Prof.ª Ana Cristina Barreiras Kochem Vendramin
Universidade Tecnológica Federal do Paraná (UTFPR), Câmpus Curitiba
Departamento Acadêmico de Informática (DAINF)

Avaliação (valor 2,5)
PyRO. Arquitetura Processos Pares.

Exclusão Mútua

Desenvolver uma aplicação para coordenação distribuída de acesso a recursos compartilhados.

- (1,0) Implemente o algoritmo proposto por Ricart e Agrawala (slides 7-11 do arquivo CoordenacaoAcordo.pdf) e utilize o *middleware* **PyRO** (*Python Remote Objects*) para prover a comunicação entre os processos.
    - Considere quatro processos pares (ex: PeerA, PeerB, PeerC, PeerD) que serão executados na mesma máquina.
    - Quando um processo precisar acessar um recurso, ele primeiro precisa obter permissão dos demais processos 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 e sair de uma seção crítica precisam ser bem-sucedidos; (3) Ordenação: ordenar as mensagens que solicitarem a entrada na SC.

- (1,5) Implemente as seguintes adaptações:

    - **Uso da comunicação *unicast*** em vez de *multicast*;
    - (0,3) **Uso do servidor de nomes do PyRO para armazenar as referências dos objetos remotos**. Haverá apenas um servidor de nomes PyRO na máquina. Logo, apenas um processo será capaz de criar esse servidor de nomes. Adicione no código o tratamento adequado para evitar múltiplas instâncias desse servidor. Caso o servidor de nomes já exista, será apenas obtida sua referência. Caso contrário, o servidor de nomes será criado e sua referência obtida. Assuma que cada processo conhece o nome das aplicações dos demais processos (exemplo: “PeerA”, “PeerB”, “PeerC” e “PeerD”) para poder obter junto ao servidor de nomes a referência de cada objeto remoto através do nome da sua aplicação;

- (0,2) **Controle de Tempo de Acesso ao Recurso**. O acesso ao recurso será limitado a um tempo máximo definido. Após o tempo expirar, o processo deverá liberar automaticamente o recurso;
- (0,5) ***Heartbeat* para detecção de falhas nos processos**. Cada processo deve enviar periodicamente uma mensagem de "vida" (*heartbeat*) para os demais processos. Se um processo não receber o *heartbeat* de outro dentro de um intervalo de tempo, deverá considerá-lo falho e removê-lo da lista de processos ativos. Isso evita que processos inativos permaneçam nas filas de acesso a recursos, ou causem bloqueios indevidos. Antes de enviar a autorização para uso de um recurso (token ou confirmação), o processo que responde deverá: verificar se o solicitante ainda está ativo, com base no último *heartbeat* recebido ou interação recente. Essa verificação previne que um recurso seja alocado a um processo que já falhou, o que poderia levar ao bloqueio do sistema;
- (0,5) **Envio de respostas aos pedidos e uso de temporizadores para esperar as respostas**. Sempre que um processo enviar um pedido para entrar na seção crítica, ele deverá iniciar um temporizador. Se não receber resposta dentro do tempo esperado (*timeout*), o processo deverá considerar o destino inativo e removê-lo da lista de processos ativos. Isso garante que não se espere indefinidamente por processos inativos. O receptor de um pedido sempre enviará uma resposta, concedendo ou negando a permissão.

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 é individual.

Profª. Ana Cristina Barreiras Kochem Vendramin
DAINF/UTFPR