Department of Computer Science & Applications
Panjab University, Chandigarh
MCA 1st Semester - October 2018 (By Ravinder Kumar Singla)
Submission Date: End of Semester CS-56: Linux Operating System

1) **Write a shell script called '*hwt*' that given a person's uid, tell you how many times that person is logged on.**
2) **Write a shell script called *lsdirs* that lists just the directories in the current directory.**
3) **Write a shell script called '*see'* taking a filename as argument which ls's the file if it's a directory and more's the file otherwise.**
4) **Write a shell script '*rfiles*' that reports how many bytes are used by regular files in a directory.**
5) **In many versions of Linux, there is an -i argument for cp so that you will be prompted for confirmation if you are about to overwrite a file. Write a script called '*cpi'* which will prompt if necessary without using the -i argument.**
6) **Write a shell program to find the largest number among the given 3 numbers as input.**
7) **Write a shell program to reverse a given 5-digit number.**
8) **Write a shell program to check whether a given string is a palindrome or not (without actually reversing the string).**
9) **Write a shell program to find the greatest common divisor (GCD) for the two given numbers.**
10) **Write script called sayhello, put this script into your startup file called, .bash_profile, the script should run as soon as you logon to system, and it print any one of the following message in infobox using dialog utility, if installed in your system, If dialog utility is not installed then use echo statement to print message : -**

    Good Morning
    Good Afternoon
    Good Evening, according to system time.

11) **Write a shell script that**
    a) **Takes as command-line options any number of text files**
    b) **Reads in each of these files, converts all the letters to uppercase, and then stores the results in a file of the same name but with a .caps extension**
12) **Write a shell script, fifteen, which prints out the fifteenth argument on the command line.**
13) **Write a shell script for adding first 10 integers.**
14) **Write a shell script to count number of characters, white spaces, words and tabs in a given text.**
15) **Write a shell script that lists the command-line arguments that were passed to the script.**
16) **Write a shell script that checks to see if the first command-line option is –i or –e. If it is –i, the program counts the number of lines in the file specified by the second command-line option that begins with the letter i. If the first option is –e, the program counts the number of lines in the file specified by the second command-line option that begins with the letter e. If the first command-line option is not –i or – e, the program prints a brief error message to the screen.**
17) **Write a shell script that reads the input file, translates all the characters in the input file into the uppercase, then stores the results in the specified output file.**
    Synopsis: upper –i infile –o outfile

18) **Write a shell script that reads all the files that are passed on the command-line and – depending on the option that was used – writes the files out in all uppercase letters, writes the files out in all lowercase letters or prints the files. [Write your own functions upper (), lower () and print ()]**

Please Turn Over

19) **Write a Linux shell script (/bin/bash) which has two command line parameters, a username and a file name. The script should determine the number of times the user with that username is logged into the system and write that number to the given file. If the file already exists, the script should output an error message. Some commands you might like to use are '*grep*', '*wc*' and '*who*'.**
20) **Write a Bourne shell program that prints the name and size of all files in the current directory that are larger than x bytes, where x is a command line argument. The output will include the names and sizes of files but not directories. Do not use "ls".**

    For example, if the program is saved to a file called bigfiles, and we run it as "bigfiles 4096", it will list all files larger than 4096 bytes.

21) **Write a C program which uses Linux system calls to create a new process. The child process should create an empty file called 'abc' and then terminate. The parent process should wait for the child process to terminate and then output the process number of the child process. Don't forget to check for error conditions.**
22) **Write a C program that takes a command and its arguments as arguments, runs the command piped into sort, and returns the status of that command. For example:**

    ./a.out diff file1 file2
    is equivalent to:
    diff file1 file2 | sort and returns the return status of diff.

23) **Give an example of a Linux file management system call and describe what it does.**
24) **Write a C program to run "*cmp file1 file2* > */dev/null*" using Linux system calls and then print the exit status of *cmp* to the screen. If status = 0 then files are same otherwise they are different.**
25) **Write a C program that illustrates the creation of child process using *fork* system call. One process finds sum of even series and other process finds sum of odd series.**
26) **Write a 'C' program to create a child process so that it can generate Fibonacci Series using command line arguments.**
27) **Write a C program to simulate cp command (File copy program) of Linux.**
28) **Write a simple C program in which a parent process sends the string "I love you" to a child process using a Linux pipe, and the child process, on receiving the string, will print it to standard output.**