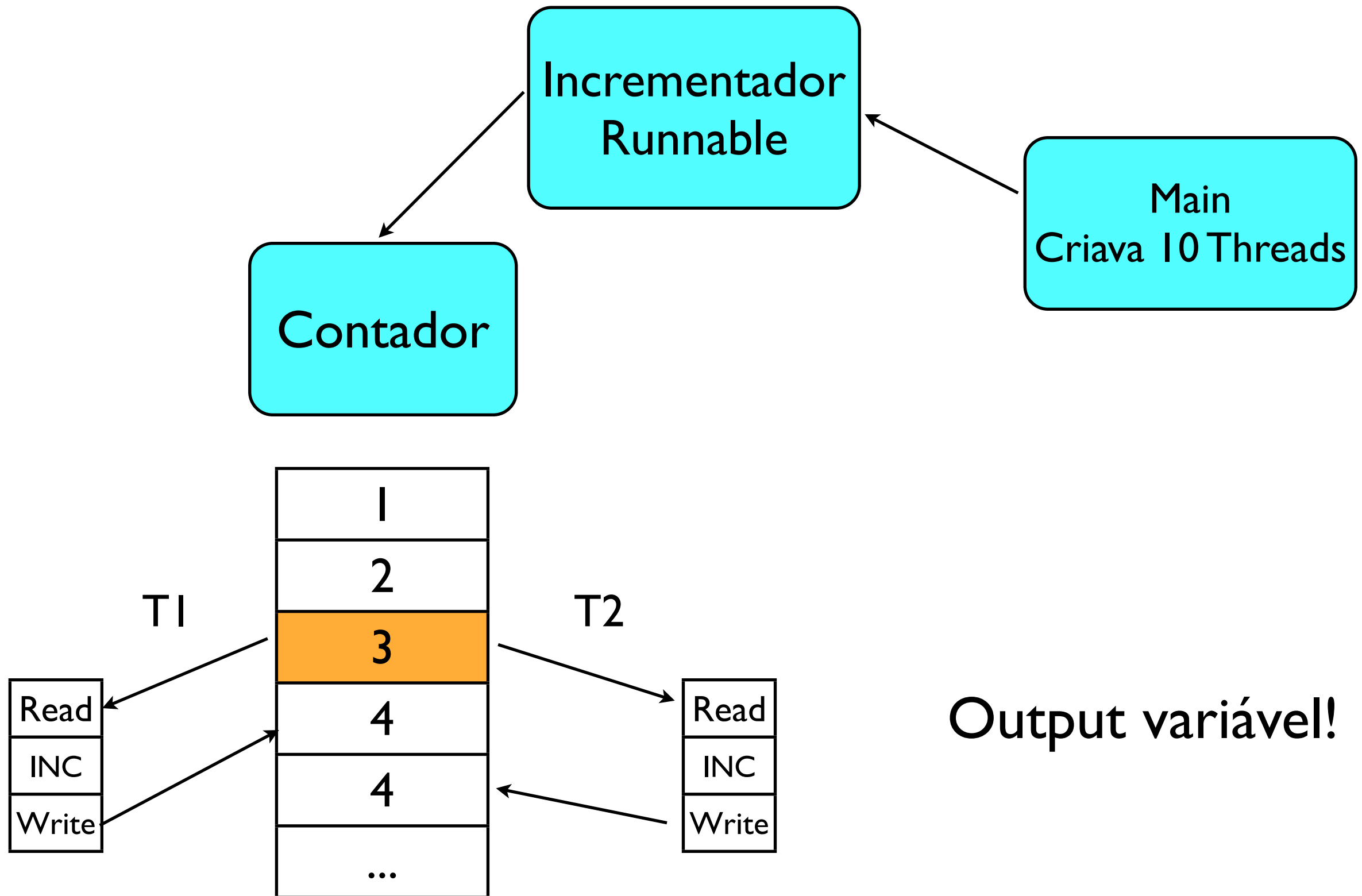


Sistemas Distribuídos

Universidade do Minho
2011/2012



- `java.lang.Runnable`
- `java.lang.Thread`
- classes que extendam `Thread` ou implementem `Runnable` devem reimplmentar o método `run()`
- métodos relevantes: `Thread()`, `start()`, `sleep()`, `join()`;



- Exclusão mútua: garantir que dois processos ou threads não tenham acesso simultaneamente a um recurso partilhado.
- Proteger secções críticas do código.

- Em JAVA:
 - Monitor locks para garantir exclusão mútua;
 - mecanismo intrínseco de exclusão mútua disponível em todos os objectos
 - `synchronized metodo (argumentos) { statements }`
 - `synchronized (objecto) { statements }`

Exemplo:

```
public synchronized void metodo() {  
    i++;  
}
```

```
synchronized (objecto){  
    i++;  
}
```

- 1. Modifique o exercício anterior — incremento concorrente de um contador partilhado — de modo a garantir a execução correcta do programa.

- 2. Implemente uma classe Banco que ofereça os métodos de consulta, crédito e débito de valores sobre um número fixo de contas (com saldo inicial nulo). Utilize exclusão mútua ao nível do objecto Banco.

- 3. Acrescente o método transferir à classe Banco como composição das operações de débito e crédito de um valor sobre duas contas.