

Dan Otieno.

CPE434-01.

HOMEWORK 2.

02/15/23.

1. Please explain what thread pools are. What purpose do they serve in the OS or user application?

In an operating system, "A thread pool is a collection of worker threads that efficiently execute asynchronous callbacks on behalf of the application. The thread pool is primarily used to reduce the number of application threads and provide management of the worker threads." Some of the advantages of thread pools are:

- i. They service requests faster with existing threads, as opposed to creating new ones.*
- ii. They allow all threads within applications to be bound by the pool size.*
- iii. The mechanics of creating a task performs a separating task that allows for different strategies for the running task.*

2. When you have multiple threads in a multi-core system under the ubuntu 20.04 operating system you are using you may use "thread safe" libraries. What is thread safe?

When a process is logically correct while executed alongside several other threads, it is deemed "thread safe". We can say that thread safety is the avoidance of a situation where data values are inaccurate, depending on the order the data is accessed and modified by multiple threads. There are three levels of thread safety; unsafe, Thread Safe – Serializable, and Thread Safe – MT-Safe.

3. When you have a multithreaded application running on Ubuntu 20.04 that you may be using for this class:

- a. how do you know or determine which thread(s) a signal is sent to?

i. Each thread is associated with a signal mask that has a signal number. The signal number contains a list of actions, which are then shared among all the threads in the process. Actions that are related to a specific thread are directed to that specific thread, or rather the specific thread that invoked the generation of that signal.

- b. when multiple threads write to the same file how do you determine the order the data is written to a file?

i. There is an execution order assigned to each action in a queue. The execution of write actions adheres to synchronization-order consistency.

4. What is priority inversion safe. Give a complete description of what the problem is and what the solution is.

According to a definition in Wikipedia, “priority inversion is a scenario in scheduling in which a high priority task is indirectly superseded by a lower priority task effectively inverting the assigned priorities of the tasks. This violates the priority model that high-priority tasks can only be prevented from running by higher-priority tasks.” An example of priority inversion creating a problem is found [here](#), where the scheduling process inside the operating system of a space vehicle sent to Mars caused a computer reset, which, despite not resulting in data loss, hampered further operations planned for that day. The article also described how changes were made to certain flags within the system to enable priority inheritance, which resolved the issue.

5. You have three processes arriving in the following order: P1, P2, P3, P4. Assume that they arrive at time = 0 to your scheduler and a lower number priority is actually a higher priority. You are given the following information about them in this table:

	Burst Time	Priority
P1	7	3
P2	8	2
P3	2	1
P4	4	5

What is the average wait time with SJF? What is the average turnaround with SJF

P3		P4		P1		P2	
0	2	6	13	21			

- *Total Waiting time = 21.*
- *Total Turnaround time = 42.*
- *Number of processes = 4.*
- *Average Wait time with SJF = $21/4 = 5.25$.*
- *Average Turnaround time with SJF = $42/4 = 10.5$.*

SOURCES:

1. <https://learn.microsoft.com/en-us/windows/win32/procthread/thread-pools>
2. <https://docs.oracle.com/cd/E19683-01/806-6867/6jfpdco5/index.html#:~:text=Thread%20safety%20is%20the%20avoidance,private%20copy%20of%20the%20data.>
3. <https://www.ibm.com/docs/en/aix/7.2?topic=threads-thread-signal-handling>
4. <https://learn.microsoft.com/en-us/dotnet/api/system.threading.thread.priority?view=net-7.0>
5. <https://docs.oracle.com/javase/specs/jls/se7/html/jls-17.html>
6. <https://users.cs.duke.edu/~carla/mars.html>
7. https://en.wikipedia.org/wiki/Priority_inversion