



1) Sincronização de processos

Acesse o link: <https://codeboard.io/projects/385417>

2) Faça a associação de termos e conceitos:

- (A) Algoritmo baseado em ficha
- (B) Algoritmo baseado em permissão

- (A) Algoritmo centralizado
- (B) Algoritmo descentralizado
- (A/B) Algoritmo distribuído
- (A) Algoritmo do anel

3) Avalie as seguintes propriedades dos algoritmos distribuídos que seguem:

3.1) Algoritmo centralizado:

- (a) (V) Exclusão mútua (como acontece?): o coordenador é responsável pela liberação do acesso à região crítica ou ao recurso compartilhado.
- (b) (V) Segurança: ao menos um processo pode executar na região crítica por vez.
- (c) (V) Vivacidade (eventualmente, todos acessam a região crítica): Se o coordenador verifica que a região crítica foi liberada pelo processo, ele autoriza o acesso de outros processos.
- (d) (F) Ordenação (como acontece?): Não. O coordenador não garante a ordem. Ele mantém a fila de processos que não foram atendidos.

3.2) Algoritmo descentralizado:

- (e) (V) Exclusão mútua (como acontece?): Os coordenadores devem concordar com a maioria dos votos para liberar o acesso à região crítica (ou recurso compartilhado) para o processo.
- (f) (V) Segurança (ao menos um processo pode executar na região crítica por vez): Sim. Porém, se muitos processos

chegarem ao mesmo tempo, pode demorar para eles terem permissão de acesso ao recurso compartilhado.

(g) (V) Vivacidade: eventualmente, todos acessam a região crítica.

(h) (V) Ordenação (como acontece?): Não. Porém, a fila pode ser usada por cada coordenador para implementar a ordenação.

4) Acesse o link: [SDCO8A - Algoritmo de Ricart-Agrawala](#)

Preencha as tabelas a seguir de acordo com o algoritmo Ricart-Agrawala:

	Quer RC?	FIFO_REQUEST	FIFO_REPLY
P1	TRUE	P2, P3	P3
P2	FALSE		P1, P3
P3	TRUE	P1, P2	P1,

P2 REPLY P1

P2(origem) tira da REPLY

P1(destino) tira da REQUEST

	Quer RC?	FIFO_REQUEST	FIFO_REPLY
P1	TRUE	P2, P3	P3
P2	FALSE		P1, P3
P3	TRUE	P1, P2	P1,

P2 REPLY P3

P2(origem) tira da REPLY

P3(destino) tira da REQUEST

	Quer RC?	FIFO_REQUEST	FIFO_REPLY
P1	TRUE	P2, P3	P3
P2	FALSE		P1, P3
P3	TRUE	P1, P2	P1,

P3 REPLY P1

P3 (origem) tira da REPLY

P1 (destino) tira P3 REQUEST

	Quer RC?	FIFO_REQUEST	FIFO_REPLY
P1	TRUE	P2, P3	P3
P2	FALSE		P1, P3
P3	TRUE	P1, P2	P1,

P1 pode acessar a Região Crítica

P1 REPLY P3

P1 (origem) tira P3 REPLY

P3 (destino) tira P1 da REQUEST

	Quer RC?	FIFO_REQUEST	FIFO_REPLY
P1	TRUE	P2, P3	P3
P2	FALSE		P1, P3
P3	TRUE	P1, P2	P1,

P3 pode acessar a região crítica

Quem quiser acessar manda requisição para todos quem receber primeiro

5) Supondo que todos os N nós mantenham a lista de todos os N nós do anel no algoritmo do anel:

a) Como é possível detectar que um nó caiu?

Timeout na resposta. O nó não recebe uma confirmação do recebimento do token do próximo nó

b) Como é possível corrigir essa falha?

O nó atual atualiza a sua lista de nós com o próximo índice do nó que houve a falha.

c) Até quantas mensagens precisam ser trocadas entre os hosts para remover o nó do anel lógico que caiu? Justifique sua resposta.

Até N-1 nós para identificar o nó da falha.

6) Ordene os passos do Algoritmo de Eleição do valentão:

(1) O nó que reconheceu a queda do coordenador envia uma mensagem ELEIÇÃO para todos os nós com índices maiores que o dele.

(2) Os nós de índices mais altos respondem e pedem para o nó que enviou ELEIÇÃO parar de enviar mensagens.

(3) Nó não recebe resposta, marca a si mesmo como o novo coordenador e envia COORDENADOR para todos os outros nós.

7) Ordene os passos do Algoritmo de Eleição de anel:

(7) Todos os processos marcam a si mesmos como não-participante.

(1) Nó detecta falha no coordenador, marca a si mesmo participante e envia ELEICAO com o seu ID para o vizinho.

(2) Par recebe ELEICAO com ID maior que o seu, encaminha a mensagem e marca a si mesmo como participante.

(3) Par recebe ELEICAO pela primeira vez com ID menor que o seu, substitui o ID na mensagem com o seu próprio ID e encaminha para o vizinho.

(4) Par recebe ELEICAO pela segunda vez com ID menor que o seu, e descarta a mensagem.

(5) Par recebe o mesmo ID que enviou, marca a si mesmo como ELEITO.

(6) Par recebe ELEITO e marca a si mesmo como não-participante.