## CS282 – Spring 2019

# Lab 17 – Unix – Shell Scripts – Loops and Positional / Command-Line Parameters

### Objectives

The objective of this lab is for students to write Unix Shell Scripts that involve loops and case structure in Unix along-with usage of Positional / Command-Line parameters in Shell scripts. Students are advised to refer to the slides “***Unix - Commands.pptx”*** uploaded on Blackboard and Chapters 2 and 3 of textbook ***Beginning Modern Unix***. Also, students are advised to further research relevant online material and Library resources for the same.

### You will be assigned a partner to work with or you may work with a partner of your choice

The instructor will randomly assign lab partners to work together on each lab. You have the choice to work independently.

Ask the instructor if you are having problems that you cannot solve together.

### Part 1 – Unix Shell Script Description – Usage of for loop with Positional /Command-line Parameters (20 Minutes)

Write a Shell Script in your Unix-based operating system (FreeBSD, XCode, etc.) that accepts numbers from the command line and prints the sum of those numbers. This program is for students to learn Syntax and usage of Positional parameters / Conditional Parameters along-with for loop in Shell Scripts.

Remember, following steps have to be followed to write the Shell Script:

1. At the $ prompt, make sure that you are in your home directory by typing the command pwd.
2. Then change to this directory, as follows:

cd CS282

You may confirm that you are inside CS282, by giving the command, pwd.

1. Then open a new file sum\_cmdLine.sh through vi editor, for writing your Shell Script.

vi sum\_cmdLine.sh

Program is provided here: Understand and trace out the program logic before you start typing it out.

#Shell Script to find sum of numbers entered at command-line

clear

sum=0

for i in $\*

do

sum=`expr $sum + $i`

done

echo “Sum is $sum”

exit 0

Save and exit by :wq

1. Give execute permission to reverse.sh Shell Script, by the following command:

chmod +x sum\_cmdLine.sh

1. Execute your Shell Script as follows:

./sum\_cmdLine.sh 10 20 30

You may give any numbers and any amount of numbers after the program name.

### Part 2 – Unix Shell Script Description – Usage of case...esac structure along-with shift statement and Positional Parameters - Write a Shell Script with your own command options (20 Minutes)

Write a Shell Script in your Unix-based operating system (FreeBSD, XCode, etc.) that has custom command options. As a Unix administrator, this Shell Script will become your one place handy utility to access all commands based on the command option entered. This program is for students to learn Syntax of case..esac structure along-with shift statement and Positional Parameters. Depending upon the option entered the files following the option are either moved (dragged) to temp directory or removed (interactively).

Program is provided here: Understand and trace out the program logic before you start typing it out.

#Shell Script to demonstrate switch..case and command line parameters

clear

optX=false

case “$1” in

-drag)

optX=true

shift

;;

-del)

shift

;;

\*)

echo “Invalid Argument: $1” 1>&2

exit 1

;;

esac

if test $optX = true

then

for i in $\*

do

mv $i ./temp

done

else

for i in $\*

do

rm -i $i

done

fi

exit 0

Students are advised to make few empty files, say, temp1, temp2, temp3, etc. Also create a temporary directory, temp.

If you have named your Shell Script as MyCmd.sh, then it will be executed as:

./MyCmd.sh -drag temp1 temp2

Or

./MyCmd.sh -del temp1 temp2

### Deliverables

Show the execution of all your Shell Scripts to your instructor, once all are completed. For this Lab17, you will be graded based on showing the execution of your Shell Scripts to the Instructor.

### Additional Programs for Practice at Home – Non-Deliverables

Practice writing more Shell Scripts, Eg:

* 1. Write a Shell Script that has menu driven sub-routines.
  2. Practice the Additional Programs of previous Unix Labs.