Distributed Systems

COMP90015 2021 Semester 1
Tutorial 03

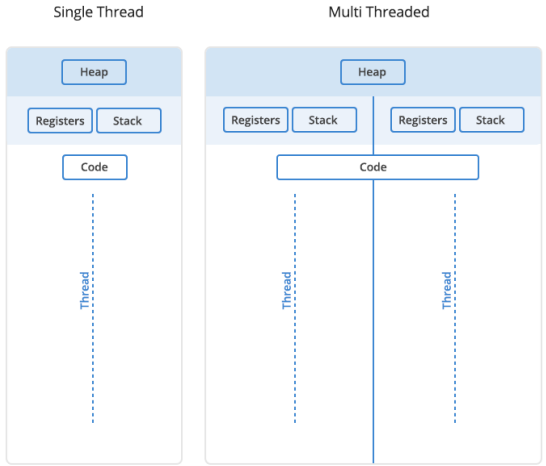
Today’s Agenda

- Quickly go through the thread slides
- Concept/question discussion
    1. What is a thread and life cycle of a thread
    2. Synchronous access to shared resources
    3. Comparison of worker pool multi-threading architecture with the thread-per-request architecture
- Code demonstration of thread Sleep, Join and Synchronization and Multithreaded Server and Client

Q1. What is Thread?

Q1. What is Thread ?

- A Thread is a piece of code that runs in concurrent with other threads.
- Each thread is a statically ordered sequence of instructions.
- Threads are used to express concurrency on both single and multiprocessors machines.

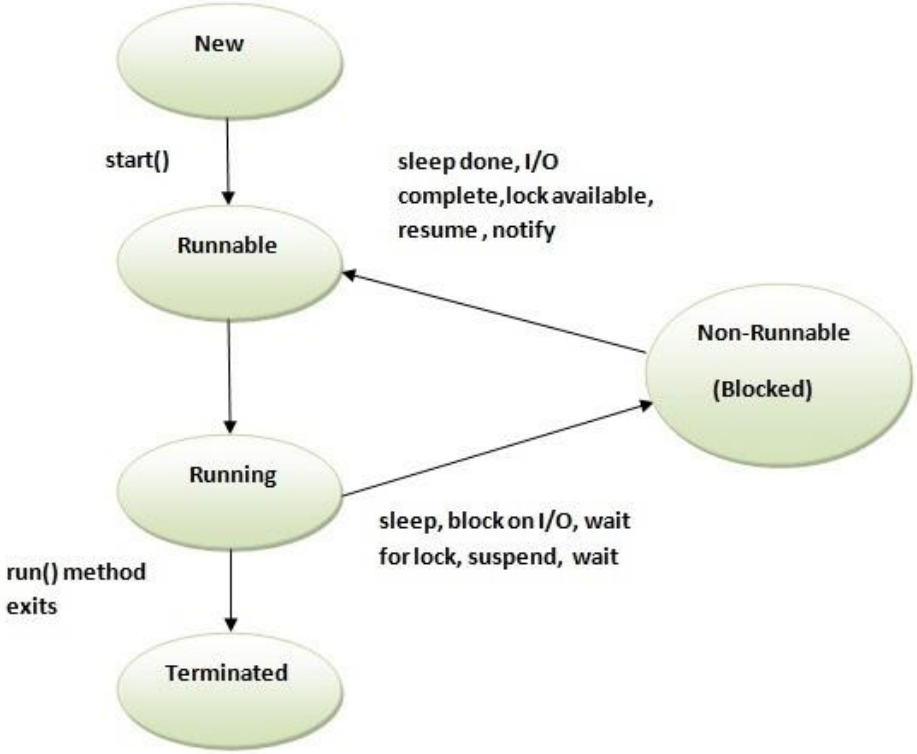


Thread vs Process

- Advantages of thread based parallelism
  - Threads share the same address space.
  - Context-switching between threads is normally inexpensive.
  - Communication between threads is normally inexpensive.

Q3. What is life cycle of a thread?

Thread Lifecycle



Q4. What do you mean by synchronous access to shared resources and how can we achieve it?

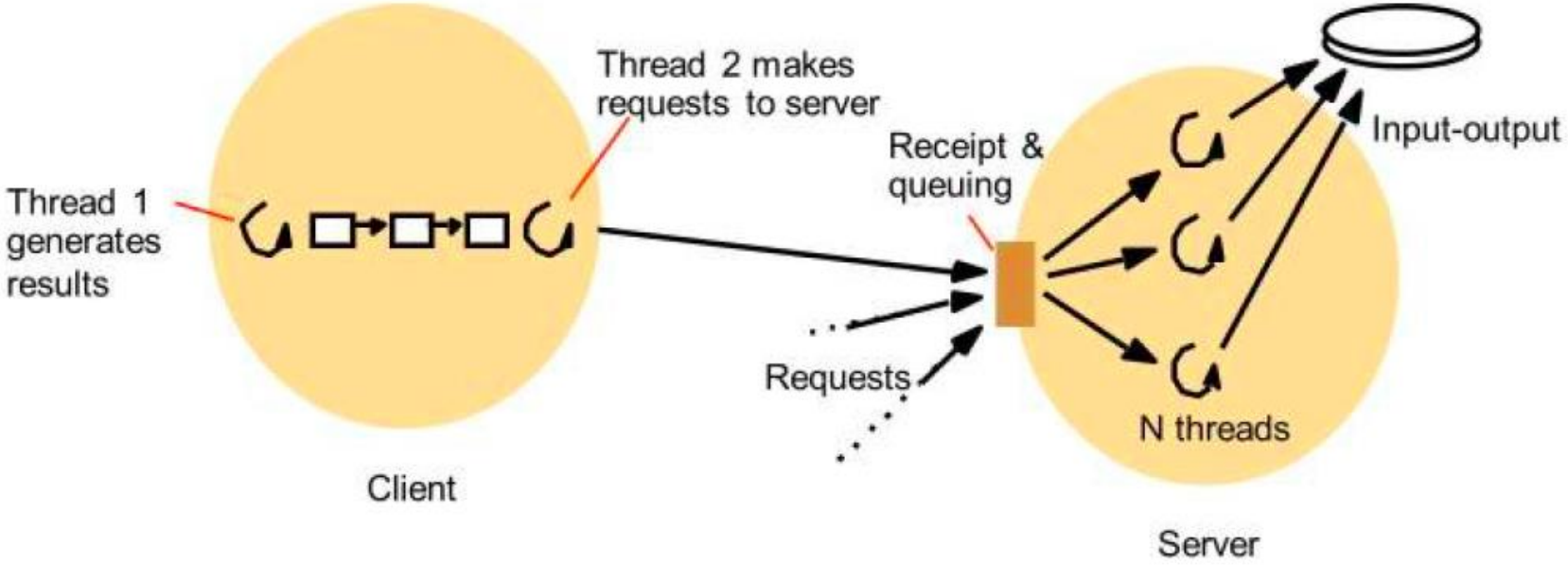
Q4. What do you mean by synchronous access to shared resources and how can we achieve it?

- If one thread tries to read the data and other thread tries to update the same data, it leads to inconsistent state.
- This can be prevented by synchronising access to the data.
- Use “synchronized” to methods or objects:

```
public synchronized void update()
{
                    …
}
```

Q5. Compare the worker pool multi-threading architecture with the thread-per-request architecture.

Worker pool architecture



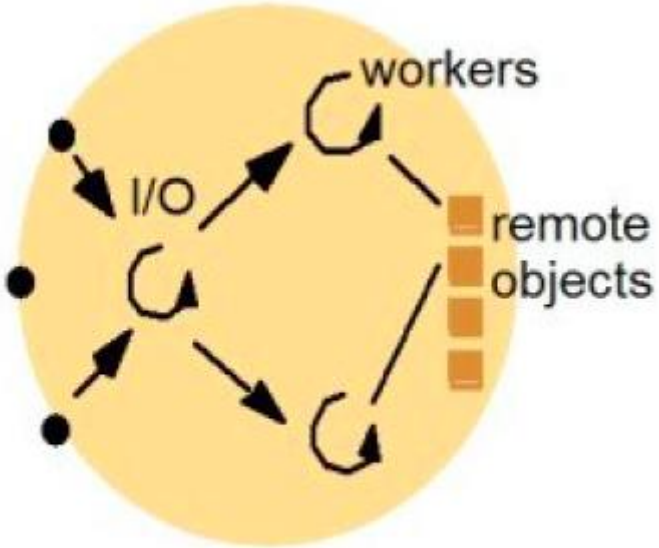
Worker pool architecture

The server creates a fixed number of threads called a worker pool. As requests arrive at the server, they are put into a queue by the I/O thread and from there assigned to the next available worker thread.

Server creates worker pool ➜ request comes in, put into a queue ➜ assigned to an available worker thread.

Useful in highly concurrent system

Thread-per-request



a. Thread-per-request

Thread-per-request architecture

Thread created for each request, when the request is finished, the thread is deallocated.

Code Demonstration

- Multi-threading in Java – Synchronization
  - In java, multithreading can be implemented in two ways
    - Extending Thread class
    - Implementing Runnable interface
- Multithreaded Server and client with Sockets

End of Tutorial