Linux System Programming

Lab Assessment on Thread Synchronization

Authored and Compiled By: **Boddu Kishore Kumar**

Email: kishore@kernelmasters.org

Reach us online: www.kernelmasters.org

Contact: 9949062828

Lab Assignments: (Mandatory Questions to everyone)

1. What is the difference between a Task, Process and thread?
2. How threads communicate with each other?
3. Do you have any idea about thread safe? How can you implement it?
4. What is Critical section, deadlocks, race around condition are problems that happen in thread synchronization?
5. WAP producer thread and consumer thread synchronization using pthreads without synchronization mechanisms?
6. WAP create two threads using pthreads and print even no and odd no alternatively.

Mini Project (Real Assignment):

1. WAP Producer thread and consumer thread synchronization using Pthreads and Signals? See the below Pseudo code.

Producer Thread synchronization flow:

Step 1: Initialize producer signal handler.
Use signal () system call.
Step 2: Producer start produce the data until the buffer is FULL.
Use while () loop until Buffer is FULL.
Step 3: Whenever buffer is FULL raise a signal to consumer thread.
Use tkill () system call (or) pthread_kill library to raise a signal to consumer thread.
Step 4: Waiting for a signal from consumer thread.
Use pause () system call wait for a signal from consumer thread.
Step 5: Whenever signal received from consumer thread, Enter step 2.

Consumer Thread synchronization flow:

Step 1: Initialize consumer signal handler & Wait for a Signal.
Use signal () system call for signal handler initialization and pause () system call wait for signal.
Step 2: Whenever signal received from producer thread, start read the buffer until buffer is empty.
Use while () loop read Buffer until Buffer is EMPTY.
Step 3: Whenever the buffer is EMPTY, Consumer raise a signal to producer thread.
Use tkill () system call (or) pthread_kill library to raise a signal to producer thread.
Step 4: Waiting for a signal from producer thread.
Use pause () system call wait for a signal from producer thread.
Step 5: Whenever signal received from producer thread, Enter step 2.