).

1. Pse nevoiten argumentet e komandë-rreshtave?

- a) Komandë-rreshtat (linjat) mundësojnë që komanda të përdorë opsionet e caktuara.
- b) Informojnë komandat që përdorën, se cilin fajll ose grup (fajllash) duhet të procesojnë.

Komanda *rm* përdorët për të fshirë fajllin. Nëse shënojmë *\$rm {emri-fajllit}*, *rm* paraqet komandën, ndërsa *{emri-fajllit}* është fajlli të cilin dëshirojmë të fshim. Pra, siç vërehet komandën mund të shoqërojmë në rresht me opsione tjera. Por, si do t'i çasemi argumenteve të komandë-rreshtit (command line) përmes skriptës? Një rast është përmes komandës *ls*: *\$ls -a/**. Kjo komandë-rresht ka dy argumente: “-a” dhe “/*”.

2. Komandë-rreshtat dhe disa informacione themelore

Paraqitja e emrave të komandave në Shell Script, përkatësisht emri i fajllit/programit të memorizuar (ruajtur) realizohet përmes komandës \$0. Numri i argumenteve gjendet përmes komandës \$#. Argumentet aktuale gjenden përmes komandës: \$1, \$2, ... , etj.

2.1. Shtypja dhe qasja e argumenteve të komandë-rreshtave

```
$cat >prog1
#
# Skripta e cila demonstroi argumentet e komandë-rreshtave
#
echo "Numri i përgjithshëm i argumenteve të komandë-rreshtave është $#"
```

- Shtypim **CTRL+D**, mundësohet ruajtja e programit.
- Pastaj shtypim **chmod 777 prog1** për aktivizim të programit.
- Pastaj shtypim **./prog1** që të ekzekutohet programi.

Në dalje të Shell Script-ës fitohet rezultati:

```
Numri i përgjithshëm i argumenteve të komandë-rreshtave është 0
./prog1 është emri i skriptës
është argumenti i parë
është argumenti i dytë"
Të gjitha nga ato janë: -0 ose
```

Shënim: komandë-rreshtat mund të përdoren për llogaritje të caktuara, prandaj në program mund të inkorporohen si: \$1, \$2, \$3, etj., dhe përmes ekzekutimit të tyre jepet vlera e komandë-rreshtave. P. sh., $R = \$2 + \3 , në dalje jepen vlerat p. sh.: `./prog2 30 40` dhe rezultati do të jetë $R = 70$.

3. Komanda për testim të numrave pozitiv dhe negativ

Komanda për testim të shprehjes përdoret për të vërtetuar nëse ndonjë shprehje është e vërtetë (*true*) dhe nëse është e vërtetë shfaqet si vlerë zero (0), nëse nuk është e vërtetë fitohet rezultati jo zero, përkatësisht kur shprehja është jo e vërtetë (*false*).

Sintaksa: `Test expression OR [expression]`.

Në vijim është dhënë një shembull me komandën për testim:

```
$cat >prog2
#
# Gjetja e vlerave pozitive ose negative
```

```
#
clear
echo "Në vazhdim kemi dy variabla të pavarura"
a=1
b=0
if test $a -gt $b
then
echo "Numri $a është numër pozitiv"
fi
```

- Shtypim **CTRL+D**, ruhet programi.
- Pastaj shtypim **chmod 777 prog2** për aktivizim të programit.
- Pastaj shtypim **./prog2** që programi të ekzekutohet.

Rezultati në dalje të Shell Script do të jetë:

```
Numri 1 është numër pozitiv
```

Shënim: në detyra të ngjashme sikurse e mësipërmja, në vend të komandës *if test \$a -gt \$b*, mund të bëhet deklarimi i testimit edhe përmes komandës *if [\$a -gt \$b]* dhe rezultati do të jetë i njëjtë. Duhet të kemi kujdes në shënimin e kushtit dhe krijimin e vendit të zbrazët ndërmjet variablave dhe komandave, gjatë shkruarjes së kushtit përmes komandës *if*!

Kushti *if test \$a -gt \$b* teston nëse argumenti i komandë-rreshtit të parë a është më i madh se b.

Testi ose *[expr]* funksionon me:

- integjer (numra të plotë),
- lloje të fajllave dhe
- stringje (karaktere).

4. Krahاسimi i numrave përmes komandës *read*

Nëse vlerat e variablave i japim përmes komandës *read* kemi mundësi të bëjmë krahasimin e numrave të ndryshëm.

```
$cat >prog3
#
# Krahاسimi i vlerave të dy variablave të ndryshme
#
clear
echo "Në vijim kemi variablat a dhe b"
echo "Jepni vlerën e variables a: "
```

```

read a
echo "Jepni vlerën e variables b: "
read b
echo "Pra, vlera e a=$a dhe b=$b"
if test $a -gt $b
then
echo "Numri $a është më i madh se numri $b"
else
echo "Numri $a është më i vogël se numri $b"
fi

```

- Shtypim **CTRL+D**, ruajm programin.
- Pastaj shtypim **chmod 777 prog3** për aktivizim të programit.
- Pastaj shtypim **./prog3** që programi të ekzekutohet.

Rezultati në dalje të Shell Script-ës do të jetë:

```

Në vijim kemi variablat a dhe b
Jepni vlerën e variablës a:
8
Jepni vlerën e variablës b:
12
Pra, vlera e a=8 dhe b=12
Numri 8 është më i vogël se numri 12

```

5. Përdorimi i operatorëve matematikor në Shell Script

Në vazhdim është dhënë tabela me operatorët matematikor që përdoren në Shell Script. Shkurtesat eq, ne, it, le, gt, ge janë shkurtesa të shprehjeve në gjuhën angleze për operatorët matematikor¹.

Operatorët matematikor në Shell Script	Roli dhe kuptimi	Aritmetika dhe deklarimi matematikor	Përdorimi në Shell Script	
			Për deklarimin e testimit me komandën if	Për deklarimin e testimit me komandën if
-eq	Është i barabartë me...	5==5	if test 5 -eq 5	if [5 -eq 5]

¹ Ku eq = is equal to; ne = is not equal to; lt = is less than; le = is less than or equal to; gt = is greater than, ge = is greater than equal to.

-ne	Është jo i barabartë me...	$4 \neq 5$	if test 4 -ne 5	if [4 -ne 5]
-lt	Është më i vogël se...	$4 < 5$	if test 4 -lt 5	if [4 -lt 5]
-le	Është më i vogël ose i barabartë me...	$4 \leq 4$	if test 4 -le 4	if [4 -le 4]
-gt	Është më i madh se...	$5 > 4$	if test 5 -gt 4	if [5 -gt 4]
-ge	Është më i madh ose i barabartë me...	$5 \geq 5$	if test 5 -ge 5	if [5 -ge 5]

6. Operatorët logjik

Operatorët logjik përdorën për të kombinuar përnjëherë dy ose më shumë shprehje. Në vijim do të paraqesim rolin dhe funksionimin e këtyre operatorëve.

Operatori	Roli dhe kuptimi
! shprehja	JO logjik (NOT logic)
Shprehja1 -a shprehja2	DHE logjik (AND logic)
Shprehja1 -o shprehja2	OSE logjik (OR logic)

7. Kushti if...else...fi

Nëse kushti i dhënë është i vërtetë, atëherë ekzekutohet *komanda1* (*command1*), përndryshe ekzekutohet *komanda2* (*command2*).

Sintaksa:

if kushti

then

 kushti është zero (e vërtetë – 0)

 ekzekutohen të gjitha komandat deri në deklarimin *else*

else

```
nëse kushti nuk është i vërtetë atëherë  
ekzekutohen të gjitha komandat deri në fi  
fi
```

Detyra:

1. Krahasoni i numrave të caktuar kur vlerat e variablave jepen përmes urdhërit *read* si dhe gjetja e shumës ose zbritjes

Nëse leximin e vlerave të variablave e realizojmë përmes komandës (urdhërit) *read* dhe pastaj bëjmë krahasimin e numrave, atëherë programi për krahasimin e variablave mund të jepet si në vazhdim.

```
$cat >prog4  
#  
# Përdorimi i urdhërit if...then...else  
#  
clear  
a=3  
b=4  
if test $a -gt $b  
then  
echo "Variabla të pavarura janë a dhe b"  
#formula për llogaritjen e mbledhjes shënohet si në vazhdim  
# Nëse a është më e madhe se b realizohet mbledhja, nëse jo realizohet zbritja  
c=`expr $a + $b`  
echo "Rezultati i shumës për variablat e pavarura a=$a dhe b=$b është: "  
echo "Shuma c=a+b është: c=$c"  
else  
#formula për llogaritjen e zbritjes shënohet si në vazhdim  
d=`expr $b - $a`  
echo "Për vlerat e argumenteve të dhëna: a=$a dhe b=$b"  
echo "Pra, argumenti a është më i vogël se b"  
echo "Rezultati i zbritjes është: d=b-a, prandaj d=$d"  
fi
```

- Shtypim **CTRL+D**, ruajm programin.
- Pastaj shtypim **chmod 777 prog4** për aktivizim të programit.
- Pastaj shtypim **./prog4** që të ekzekutojmë programin.

Rezultati në dalje do të jetë:

Për vlerat e argumenteve të dhëna: a=3 dhe b=4
 Pra, argumenti a është më i vogël se b:
 Rezultati i zbritjes është: d=b-a, prandaj d=1

2. Përdorimi i komandës if...then...else dhe përdorimi alternativ i saj për krahasim të variablave

Krahasimi i vlerave të variablave realizohet dhe përmes sintaksës: *if [variabla –OperatoriLogjik variabla2]*.

Për këtë rast është dhënë një shembull si në vazhdim:

```
$cat >prog5
#
# Përdorimi i komandës if (me kllapa të mesme) në vend të test
#
clear
a=10
b=20
if [ $a -lt $b ]
then
echo "Numri a është më i vogël se numri b"
else
echo "Numri a është me i madh se numri b"
fi
```

- Shtypim **CTRL+D**, ruhet programi.
- Pastaj shtypim **chmod 777 prog5** për aktivizim të programit.
- Pastaj shtypim **./prog5** që të ekzekutohet programi.

Rezultati në dalje të Shell Script-ës do të jetë:

Numri a është më i vogël se numri b

3. Përdorimi i komandës if...then...else për variabla të ndryshme

Përdorimi i komandës (urdhërit) if...then...else për variabla të ndryshme është dhënë përmes shembullit në vazhdim:

```
$cat >prog6
#
# Përdorimi i komandës if...then...else përmes komandës read
#
clear
echo "Jepni vlerat e variablave a dhe b: "
read a b
echo "Vlerat e variablave janë: a=$a dhe b=$b."
if [ $a -lt $b ]
then
echo "Numri a është më i vogël se numri b"
else
echo "Numri a është më i madh se numri b"
fi
```

- Shtypim **CTRL+D**, ruhet programi.
- Pastaj shtypim **chmod 777 prog6** për aktivizim të programit.
- Pastaj shtypim **./prog6** që të ekzekutohet programi.

Rezultati në dalje do të jetë i ndryshëm për variabla të ndryshme:

```
Jepi vlerat e variablave a dhe b:
-30 -70
Vlerat e variablave janë: a=-30 dhe b=-70.
Numri a është më i madh se numri b
```

Urdhëri if...then...else mund të përdoret edhe sipas sintaksës:


```

if kushti
then
    if kushti
    then
        ...
        ...
        urdhëri
    else
        ...
        ...
        urdhëri
    fi
else
    ...
    ...
    urdhëri
fi

```

4. Përdorimi shumënivelësh i if...then...else

Për përdorimin shumënivelësh të kushtit if...then...else shfrytëzohet sintaksa:

```

if kushti
then
    ekzekutohen të gjitha komandat deri te deklarimi elif (kur kushti është i vërtetë)
elif kushti1
then
    ekzekutohen të gjitha komandat deri te deklarimi elif (kur kushti është i vërtetë)
elif kushti2
then
    ekzekutohen të gjitha komandat deri te deklarimi elif (kur kushti është i vërtetë)
...
else
    ky kusht plotësohet kur nuk plotësohen kushtet paraprake
fi

```

Pra, siç vërehet mund të krijojmë një varg kushtesh të caktuara me ndihmën e urdhërit *elif*.

```

$ cat> prog7
#
# Skripta për testim të if...elif...else

```

```
# Kur variablat kanë vlera konstante
a=1
b=0
if [ $a -gt $b ]; then
echo "Numri $a është numër pozitiv"
elif [ $a -lt $b ]
then
echo "Numri $a është numër negativ"
elif [ $a -eq $b ]
then
echo "Numri $a është zero "
else
echo "Kujdes! A janë numra $a dhe $b, jepni numrat?!"
fi
```

- Shtypet **CTRL+D**, ruhet programi.
- Pastaj shtypim **chmod 777 prog7** për aktivizim të programit.
- Pastaj shtypim **./prog7** që të ekzekutohet programi.

Rezultati në dalje do të jetë:

```
Numri 1 është numër pozitiv
```

5. Përdorimi shumënivelësh i if...then...else (elif)

Shembulli në vazhdim jep një rast për përdorimin shumënivelësh të if...then...else.

```
$ cat > prog8
#
# Përdorimi i if...elif...else kur vlerat e variablave i japim përmes
# komandës read
#
clear
echo "Jepni vlerat për variablat a dhe b:"
read a b
echo "Vlerat e variablave janë: a=$a dhe b=$b"
if [ $a -gt $b ]; then
echo "Numri $a është numër më i madh se $b"
elif [ $a -lt $b ]
```

```
then
echo "Numri $a është numër më i vogël se $b"
elif [ $a -eq $b ]
then
echo "Numri $a është i barabartë me numrin $b"
else
echo "Kujdes! A janë numra $a dhe $b? Ju lutemi jepni numrin!"
fi
```

- Shtypim **CTRL+D**, ruhet programi.
- Pastaj shtypim **chmod 777 prog8** për aktivizim të programit.
- Pastaj shtypim **./prog8** që të ekzekutohet programi.

Rezultati në dalje do të jetë (për vlera arbitrare të a dhe b):

```
Jepni vlerën e variablave a dhe b:
10 10
Vlerat e variablave janë: a=10 dhe b=10
Numri 10 është i barabartë me numrin10
```