# MultiProcessingAssignments -1

1. Refer the code snippet below and answer the queries.

int g\_value =10; /\* global variable \*/

int main()

{

int pid;

int l\_value =5;

printf(“Writing a sample code\n”);

pid = fork(); /\* fork() returns 0 to child process and process id of child to parent process\*/

if(0 == pid)

{

printf(“For child Local variable value=%d\n and global variable value=%d\n”,l\_value,g\_value);

exit(0);

}

else

{

printf(“For parent Local variable value=%d\n and global variable value=%d\n”,l\_value,g\_value);

}

printf(“Code common for both parent process and child process\n);

return 0;

1. What will be the output of parent process and child processes?

Ans- For child Local variable valued and global variable value are:5 and 10

For Parent Local variable valued and global variable value are:5 and 10

1. Find out whether the value of local variable and global variable value will be same for both parent process and child process

Ans- The value of local variable and global variable value will be same for both parent process and child process.

1. Will the order of execution be same always or could be different? Will it impact the output?

Ans-: Yes, order of execution depends on which process gets the first thread. Hence it will impact the output as to which process prints first.

1. Why the first printf() statement will be executed only by parent process and not by child process?

Ans-: Because child process doesn’t exist until then, hence parent process only will print it.

1. Implement a function below to use system call sendfile() to copy from input file descriptor to output file descriptor

size\_t sendfile\_copy(<destination fd>, <source fd>, <offset position in input buffer source>, <number of bytes to copy>)

1. Calculate the execution time for the above step b)
2. Implement a function below to use system call read() and write() to copy from input file descriptor to output file descriptor

size\_t read\_write\_copy(<buffer containing data to be written>, <input buffer source>, <offset position in input buffer source>, <number of bytes to copy>)

1. Calculate the execution time for the above stepd)
2. Compare execution time in c) and e). Which one is better?
3. Run program at console, verify if the “fout.txt” is same as “fin.txt”
4. Run command below and capture and view the internal system calls

Strace <prgname>

1. Try the above program using a non-existing input file name. Capture the errors thrown and display using strerror(), perror()