

Redes de Computadores

Aulas Práticas

Material de suporte às aulas de Redes de Computadores – Copyright DI – FCT/ UNL /

Programação Concorrente com Threads em Java

Aula #2

Material de suporte às aulas de Redes de Computadores – Copyright DI – FCT/ UNL /

Sumário

- Java Threads
- Exemplo em Java (ConcorrentEchoClient.java)
- Exercício (“ChatRoom / Talk a N”)

Java Threads

- A linguagem Java permite ter vários fios de execução independentes (**threads**) num mesmo programa.
- Além do **thread principal** que arranca no método **main()**, um programa Java pode criar tantos *threads* auxiliares quantos os necessários, desde que não esgote os recursos da máquina...
- Os threads são representados por objectos da classe **java.lang.Thread**

Criação e arranque de Threads...

- Há várias possibilidades para criar threads...
 - `class myClass extends Thread { public void run() {...} }`
 - **Criação:** `myClass a = new myClass(...);`
 - **Arranque:** `a.start()`
 - o novo thread arranca dentro do método `run()` de `myClass`
 - `class myClass implements Runnable { public void run() {...} }`
 - **Criação:** `Thread t = new Thread(new myClass(...));`
 - **Arranque:** `t.start()`
 - o novo thread arranca dentro do método `run()` de `myClass`
 - E, ainda, usando classes anónimas...
 - Programação Java avançada...
(os curiosos podem estudar o código de exemplo na página da cadeira...)

Terminação de Threads

Os threads **não se podem matar** directamente...

Os threads **terminam** quando chegam ao fim do método **run()** onde arrancaram.

Comunicação e Threads

Os threads são necessários quando um programa precisa de fazer, **em simultâneo**, várias operações de leitura(*), as quais são **bloqueantes** por natureza.

Por exemplo, só com o thread principal não é possível estar à espera de uma mensagem num socket (**receive**) e, ao mesmo tempo, querer ler da consola (**System.in**). Ou bloqueia numa operação ou na outra...

Criando um thread auxiliar resolve-se esse problema...

(*) sobre sockets e outros canais de leitura, como por exemplo o System.in)

Material de suporte às aulas de Redes de Computadores – Copyright DI – FCT/ UNL /

Exemplo...

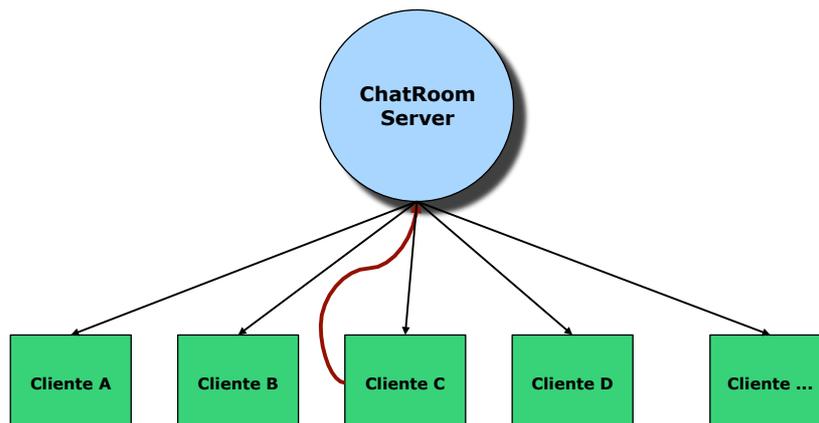
Um cliente do serviço ECHO concorrente, onde se utiliza um thread para ler da consola e outro para ler as mensagens de ECHO re-enviadas pelo servidor...

Material de suporte às aulas de Redes de Computadores – Copyright DI – FCT/ UNL /

Programação em Java com Threads

- Pacotes
 - java.lang
- Classes específicas
 - java.lang.**Thread** → fio de execução independente
 - java.lang.**Runnable** → interface auxiliar para lançar Threads dentro de outras classes...

Exercício: “ChatRoom / Talk a N”



Sugestões...

No servidor do "ChatRoom" pode-se manter os clientes conhecidos num `java.util.Vector` ou num `java.util.Set`

- Cada cliente tem que ser identificado por um par <máquina/IP, porto>. Para tal, a classe `java.net.InetSocketAddress` pode ser usada para guardar e aceder a essa informação.

```
import java.net.* ;
```

```
import java.util.* ;
```

```
Vector<InetSocketAddress> clients = ...
```