**Trabalho prático 1**

Disciplina: Sistemas Distribuídos

Professor: Aurélio Faustino Hoppe

Alunos: Antônio José Brogni, Lucas Bauchspiess, Guilherme William Back

1)

O algoritmo de exclusão mútua centralizado caracteriza-se pela presença de um processo principal chamado de coordenador. Quando um processo precisa acessar algum recurso compartilhado com outros processos, envia uma requisição para o coordenador. Caso o recurso não esteja sendo utilizado por outro processo, o coordenador concede permissão de acesso ao processo solicitante. Caso o recurso esteja sendo utilizado, o processo solicitante entra em uma fila de espera. Quando o processo que está utilizando o recurso termina a execução, ele libera o recurso e envia uma mensagem de liberação ao coordenador, que, então, permite que o próximo processo da fila seja ou um novo processo (caso a fila esteja vazia) utilize o recurso.

- Vantagens desta utilização: Garante exclusão mútua na utilização do recurso, garantindo que somente um processo por vez acesse o recurso. Além disso, somente 3 mensagens são necessárias para cada utilização do recurso (requisição, concessão, liberação)

- Desvantagens: Como todos os recursos compartilhados passam pelo coordenador, este pode tornar-se um gargalo no desempenho, além de, em caso de falha, poder derrubar todo o sistema.

2)

Considerando somente estes processos, temos a seguinte sequência de passos:

P1 envia uma mensagem para P1, P2, P3, contendo: (recurso1, P1, 5)

P2 envia uma mensagem para P1, P2, P3, contendo: (recurso1, P2, 1)

P3 envia uma mensagem para P1, P2, P3, contendo: (recurso1, P3, 4)

Pela exclusão mútua, o menor tempo recebe prioridade

P1 envia uma mensagem de ok para P2

P3 envia uma mensagem de ok para P2

P2 utiliza o recurso e adiciona os outros processos à fila. Após utilização do recurso:

P2 envia uma mensagem de ok para P1 e P3

P3 envia uma mensagem para P1 e, P3, contendo: (recurso1, P1, 5)

P3 envia uma mensagem para P1 e, P3, contendo: (recurso1, P3, 4)

P1 envia uma mensagem de ok para P3

P3 utiliza o recurso e adiciona P1 à fila. Após utilização do recurso:

P3 envia uma mensagem de ok para P1

P1 utiliza o recurso

P1 libera o recurso.

3)

- Algoritmo de Valentão (Bully):

P3 envia mensagem de eleição para os processos com id maior, neste caso, somente P4. P4 não responde. P3 se torna o coordenador, por ser o processo ATIVO com maior id.

- Algoritmo do anel lógico:

E[ ] é a lista com todos os id’s. Cada processo adiciona seu próprio id.

P3 percebe que P4 caiu. Como P4 era sucessor de P3, para formar um cliclo, P0 passa a ser o sucessor de P3.

P3 envia para **P0** – E[3]

P0 envia para P1 – E[3, 0]

P1 envia para P2 – E[3, 0, 1]

P2 envia para P3 – E[3, 0, 1, 2]

P3 recebe a mensagem de P2, verifica que 3 é o maior id ativo e envia novamente uma mensagem para P0, informando que P3 é o coordenador. Esta mensagem é repassada da mesma forma que a mensagem anterior.

4)

- Vantagens do algoritmo de Bully:

Sempre será possível obter um novo coordenador quando um determinado coordenador cair;

Quanto maior o id do processo que perceber a queda do coordenador, menos mensagens são trocadas.

- Desvantagens do algoritmo de Bully:

Em sistemas muito grandes, caso um processo com id muito pequeno perceba a falha do coordenador o número de mensagens trocadas pode escalar de forma demasiada.

- Vantagens do algoritmo de Anel Lógico:

O número de mensagens trocadas sempre será 2x número de processos, oque pode evitar muito consumo de recursos em sistemas grandes.

- Desvantagens do algoritmo de Anel Lógico:

Em sistemas pequenos, pode ter uma troca de mensagens maior que o algoritmo de bully.

A mensagem enviada pode não retornar à origem, demandando uma nova eleição.

- Algoritmo escolhido: Utilização do algoritmo do Anel Lógico para sistemas maiores, afim de evitar consumo de recursos causado pela escalabilidade do algoritmo De Bully. Porém, para sistemas menores, optou-se pelo algoritmo de Bully, por sua maior segurança e possibilidade de menor troca de mensagens.