

Praktikum 5

SKRIPTID

Praktikumi ülesanded

- 1) Windows-i käsureaskriptid
- 2) PowerShell skriptid
- 3) BASH-i skriptid

cmd.exe – lihtne skriptifail

```
@ Echo off
```

```
REM See on esimene lihtne skriptifail, mis salvestab argumendina  
REM etteantud kataloogis olevate  
REM failide nimekirja argumendiga etteantud faili.  
Dir %1 > %2
```

@echo off – vaikimisi kuvatakse kõik skriptifailis olevad käsud command prompt aknas, „echo off“ keelab sellise tegevuse. @ märk tähendab, et ei kuvata ka „echo off“ käsku. Skriptifailide sellise käsuga alustamine on hea tava.

REM – selle käsuga algavat rida käsuinterpretaator ei loe – kasutatakse kommentaaride kirjutamiseks

dir – kuvab etteantud kataloogi sisu.

> - suunab dir käsu väljundi etteantud faili.

%1 ja **%2** on vastavalt esimene ja teine argument, millega skriptifail käivitatakse. Argumente saab olla kuni üheksa.

Seega tuleb ülaltoodud skriptifaili käivitada järgmiselt: „**skriptifail.bat kataloog faililisting.txt**“

cmd.exe – muutujate defineerimine

Kasutaja sisestuse küsimine skripti täitmise ajal käib käsu **SET** abil. SET käsk on muutujate väärtustamiseks.

Näiteks: **SET muutuja=12**

Muutujale nimega „**muutuja**“ omistatakse väärtus **12**. Edasi saaks „muutuja“ väärtust kasutada. Näitkes: **echo %muutuja%**

Kuvab command prompt aknasse väärtuse 12.

SET käsul on kaks võtit **/a** ja **/p**.

/a võti tähendab, et muutujale omistatakse arvuline väärtus, mis arvutatakse võrdusmärgile järgnevast avaldisest.

Näiteks: **SET /a muutuja=2+3*5**

Sellisel juhul saab muutuja väärtuseks 17 (järgitakse tehete järjekorda).

/p võti võimaldab muutuja väärtuse küsida kasutajalt.

Näiteks: **SET /p muutuja=Sisesta oma nimi:**

Selline käsk kuvab command prompt aknasse rea „Sisesta oma nimi:“ ja jääb ootama kasutaja sisestust, mille lõpetab ENTER klahvi vajutamine.

cmd.exe - valiklause

Valiklauseks on klassikaline IF lause.

Süntaks:

IF [tingimus] (käsk1) ELSE (käsk2)

Näiteks: **IF %a% EQU %b% (ECHO a == b) ELSE (ECHO a!=b)**

Tingimuse võrdlusoperaatorid on:

EQU – võrdub

NEQ – ei võrdu

LSS – on väiksem kui

LEQ – on väiksem või võrdne kui

GTR – on suurem kui

GEQ – on suure, või võrdne kui

Veel saab tingimuseks kasutada:

„**defined**“ ja „**not defined**“ - kas muutuja on defineeritud

„**exist**“ ja „**not exist**“ - näiteks faili või kataloogi olemasolu kontrollimiseks.

„**errorlevel**“ - errorlevel on eriline muutuja, kus hoitakse viimase käsu lõpukoodi (mis käsu eduka täitmise korral on 0), kui käsku edukalt ei täidetud, siis on see midagi muud olenevalt käsust ja selle täitmisel tekkinud veast.

cmd.exe – valiklause ja GOTO

Sageli kasutatakse koos IF lausega ka **GOTO** käsku.

GOTO käsk võimaldab osa koodiridu vahele jätta ja „hüpata“ kohe vajaliku koodiosa juurde.

Näiteks:

```
@ Echo off
Set a = %1
IF defined a goto :olemas
set a=vaikimisi v22rtus
:olemas
echo %a%
```

cmd.exe – tsükkel (1)

Korduvate tegevuste sooritamiseks on olemas FOR lause, mille süntaks on:

for %%muutuja in (kogumik) do (käsud)

Muutuja väärtustatakse järjest kõigi kogumikku kuuluvate objektide või väärtusega. Vaikimisi eeldatakse, et kogumikku kuuluvad failid või stringid.

Näiteks:

For %%x in (*) do (echo %%x)

For %%x in („üks“, „kaks“, „kolm“) do (echo %%x)

Esimene käsk kirjutab command prompt aknasse kõik aktiivses kataloogis olevad failinimed. Teine käsk kirjutab command prompt aknasse stringid „üks“, „kaks“, „kolm“

Sama käsk kataloogide nimede saamiseks nõuab võtit /d

For /d %%x in (*) do (echo %%x)

cmd.exe – tsükkel (2)

Kui soovimes saada failide nimekirja ka etteantud kataloogi alamkataloogides olevatest failidest, siis tuleb kasutada **võtit /r**

For /r c:\windows %%x in (*) do (echo %%x)

Kui soovime, et kogumik ei sisaldaks mitte faile või katalooge vaid hoopis mingit väärtuste seeriat (mis on programmeerijatele harjumuspärasem), siis kasutame **võtit /l**

For /l %%muutuja in (algus, samm, lõpp) do (käsud)

Näiteks:

for /l %%x in (1,1,10) do (echo %%x)

x väärtustatakse arvudega 1 kuni 10, sammuga 1.

Harjutused

1) Looge skriptifail, mis võimaldab argumentina sisse anda nime. Kui nime ei antud, siis küsitakse seda kasutaja käest skripti täitmise ajal. Lõpuks kasutajat tervitatakse nimepidi.

2) Looge skriptifail, mis lihtsustaks xcopy käsu kasutamist. Eesmärgiks on, et esimese argumentina antud kataloogi sisu kopeerida koos alamkataloogide ja peidetud failidega teise argumentiga antud asukohta. NB! Vaikimisi on olemas selline silt nagu :eof

3) Looge skriptifail, mis võimaldab kasutajal argumentidena sisestada kolmnurga külgede pikkused ja arvutab kolmnurga übermõõdu. Kui kasutaja sisestab vale arvu argumente, siis teavitatakse teda sellest. Samuti kontrollitakse, kas selline kolmnurk saab olemas olla (iga kahe külje summa peab olema suurem kui kolmas külge).

PowerShell – muutujate defineerimine

Muutuja võib defineerida käigult.

Näiteks:

\$muutuja = „Muutuja väärtus“

Muutuja täpsemaks defineerimiseks on olemas, kommandlet: **Set-Variable**

Näiteks:

Set-Variable -Name „muutuja“ -Value „Muutuja väärtus,, - Description „See on näidismuutuja“

Kuna muutuja on objekt, siis saab küsida tema meetodeid ja omadusi:

Get-Member –InputObject \$muutuja

PowerShell - valiklause

Klassikaline IF lause:

```
If ($a -eq $b) {  
    'a == b'  
} Else {  
    'a !=b'  
}
```

-eq	On võrdne
-ne	Ei ole võrdne
-gt	On suurem kui
-ge	On suurem või võrdne
-lt	On väiksem kui
-le	On väiksem või võrdne
-like	Vastab metamärkidega stringile
-notlike	Ei vasta metamärkidega stringile
-match	Vastab regulaaravaldisele
-notmatch	Ei vasta regulaaravaldisele
-contains	Kollektsioon sisaldab objekti
-notcontains	Kollektsioon ei sisalda objekti
-in	Objekt sisaldub kollektsioonis
-notin	Objekt ei sisaldu kollektsioonis

PowerShell - tsükkel

Klassikaline for tsükkel:

```
For (initsialiseerimine; tingimus; kordus) {  
    tsüklisisesed käsud  
}
```

Näiteks:

```
For ($i=1; $i -le 10; $i++){  
    Write-Host $i  
}
```

initsialiseerimine – komaga eraldatud käsud, mida käivitatakse enne tsükli alustamist.

tingimus – tsükkel kordub, kui selle avaldise väärtus on TÕENE. Avaldise väärtust kontrollitakse iga korduse järel.

kordus – komaga eraldatud käsud, mida käivitatakse iga kord kui tsükkel kordub.

Harjutus

Looge skriptifail, mis leiab esimese argumendiga etteantud kataloogist kõik .mp3 laiendiga failid ja kopeerib need teise argumendiga etteantud kataloogi uue nimekujuga

„muusika-[vanafailinimi]“.

BASH – lihtne skriptifail

```
#!/bin/bash  
muutuja = „Tere Maailm!”  
echo $muutuja
```

Esimene rida näitab, millise interpretaatoriga tuleb skriptifailis olevaid käskke täita. Ülejäänud käskke täidetakse määratud interpretaatoriga järjest.

BASH - muutujate defineerimine

Muutujate formaalne defineerimine pole vajalik, uusi muutujaid võib tekitada käigult.

Näiteks:

\$muutuja = „Muutuja väärtus“

BASH - valiklause

Klassikaline IF lause:

```
#!/bin/bash
A=1
B=„1“
if [ "$A" = "$B" ]; then
    echo On võrdsed
else
    echo Ei ole võrdsed
fi
```

Võrdlusoperaatorid:

-lt (<) – on väiksem kui

-gt (>) – on suurem kui

-le (<=) – on väiksem või võrdne

-ge (>=) – on suurem või võrdne

-eq (==) – on võrdne

-ne (!=) – ei ole võrdne

BASH - tsükkel

Klassikaline for tsükkel:

```
For ((initsialiseerimine; tingimus; kordus) )  
do  
    tsüklisisesed käsud  
done
```

Näiteks:

```
#!/bin/bash  
for (( i=1; i<=5; i++ ))  
do  
    echo "$i"  
done
```

initsialiseerimine – komaga eraldatud käsud, mida käivitatakse enne tsükli alustamist.

tingimus – tsükkel kordub, kui selle avaldise väärtus on TÕENE. Avaldise väärtust kontrollitakse iga korduse järel.

kordus – komaga eraldatud käsud, mida käivitatakse iga kord kui tsükkel kordub.