# **MultiProcessing Assignments - 1**

2. 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);

}

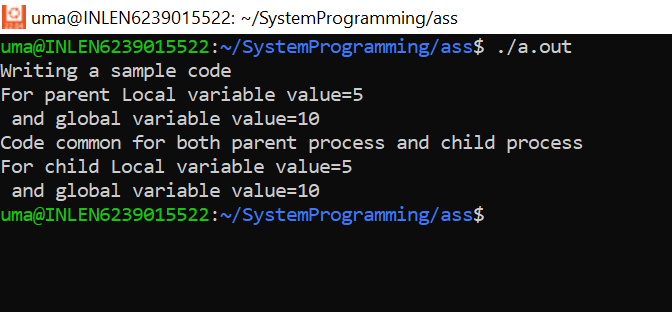
printf(“Code common for both parent process and child process\n);

return 0;

}

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

Solution: The output for the above program.



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 above image states that the local and the parent variable is the same for both parent and child processes.

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

Ans: Maybe it could be different because the execution happens based on the vacancy of the thread and it also clearly shows that the pid has some value which is not equal to 0.

For eg:

if the 1st thread is free then it executes the parent process or if 2nd thread is free it execute child process.

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

In this program, wait is not used in parent execution so the first printf() statement is executed by the parent. If we use wait then the first printf() statement will be executed by the child and then the parent will execute.