



**Prof.<sup>a</sup> Ana Cristina Barreiras Kochem Vendramin**

Universidade Tecnológica Federal do Paraná (UTFPR), Câmpus Curitiba  
Departamento Acadêmico de Informática (DAINF)

**Avaliação (valor 2,5)**  
**Arquitetura Cliente-Servidor, Eventos e Notificações,**  
**JAVA RMI ou PYRO**

**Exclusão Mútua**

Desenvolver uma aplicação que garanta a consistência no acesso a recursos compartilhados entre processos. Utilize a *middleware* **Java RMI** (*Remote Invocation Method*) ou **PyRO** (*Python Remote Objects*) para prover a comunicação entre os processos clientes e servidor.

- Quando um processo precisar acessar um recurso, ele precisa obter permissão do servidor para entrar em uma seção crítica (SC) e acessar o recurso desejado. Após entrar na SC e acessar o recurso desejado, o processo sairá da SC e liberará o recurso. Requisitos para exclusão mútua: (1) Segurança: no máximo um processo por vez pode ser executado na SC; (2) Subsistência: os pedidos para entrar/sair de seção crítica precisam ser bem-sucedidos; (3) Ordenação: ordenar as mensagens que solicitarem a entrada na SC.
- Implemente o algoritmo do servidor central (slides 4 e 5 da aula 5\_Coordenação&Acordo) com os seguintes requisitos (**valor 2,5**):
  - Quatro processos clientes e um processo servidor que poderão ser executados na mesma máquina;
  - Dois recursos que serão compartilhados entre os quatro processos clientes. Servidor deve possuir duas filas para controlar o acesso aos recursos;
  - Não ocorrerão falhas nos processos e no canal de comunicação;
  - O pedido para acessar um recurso (pedido de entrada na seção crítica) indica um registro de interesse em receber uma notificação de evento quando o recurso desejado for liberado (caso o mesmo não esteja disponível no momento da solicitação). Quando um processo liberar um recurso, ele deverá avisar o servidor e este, por sua vez, enviará uma notificação dessa liberação (concessão do *token*) ao primeiro processo da fila do recurso correspondente e removerá o pedido desse processo da fila do recurso correspondente. **O envio da notificação deverá ser feito através de uma chamada de método do servidor no cliente** (isto é, o cliente deve implementar um método para receber as notificações do servidor). Então, haverá chamada bidirecional de métodos (valor 1,5);

- Existe um limite de tempo fixo para uso de cada recurso. Caso o tempo expire, o recurso será automaticamente liberado e o servidor deve tratar do envio da notificação do evento (valor 0,25);
- Apenas no primeiro pedido de acesso a um recurso, o servidor retornará ao cliente a sua chave pública (valor 0,25);
- Toda mensagem de notificação (concessão do *token*) enviada pelo servidor a um cliente deve conter uma assinatura digital. Para isso, o servidor utilizará a sua chave privada. Quando o processo cliente receber a mensagem, ele deverá validar a assinatura utilizando a chave pública correspondente que possui do servidor (valor 0,5).

Observações:

- Desenvolva uma interface com recursos de interação apropriados;
- É obrigatória a defesa da aplicação para obter a nota.
- O desenvolvimento da aplicação pode ser individual ou em dupla. Porém, a defesa da aplicação e a nota são individuais.