TUTORIAL #6 POSIX Threads Part II

Group 8

Conceptual Questions:

1) A fork means creating a new process and all process will execute the instruction when fork() . While a thread means creating a thread in the same process

2) IPC is the activity that share data and resources between different process.

3) Semaphore is a critical section in each process that can check and change the value. It starts with waiting a condition and if the condition is true, it will run the critical section and post the new value. Mutex will lock the thread and only that thread can unlock it while Semaphore doesn't tied up with a thread. Other thread can change the value as well.

4) Wait: if the value is greater or equal to 0, decrements it by 1. After the decrement, if the value is less than zero, the process will blocked until the value is greater or equal to 1.

Signal: Increment the value by 1.

5) int sem\_close(sem\_t \*); - to indicate that the calling process is finished

int sem\_destroy(sem\_t \*); - to destroy the unnamed semaphore

int sem\_getvalue(sem\_t \*, int \*sval); - updates the location

int sem\_init(sem\_t \*, int, unsigned int); - to initialize the unnamed semaphore

sem\_t \*sem\_open(const char \*, int, ...); - establishes a connection between a named semaphore and a process

int sem\_post(sem\_t \*); - unlocks the semaphore

int sem\_trywait(sem\_t \*); - locks the semaphore

int sem\_unlink(const char \*); - removes the semaphore.

int sem\_wait(sem\_t \*); - locks the semaphore