Redes de Computadores

Aulas Práticas

Sincronização em Java

Aula #5

Sumário

- Sincronização em Java
- Exercício

Sincronização (1)

- O funcionamento concorrente de múltiplos fluxos de execução (*threads*) pode causar problemas no acesso a estruturas de dados partilhadas

- A linguagem Java permite sincronizar os acessos concorrentes usando as seguintes primitivas baseadas em monitores:
  - bloco ***synchronized*** : obtém explicitamente o monitor do objecto especificado

```
synchronized( obj) {
   // código a executar em exclusão mútua
}
```

  - método ***synchronized*** : obtém (implicitamente) o monitor do objecto **(this)** em que o método está definido

```
synchronized void f( int n) {
   // código a executar em exclusão mútua
}
```

Sincronização – bloco *synchronized*

```
class Exemplo {
   List<String> msgs = new LinkedList<String>();

   void add(String s) {
     synchronized( msgs) {
           msgs.add( s);
     }
   }
   void remove(String s) {
     synchronized( msgs) {
           msgs.remove( s);
     }
   }
   void dump() {
     synchronized( msgs) {
           Iterator<String> it = msgs.iterator();
           while( it.hasNext())
                 System.out.println( it.next());
     }
   }
}
```


Sincronização – método *synchronized*

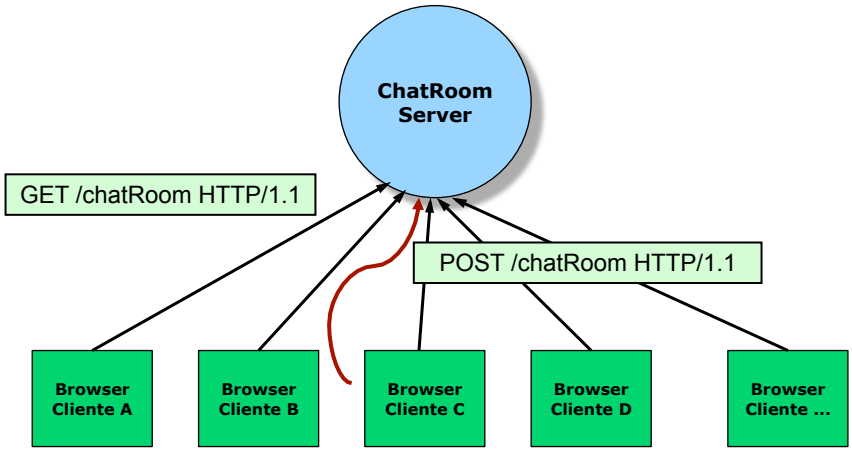
```
class Exemplo {
   List<String> msgs = new LinkedList<String>();

   synchronized void add(String s) {
     msgs.add( s);
   }
   synchronized void remove(String s) {
     msgs.remove( s);
   }
   synchronized void dump() {
     Iterator<String> it = msgs.iterator();
     while( it.hasNext())
           System.out.println( it.next());
   }

}
```

Necessidade da sincronização em programas cliente/servidor

- Servidores concorrentes
  - Acesso a estruturas de dados partilhadas por múltiplos pedidos.
    - Modificação-modificação ou Modificação-leitura
- NOTA: pode acontecer aquando da execução por diferentes *threads* da mesma operação (tipo de pedido) ou de operações diferentes

Exercício: "ChatRoom / Talk a N" concorrente

Tornar o servidor de ChatRoom em HTTP criado na última aula num servidor concorrente



Exercício: Sincronização

- "ChatRoom" HTTP concorrente
  - *Guia* da solução
    - Criar *threads* para atenderem pedidos concorrentes
    - Analisar quais as estruturas de dados partilhadas e identificar as instruções que devem ser executadas em exclusão mútua
- Problemas anteriores
  - Analisar quais os programas anteriores que têm problemas de sincronização