

Laboration 4 - BASH

During this laboration we will learn and practice Bourne Again Shell scripting.

Goals

- Create and execute scripts
- Use variables
- Operate on users' input
- Use decision and loop control structures

Required material

You are required to print out this document. It will serve as guide during the laboration.

You are going to need your student user name and the password in order to login in the lab computers.

*Take with you the material from **previous** laboration exercises.*

If you have the course book, please take it with you to the laboration.

Preparation

BEFORE each laboration exercise, you will be required to prepare the PREPARATION assignment (Page 2).

Doing this preparation before the exercise will help you solve the exercises and, more important, understand what you are doing in the exercise.

Preparations

1. Read about all the commands needed in this laboration. Document their function and arguments for typical use.

shebang/hashbang

if

if...else

for

-ne, -eq, -gt, -ge, -lt, -le

-d, -e, -f, -x, -r, -s

while

variables

\$#

\$0

\$1-\$n

test

shift

Laboration

1. Setting up the environment

- 1.1. Check the course web page for information about where the virtual machine repository is located.
- 1.2. Copy the virtual machine folder called DO2003 from the repository to D:\VMW\DO2003.
- 1.3. Execute (double click) the file do2003.vmx (VMware virtual machine configuration).
- 1.4. Log in to the system
User: ide
Password: ide

2. Creating and executing BASH Scripts

- 2.1. Create a skeleton for a bash script and save it as *template.sh*

2.2. Change the file permissions to read-only (avoids overwriting the template) and executable.

2.3. Open the template using nano or vi. Add to the template necessary to display “Hello World!” on the screen. Save the template with another file name and execute it.

2.4. Modify your script to “Hello world!” in a variable and to display the variable on screen.

2.5. Create a new script file that asks for the users name and prints “Hello <user name>! How are you?”. Example:

```
Enter your name: Wagner
Hello Wagner! How are you?
$
```

2.6. Create a script that prints the arguments \$0 - \$3.

2.7. Create a script that prints the number of arguments.

2.8. Create a script that asks for two numbers, a and b, and print “a > b” if a>b or “a < b” otherwise using the if-construct. Example:

```
Enter a: 7
Enter b: 3
a > b
$
```

2.9. Update the script in 2.8 and make it output “a = b” if a = b. Use the if-construct.

2.10. Create a script that prints all arguments, no matter how many arguments there are. Use a while loop, the “number of arguments” variable and the shift command. Don’t forget \$0! Example:

```
script6.sh this is a "little test"
script6.sh
this
is
a
little test
$
```

- 2.11. Make a script that lists all files and folders in the current folder using a for loop and a suitable wildcard.
- 2.12. Update the script so it lists only files.
- 2.13. Update the script so it lists files according to a wildcard given as an argument to the script. Example:

```
script7.sh *.sh
script1.sh
script2.sh
script3.sh
...
$
```

Hint: Print the number of arguments. To get a better look at what is happening, run the script 2.10 with a few different wildcards. Eg. *.sh, *, or something similar.

- 2.14. Update the script so it performs md5sum for each file instead of printing the filename to the screen. The syntax for this must be `md5sum filename`
 - 2.15. You have now written a very useful script that can generate lists of checksums for files only that can be used to detect if the content of a file has changed. (Compare the output from your script and md5sum given the wildcard * and a folder of files and subfolders.)
- 3. Creating scripts to assist with administrative tasks**
- 3.1. A SSH server can be installed and configured using scripts. Modify the following script code according to the commented instructions.

```
#!/bin/sh
# 1. Define a variable CONF and assign to it the absolute pathname of the sshd configuration file
ORIG=$CONF-original
# 2. Add the command necessary to install the SSH server
if # 3. Add a condition to check if the previous command succeeded
then
    # 4. Display error message "Error installing SSH"
    exit 1
fi
mv -n $CONF $ORIG # Explain what this command does
grep -v "Port 22" $ORIG > $CONF # Explain what this command does
# 5. Add "Port 1022" line at the end of sshd configuration file (use variable)
# 6. Add the command to restart the SSH server
# 7. Display message "SSH installed and configured successfully"
```

3.2. Save and execute the script. Use the space below to write down comments.

3.3. Make a copy of the previous script. Modify the script so it installs a FTP server, changes the ftp configuration file to enable anonymous, and restarts the FTP server.

3.4. Save and execute the script. Use the space below to write down comments.

4. Finishing the exercise

4.1. Ask laboration assistant to check your work.

4.2. Shut down your virtual machine. Which command do you use to shut down?

4.3. Did it succeed? Is there any problem? Remember, the command shutdown cannot be executed by the user so you will have to execute it as administrator.

5. After the virtual machine is closed down, REMOVE it from the host.

Congratulations