Sistemas Distribuídos

Universidade do Minho

2011/2012

Apresentação

- Francisco Cruz.
- Email: fmcruz@di.uminho.pt
- Laboratório SD 2.20 no DI.

Materiais Aulas

- Java
- SDK 6

- Eclipse
- IntelliJ
- ...

- Fios de execução concorrentes de um programa.
- Um mesmo processo pode ter várias threads.
- Partilham recursos como memória.

- Analogia:
  - Uma thread de um processo são como vários cozinheiros que seguem as instruções do mesmo livro de culinária, mas não necessariamente todos na mesma página.

- Classes, interfaces e métodos mais relevantes:
  - java.lang.Runnable
    – interface implementado por classes cujas instâncias são executadas por uma thread

    – classes que implementem o interface têm que implementar o método run();
  - java.lang.Thread

    – implementa java.lang.Runnable
    – classes que extendam Thread devem reimplmentar o método run()

    – outros métodos relevantes: Thread(), start(), sleep(), join();

- Exemplo HelloWorld:

```
public class HelloRunnable implements Runnable {

    public void run() {
        System.out.println("Hello from a thread!");
    }

    public static void main(String args[]) {
        (new Thread(new HelloRunnable())).start();
    }

}
```

Exemplo HelloWorld 2:

```
public class HelloRunnable2 implements Runnable {
        int n;
    public void run() {
        System.out.println(n);
    }

   HelloRunnable2(int a) {
       n=a;
   }
    public static void main(String args[]) {
       HelloRunnable2 r222 = new HelloRunnable2(222);
       HelloRunnable2 r111 = new HelloRunnable2(111);
       Thread t1=new Thread(r222);
       Thread t2=new Thread(r111);;
      System.out.println("Antes");
       t1.start();
       t2.start();
      System.out.println("Depois");
       try {
           t2.join();
           t1.join();
       } catch (InterruptedException e) {}
      System.out.println("Fim");
    }

}
```

Exemplo HelloWorld 3:

```
public class HelloRunnable3 implements Runnable {
    int n;
     public void run() {
         System.out.println(n);
           this.set(111);
     }
   HelloRunnable3(int a) {
       n=a;
   }

   public void set(int b) { n=b; }

    public static void main(String args[]) {
       HelloRunnable3 r=new HelloRunnable3(222);
       Thread t1=new Thread(r);
       Thread t2=new Thread(r);
      System.out.println("Antes");
       t1.start();
       t2.start();
      System.out.println("Depois");
       try {
          t2.join();
          t1.join();
       } catch (InterruptedException e) {}
    }
}
```

Exercicios

- 1. Crie uma thread que imprima e incremente um contador a cada segundo, enquanto outra thread repete o standard input para o standard output.

- Nota: o corpo (ou thread) principal do programa espera implicitamente que as threads terminem a sua execução.

Exercicios

- 2. Crie 10 threads que incrementem dez milhões de vezes um contador partilhado. No final da execução das threads, o corpo ou thread principal do programa deve imprimir o valor final do contador.
- Nota: a thread principal do programa tem que esperar que as threads criadas terminem, antes de imprimir o valor do contador.