Home assignment – Thread

Name: Aakanksha Agarwal

Roll no.: 121002

1. Assuming that child process is always scheduled before parent process, the output is as follows:

Parent sees i = 0

Parent sees i = 1

Child sees i = 1

Parent sees i = 2

Child sees i = 2

Child sees i = 0

The child process, being always scheduled before the parent process, shares the same address space with the parent process. Until the child process calls for exec() or exit(), execution of the calling process is blocked. Since the same address space is shared, anything that affects parent process should not be used in child process.

1. Assuming that the threads are scheduled completely before the parent process, the output will be as follows:

Parent says a: 1

id: 1 a: 2 b: 1

id: 2 a: 3 b: 1

Thread 1 and 2 complete

1. In some multi-threaded applications, mapping of m user-level threads to n kernel-level threads is a good idea because in this implementation, the threading library schedules user threads on available schedulable entities. This is beneficial since it would make context switching very quick, ignoring system calls.

M > n is the best choice.