MODULE: Operating System (DAC-Feb-2017)
TOPIC: Linux Commands
DEADLINE:18/03/2017 (4 PM)

Day 3 Assignments

Note 1: Refer following documents for examples and code templates

1) **ShellIntro.pdf**
2) **Beginning.Linux.Programming.4th.Edition.pdf**
3) **UNIX, concepts and applications**
   Book by Sumitabha Das

Part 1 (Shell scripting)

Note1: For the following assignments, accept only integer values from user.If user enters value other than integer display error message and terminate the program.
Note2: If you want to perform the following arithmetic operations with Floating point values make use of **Basic calculator** in your script.
(**Hint:** use **bc** command for Basic calculator)

1. Write a shell script to Print prime numbers from 1 to **n**. Read the value of **n** from user.
2. Write a script to check given number is even or odd.
3. Write a shell script to convert a decimal number to binary number.
4. Write a script to **swap** 2 numbers using intermediate variable.
5. Write a script to **swap** 2 numbers without using intermediate variable.
6. Write a script to reverse a number using while loop.
   Example:
   input : 12 output: 21
   input : 213 output: 312
   input : 125634 output:436521
7. print multiplication table of integer using while loop.
   Example:
   2 x 1 = 2
   2 x 2 = 4
   …….
8. Get year as an input from user and find whether year is leap year or not.
9. Write a script to read the number of rows to be displayed in the pattern and print following pattern using for loop:

1
2 3
4 5 6
……..

10. Write a script using **case** condition to do the following
    - Display “Press any key of keyboard and then press enter key”
    - If the given input is number display “The input is digit.” message
    - If the given input is lowercase letter then display “The input is lowercase Letter.” message
    - If the given input is uppercase letter then display “The input is Uppercase letter.” message
11. Write a for loop to display the outputs of **Date,pwd, df** commands. **df** command displays system disk usage details.
    (**Hint:** give these commands as input to for loop)
12. Write a script to take filename as argument and display the file exists or not
    Note: If the file exists in current working directory just give filename as argument, If not give absolute path of that file as argument.
13. Write a script to take directory name as argument and display the directory exists or not.
    Note: If the directory exists in current working directory just give filename as argument, If not give absolute path of that file as argument.
14. Read a file and display the contents of the file line by line using for loop and pass the file as command line argument to the script.
15. Read a file and display the contents of the file line by line using while loop and pass the file as command line argument to the script.
16. Write a shell script to read array elements as command line arguments assign the arguments to array and do the following
    - Display the length of the array.
    - Display the all elements and their index values.

    Example :

    let the array elements are as follows
    arr[0]=”zero”, arr[1]=”one”, arr[2]=”two”,

    output should be :
    length of the array : 3
    index 0 element is “zero”
    index 1 element is “one”
    index 2 element is “two”

17. Write a program to read array of 10 integers from user and find the smallest number in the array and print it.

18. Write a single shell script to do the following
    - read two numbers as input from terminal.
    - write Add,Sub and Mul functions to perform addition,subtraction and multiplication between two integers.
    - Display the results(**Note:** To display float results using basic calculator.)

Part 2
Question : 1
Description : Understanding and Controlling the process execution.

Write a C/Shell program which will take more than 10 minutes to execute(use **sleep** or write for/while loop). Run the program on the terminal. Stop the process using **Ctrl+z** key combination. Send that program to execute in background. List the processes executing in background. Bring that process to foreground.

Hint : use **jobs**,**bg** and **fg** command. Use man page for references or use google for examples.

<u>Part 3</u> **Read the following before you start the OS Assignments**

To understand the parent & child process concept.

- ❏ what is **getpid()** and how to use it..??
- ❏ what is **getppid()** and how to use it..??
- ❏ what is **fork()**? what happens when you use fork() in your program..??
- ❏ what is **wait()** / **waitpid()** and how to use it..??
- ❏ what are **zombie processes**? when they will get created.? How to kill the zombie processes..?
- ❏ what are **orphan processes**? when they will get created.?
- ❏ what is user space?
- ❏ what is kernel space / system space?
- ❏ What is **IPC**(inter process communication) and what is the need of it.
- ❏ what is **Unnamed PIPE?** how to create it? When we need this??
- ❏ what is **named PIPE**? how to create it? When we need this??
- ❏ What is the difference between **Unnamed PIPE** and **named PIPE**?
- ❏ what are execl,exec,execv and difference between them..??
- ❏ what is **semaphore**? Explore sem_init(), sem_create(), sem_destroy(), sem_unlink(), sem_wait(), sem_post().
- ❏ what is **mutex**? Difference between semaphore and mutex.

To understand the thread concepts

- ❏ what is **pthread_create()** and how to use it..?
- ❏ what is **pthread_exit()** and how to use it..?
- ❏ what is **pthread_join()** and how to use it..?