UCS1304 Unix and Shell Programming
AY: 2021-22

Name: *Krithika Swamninathan*
Reg. No.: *205001057*

Assignment 7 – Simple Bash Commands

1. Write a shell script that prints “Shell Scripting is Fun!” on the screen. Modify the shell script above to include a variable. The variable will hold the contents of themessage “Shell Scripting is Fun!”

```
asec20@sel20-HP-Compaq-Pro-6305-SFF:~$ cat> script.sh
echo "Shell Scripting is Fun!"
^C

asec20@sel20-HP-Compaq-Pro-6305-SFF:~$ cat script.sh
echo "Shell Scripting is Fun!"

asec20@sel20-HP-Compaq-Pro-6305-SFF:~$ bash ./script.sh
Shell Scripting is Fun!
```

Modified:

```
asec20@sel20-HP-Compaq-Pro-6305-SFF:~$ cat> script.sh
msg="Shell Scripting is Fun!"
echo $msg
^C

asec20@sel20-HP-Compaq-Pro-6305-SFF:~$ cat script.sh
msg="Shell Scripting is Fun!"
echo $msg

asec20@sel20-HP-Compaq-Pro-6305-SFF:~$ bash ./script.sh
Shell Scripting is Fun!
```

2. Print the values of the environment variables HOME, USER, SHELL and PATH with set, printenv and echo.

```
asec20@sel20-HP-Compaq-Pro-6305-SFF:~$ echo $HOME
/home/cd6a
asec20@sel20-HP-Compaq-Pro-6305-SFF:~$ echo $USER
asec20
asec20@sel20-HP-Compaq-Pro-6305-SFF:~$ echo $SHELL
/bin/bash
asec20@sel20-HP-Compaq-Pro-6305-SFF:~$ echo $PATH
/home/cd6a/bin:/home/cd6a/.local/bin:/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin:/
usr/games:/usr/local/games:/snap/bin

asec20@sel20-HP-Compaq-Pro-6305-SFF:~$ printenv HOME
/home/cd6a
asec20@sel20-HP-Compaq-Pro-6305-SFF:~$ printenv USER
asec20
asec20@sel20-HP-Compaq-Pro-6305-SFF:~$ printenv SHELL
/bin/bash
asec20@sel20-HP-Compaq-Pro-6305-SFF:~$ printenv PATH
```

/home/cd6a/bin:/home/cd6a/.local/bin:/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin:/usr/games:/usr/local/games:/snap/bin

3. Store the output of the command "hostname" in a variable. Display "This script is running on _." where "_" is the output of the "hostname" command.

```
asec20@sel20-HP-Compaq-Pro-6305-SFF:~$ cat> script3.sh
var=$HOSTNAME
echo "This script is running on $var."
^C

asec20@sel20-HP-Compaq-Pro-6305-SFF:~$ cat script3.sh
var=$HOSTNAME
echo "This script is running on $var."

asec20@sel20-HP-Compaq-Pro-6305-SFF:~$ bash ./script3.sh
This script is running on sel20-HP-Compaq-Pro-6305-SFF.
```

4. Get two numbers a and b from user using read statement. Do the following:

- **a. Add the two numbers**
- **b. Subtract the numbers**
- **c. Multiply the numbers**
- **d. Divide the numbers.**

Print the result.

```
asec20@sel20-HP-Compaq-Pro-6305-SFF:~$ cat> script4.sh
echo "Enter two numbers: "
read a b
sum=$(($a+$b))
echo "Sum: $sum"
diff=$(($a-$b))
echo "Difference: $diff"
prod=$(($a*$b))
echo "Product: $prod"
div=$(($a/$b))
echo "Quotient: $div"
rem=$(($a%$b))
echo "Remainder: $rem"
^C

asec20@sel20-HP-Compaq-Pro-6305-SFF:~$ cat script4.sh
echo "Enter two numbers: "
read a b
sum=$(($a+$b))
echo "Sum: $sum"
diff=$(($a-$b))
echo "Difference: $diff"
prod=$(($a*$b))
echo "Product: $prod"
```

```
div=$(($a/$b))
echo "Quotient: $div"
rem=$(($a%$b))
echo "Remainder: $rem"

asec20@sel20-HP-Compaq-Pro-6305-SFF:~$ bash ./script4.sh
Enter two numbers:
20 15
Sum: 35
Difference: 5
Product: 300
Quotient: 1
Remainder: 5
```

5. Get length and breadth for a rectangle and radius for a circle using command line argument. Calculate area and perimeter of the rectangle and also area and circumference of a Use the special character data types and display the arguments using them.

```
asec20@sel20-HP-Compaq-Pro-6305-SFF:~$ cat> script5.sh
area1=$(($1*$2))
echo "Area of rectangle: $area1"
peri1=$((2*($1+$2)))
echo "Perimeter of rectangle: $peri1"
area2=$((3*$3*$3))
echo "Area of circle: $area2"
circ=$((2*3*$3))
echo "Circumference of circle: $circ"
^C

asec20@sel20-HP-Compaq-Pro-6305-SFF:~$ cat script5.sh
area1=$(($1*$2))
echo "Area of rectangle: $area1"
peri1=$((2*($1+$2)))
echo "Perimeter of rectangle: $peri1"
area2=$((3*$3*$3))
echo "Area of circle: $area2"
circ=$((2*3*$3))
echo "Circumference of circle: $circ"

asec20@sel20-HP-Compaq-Pro-6305-SFF:~$ bash ./script5.sh 12 4 5
Area of rectangle: 48
Perimeter of rectangle: 32
Area of circle: 75
Circumference of circle: 30
```

6. Temperature of a city in Fahrenheit degree is input through the keyboard. Write a program to convert this temperature into Centigrade degrees.

```
asec20@sel20-HP-Compaq-Pro-6305-SFF:~$ cat> script6.sh
```

```
read -p "Enter temperature in Fahrenheit: " f
c=$(((($f-32)*5/9))
echo "Temperature in Celsius: $c"
^C

asec20@sel20-HP-Compaq-Pro-6305-SFF:~$ cat script6.sh
read -p "Enter temperature in Fahrenheit: " f
c=$(((($f-32)*5/9))
echo "Temperature in Celsius: $c"

asec20@sel20-HP-Compaq-Pro-6305-SFF:~$ bash ./script6.sh
Enter temperature in Fahrenheit: 95
Temperature in Celsius: 35
```

7. Write a shell script to calculate the net salary of an employee in a particular month considering various allowances (TA, DA, HRA) and deductions (INCOME TAX, PROVIDEND FUND) as:
a. TA=15 percent of basic salary
b. DA=2 percent of basic salary
c. HRA=10 percent of basic salary
d. INCOME TAX=5 percent of salary
e. PROVIDEND FUND=10 percent of salary

```
asec20@sel20-HP-Compaq-Pro-6305-SFF:~$ cat> script7.sh
read -p "Enter Basic Pay: " BP
TA=$((15))
DA=$((2))
HRA=$((10))
TAX=$((5))
FUND=$((10))
allowances=$(($TA + $DA + $HRA))
deductions=$(($TAX + $FUND))
sub=$(($allowances - $deductions))
mult=`echo \(\($sub \* 0.01\) \* $BP\) + $BP|bc`
echo $mult
^C

asec20@sel20-HP-Compaq-Pro-6305-SFF:~$ cat script7.sh
read -p "Enter Basic Pay: " BP
TA=$((15))
DA=$((2))
HRA=$((10))
TAX=$((5))
FUND=$((10))
allowances=$(($TA + $DA + $HRA))
deductions=$(($TAX + $FUND))
sub=$(($allowances - $deductions))
mult=`echo \(\($sub \* 0.01\) \* $BP\) + $BP|bc`
```

```
echo $mult
asec20@sel20-HP-Compaq-Pro-6305-SFF:~$ bash ./script7.sh
Enter Basic Pay: 10000
11200.00
```

8. In a town, the percentage of men is 52. Rest all are women. The percentage of total literacy is 48. If total percentage of literate men is 35 of the total population, WAP to find the total number of illiterate men and women. The population of the town is 80,000.

```
asec20@sel20-HP-Compaq-Pro-6305-SFF:~$ cat> script8.sh
population=$((80000))
literate=`echo \(48 \* 0.01\) \* $population|bc`
echo "Number of literate people: $literate"
men=`echo \(35 \* 0.01\) \* $population|bc`
women=`echo $literate - $men|bc`
echo "Number of literate men: $men"
echo "Number of literate women: $women"
^C

asec20@sel20-HP-Compaq-Pro-6305-SFF:~$ cat script8.sh
population=$((80000))
literate=`echo \(48 \* 0.01\) \* $population|bc`
echo "Number of literate people: $literate"
men=`echo \(35 \* 0.01\) \* $population|bc`
women=`echo $literate - $men|bc`
echo "Number of literate men: $men"
echo "Number of literate women: $women"

asec20@sel20-HP-Compaq-Pro-6305-SFF:~$ bash ./script8.sh
Number of literate people: 38400.00
Number of literate men: 28000.00
Number of literate women: 10400.00
```

9. Write a shell script that displays “man”,”bear”,”pig”,”dog”,”cat”,and “sheep” on the screen with each appearing on a separate line. Use special characters to display the filename, no of parameters, display the arguments each on one line, use appropriate command to dispaly the differences between $@, $*. Explain how about the status code of the script.

```
asec20@sel20-HP-Compaq-Pro-6305-SFF:~$ cat> script9.sh
echo "using @"
for var in "$@"
do
	echo $var
done
echo "using *"
for var in "$*"
do
	echo $var
```

```
done
echo "filename: $0"
echo "number of param: $#"
echo "exit status: $?"
^C

asec20@sel20-HP-Compaq-Pro-6305-SFF:~$ cat script9.sh
echo "using @"
for var in "$@"
do
	echo $var
done
echo "using *"
for var in "$*"
do
	echo $var
done
echo "filename: $0"
echo "number of param: $#"
echo "exit status: $?"

asec20@sel20-HP-Compaq-Pro-6305-SFF:~$ bash ./script9.sh "man" "bear" "pig" "dog" "cat"
"sheep"
using @
man
bear
pig
dog
cat
sheep
using *
man bear pig dog cat sheep
filename: ./script9.sh
number of param: 6
exit status: 0
```

10. Write a shell script that prompts the user for a name of a file or directory and reports if it is a regular file, a directory, or another type of file. Also perform an ls command against the file or directory with the long listing option.

```
asec20@sel20-HP-Compaq-Pro-6305-SFF:~$ cat> script10.sh
read -p "Enter filename: " filename

if [ -e "$filename" ]; then
	if [ -f "$filename" ]; then
		echo "$filename is a normal file"
		ls -l $filename
	elif [ -d "$filename" ]; then
		echo "$filename is a directory"
```

```
                ls -l $filename
        else
                echo "$filename is a spl file"
        fi
fi
^C

asec20@sel20-HP-Compaq-Pro-6305-SFF:~$ cat script10.sh
read -p "Enter filename: " filename

if [ -e "$filename" ]; then
        if [ -f "$filename" ]; then
                echo "$filename is a normal file"
                ls -l $filename
        elif [ -d "$filename" ]; then
                echo "$filename is a directory"
                ls -l $filename
        else
                echo "$filename is a spl file"
        fi
fi

asec20@sel20-HP-Compaq-Pro-6305-SFF:~$ bash ./script10.sh
Enter filename: dictionary
dictionaryis a directory
total 0
```

11. Modify the previous script to that it accepts unlimited number of files and directories as arguments and display the information about it. (use cat for files and ls-l for directories)

```
asec20@sel20-HP-Compaq-Pro-6305-SFF:~$ cat> script11.sh
for filename in "$@"
do
   if [ -e "$filename" ]; then
        if [ -f "$filename" ]; then
                        echo "$filename is a normal file"
        cat $filename
                elif [ -d "$filename" ]; then
                echo "$filename is a directory"
                ls -l $filename
                else
        echo "$filename is a spl file"
                fi
   fi
done
^C

asec20@sel20-HP-Compaq-Pro-6305-SFF:~$ cat script11.sh
for filename in "$@"
```

```
do
   if [ -e "$filename" ]; then
        if [ -f "$filename" ]; then
                        echo "$filename is a normal file"
        cat $filename
                elif [ -d "$filename" ]; then
                echo "$filename is a directory"
                ls -l $filename
                else
        echo "$filename is a spl file"
                fi
   fi
done
```

```
asec20@sel20-HP-Compaq-Pro-6305-SFF:~$ bash ./script11.sh shapes.sh newfile.sh
shapes.sh is a normal file
#!/bin/bash
echo "rectangle"
perim=`echo $1 + $1 + $2 + $2|bc`
echo "perimenter: $perim"
area=`echo $1 \* $2|bc`
echo "area: $area"
echo "circle"
circum=`echo 2 \* 3.14 \* $3|bc`
echo "circumference: $circum"
ar=`echo 3.14 \* $3 \* $3|bc`
echo "area: $ar"
newfile.sh is a normal file
#!/bin/bash
echo $SHELL

dictionary is a directory
total 0
```

12. Write a shell script to display the current date and cut down the month of the date and store it in the file date.txt. Use ` ` in the command to store the content in the file and display the file. Also use an alias function to cut down the day of the week and execute the command.

```
asec20@sel20-HP-Compaq-Pro-6305-SFF:~$ cat> script12.sh
date
month=`date | cut -d ' ' -f3 > date.txt`
echo "month stored in date.txt"
alias d="date"
date | cut -d ' ' -f1
^C

asec20@sel20-HP-Compaq-Pro-6305-SFF:~$ cat script12.sh
date
month=`date | cut -d ' ' -f3 > date.txt`
```

```
echo "month stored in date.txt"
alias d="date"
date | cut -d ' ' -f1

asec20@sel20-HP-Compaq-Pro-6305-SFF:~$ bash ./script12.sh
Friday 03 December 2021 03:00:17 PM IST
Month stored in date.txt...
Friday

asec20@sel20-HP-Compaq-Pro-6305-SFF:~$ cat date.txt
December
```

13. Create the following files and change the permissions specified
File1 701
File2 400
File3 300
File4 676
File5 045
File6 177
File7 234
File8 507
Write a shell script to find the number of readable, writable and executable files.

```
kri@kri-ubuntu:~/workspace/asst7$ cat> script13.sh
touch file1 file2 file3 file4 file5 file6 file7 file8
chmod 701 file1
chmod 400 file2
chmod 300 file3
chmod 676 file4
chmod 045 file5
chmod 177 file6
chmod 234 file7
chmod 507 file8
ls -l file1 file2 file3 file4 file5 file6 file7 file8

r=$((0))
w=$((0))
x=$((0))

for var in file1 file2 file3 file4 file5 file6 file7 file8
do
        if [ -r "$var" ]; then
                r=$(($r+1))
        elif [ -w "$var" ]; then
                w=$(($w+1))
        else
                x=$(($x+1))
        fi
done
```

Name: *Krithika Swamninathan*
Reg. No.: *205001057*

```
kri@kri-ubuntu:~/workspace/asst7$ cat script13.sh
touch file1 file2 file3 file4 file5 file6 file7 file8
chmod 701 file1
chmod 400 file2
chmod 300 file3
chmod 676 file4
chmod 045 file5
chmod 177 file6
chmod 234 file7
chmod 507 file8
ls -l file1 file2 file3 file4 file5 file6 file7 file8

r=$((0))
w=$((0))
x=$((0))

for var in file1 file2 file3 file4 file5 file6 file7 file8
do
        if [ -r "$var" ]; then
                r=$(($r+1))
        elif [ -w "$var" ]; then
                w=$(($w+1))
        else
                x=$(($x+1))
        fi
done

kri@kri-ubuntu:~/workspace/asst7$ bash ./script13.sh
-rwx-----x 1 kri kri 0 Dec  9 19:50 file1
-r-------- 1 kri kri 0 Dec  9 19:50 file2
--wx------ 1 kri kri 0 Dec  9 19:50 file3
-rw-rwxrw- 1 kri kri 0 Dec  9 19:50 file4
----r--r-x 1 kri kri 0 Dec  9 19:50 file5
---xrwxrwx 1 kri kri 0 Dec  9 19:50 file6
--w--wxr-- 1 kri kri 0 Dec  9 19:50 file7
-r-x---rwx 1 kri kri 0 Dec  9 19:50 file8
```

14. Write the shell script that renames all files in the current directory that end in ".jpg" to begin with today's date in the following format: YYYY-MM-DD. For example, if a picture of my cat was in the current directory and today was October 31,2016 it would change name from "mycat.jpg" to "2016–10–31-mycat.jpg".

```
kri@kri-ubuntu:~/workspace/asst7$ cat> script14.sh
d=$(date '+%Y-%m-%d')
```

Name: *Krithika Swamninathan*
Reg. No.: *205001057*

```
echo "before"
ls -l
for file in *.jpg
do
        mv "$file" "$d\_$file"
done
echo "after"
ls -l
```

```
kri@kri-ubuntu:~/workspace/asst7$ cat script14.sh
d=$(date '+%Y-%m-%d')
echo "before"
ls -l
for file in *.jpg
do
        mv "$file" "$d\_$file"
done
echo "after"
ls -l
```

```
kri@kri-ubuntu:~/workspace/asst7$ bash ./script14.sh
before
total 8
-rwx-----x 1 kri kri   0 Dec  9 19:50 file1
-r-------- 1 kri kri   0 Dec  9 19:50 file2
--wx------ 1 kri kri   0 Dec  9 19:50 file3
-rw-rwxrw- 1 kri kri   0 Dec  9 19:50 file4
----r--r-x 1 kri kri   0 Dec  9 19:50 file5
---xrwxrwx 1 kri kri   0 Dec  9 19:50 file6
--w--wxr-- 1 kri kri   0 Dec  9 19:50 file7
-r-x---rwx 1 kri kri   0 Dec  9 19:50 file8
-rw-rw-r-- 1 kri kri 483 Dec  9 19:49 script13.sh
-rw-rw-r-- 1 kri kri 115 Dec  9 19:51 script14.sh
mv: cannot stat '*.jpg': No such file or directory
after
total 8
-rwx-----x 1 kri kri   0 Dec  9 19:50 file1
-r-------- 1 kri kri   0 Dec  9 19:50 file2
--wx------ 1 kri kri   0 Dec  9 19:50 file3
-rw-rwxrw- 1 kri kri   0 Dec  9 19:50 file4
----r--r-x 1 kri kri   0 Dec  9 19:50 file5
---xrwxrwx 1 kri kri   0 Dec  9 19:50 file6
--w--wxr-- 1 kri kri   0 Dec  9 19:50 file7
-r-x---rwx 1 kri kri   0 Dec  9 19:50 file8
-rw-rw-r-- 1 kri kri 483 Dec  9 19:49 script13.sh
-rw-rw-r-- 1 kri kri 115 Dec  9 19:51 script14.sh
```

```
kri@kri-ubuntu:~/workspace/asst7$ ls
file1  file3  file5  file7  script13.sh
```

```
file2  file4  file6  file8  script14.sh
```

15. Write a script that executes the command “cat/etc/shadow”. If the command return a 0 exit status, report “command succeeded” and exit with a 0 exit status. If the command returns a non-zero exit status, report “Command failed” and exit with a 1 exit status.

```
asec20@sel20-HP-Compaq-Pro-6305-SFF:~$ cat> script15.sh
d=$(date '+%Y-%m-%d')
echo "before"
ls -l
for file in *.jpg
do
        mv "$file" "$d\_$file"
done
echo "after"
ls -l
^C

asec20@sel20-HP-Compaq-Pro-6305-SFF:~$ cat script15.sh
d=$(date '+%Y-%m-%d')
echo "before"
ls -l
for file in *.jpg
do
        mv "$file" "$d\_$file"
done
echo "after"
ls -l

asec20@sel20-HP-Compaq-Pro-6305-SFF:~$ bash ./script15.sh
cat: /etc/shadow: Permission denied
command succeeded
```