

EXPERIMENT NO. 2

Student Name and Roll Number: Chirag Sardana and 19CSU071
Semester /Section: 5 th / FS A1
Link to Code: https://github.com/chiragsardana/Skill-Developement/tree/master/OperatingSystem/Experiment-2nd
Date: 10 th August 2021
Faculty Signature:
Marks:

Objective: To write the shell programming code for the following.
Outcome: Student is able to write code in shell programming
Problem Statement: a) Write A Shell Program of Hello World b) Write a shell program to find factorial of a number. c) Write a shell program to find gross salary of an employee. d) Write a shell program to display the menu and execute instructions accordingly (i)List of files (ii)Process Status (iii) Date (iv) users in program (v) Quit
Background Study: A shell script is a file with a set of commands in it. The shell reads this file and executes the instructions as if they were input directly on the command line. A shell is a command-line interpreter and operations such as file manipulation, program execution and text printing is performed by shell script. So, we will use vi editor to edit our files.
Question Bank: 1. What is a shell? Solution: the shell is a program that takes commands from the keyboard and gives them to the operating system to perform. On most Linux systems a program called bash (which stands for Bourne Again SHell, an enhanced version of the original Unix shell program, sh , written by Steve Bourne) acts as the

shell program. Besides **bash**, there are other shell programs available for Linux systems. These include: **ksh**, **tcsh** and **zsh**.

2. What is the significance of \$#?

Solution: \$# is **the number of arguments**, but remember it will be different in a function. \$# is the number of positional parameters passed to the script, shell, or shell function. This is because, while a shell function is running, the positional parameters are temporarily replaced with the arguments to the function.

3. What are the different types of commonly used shells on a typical Linux system?

Solution:

Shell	Complete path-name	Prompt for root user	Prompt for non root user
Bourne shell (sh)	/bin/sh and /sbin/sh	#	\$
GNU Bourne-Again shell (bash)	/bin/bash	bash-VersionNumber#	bash-VersionNumber\$
C shell (csh)	/bin/csh	#	%
Korn shell (ksh)	/bin/ksh	#	\$
Z Shell (zsh)	/bin/zsh	<hostname>#	<hostname>%

4. How will you pass and access arguments to a script in Linux?

Solution: The shell command and any arguments to that command appear as *numbered* shell variables: \$0 has the string value of the command itself, something like script, ./script, /home/user/bin/script or whatever. Any arguments appear as "\$1", "\$2", "\$3" and so on. The count of arguments is in the shell variable "\$#".

Common ways of dealing with this involve shell commands getopts and shift. getopts is a lot like the C getopt() library function. shift moves the value of \$2 to \$1, \$3 to \$2, and so

on; \$# gets decremented. Code ends up looking at the value of "\$1", doing things using a case...esac to decide on an action, and then doing a shift to move \$1 to the next argument. It only ever has to examine \$1, and maybe \$#.

5. Use sed command to replace the content of the file (emulate tac command)

Solution:

Eg.

if cat file1

ABCD

EFGH

Then O/p should be

EFGH

ABCD

sed '1! G; h;\$!d' file1

Post Your Answer

Here G command appends to the pattern space,

h command copies pattern buffer to hold buffer

and d command deletes the current pattern space.

Student Work Area

Algorithm/Flowchart/Code/Sample Outputs

→ Hello World Program

```
(base) chiragsardana@MacBook new % vim HelloWorld.sh
```

```
#!/bin/bash
echo "Hello World"
```

~~~~~

: wq

```
(base) chiragsardana@MacBook new % zsh HelloWorld.sh
HelloWorld.sh:1: no such file or directory: !/bin/bash
Hello World
(base) chiragsardana@MacBook new %
```

```
#!/bin/sh
echo "Enter a number"

read num

fact=1

while [ $num -gt 1 ]
do
    fact=$((fact * num)) #fact = fact * num
    num=$((num - 1))     #num = num - 1
done

echo "The Factorial of a number $num: $fact"
"
```

"Factorial.sh" 15L, 209B

```
[(base) chiragsardana@MacBook new % zsh Factorial.sh
Factorial.sh:1: no such file or directory: !/bin/sh
Enter a number
4
The Factorial of a number 1: 24

[(base) chiragsardana@MacBook new % zsh Factorial.sh
Factorial.sh:1: no such file or directory: !/bin/sh
Enter a number
5
The Factorial of a number 1: 120

(base) chiragsardana@MacBook new % █
```

## →Gross Salary Program

```
((base) chiragsardana@MacBook new % vim GrossSalary.sh
```

```
echo -e "Enter ur basic salary \c"
read sal
if [ $sal -ge 1000 ]
then
    da=`expr $sal \* 40 / 100`
    ha=`expr $sal \* 20 / 100`
    Nsal=`expr $sal + $da + $ha`
    echo "ur Basic Salary      $sal "
    echo "ur Dearness Allowance  $da "
    echo "Ur House rent           $ha "
    echo "                        _____"
    echo "Ur Net Salary is    Rs. $Nsal "else
echo "Pls enter basic salary greater than 1000 "
fi
```

```
"GrossSalary.sh" 14L, 502B
```

```
(base) chiragsardana@MacBook new % zsh GrossSalary.sh
GrossSalary.sh:1: no such file or directory: !/bin/sh
enter the basic salary:
1000
GrossSalary.sh:13: bad pattern: baseSalary+((40/100)*baseSalary)+((20/100)*baseS
alary)
(base) chiragsardana@MacBook new % zsh GrossSalary.sh
GrossSalary.sh:1: no such file or directory: !/bin/sh
enter the basic salary:
1000
GrossSalary.sh:13: bad pattern: baseSalary+((40/100)*baseSalary)+((20/100)*baseS
alary)
(base) chiragsardana@MacBook new % vim GrossSalary.sh
(base) chiragsardana@MacBook new % zsh GrossSalary.sh
Enter ur basic salary 1000
ur Basic Salary          1000
ur Dearness Allowance    400
Ur House rent            200
-----
Ur Net Salary is Rs. 1600 else
Pls enter basic salary greater than 1000
(base) chiragsardana@MacBook new %
```



## →Display Menu Program

```
(base) chiragsardana@MacBook new % vim listProgram.sh
```

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

```
echo "Menu:"  
echo "1. List of files "  
echo "2. Process Status "  
echo "3. Date "  
echo "4. users in program "  
echo "5. Quit"  
  
while :  
do  
echo "Enter your choice: "  
read ch  
  
case $ch in  
    1) ls;;  
    2) pgrep chrome;;  
    3) date;;  
    4) users;;  
    5) exit;;  
    *) echo "Invalid Choice"
```

```
esac  
done
```

```
~  
~  
~  
~  
~  
~  
~  
~  
~  
~  
~
```

```
"listProgram.sh" 23L, 293B
```

```
[(base) chiragsardana@MacBook new % zsh listProgram.sh
listProgram.sh:1: no such file or directory: !/bin/bash
Menu:
1. List of files
2. Process Status
3. Date
4. users in program
5. Quit
Enter your choice:
1
Factorial.sh      HelloWorld.sh      file_name2      listProgram.sh
GrossSalary.sh   file_name1        file_name3      trail.sh
Enter your choice:
2
12736
13099
14759
32973
37267
38856
Enter your choice:
3
Tue Aug 17 09:11:11 IST 2021
Enter your choice:
4
chiragsardana
Enter your choice:
5
```