Assignment for 3rd Year 1stSemester Students

IT/S/313

[**Preliminaries:** Make sure you know the following bash shell command:-

cd, ls, cat, cp, man, pwd, time, env, expr, export PATH, gedit, vi, date, grep, gcc, diff, cmp, sed, awk, cal, wc, echo, if, while, watch, find, ps, uname, sort, tail, top, netstat, df, du, chmod, history, bc etc.]

1. **Shell Script (CO1)**

**a)** Write a ‘bash’ shell Script, using case statement to perform basic math operation as  
follows:  
+ addition  
- subtraction  
x multiplication  
/ division  
The name of script must be '<Roll No>\_script' which works as follows  
$> ./<Roll No>\_script 20 / 3

[The answer should be 6.66]

Also check for sufficient command line arguments and its correctness.

**b)** Use sed to replace a pattern with another pattern in a big file. Use grep to verify.

**c)** Find out how many people have logged in the system and how many processes are running in the system.

1. **File IO (CO2)**

**a)** Write a program (using open/read/write call) to copy a source file to a destination file. File Size should be at least 1GB. Use gettimeofday call to note down the time. Also, always use “perror” call to check the return status of a library/system call.

**b)** Repeat the same assignment above, using fopen/fread/fwrite call.

1. **SIGNAL Handling (CO3)**

**a)** Catch the signal ‘SIGINT’ and display “Ha Ha, Not Stopping”. Use ‘signal’ system call.

1. **Thread Programming (CO4)**

**a)** Using thread programming technique, write a parallel program which will find the minimum number from a huge unsigned character array. The array size should be at least 3x10^9. Use ‘gettimeofday’ to note down the timing. As you increase the number of threads, your timing should show improvement.

You have to use various ‘Pthread’ library calls. Use “-lpthread” option while compiling your program. Your time should not include the array initialization time.

For this problem, boot your Lab Desktop machine in linux mode and do the assignment there. Please keep the timing with you and show this during the viva.

1. **a) Fork & Exec (CO5)**

Create two processes (one child process and one parent process) using fork call. Now print the messages "In child process" and "In parent process" in the child process and the parent process respectively.

Introduce 10 sec sleep for both child and parent process. Use 'ps' command to see both these process in the process table entry. Use “getpid” call to get the process id for both parent & child and verify it using ‘ps’ command.

Now use “exec” call in the child process to display the content of the “/tmp” directory.

**b) IPC using FIFO**

Transfer 1GB file from a client process to a server process. Note down the time. Please keep the timing with you and show this during the viva.

**c) IPC using SOCKET**

Transfer 1GB file from a client process to a server process. Note down the time. Please keep the timing with you and show this during the viva.