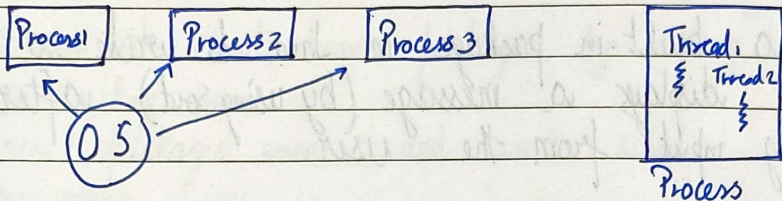


Chapter 13 - Multithreading

Multiprocessing and multithreading both are used to achieve multitasking



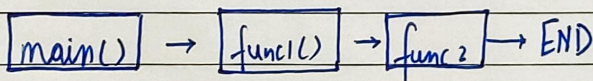
In a nut shell...

- Threads use shared memory area.
- Threads \Rightarrow Faster context switching
- A Thread is light-weight whereas a process is heavyweight

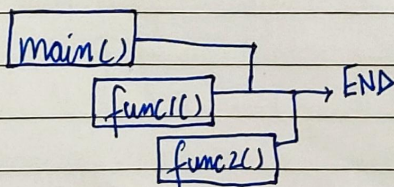
For Example \rightarrow A word processor can have one thread running in foreground as an editor and another in the background auto saving the document!

Flow of control in Java

1. Without threading:



2. With threading:

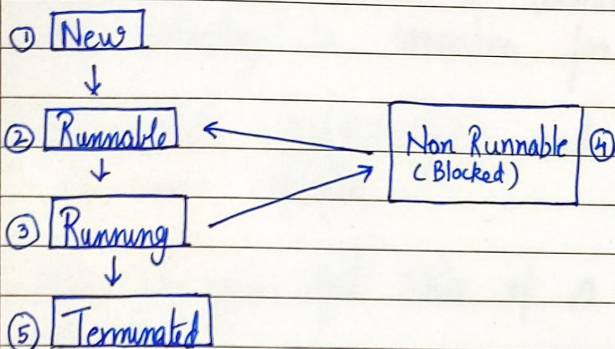


Creating a Thread

There are two ways to create a thread in Java.

1. By extending Thread class
2. By implementing Runnable interface

Life cycle of a Thread



- ① New → Instance of thread created which is not yet started by invoking start()
- ② Runnable → After invocation of start() & before it is selected to be run by the scheduler.
- ③ Running → After thread scheduler has selected it.
- ④ Non Runnable → Thread alive, not eligible to run.
- ⑤ Terminated → run() method has exited

The Thread class

Below are the commonly used constructors of Thread class:

- ① Thread()
- ② Thread(String name)
- ③ Thread(Runnable r)
- ④ Thread(Runnable r, String name)

Methods of Thread class

Thread class offers a lot of methods such as `run()`, `start()`, `join()`, `getPriority()`, `setPriority()` etc. More can be found on visiting Java docs

