

UF 2: Multi-thread programming

Maribel
Madueño

Thread

- ❑ A thread is the smallest PE an OS may schedule.
- ❑ A thread it is a **lightweight** PE.
- ❑ The threads of a process **share memory** space.

Multi-thread programming

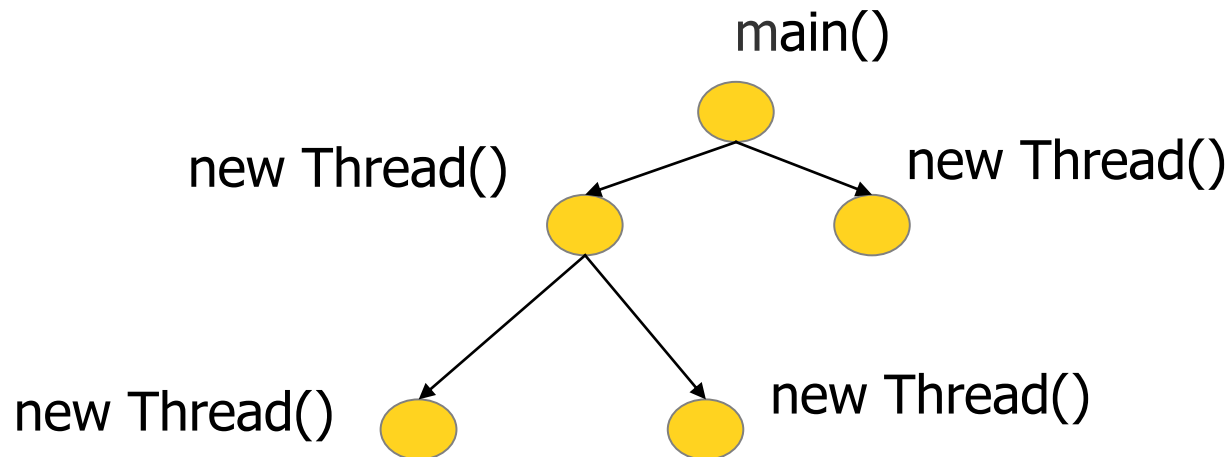
❑ Thread **synchronization**: Manage access to shared resources so as to **avoid** data **inconsistencies** or **interferences**.

Threads in Java

❑ In Java, thread scheduling is performed by the **Java VM** and it is **not CPU consuming** (not significant).

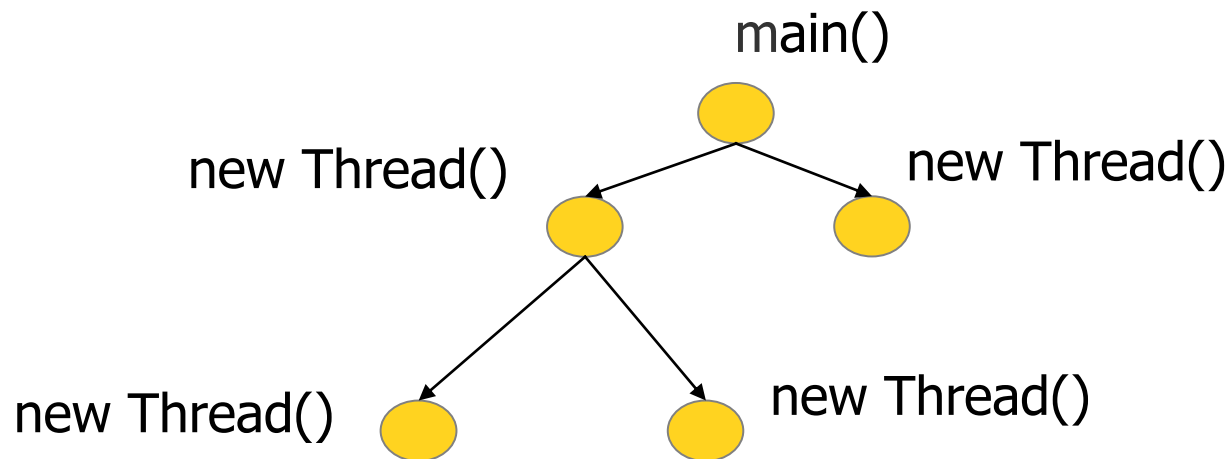
Threads in Java

- ❑ `main()` method: **Main thread** of a process.
- ❑ Additional execution threads may be created in the main thread.
- ❑ Any thread may create additional threads.

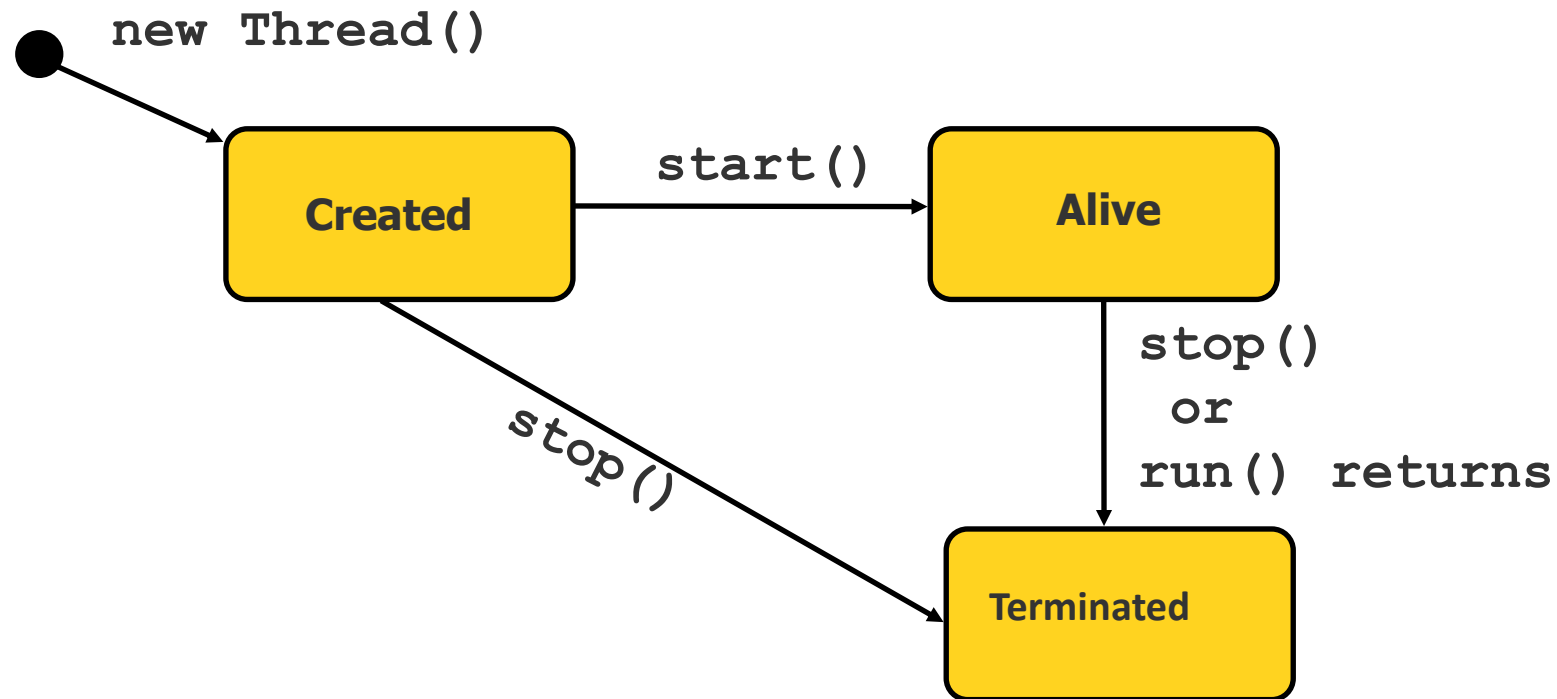


Threads in Java

All threads belong to a group. A default group is created named main.

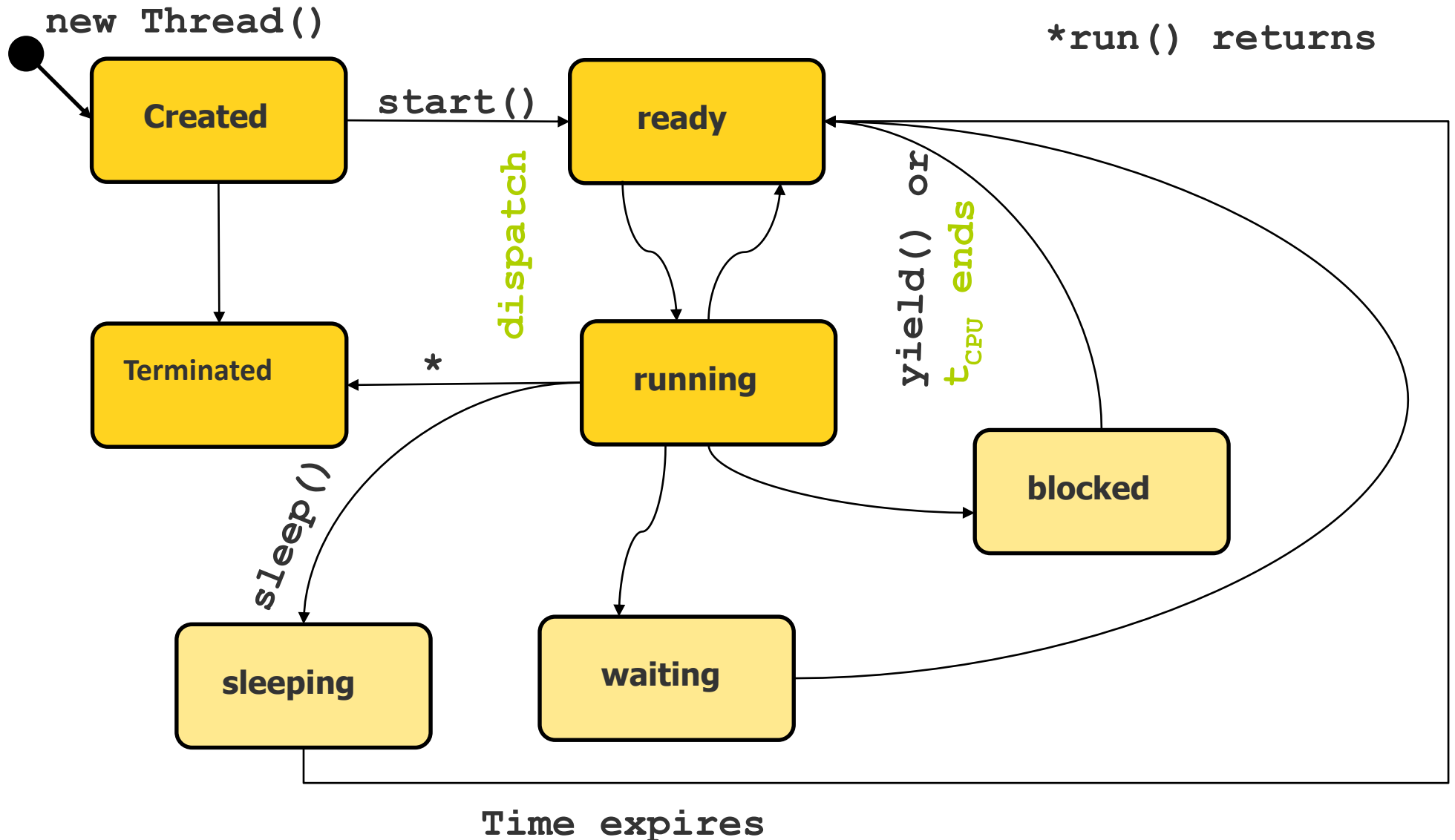


Thread life cycle in Java



start() causes the thread to call its `run()` method.

Thread life cycle in Java



Thread states in Java

- ❑ **Created** or new.
- ❑ **Ready**: at the cue waiting for CPU use.
- ❑ **Running**: using the CPU.
- ❑ Blocked, sleeping, waiting: are **idle states**.
- ❑ Terminated or **dead**: ends execution.

Thread states in Java

Blocked:

- ☐ An I/O operation occurs
- ☐ Unable to access to synchronized code*
- ☐ Wait for another thread to end.

*Related to access to shared resources

Thread states in Java

❑ Sleep:

used, for instance, to manage periodic execution of code.

<https://docs.oracle.com/javase/tutorial/essential/concurrency/sleep.html>

❑ Waiting:

used, for thread communication.