



CST365-3

SYSTEMS LEVEL PROGRAMMING

# **PRACTICAL ASSIGNMENT-03**

UWU/CST/17/054  
Shindujah Arudchelvan  
Computer Science and Technology  
Faculty of Applied Sciences

- 1) i. What is the difference between echo \$HOME and echo HOME?

echo \$HOME

\$HOME is an **environment variable**, In this case, the shell evaluates \$HOME and passes its value as an argument to echo. It will print path of the user's home directory

echo HOME

In this case it will print 'HOME' .

```
user@user-HP-Pavilion-Laptop-14-bk0xx:~$ echo $HOME
/home/user
user@user-HP-Pavilion-Laptop-14-bk0xx:~$ echo HOME
HOME
```

- ii. Create a script named "userinfo" to print the user's login name, path to home directory and user's name.

```
#!/bin/bash
#userinfo Script
#
echo $LOGNAME
echo $HOME
echo $USERNAME
~
~
```

```
user@user-HP-Pavilion-Laptop-14-bk0xx:~$ vi userinfo
user@user-HP-Pavilion-Laptop-14-bk0xx:~$ ./userinfo
bash: ./userinfo: Permission denied
user@user-HP-Pavilion-Laptop-14-bk0xx:~$ chmod 700 userinfo
user@user-HP-Pavilion-Laptop-14-bk0xx:~$ ./userinfo
user
/home/user
user
user@user-HP-Pavilion-Laptop-14-bk0xx:~$
```

- iii. Create a script named "environment" to print the height and width of the terminal, operating system type, shell name and version.

```
#!/bin/bash
#environment script
#
clear
echo $COLUMNS,$LINES
echo $OSTYPE
echo $SHELL,$BASH_VERSION
```

```
142,38
linux-gnu
/bin/bash,5.0.3(1)-release
user@user-HP-Pavilion-Laptop-14-bk0xx:~$
```

- 2) Create a script that defines variable containing the your postal address and prints this variable prefixed with the string “My address is : “

```
#!/bin/bash
#myaddress script
#
address="JAFFNA"
echo "My address is ${address}"
```

```
user@user-HP-Pavilion-Laptop-14-bk0xx:~$ vi address
user@user-HP-Pavilion-Laptop-14-bk0xx:~$ chmod 700 address
user@user-HP-Pavilion-Laptop-14-bk0xx:~$ ./address
My address is JAFFNA
user@user-HP-Pavilion-Laptop-14-bk0xx:~$
```

- 3) i. How to Define variable x with value 10 and print it on screen?

```
#!/bin/bash
#print value Script
#
x=10
echo ${x}
```

```
user@user-HP-Pavilion-Laptop-14-bk0xx:~$ vi print
user@user-HP-Pavilion-Laptop-14-bk0xx:~$ chmod 700 print
user@user-HP-Pavilion-Laptop-14-bk0xx:~$ ./print
10
user@user-HP-Pavilion-Laptop-14-bk0xx:~$
```

- ii. How to Define variable xn with your name and print it on screen?

```
#!/bin/bash
#print name script
xn="Shindujah"
echo "My name is ${xn}"
```

```
user@user-HP-Pavilion-Laptop-14-bk0xx:~$ vi name
user@user-HP-Pavilion-Laptop-14-bk0xx:~$ chmod 700 name
user@user-HP-Pavilion-Laptop-14-bk0xx:~$ ./name
My name is Shindujah
```

iii. How to print sum of two numbers?

```
#!/bin/bash
# print sum script
x=5
y=6
z=`expr $x + $y`
echo ${z}
```

```
user@user-HP-Pavilion-Laptop-14-bk0xx:~$ vi sum
user@user-HP-Pavilion-Laptop-14-bk0xx:~$ chmod 700 sum
user@user-HP-Pavilion-Laptop-14-bk0xx:~$ ./sum
11
```

iv. Point out errors if any in following script

```
#!/bin/bash
```

```
#
```

```
# Script to test MY knowledge about variables!
```

```
#
```

```
myname=Praboda //When adding String we want to use double or single quote
```

```
myos = MacOS //This is String so we want to add double or single quote & variable want to declare without any space
```

```
myno=5
```

```
echo "My name is $myname" //Curly bracket missing when calling the variable
```

```
echo "My OS is $myos" //Curly bracket missing when calling the variable
```

```
echo "My number is myno, can you see this number" //missing dollar sign and curly bracket
```

- 4) Create a script that asks you to enter your name and print a welcome message personalized with your name.

```
#!/bin/bash
#Get name Script
echo "Enter Your Name";
read name
echo "Welcome ${name}"
```

```
user@user-HP-Pavilion-Laptop-14-bk0xx:~$ ./Getname
Enter Your Name
Shindujah
Welcome Shindujah
user@user-HP-Pavilion-Laptop-14-bk0xx:~$
```

5)

- i. Create a script to print the number of command line arguments it has received.

```
$cat>demo
#!/bin/bash
#
#Script for no of arguments
echo "Total no of arguments : $#"
```

```
echo "1st argument : $1"
echo "2nd argument : $2"
```

```
user@user-HP-Pavilion-Laptop-14-bk0xx:~$ vi commandline
user@user-HP-Pavilion-Laptop-14-bk0xx:~$ chmod +x commandline
user@user-HP-Pavilion-Laptop-14-bk0xx:~$ ./commandline shindu 25
Total no of arguments : 2
1st argument : shindu
2nd argument : 25
```

- ii. Create a script that prints the value of the first command line argument followed by an asterisk and then the second command line argument.

```
#!/bin/bash
#Script for command line argument followed by an asterisk and then second commandlineargument
echo "1st argument: $1"
echo "2nd argument: $2"

echo "All of them are:$*"
```

```
user@user-HP-Pavilion-Laptop-14-bk0xx:~$ ./asterisk 'Shindujah' 'Arudchelvan'
1st argument: Shindujah
2nd argument: Arudchelvan
All of them are:Shindujah Arudchelvan
```

- 6)
- i. Write a script to show the contents of your home directory to a file called YourName\_files

```
#!/bin/bash
#shindufiles
cd /home
ls
```

```
user@user-HP-Pavilion-Laptop-14-bk0xx:~$ vi shindufiles
user@user-HP-Pavilion-Laptop-14-bk0xx:~$ chmod 700 shindufiles
user@user-HP-Pavilion-Laptop-14-bk0xx:~$ ./shindufiles
lost+found user
user@user-HP-Pavilion-Laptop-14-bk0xx:~$
```

- ii. Write a script to append names of the files listed in the /etc directory to YourName\_files

```
#!/bin/bash
ls /home/user/ > shindujah_files
cp /etc >>shindujah_files
cat shindujah
```

```
user@user-HP-Pavilion-Laptop-14-bk0xx:~$ vi Shindujah_files
user@user-HP-Pavilion-Laptop-14-bk0xx:~$ chmod 700 Shindujah_files
user@user-HP-Pavilion-Laptop-14-bk0xx:~$ ./Shindujah_files
./Shindujah_files: line 4: ommandline: command not found
./Shindujah_files: line 5: commandlines: command not found
./Shindujah_files: line 6: demo: command not found
./Shindujah_files: line 7: Desktop: command not found
./Shindujah_files: line 8: dir1: command not found
./Shindujah_files: line 9: Documents: command not found
./Shindujah_files: line 10: Downloads: command not found
./Shindujah_files: line 11: environment: command not found
Usage: file [-bcCdEhikLlNnprsvZ0] [--apple] [--extension] [--mime-encoding]
        [--mime-type] [-e <testname>] [-F <separator>] [-f <namefile>]
        [-m <magicfiles>] [-P <parameter=value>] <file> ...
        file -C [-m <magicfiles>]
        file [--help]
./Shindujah_files: line 13: file1: command not found
./Shindujah_files: line 14: file5: command not found
./Shindujah_files: line 15: filename1: command not found
./Shindujah_files: line 16: filename2: command not found
./Shindujah_files: line 17: getname: command not found
./Shindujah_files: line 18: Getname: command not found
./Shindujah_files: line 19: hello1: command not found
./Shindujah_files: line 20: hello2: command not found
./Shindujah_files: line 21: helloworld1: command not found
./Shindujah_files: line 22: Music: command not found
```

iii. Write a script that takes filename1 and filename2 as command line arguments. The script must use the “tr” command to convert the contents in the file filename1 to uppercase and store it in the file filename2. [see the man page of the tr command]

```
#!/bin/bash
cat > filename1

cat > filename2

cat filename1 | tr "[a-z]" "[A-Z]" >> filename2
```

```
user@user-HP-Pavilion-Laptop-14-bk0xx:~$ vi commandargument
user@user-HP-Pavilion-Laptop-14-bk0xx:~$ chmod 700 commandargument
user@user-HP-Pavilion-Laptop-14-bk0xx:~$ ./commandargument
hello shindujah
user@user-HP-Pavilion-Laptop-14-bk0xx:~$ cat filename1
hello shindujah
user@user-HP-Pavilion-Laptop-14-bk0xx:~$ cat filename2
HELLO SHINDUJAH
```

7) Assume that you have file called catFile (permissions 700), which includes following commands.

```
user@user-HP-Pavilion-Laptop-14-bk0xx:~$ cat>catFile
#!/bin/bash
#
#Script to print file
#
if cat $1
then
echo -e "\n\nFile $1, found and successfully echoed"
fi
user@user-HP-Pavilion-Laptop-14-bk0xx:~$ chmod 700 catFile
user@user-HP-Pavilion-Laptop-14-bk0xx:~$ ./catFile file1
hello4
hello3
hello2
hello1

File file1, found and successfully echoed
user@user-HP-Pavilion-Laptop-14-bk0xx:~$
```

- 8) Assume that you have a file called rmFile (permissions 700) containing following commands

```
#!/bin/bash
#Script to test rm command and exist status
#
if rm $1
then
echo "$1 file deleted"
fi
```

- i. catFile file exists on your disk and you give command, \$ ./rmFile catFile what will be the output?

```
user@user-HP-Pavilion-Laptop-14-bk0xx:~$ vi rmFile
user@user-HP-Pavilion-Laptop-14-bk0xx:~$ chmod 700 rmFile
user@user-HP-Pavilion-Laptop-14-bk0xx:~$ ./rmFile catFile
catFile file deleted
user@user-HP-Pavilion-Laptop-14-bk0xx:~$
```

- ii. If bar file not present on your disk and you give command, \$ ./rmFile bar what will be output?

```
user@user-HP-Pavilion-Laptop-14-bk0xx:~$ ./rmFile bar
rm: cannot remove 'bar': No such file or directory
user@user-HP-Pavilion-Laptop-14-bk0xx:~$
```

- iii. And if you type \$ ./rmFile What will be output?

```
user@user-HP-Pavilion-Laptop-14-bk0xx:~$ ./rmFile
rm: missing operand
Try 'rm --help' for more information.
user@user-HP-Pavilion-Laptop-14-bk0xx:~$
```



9) i. Write a script that prints out whether the 1st argument provided is positive, negative or zero. If no arguments are provided by the user, then print an error message.

```
#!/bin/bash
#Script for check argument whether positive or negative or zero
#
if test $# -eq 0
then
echo "$0 :Error no arguments"
exit 1
fi

if test $1 -gt 0
then
echo "$1 number is positive"

elif test $1 -lt 0
then
echo "$1 number is negative"
else
echo "$1 is Zero"
fi
```

```
user@user-HP-Pavilion-Laptop-14-bk0xx:~$ vi checkargu1
user@user-HP-Pavilion-Laptop-14-bk0xx:~$ chmod 700 checkargu1
user@user-HP-Pavilion-Laptop-14-bk0xx:~$ ./checkargu1 3
3 number is positive
user@user-HP-Pavilion-Laptop-14-bk0xx:~$ ./checkargu1 -1
-1 number is negative
user@user-HP-Pavilion-Laptop-14-bk0xx:~$ ./checkargu1 0
0 is Zero
user@user-HP-Pavilion-Laptop-14-bk0xx:~$ ./checkargu1
./checkargu1 :Error no arguments
```

- ii. Write a script that prints out whether the 1st argument provided is odd, even or zero. If no arguments are provided by the user, then print an error message.

```
#!/bin/bash
#Script for check argument whether odd or even or zero
#
if test $# -eq 0
then
echo "$0:Error noarguments"
exit 1
fi

if test `expr $1 % 2` -eq 0
then
echo "$1 is even number"
elif test `expr $1 % 2` -ne 0
then
echo "$1 is odd number"
else
echo "$1 is zero"
fi
```

```
user@user-HP-Pavilion-Laptop-14-bk0xx:~$ vi checkargu2
user@user-HP-Pavilion-Laptop-14-bk0xx:~$ chmod 700 checkargu2
user@user-HP-Pavilion-Laptop-14-bk0xx:~$ ./checkargu2 6
6 is even number
user@user-HP-Pavilion-Laptop-14-bk0xx:~$ ./checkargu2 3
3 is odd number
user@user-HP-Pavilion-Laptop-14-bk0xx:~$ ./checkargu2 0
0 is even number
user@user-HP-Pavilion-Laptop-14-bk0xx:~$ ./checkargu2
./checkargu2:Error noarguments
```