

Multithreaded Programming Basics Assignments

1. Which applications can be multithreaded?

- Web Browsers
- Web Servers
- Computer Games
- Text Editors

2. Which applications cannot be multithreaded?

- On a single processor machine and a desktop application, you use multi threads so you don't freeze the app but for nothing else really. On a single processor server and a web based app, no need for multi threading because the web server handles most of it.

3. How to create or use POSIX threads?

- The C/C++ languages provide the POSIX thread(pthread)

4. How to compile and execute program using POSIX threads API. Mention the statements to compile a file mythread.c and generate an executable "mythread".

- gcc -lpthread mythread.c

5. What is the difference between mutex and semaphore? Can we use mutex as a semaphore and vice versa?

Semaphore	Mutex
It is a type of signaling mechanism.	It is a locking mechanism.
Semaphore is an integer variable.	Mutex is just an object.
The wait and signal operations can modify a semaphore.	It is modified only by the process that may request or release a resource.

6. What are the 2 thread standards? Which one is commonly used?

- NPS: Straight threads, meaning the male and female threads are parallel. NPT: Tapered threads, meaning the male and female threads are tapered. NPT threads are more commonly used. NPT Thread specifications are based on ANSI B1.

7. What do you mean by starvation and deadlock? When will it occur? What are the solutions to handle that?
- A fair system prevents starvation and deadlock. Starvation occurs when one or more threads in your program are blocked from gaining access to a resource and, as a result, cannot make progress. Deadlock, the ultimate form of starvation, occurs when two or more threads are waiting on a condition that cannot be satisfied.
 - A possible solution to starvation is to use a scheduling algorithm with priority queue that also uses the aging technique. Aging is a technique of gradually increasing the priority of processes that wait in the system for a long time.
8. What all are shared across threads?
- The items that are shared among threads within a process are: Text segment (instructions) Data segment (static and global data) BSS segment (uninitialized data)
9. What is the difference between thread, process and a program?
- A thread shares information like data segment, code segment, files etc. with its peer threads while it contains its own registers, stack, counter etc. A process is a program under execution i.e an active program. A thread is a lightweight process that can be managed independently by a scheduler.
10. Is thread id same as PID? How will you get thread id?
- The PID is the process ID, TID is the thread ID. The thing is that for the first thread created by fork(), PID=TID. If you create more threads within the process, with a clone() command, then PID and TID will be different, PID will always be smaller than TID.
11. Name atleast 5 thread attributes.
- Priority
 - Stack size
 - Thread group
 - Detach state
 - Scheduling policy
12. Refer link below and answer the question.
- https://man7.org/linux/man-pages/man3/pthread_attr_setscope.3.html
- what are the 2 thread scopes, if there are 4 processes, each with 3 threads then comment on %allocation of CPU to every thread in the 2 thread scope cases.
13. Which one is memory intensive multithreading or multiprocessing?

- Multiprocessing is used to create a more reliable system, whereas multithreading is used to create threads that run parallel to each other. Multiprocessing requires a significant amount of time and specific resources to create, whereas multithreading is quick to create and requires few resources.

14. Why are threads referred as LWP?

- User level thread are those that can be handled by thread library. On the other hand kernel level thread (which needs to deal with hardware) are also called LWP(light weight process) to maximize the use of system and so the system does not halt upon just one system call.

15. Can threads of different process communicate with each other? If yes, what are the mechanisms available?

- Inter-process communication (IPC) is a mechanism that allows processes to communicate with each other and synchronize their actions. The communication between these processes can be seen as a method of co-operation between them. Processes can communicate with each other through both:
 - Shared Memory
 - Message passing

16. What is the difference between concurrent and parallel processing?

- Concurrency is the task of running and managing the multiple computations at the same time. While parallelism is the task of running multiple computations simultaneously