JUNAID GIRKAR
60004190057
SE COMPS A-3

OPERATING SYSTEMS
EXPERIMENT - 3

AIM: Building multi-threaded and multi-process applications

THEORY:-

Multiprocessing:

A multiprocessing system has more than two processors. The CPUs are added to the system that helps to increase the computing speed of the system. Every CPU has its own set of registers and main memory.

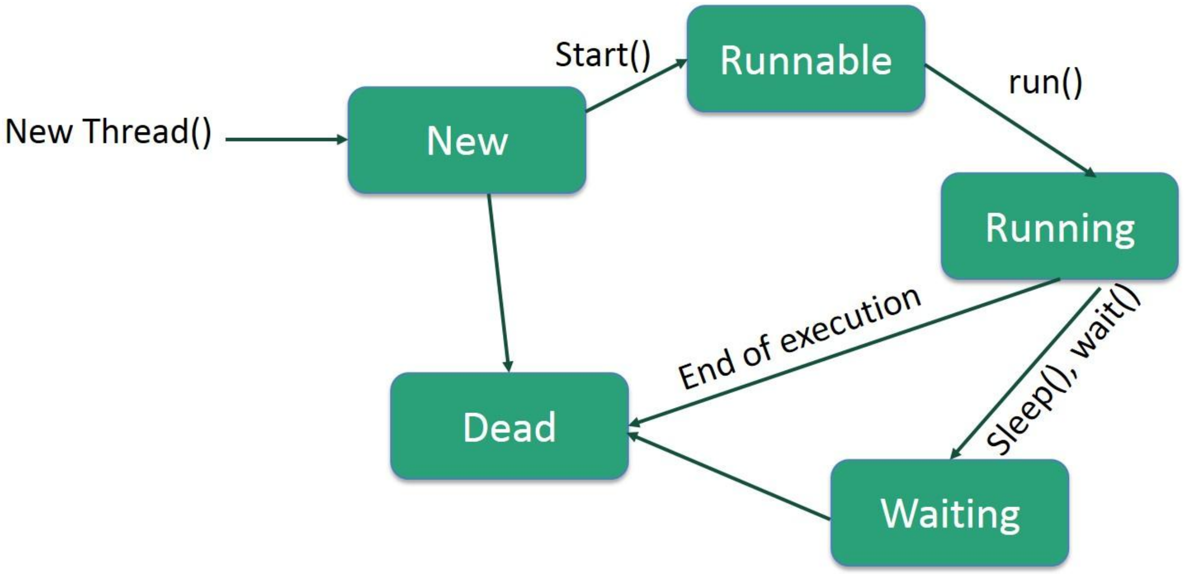
However, because each CPU are separate, it may happen that one CPU may not have anything to process. One processor may sit idle, and the other may be overloaded with the specific processes. In such a case, the process and resources are shared dynamically among the processors.

Multithreading:

Multithreading is a program execution technique that allows a single process to have multiple code segments (like threads). It also runs concurrently within the "context" of that process. Multi-threaded applications are applications that have two or more threads that run concurrently. Therefore, it is also known as concurrency.

Life Cycle of a Thread

A thread goes through various stages in its life cycle. For example, a thread is born, starts, runs, and then dies. The following diagram shows the complete life cycle of a thread.



Following are the stages of the life cycle −

- New − A new thread begins its life cycle in the new state. It remains in this state until the program starts the thread. It is also referred to as a born thread.
- Runnable − After a newly born thread is started, the thread becomes runnable. A thread in this state is considered to be executing its task.
- Waiting − Sometimes, a thread transitions to the waiting state while the thread waits for another thread to perform a task. A thread transitions back to the runnable state only when another thread signals the waiting thread to continue executing.
- Timed Waiting − A runnable thread can enter the timed waiting state for a specified interval of time. A thread in this state transitions back to the runnable state when that time interval expires or when the event it is waiting for occurs.
- Terminated (Dead) − A runnable thread enters the terminated state when it completes its task or otherwise terminates.

Thread Priorities

Every Java thread has a priority that helps the operating system determine the order in which threads are scheduled.

Threads with higher priority are more important to a program and should be allocated processor time before lower-priority threads. However, thread priorities cannot guarantee the order in which threads execute and are very much platform dependent.

CREATING A THREAD:

A thread can be created in two ways:-

1. **Extending Thread class:** This approach provides more flexibility in handling multiple threads created using available methods in Thread class.
2. **Implementing a Runnable Interface:** This method provides an entry point for the thread and you will put your complete business logic inside this method.

Characteristics of Multiprocessing

Here are the essential features of Multiprocessing:

- Multiprocessing is classified according to the way their memory is organized.
- Multiprocessing improves the reliability of the system
- Multiprocessing can improve performance by decomposing a program into parallel executable tasks.

Characteristics of Multithreading

Here are important aspects of multithreading:

- In the multithreading process, each thread runs parallel to each other.
- Threads do not allow you to separate the memory area. Therefore it saves memory and offers a better application performance

Advantage of Multiprocessing

- The biggest advantage of a multiprocessor system is that it helps you to get more work done in a shorter period.
- The code is usually straightforward.
- Takes advantage of multiple CPU & cores
- Helps you to avoid GIL limitations for CPython
- Remove synchronization primitives unless if you use shared memory.
- Child processes are mostly interruptible/killable
- It helps you to get work done in a shorter period.
- These types of systems should be used when very high speed is required to process a large volume of data.
- Multiprocessing systems save money compared to single processor systems as processors can share peripherals and power supplies.

Disadvantage of Multiprocessing

- IPC(Inter-Process Communication) a quite complicated with more overhead
- Has a larger memory footprint

Advantage of Multithreading

- Threads share the same address space
- Threads are lightweight which has a low memory footprint
- The cost of communication between threads is low.
- Access to memory state from another context is easier
- It allows you to make responsive UIs easily
- An ideal option for I/O-bound applications
- Takes lesser time to switch between two threads within the shared memory and time to terminate
- Threads are faster to start than processes and also faster in task-switching.
- All Threads share a process memory pool that is very beneficial.
- Takes lesser time to create a new thread in the existing process than a new process

Disadvantage of multithreading

- Multithreading system is not interruptible/killable
- If not following a command queue and message pump model then manual use of synchronization needed which becomes a necessity
- Code is usually harder to understand and increases the potential for race conditions increases dramatically.

Parameter	Multiprocessing	Multithreading
Basic	Multiprocessing helps you to increase computing power.	Multithreading helps you to create computing threads of a single process to increase computing power.

Execution	It allows you to execute multiple processes concurrently.	Multiple threads of a single process are executed concurrently.
CPU switching	In Multiprocessing, CPU has to switch between multiple programs so that it looks like that multiple programs are running simultaneously.	In multithreading, CPU has to switch between multiple threads to make it appear that all threads are running simultaneously.
Creation	The creation of a process is slow and resource-specific.	The creation of a thread is economical in time and resource.
Classification	Multiprocessing can be symmetric or asymmetric.	Multithreading is not classified.
Memory	Multiprocessing allocates separate memory and resources for each process or program.	Multithreading threads belonging to the same process share the same memory and resources as that of the process.
Pickling objects	Multithreading avoids pickling.	Multiprocessing relies on pickling objects in memory to send to other processes.
Program	Multiprocessing system allows executing multiple programs and tasks.	Multithreading system executes multiple threads of the same or different processes.
Time taken	Less time is taken for job processing.	A moderate amount of time is taken for job processing.

CONCLUSION:

We learnt the difference between multithreading and multiprocessing and the different methods that can be implemented in multithreading. For our experiment, I have extended the Thread class to implement multithreading in our bus ticket booking system.