

Trabalho Prático 2

Especificação:

O segundo trabalho prático da disciplina tem como objetivo aprofundar o conhecimento relacionado aos tópicos de exclusão mútua e eleição distribuída. O acesso à região crítica deverá ser controlado pelo par de métodos $P()$ / $V()$, e deve ser utilizado o algoritmo baseado em um coordenador central. A fim de testar a funcionalidade implementada, é necessário que os alunos escolham uma aplicação que necessite de pelo menos 3 semáforos. Para garantir que a aplicação continue funcionando mesmo na ocorrência de falha de algum processo, é solicitado que seja utilizado o algoritmo de eleição do valentão (Bully Algorithm).

As especificações da interface de linha de comando são as seguintes:

- *programa arquivo_config.txt linha*
onde *arquivo_config.txt* é o nome do arquivo de configuração e *linha* é a linha desse nodo (descrito abaixo).

O arquivo de configuração possui um número indeterminado de linhas, e cada linha possui o seguinte formato:

- *id host porta*
onde *id* é um inteiro usado para identificar um nodo, *host* é o endereço IP do nodo e *porta* é a porta onde o nodo está escutando (em uma thread). O programa aceita como parâmetro um inteiro que indica seu próprio ID (a linha no arquivo de configuração que indica qual porta o nodo deve usar). O programa deve usar as outras linhas para saber quem são os outros nodos no sistema (isso é necessário para o algoritmo Bully).

Comportamento do programa:

O nodo com o maior ID é o coordenador inicial, e será responsável por manter o estado de acesso aos semáforos. Os outros nodos enviam mensagens ao coordenador sempre que desejarem entrar e sair de uma região crítica. Se o coordenador não confirmar (a confirmação de recebimento deve ser feita mesmo quando negada a entrada na região crítica), então o deve-se iniciar uma eleição, de acordo com o algoritmo do valentão. Quando um novo coordenador for selecionado, os nodos devem começar a enviar as solicitações de entrada e saída de região crítica para o mesmo. É importante que o mecanismo seja validado eliminando-se o coordenador em algum momento. O coordenador deve ser o primeiro processo a ser carregado no ambiente para que o mecanismo funcione.

Formato de saída:

Cada programa individual deve produzir uma saída de sua execução. O seguinte formato pode ser usado:

- (uso de exclusão mútua)
- [processamento]
- *lock()*
- [ganhou lock]
- *unlock()*
- [liberou lock]

- (algoritmo Bully)
- Terminação do coordenador: *t i*, onde *i* é o ID do coordenador;
- Início da eleição: *e [v]*, onde *v* é um vetor de IDs para onde a mensagem de eleição é enviada. Essa mensagem é impressa por um nodo iniciando uma eleição.

- Fim de eleição: $c\ i$, onde i é o ID do novo coordenador. Essa mensagem é impressa por todos os nodos.

Uma máquina com um recurso compartilhado entre os outros nodos deve apresentar em um *log* as entradas e saídas de diferentes processos na região crítica, e essa será a prova de que o mecanismo está funcionando da maneira esperada.

Entrega:

O trabalho deve ser realizado em grupos de 2 ou 3 integrantes. Para a entrega é solicitado o código fonte da implementação. O trabalho será apresentado em aula, e para isso é importante que o grupo se organize para em 10 minutos (no máximo):

- Descrever e linhas gerais a abordagem adotada para solucionar o problema;
- Demonstrar a funcionalidade de sua implementação.