# Concurrency

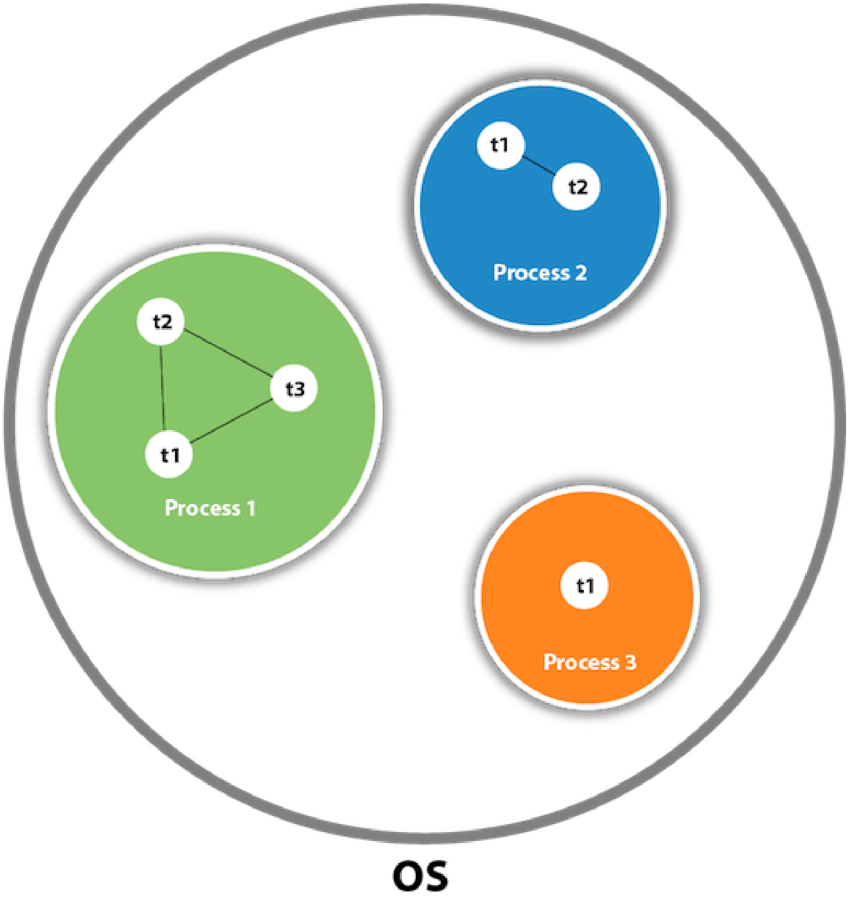
## common

# Concurrency in Java

# Definition

- multitasking
- multiprocessing
- multithreading

# Differentiation



# Class Thread

# Example

```java
public static void main(String[] args) {
    Thread t = Thread.currentThread(); // получаем главный поток
    System.out.println(t.getName()); // main
}
```

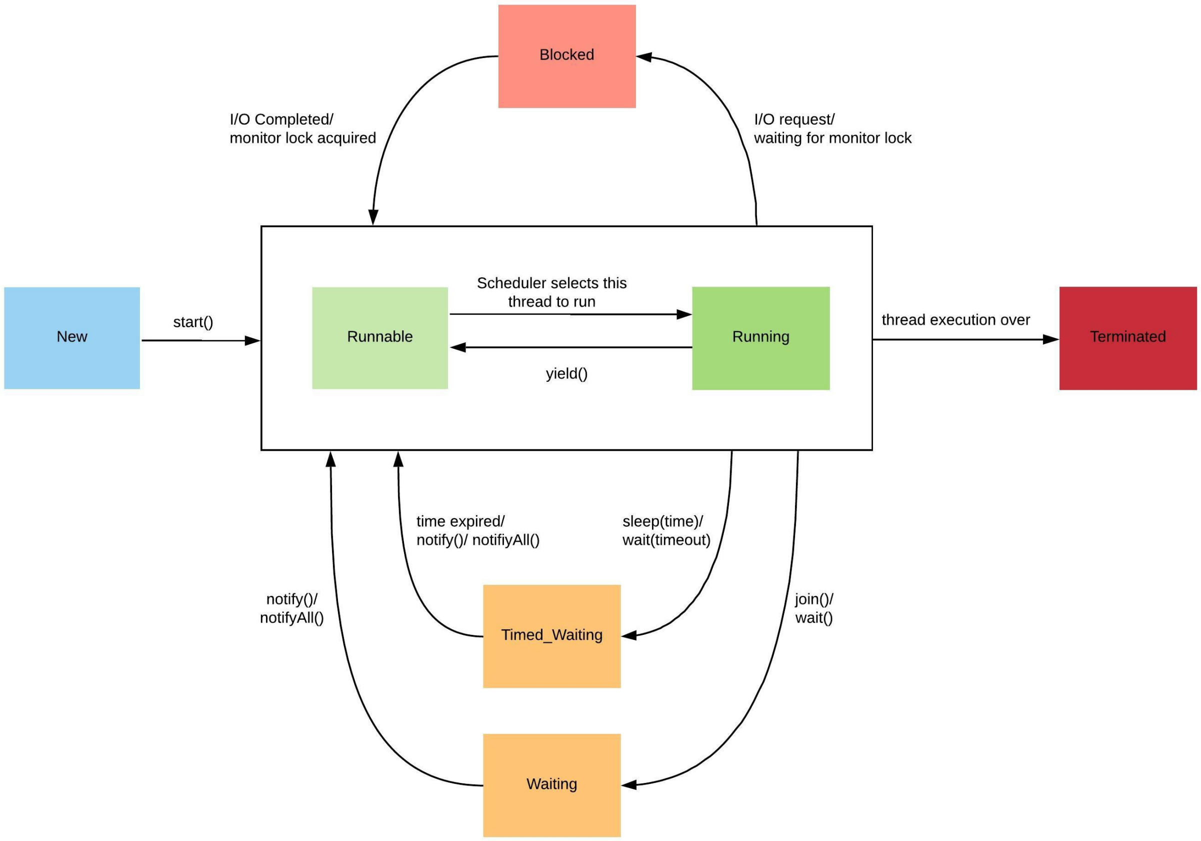
# Methods

- `getName(): String`
- `setName(String): void`
- `getPriority(): int`
- `setPriority(int): void`
- `isAlive(): boolean`

## Methods

- `isInterrupted(): boolean`
- `join(): void`
- `run(): void`
- `sleep(long): void`
- `start(): void`

# Thread Lifecycle



# Creating Thread

## Ways to create

- By extending `Thread` class
- By implementing `Runnable` interface

# Constructors

- `Thread()`
- `Thread(String name)`
- `Thread(Runnable r)`
- `Thread(Runnable r, String name)`

# By extending Thread class

```java
class Multi extends Thread {
    public void run() {
        System.out.println("thread is running...");
    }

    public static void main(String[] args) {
        Multi t1 = new Multi();
        t1.start();
    }
}
```

# Implementing Runnable interface

```
class Multi3 implements Runnable {
    public void run() {
        System.out.println("thread is running...");
    }

    public static void main(String[] args) {
        Multi3 m1 = new Multi3();
        Thread t1 = new Thread(m1);
        t1.start();
    }
}
```