



CS F213 - Object Oriented Programming

J. Jennifer Ranjani

email: jennifer.ranjani@pilani.bits-pilani.ac.in

Chamber: 6121 P, NAB

Consultation: Fridays 4 – 5 p.m.

<https://github.com/JenniferRanjani/Object-Oriented-Programming-with-Java>



BITS Pilani
Pilani Campus



Multithreaded Programming

Multi-tasking

innovate

achieve

lead



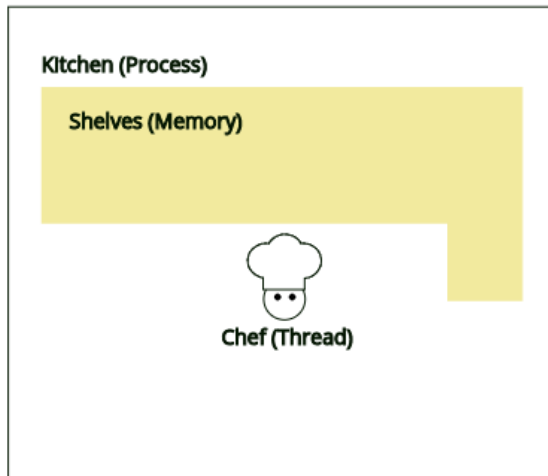
Process vs. Thread

innovate

achieve

lead

Restaurant (A CPU Core)



Order 2
Recipe

Step 1
Step 2
Step 3
...



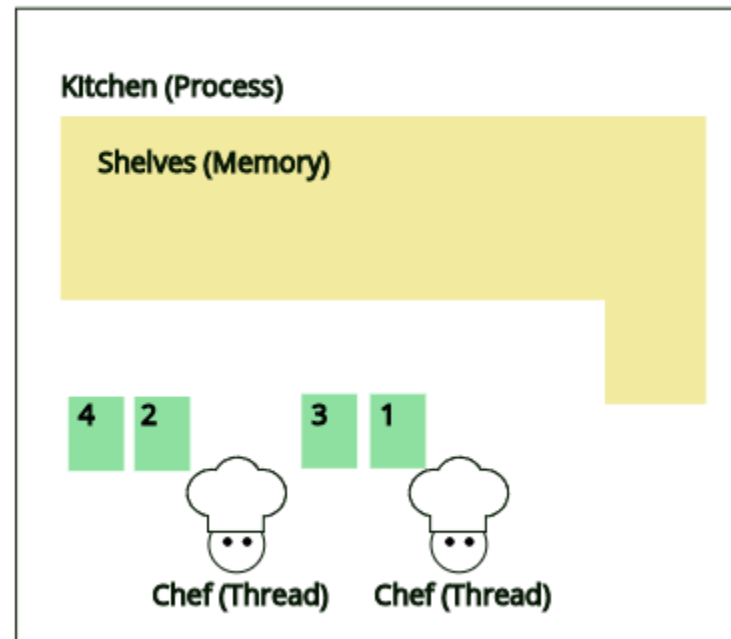
Order 1
Recipe

Step 1
Step 2
Step 3
...



Chef (Thread)

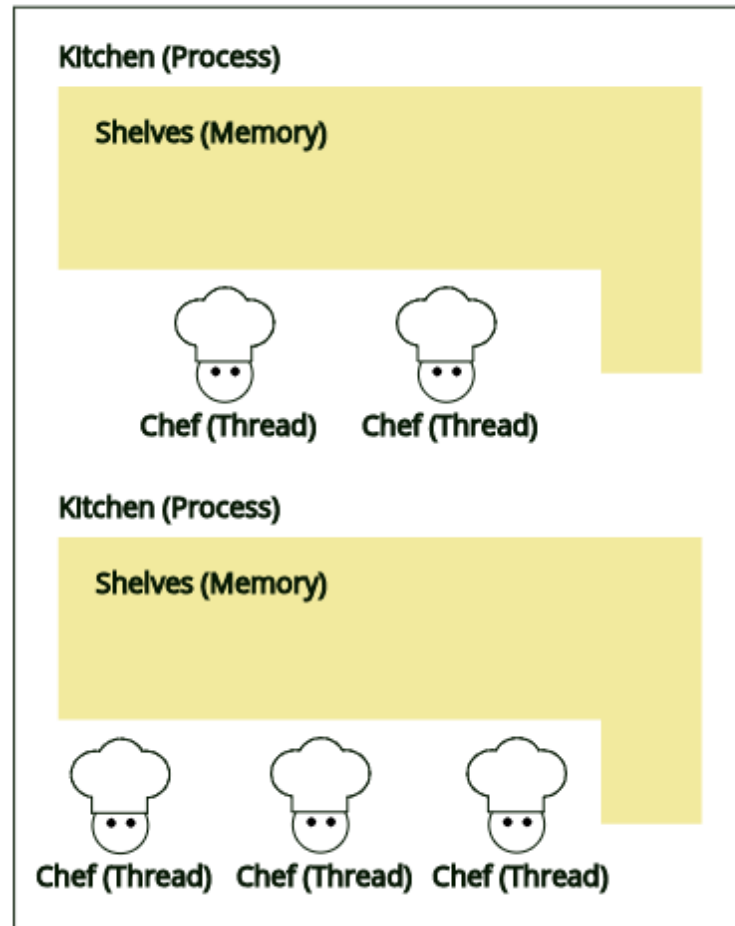
Restaurant (A CPU Core)



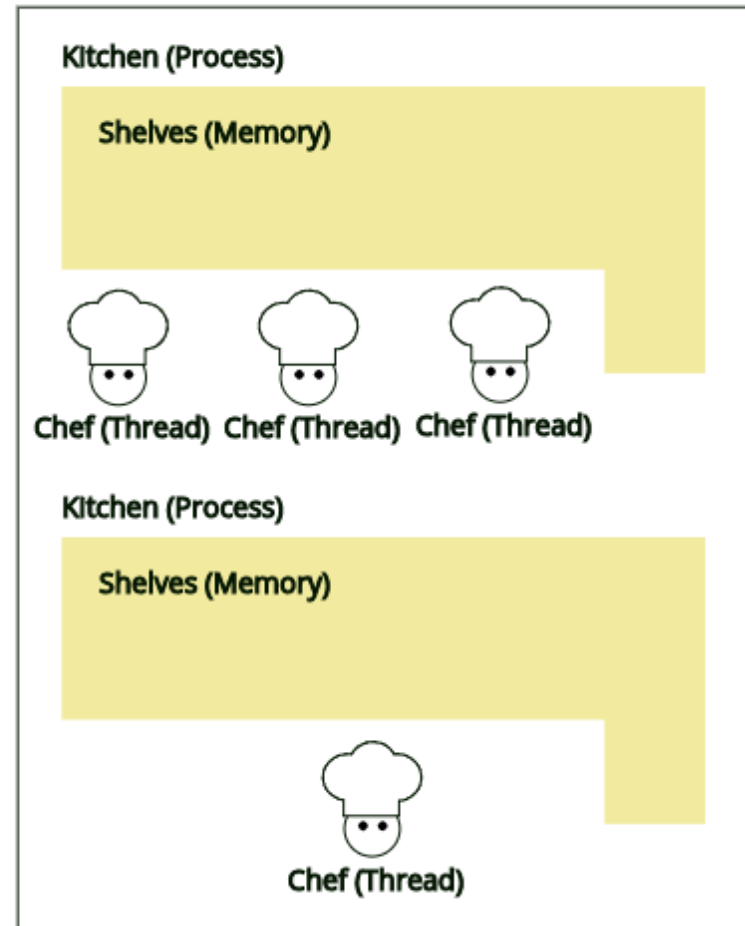
Process vs. Thread



Restaurant (CPU Core 1)



Restaurant (CPU Core 2)

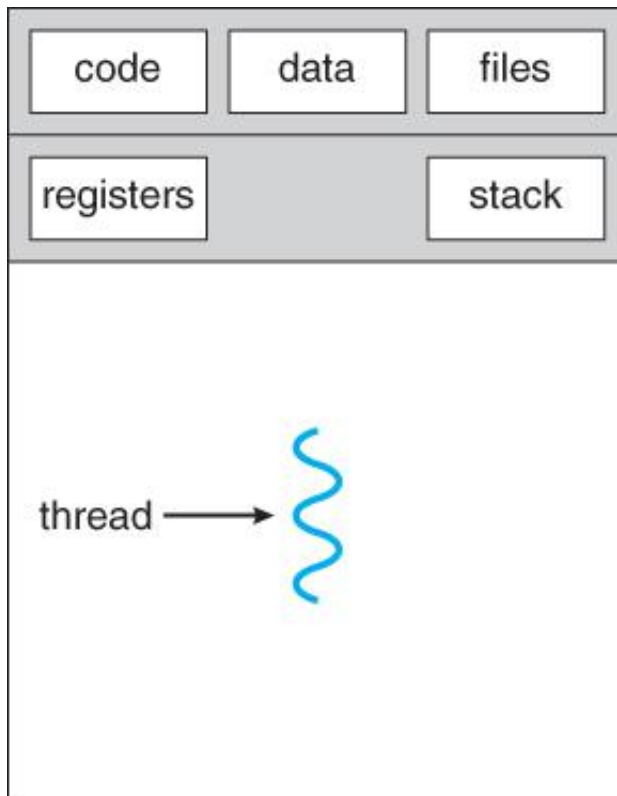


Multi-tasking

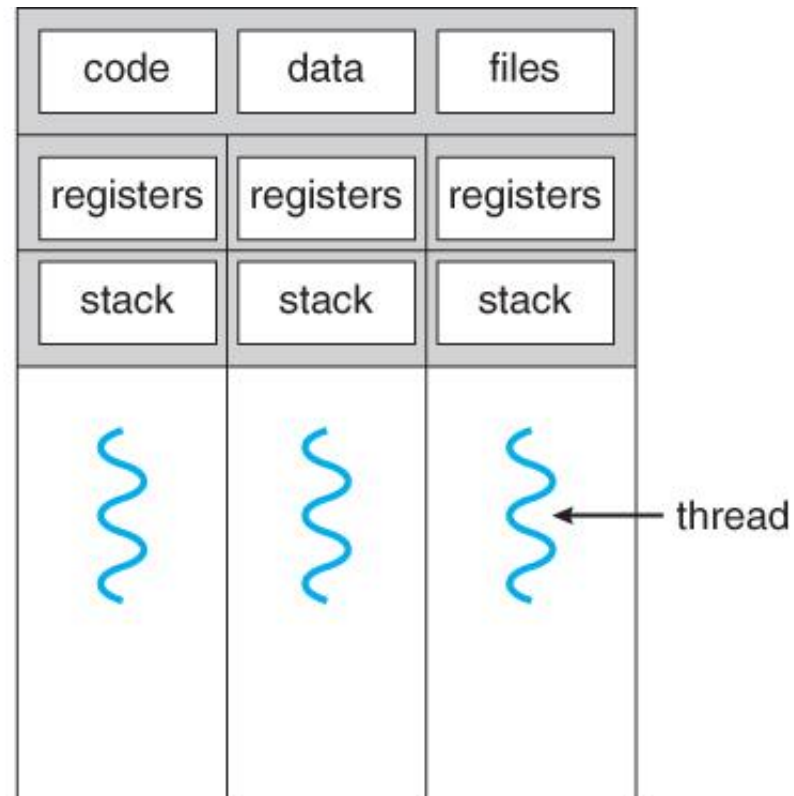


Process based	Thread based
Heavy weight	Light weight
Program in execution is called as a process.	Thread is a part of the program that has separate path of execution.
Each process has a separate space in memory	Threads share the same address space
Communication cost is high	Communication cost is low
Context switching is required for saving and loading registers, memory maps etc.	Context switching is not required.
Eg. Browsing and listening to music	Eg. Formatting using a text editor at the same time it is printing.

Single vs. Multi Threading



single-threaded process



multithreaded process

Thread Model



- Single threaded systems use event loop with polling
 - Threads run a infinite loop
 - It polls a single event queue, let say, waiting for a network file to be read
 - The program wait until the event handler returns which wastes the CPU time.
 - When a thread blocks for a resource, entire program stops running

Example



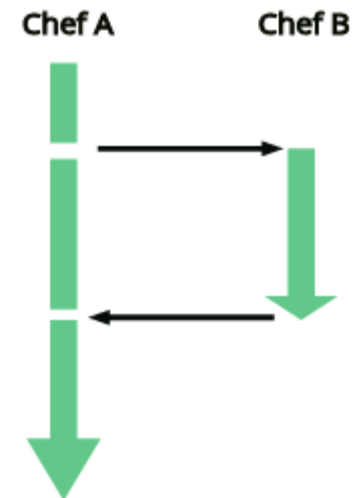
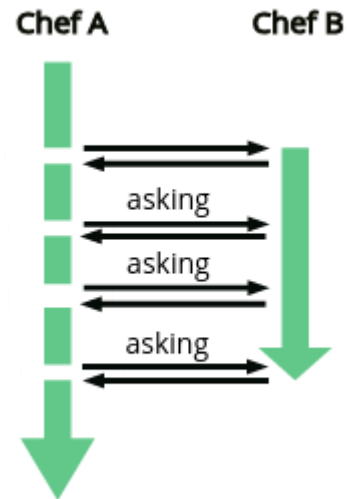
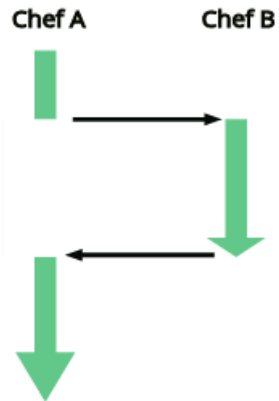
Interrupt



Polling



Polling vs. Interrupts



Thread Model

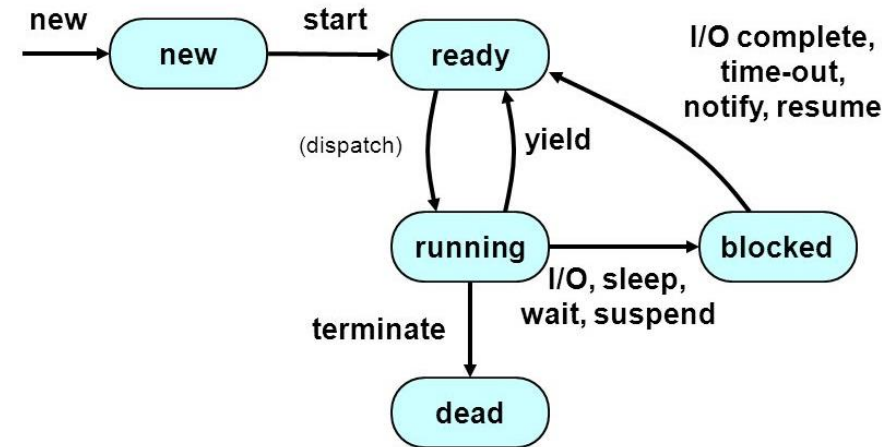


- All Java class libraries are designed with multithreading in mind.
- Java Multithreading
 - Eliminates loop/polling mechanism
 - One thread can pause without stopping the other parts of the program
 - Eg. It allows animation loops to sleep for a second without causing the whole system to pause
 - One thread that is blocked pauses.

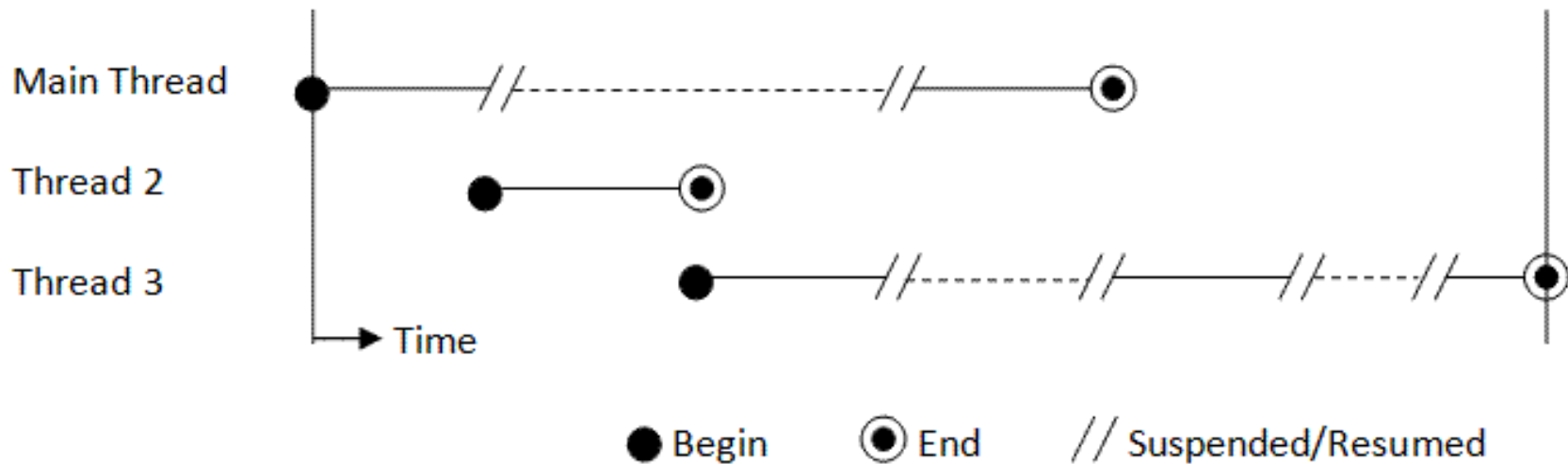
Thread States



- **Ready to run (New):** As soon as it gets CPU time it will start running.
- **Running:** Under execution.
- **Suspended:** Temporarily not active or under execution.
- **Blocked:** Waiting for resources.
- **Resumed:** Suspended thread resumed, and start from where it left off.
- **Terminated:** Halts the execution immediately and never resumes.



Multi-threading



Thread Scheduler



Thread Priorities



- Priorities determine how thread should be treated with respect to the others
- Priorities are integers that specify relative priority of one thread to another.
- Higher priority does not mean that the thread runs faster.
- When switching from one thread to the next, the priority is used for deciding which one to choose next – context switch

Rules determining Context Switch

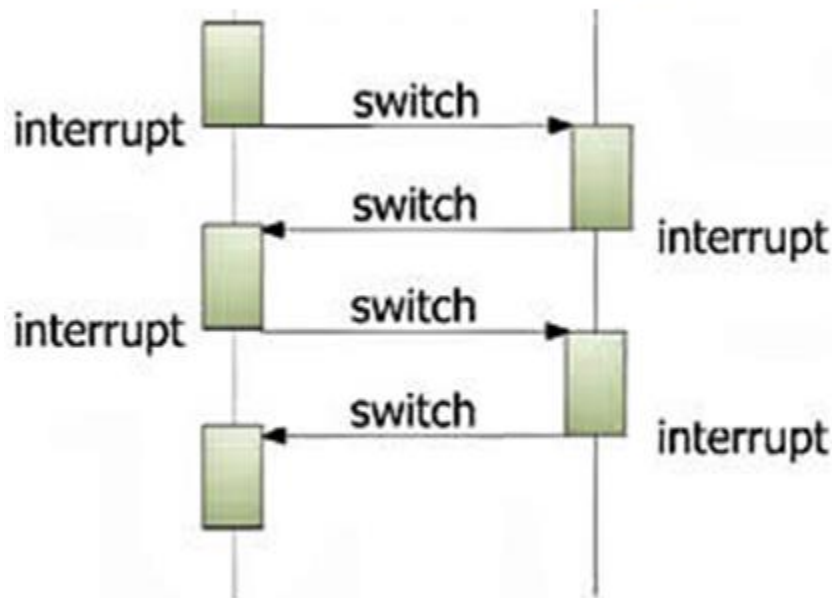


- A thread can voluntarily relinquish control
 - When explicitly yielding, sleeping or when blocked
 - The highest priority thread that is ready to run is given the CPU
 - Non-preemptive multitasking
- A thread can be preempted by a higher priority thread
 - When a lower priority thread that does not yield the processor is simple preempted by a higher priority thread no matter what it is doing
 - As soon as the higher priority thread want to run, it does
 - Preemptive Multitasking
- Some operating systems, time slice equal priority threads in round robin fashion. For others, thread should voluntarily yield otherwise it will not run.

Pre-emptive vs. Non-preemptive Scheduling

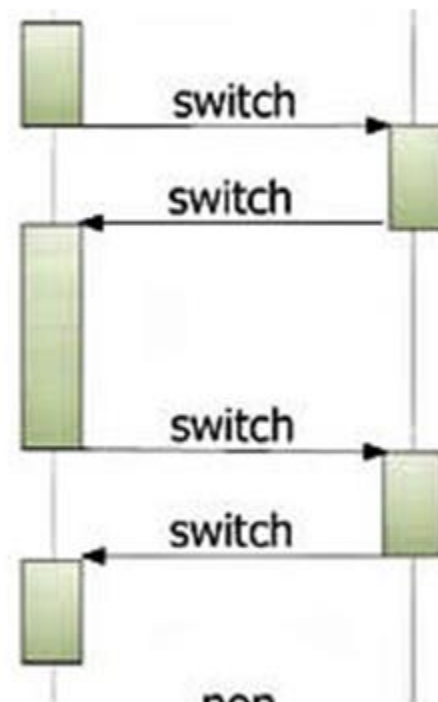


Thread1 Thread2



pre-emptive

Thread1 Thread2



non
pre-emptive