

LD1. Vartotojų paskyrų valdymo scenarijai

Aprašymas:

1. Kintamųjų sukūrimas ir spausdinimas

Kintamojo sukūrimas ir reikšmės suteikimas:

```
$ test="test"
```

Komandos rezultatų priskyrimas:

```
$ namukatalogas=`pwd`
```

Simbolių eilutės priskyrimas(\ naudojamas apsaugai nuo simbolio interpretavimo apvalkale):

```
$ string="\`Labas rytas\`" pasake zmogus."
```

Kintamojo panaudojimas:

```
$ namukatalogas=$HOME
```

```
$ cd $namukatalogas
```

Apvalkalo kintamųjų spausdinimas:

```
$ echo PS1
$ echo $PS1
$ echo $USERNAME
$ echo 'hello'
$ echo "hello"
$ echo hello
```

Kelių eilučių apjungimas:

```
$cat << EOF
labas
Dabar yra pusiaudienis
EOF
```

Komandų apjungimas:

```
$bash << EOF
echo darius
chmod 705 $HOME
EOF
```

Netiesioginis komandos iškvietai. Pirmiausia priskiriama string kintamajam, po to jis išspausdinamas apvalkale ir įvykdomas:

```
$ x='ls -l'
$ $x
```

Vykdoma komanda echo -n:

```
$ var="-n"
$ echo $var
```

Viengubose kabutėse perduodama viskas, dvigubose – viskas išskyrus kintamuosius, kurie yra interpretuojami:

```
#!/bin/sh
v1="abc "
v2="$v1 d"
echo $v1 $v2
```

```
v1="abc"
v2='$v1 d'
echo $v1 $v2
```

Atitikimas šablonams:

* - simbolių eilutėje atitinka visus simbolius

? - simbolių eilutėje atitinka vieną simbolį

```
$ ls *.dat
```

2. Globalūs ir lokalūs kintamieji ir funkcijos

```
VAR="Globalus"
```

```
function funkcija {
    local VAR="lokalus"
    #Lokalus kintamasis, matomas
    #tik funkcijos viduje
    echo $VAR
}
funkcija
echo $VAR
bash
echo $VAR
rez=$(funkcija)    #arba rez=`funkcija`

#!/bin/bash
str="vienas"
funkcija() {
    str="du" #matomas ne tik funkcijoje
    for x in $str
    do
```

Failė (pvz. funkcijos.lib) galima aprašyti tik funkcijas:

```
funkcija () {
# funkcijai galima perduoti tik jos
# viduje galiojancius argumentus
echo $1 $2 trys
}
```

Į scenarijų funkcijos importuojamos:

```
source funkcijos.lib
```

```
#!/bin/bash
source funkcijos.lib
# vienas du - perduodami argumentai
funkcija vienas du
exit 1
```

Funkcijos grąžinama reikšmė gaunama taip:

```

        echo $x
    done
}
funkcija
echo $str $x

```

```
result=$(funkcija vienas du)
```

3. Svarbiausios konstrukcijos (prisiminimui)

Sakinys if:

```

if [ $# -ne 4 ]; then
    echo Iveskite keturis argumentus
    exit $E_ARG
fi

```

Ciklas for:

```

sar="$(ls *.csv)"
for i in $sar; do
    echo $i
done

for i in $2 $4 $5; do
    echo $i
done

```

Ciklas while:

```

x=1
while [ $x -le 5 ]; do
    echo $x
    x=$(( $x + 1 )) #bash aritmetika
done

while read i; do
    echo $i
done < $1 #per argumentą perduodamas
#failas, kuris skaitomas po eilutę

```

Ciklas case:

```

case $1 in
1)echo vienas
;;
2)echo du
;;
3)echo trys
;;
esac

```

4. Masyvai

Masyvo deklaravimas:

```

#!/bin/bash
mas=( 'Darius' 'Jonas' Petras Ieva )
mas[0]=Darius
mas[1]=Jonas

```

Taip pat argumentus iš komandinės eilutės galima laikyti masyve:

```
mas=("$@")
```

Išspausdinti masyvo elementus:

```

echo ${mas[0]} ${args[1]}
KIEKIS=${#mas[@]}
for (( i=0;i<$KIEKIS;i++)); do
    echo ${mas[$i]}
done

${mas[*]}      # Visi elementai
${!mas[*]}     # Visi masyvo indeksai
${#mas[*]}     # Elementų kiekis masyve
${#mas[@]}     # Elementų kiekis masyve
${#mas[0]}     # Nulinio elemento ilgis

```

5. Argumentų perdavimas scenarijui iš komandinės eilutės

```

#!/bin/bash
echo $1 $2 $3;

```

\$@ - visi argumentai sąrašu.
 \$* - visi argumentai simboliu eilute.
 \$# - argumentų kiekis.
 \$? - paskutinės vykdytos komandos išėjimo kodas.
 \$\$ - apvalkalo proceso ID.

6. Komandų SFTP ir SSH panaudojimas

Komanda scp leidžia kopijuoti failus tarp tinklo serverių, autentifikavimui ir failų perdavimui naudoja saugų ssh protokolą:

```
$ scp loginas@serveris:failas.txt /kur/kopijuoti
```

Kopijuoti failą(-us) iš serverio:

```
$ scp vartotojo_vardas@serveris.lt:failas_serveryje.txt /lokalus/katalogas  
$ scp vartotojo_vardas@serveris.lt:~/\{failas1.txt,failas2.txt\} .
```

Kopijuoti failą į serverį:

```
$ scp failas.txt vartotojo_vardas@serveris.lt:/serverio/katalogas  
$ scp failas1.txt failas2.txt vartotojo_vardas@serveris.lt:~ #į namų katalogą  
$ scp -P 64801 -C failas.txt vartotojo_vardas@serveris.lt:/serverio/katalogas  
#64801 serverio portas, -C - naudoti duomenų suspaudimą
```

Katalogo kopijavimas į serverį:

```
$ scp -r katalogas vartotojo_vardas@serveris.lt:/serverio/katalogas
```

Komandų vykdymas nutolusiame serveryje:

```
$ ssh vartotojo_vardas@serveris.lt 'komanda'  
$ ssh vartotojo_vardas@serveris.lt 'ls -l ~/public_html'
```

7. Naudingos komandos:

```
bash -x skriptas.bash #bash debug'inimo režimas
```

\$IFS - vidinis apvalkalo stulpelių skyriklis (pagal nutylėjimą tarpas, tab'as ir naujos eilutės simbolis).
Panaudojimo pavyzdys:

```
#!/bin/bash  
issaugotasIFS=$IFS  
IFS=$' ' #skyrikliais dabar yra ; ir tarpas.  
# .....  
# Veiksmai, pvz eilutes, kurioje žodžiai atskirti tarpais ir kableliais  
# nuskaitymas į masyvą.  
# .....  
IFS=$issaugotasIFS; #arba unset $IFS  
  
$ chmod u+x scenarijus #vykdymo teisių suteikimas failui scenarijus
```

Užduotys

Scenarijų serverio adresas: **158.129.0.113**. Prisijungimas per *putty* arba *Secure Shell Client*. Vartotojo vardas – *varpava*, (t.y. trys vardo raidės ir keturios pavardės raidės), o slaptažodžio paprašykite pas dėstytoją. Prisijungę būtinai pasikeiskite slaptažodį komanda *passwd*.

Šiai užduočiai atlikti reikės prisijungimo prie antro serverio (*stud*), kurio vardas *stud.if.ktu.lt*. Prisijungimą prie šio serverio galima susikurti svetainėje <https://stud.if.ktu.lt>.

Užduotis1. Parašykite scenarijų *uzd1.sh*, kuris per *ssh* iš scenarijų serverio prisijungtų prie *stud* serverio ir iš Jūsų paskyros *public_html* katalogo grąžintų tik šakniniame kataloge esančių *.html, *.htm ir *.php failų sąrašą, o taip pat suskaičiuotų jų užimamą vietą diske. Pirmasis scenarijaus argumentas – Jūsų prisijungimo vardas *stud* serveryje, tolesnieji - failų priesagos.

Tam tikslui pirmiausia scenarijų modulyje serveryje sukurkite failą *~/ssh/config*, kurio turinys:

```
Host stud.if.ktu.lt  
    HostKeyAlgorithms+=ssh-rsa  
    PubkeyAcceptedKeyTypes+=ssh-rsa
```

O kad nereikėtų kiekvieną kartą jungiantis prie serverio suvesti slaptažodžio, reikia sugeneruoti viešą ssh raktą ir įrašyti į stud.if.ktu.lt serverio failą ~/.ssh/authorized_hosts:

```
$ ssh-keygen -t rsa
Generating public/private rsa key pair.
Enter file in which to save the key (/home/login/.ssh/id_rsa):
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/login/.ssh/id_rsa.
Your public key has been saved in /home/login/.ssh/id_rsa.pub.
The key fingerprint is:
f6:61:a8:27:35:cf:4c:6d:13:22:70:cf:4c:c8:a0:23 login@localhost
```

Sukurtas privatus raktas išsaugotas ~/.ssh/id_rsa, viešas - ~/.ssh/id_rsa.pub. Jame saugojama viena eilutė:

```
ssh-rsa AAAAB3NzaClyc2EAAAABIwAAAIEArkqv9X8eTVK4F7pM1St45pWoiakFkZMw
G9BjydOJPGH0RFNAYlQqIWBGWv7vS5K2tr+EEO+F8WL2Y/jK4ZkUoQgoi+n7DWQVOHsR
ijcS3LvtO+50Np4yJXYWJKh29JL6GHcp8o7+YKEYVUMB2CSDOP99eF9g5Q0d+1U2WVdB
WQM= login@localhost
```

Šita eilutė turi būti nukopijuota į stud.if.ktu.lt serverio failą ~/.ssh/authorized_keys ir sistema prie serverio jungsis neprašydama slaptažodžio:

```
$ ssh-copy-id -i ~/.ssh/id_rsa.pub stud_serverio_loginas@stud.if.ktu.lt
```

Naudingos komandos užduotims atlikti:

```
ssh stud_serverio_loginas@stud.if.ktu.lt 'ls ~/public_html'
#grąžina serverio public_html katalogo turinį.
ssh stud_serverio_loginas@stud.if.ktu.lt 'ls ~/public_html' 2> klaidos.txt
#klaidos vietoj ekrano nukreipiamos į failą
ls -l *.csv | awk '{print $5}' #grąžina *.csv failų dydžių sąrašą
```

Užduotis2. Pirmąją užduotį atlikite faile *uzd2.sh* panaudodami funkcijas. Tam tikslui faile *funkcijos.lib* sukurkite funkcijas:

- **suma** (skaičiuoja argumentų elementų sumą (ne kiekį!)),
- **failu_sarasas** (grąžina failų sąrašą, paduodamas prisijungimo vardas ir failų galūnės),
- **failu_dydziai** (grąžina failų dydžius, paduodamas prisijungimo vardas ir failų galūnės)

Funkcijas panaudokite scenarijuje *uzd2.sh*. Papildomai suformuokite rezultatų failą, nusiųskite į stud.if.ktu.lt serverį ir patalpinkite namų kataloge ~/.

Naudingos komandos:

```
$ echo $s | tee -a rez # rezultatai isvedami ir į ekraną ir į failą
```

Užduotis3. Parašykite ir įvykdykite scenarijų *uzd3.sh*, kuris iš faile *stud.csv* (/home/stud/stud/stud.csv) saugomo studentų sąrašo suformuotų prisijungimo vardų, susidedančių iš keturių vardo ir keturių pavardės, ir 10 simbolių slaptažodžių sąrašą, ir išspausdintų juos į ekraną. Failas nurodomas per argumentus.

Galimas užduoties vykdymo scenarijus ir naudingos komandos:

```
$ str1=`echo $i | awk -F';' '{print $2}'` #išskirti antrąjį stulpelį, -F - skyriklis
$ login=${vardas:0:3}${pavarde:0:3} #nuo dviejų string kintamųjų paimami pirmi
#trys simboliai, kurie po to sujungiami.

gen_pass() { #slaptažodžius generuojanti funkcija
[ "$2" == "0" ] && CHAR="[:alnum:]" || CHAR="[:graph:]"
cat /dev/urandom | tr -cd "$CHAR" | head -c ${1:-32}
echo
}
```

Naudojimas: `gen_pass <slaptažodžio ilgis>`

Užduotis4. Sukurkite scenarijų *uzd4.sh*, kuris naudodamasis trečios užduoties rezultatais serveryje automatizuotai sukurtų sugeneruotus vartotojus, nustatytų jų namų katalogus, taip pat kiekvienam vartotojui sukurtų jo prisijungimo vardo mysql duomenų bazę, kurios prisijungimo duomenys būtų tokie patys kaip ir prisijungimo prie serverio duomenys.

Galimas užduoties įvykdymo scenarijus ir naudingos komandos:

- Pirmiausiai reikia patikrinti ar toks vartotojas neegzistuoja. Tam filtruojame */etc/passwd* failą (grep ir po to if).
- Jei vartotojo nėra, nustatome vartotojo grupės id (\$GID) ir įtraukiame vartotoją į sistemą su komanda *adduser*. Perskaitykite ką reiškia *adduser* raktai *-m -g -c -s* ir juos panaudokite.
- Po to nustatykite sukurtam vartotojo namų katalogui teises 705 (chmod).
- Naudodami mysql komandą sukurkite naują mysql duomenų bazę, naują mysql vartotoją ir suteikite jam teises naudotis sukurta duomenų baze. Reikiamus raktus ištrinkite patys.

Svarbu. Šiai užduočiai atlikti serveryje yra reikalingos root teisės. Neturint root teisių, galima galima komandas spausdinti, o ne vykdyti.