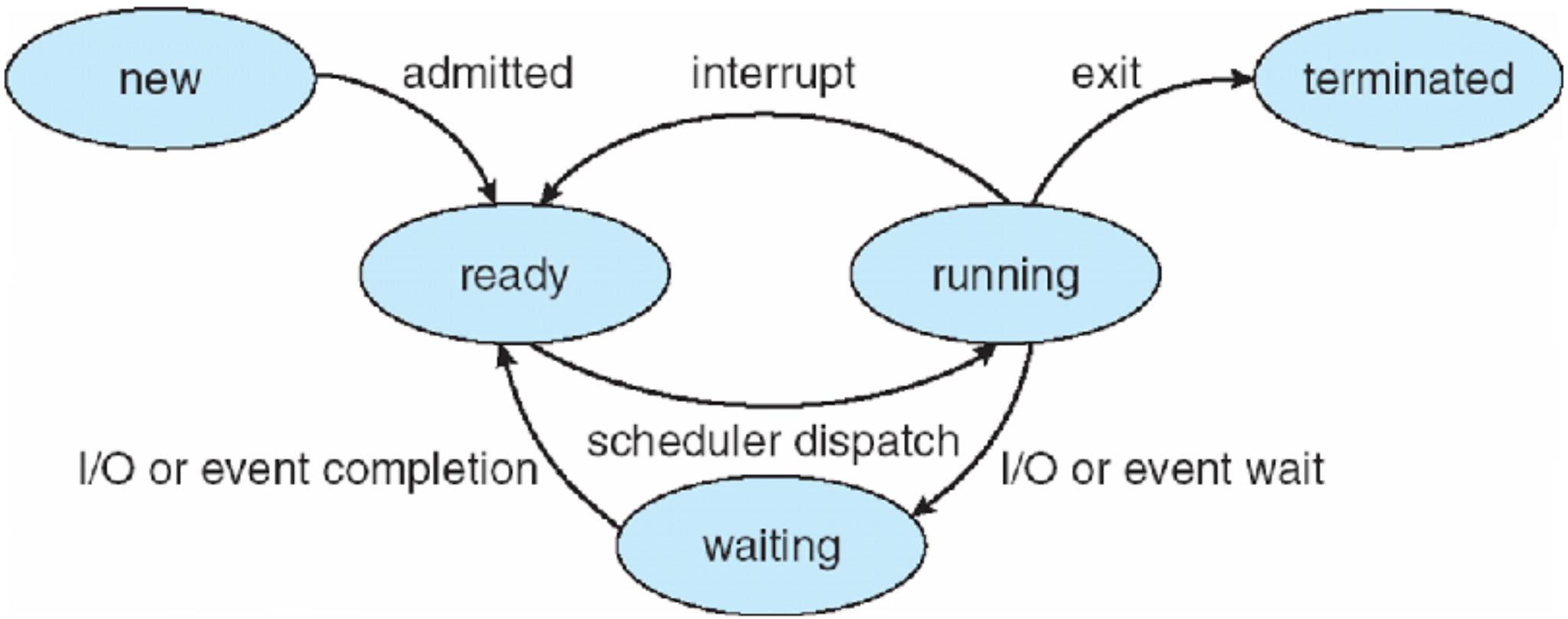
Threads em Java

Sistemas Operacionais
Gerência de processos

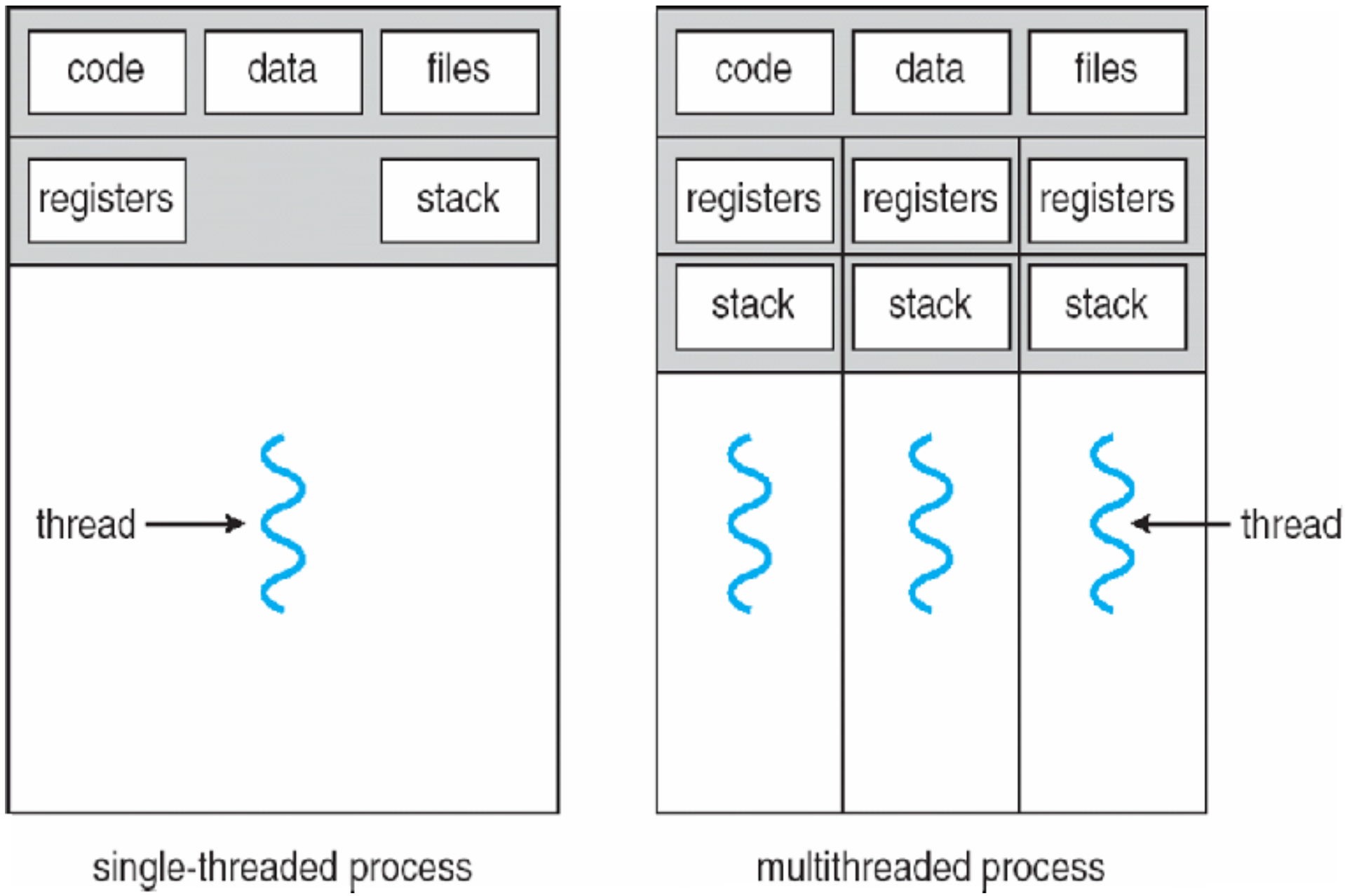
Agenda

* Contextualização
* Introdução
* Ciclo de vida
* Como criar threads
* Exemplos

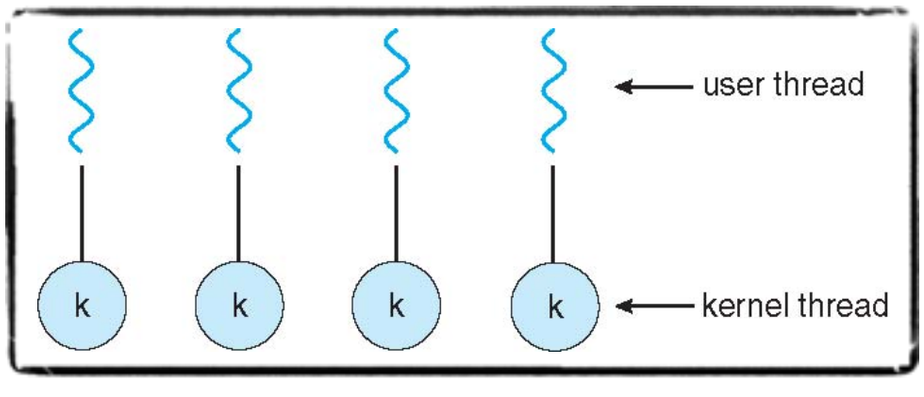
Estados de processo

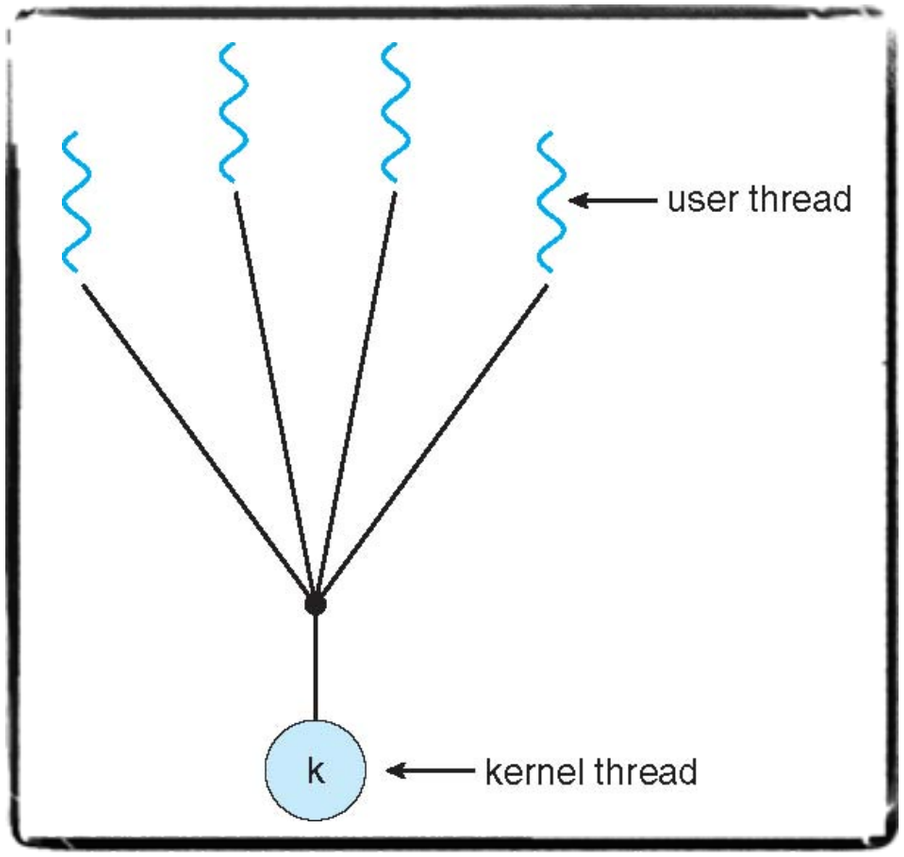


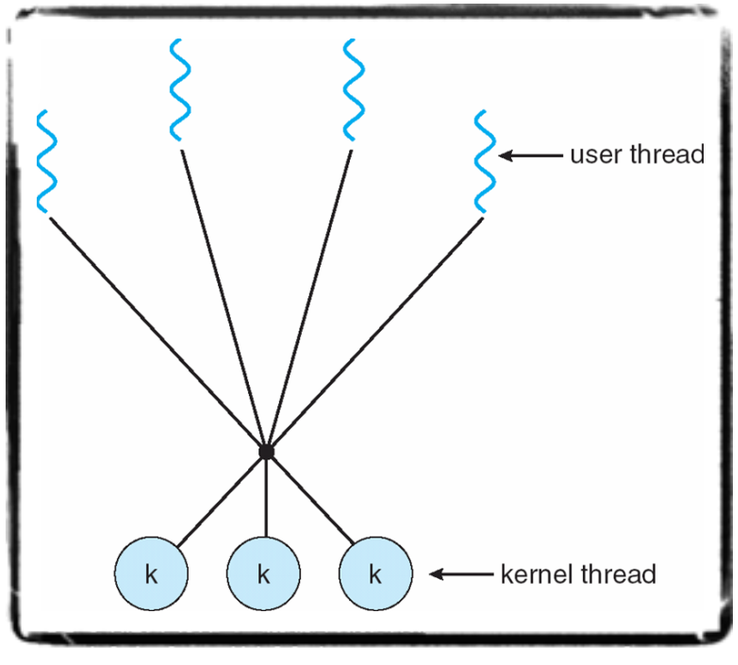
Threads

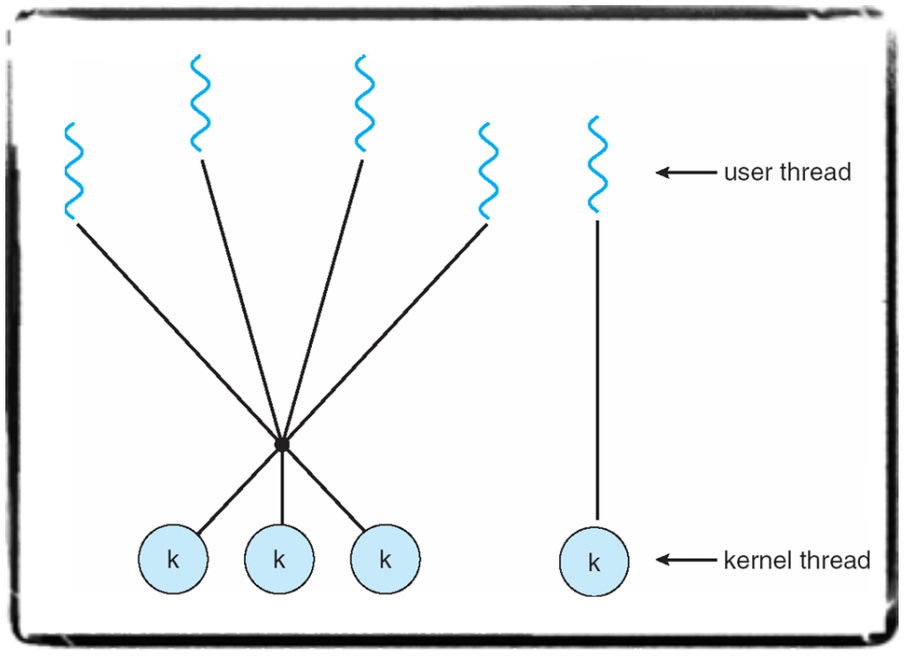

code
data
files
registers
stack
thread
single-threaded process
code
data
files
registers
registers
registers
stack
stack
stack
thread
multithreaded process

Modelos de threads









Agenda

* Contextualização
* Introdução
* Ciclo de vida
* Como criar threads
* Exemplos

Introdução

* Os mecanismos de gerenciamento, comunicação e sincronização de threads foram incorporados a linguagem
* Threads java são representadas por objetos da classe java.lang.Thread

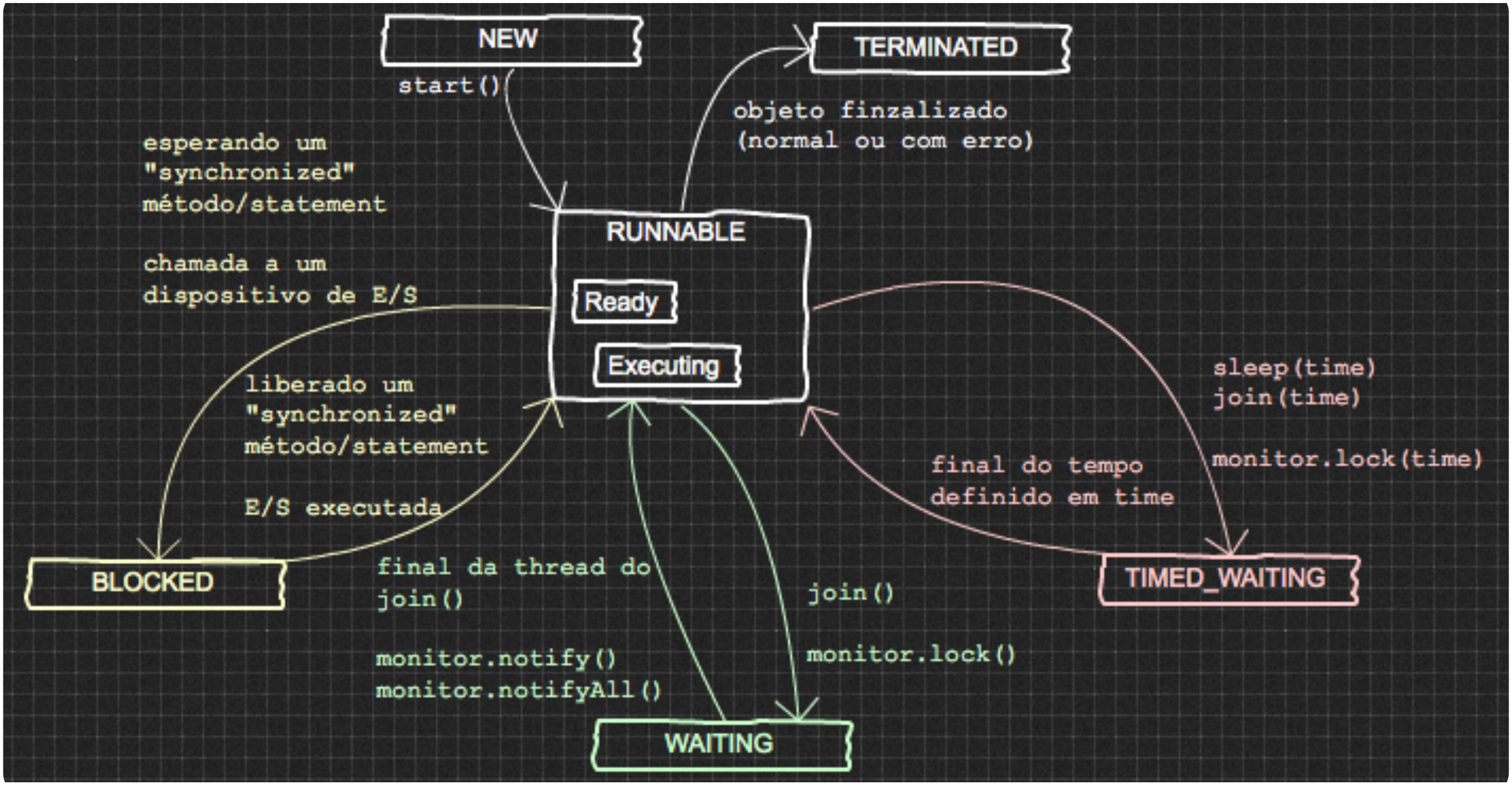
Introdução

* Quando um software Java é executado
  * A JVM cria um objeto do tipo Thread, cuja tarefa é executar o método main()
  * Este objeto Thread é iniciado automaticamente
  * As instruções descritas no método main() são executadas sequencialmente até que o método termine, finalizando a execução do objeto thread

Agenda

* Contextualização
* Introdução
* Ciclo de vida
* Como criar threads
* Exemplos

Ciclo de vida



java.lang.Thread.State

* NEW : um objeto que ainda não foi iniciado
* RUNNABLE : um objeto que esta em execução na JVM
* BLOCKED : um objeto executa chamada a um dispositivo de E/S ou "entra" em um bloco de código "synchronized"
* WAITING : um objeto esta aguardando por eventos externos (monitor.notify* ou final de execução de um objeto thread)
* TIME_WAITING : um objeto aguarda por um tempo determinado
* TERMINATED : um objeto completou sua execução (normal ou com erro)

Agenda

* Contextualização
* Introdução
* Ciclo de vida
* Como criar threads
* Exemplos

Como criar threads

* Apresento 2 formas de criar explicitamente um objeto thread em java
  * extend java.lang.Thread
  * implements java.lang.Runnable

Como criar threads

* Apresento 2 formas de criar explicitamente um objeto thread em java
  * extend java.lang.Thread
    * Cria uma classe que herde de Thread
    * Instancia um objeto desta nova classe (nomeada ou anônima)
  * implements java.lang.Runnable

Como criar threads

* Apresento 2 formas de criar explicitamente um objeto thread em java
  * extend java.lang.Thread
  * implements java.lang.Runnable
    * Cria uma classe que implemente a interface java.lang.Runnable
    * Instancia um objeto dessa nova classe
    * Cria um objeto do tipo Thread, passando como parâmetro o objeto de interface Runnable

Como criar threads

* Implementar o método run()
  * Em ambos os casos apresentados, é necessário implementar na nova classe o método run()
* Este método irá descrever o que realmente as instruções a serem executadas da thread

Agenda

* Contextualização
* Introdução
* Ciclo de vida
* Como criar threads
* Exemplos

Exemplo [extends Thread]

```java
public class SimpleThread extends Thread {

  final private int MAX = 10;
  final private int INIT = 0;

  public SimpleThread() {
    this("unnamed");
  }
  public SimpleThread(String name) {
    super(name);
  }

  public void run() {
    System.out.println("Thread [" + getName() + "] is started!");
    for (int i = INIT; i < MAX; i++) {
      System.out.println(getName() + " - " + i);
    }
    System.out.println("Thread [" + getName() + "] is done!");
  }
}
```

Exemplo [implements Runnable]

```
1  public class SimpleRunnable implements Runnable {
2
3    final private int MAX = 10;
4    final private int INIT = 0;
5    private String name;
6
7    public SimpleRunnable() {
8      this("unnamed");
9    }
10   public SimpleRunnable(String name) {
11     super();
12     this.name = name;
13   }
14
15   private String getName() {
16     return this.name;
17   }
18
19   public void run() {
20     System.out.println("Thread [" + getName() + "] is started!");
21     for (int i = INIT; i < MAX; i++) {
22       System.out.println(getName() + " - " + i);
23     }
24     System.out.println("Thread [" + getName() + "] is done!");
25   }
26 }
```

Exemplo
Instanciando e executando

```
1  public class SimpleThreadDemo {
2    public static void main(String[] args) {
3      Thread thread = new SimpleThread("SimpleThread");
4
5      Runnable runnable = new SimpleRunnable("SimpleRunnable");
6      Thread threadRunnable = new Thread(runnable);
7
8      thread.start();
9      threadRunnable.start();
10   }
11 }
```

Exemplo
Possível resultado

```
MacBook-Pro-de-Leonardo:threads leo$ javac SimpleThreadDemo.java
MacBook-Pro-de-Leonardo:threads leo$ java SimpleThreadDemo
Thread [SimpleThread] is started!
SimpleThread - 0
SimpleThread - 1
SimpleThread - 2
SimpleThread - 3
SimpleThread - 4
SimpleThread - 5
SimpleThread - 6
SimpleThread - 7
SimpleThread - 8
SimpleThread - 9
Thread [SimpleThread] is done!
Thread [SimpleRunnable] is started!
SimpleRunnable - 0
SimpleRunnable - 1
SimpleRunnable - 2
SimpleRunnable - 3
SimpleRunnable - 4
SimpleRunnable - 5
SimpleRunnable - 6
SimpleRunnable - 7
SimpleRunnable - 8
SimpleRunnable - 9
Thread [SimpleRunnable] is done!
MacBook-Pro-de-Leonardo:threads leo$
```

Bibliografia

Threads em Java
Gerência de processos
Sistemas operacionais

Bibliografia

* **The Java Tutorials: concurrency**. Disponível em http://docs.oracle.com/javase/tutorial/essential/concurrency/ (acessado em 18/01/2013)
* SILBERSCHATZ, G.; GAGNE, G. **Sistemas Operacionais com Java**. Campus, 7a Ed, 2007.

Threads em Java

Sistemas Operacionais
Gerência de processos