



1) Acesse o link: [Atividade Online](#)

1) Em sistemas distribuídos, qual a diferença entre concorrência e colaboração quanto à região crítica?

Concorrência os processos competem pelo acesso a região (não necessita de trocas de mensagens). Já a colaboração há trocas de mensagens.

2) Por que um algoritmo de exclusão mútua baseado em ficha deve ter uma única ficha?

Pois, a ficha garante que um único processo acesse por vez a região crítica.

3) No algoritmo Token Ring, porque o processo não pode acessar novamente a região crítica com a mesma ficha?

Para não causar inanição nos outros processos, pois ela poderia ficar acessando toda hora e não deixar que os outros tenham a chance de acessar.

4) No algoritmo

de exclusão mútua centralizado com passagem de permissão,

por que são necessárias três mensagens para entrar e sair da região crítica?

Uma de requisição do processo para o processo coordenador, uma do processo coordenador permitindo e uma processo para liberar.

5) No algoritmo

de exclusão mútua descentralizado com passagem de permissão,

por que são necessárias $3mk$ mensagens para entrar e sair da região crítica?

Uma de requisição do processo para o processo coordenador, uma do processo coordenador permitindo e uma processo para liberar. m o número de coordenadores e k as trocas de mensagens

6) No algoritmo

de exclusão mútua distribuído com passagem de permissão,

por que são necessárias $2(n-1)$ mensagens para entrar e sair da região crítica?

são necessárias na entrada $2 * ((\text{número de processos}) - 1)$ mensagens (uma solicitação para cada um dos processos e a permissão de cada um dos processos) e na saída $(\text{número de processos} - 1)$ mensagens, sendo uma mensagens de liberação para cada um dos processos.

- 2) As execuções concorrentes a seguir devem garantir exclusão mútua para operações de leitura e escrita. A condição de corrida produz um estado consistente?

Px = processo x; RC=região crítica; Read=leitura da RC; Write=escrita na RC; Calc=adicionar 1 ao valor lido da RC.

(a) () Verdadeiro

() Falso

P1	P2	P3	RC
Read	Read	Read	0
Calc	Calc	Calc	0
Write	Write	Write	1

(b) () Verdadeiro

() Falso

P1	P2	P3	RC
Read		Read	0
Calc	Read		0
Write		Calc	1
	Calc		1
	Write		2
		Write	3

(c) () Verdadeiro

() Falso

P1	P2	P3	RC
Read			0
Calc			0
Write	Read		1
	Calc		1
	Write	Read	2
		Calc	2
		Write	3

(d) () Verdadeiro

() Falso

P1	P2	P3	RC
Read			0
Calc			0
Write			1
	Read		1
	Calc		1
	Write		2
		Read	2
		Calc	2
		Write	3

(e) () Verdadeiro

() Falso

P1	P2	P3	RC
Read	Read	Read	0
Calc	Calc	Calc	0
Write			1
	Write		1
		Write	1

4) Marque verdadeiro ou falso sobre os algoritmos de exclusão mútua baseado em permissão:

- a) (F) O algoritmo centralizado com apenas 1 (um) coordenador é mais rápido que o algoritmo descentralizado para acessar a região crítica.
- b) (V) O algoritmo descentralizado com 'm' coordenadores requer $3mk$ mensagens (m =coordenadores, k =tentativas) para acessar e sair da região crítica.
- c) (V) O algoritmo distribuído de Ricart-Agrawala requer $2(n-1)$ mensagens para entrar e sair da região crítica.
- d) (F) O algoritmo de eleição do anel é mais rápido que o algoritmo de eleição do valentão em uma rede com 100 nós interligados.

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

- (6) 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.
- (2) 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.
- (3) Par recebe ELEICAO pela segunda vez com ID menor que o seu, e descarta a mensagem.
- (4) Par recebe o mesmo ID que enviou, marca a si mesmo como ELEITO.

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

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.

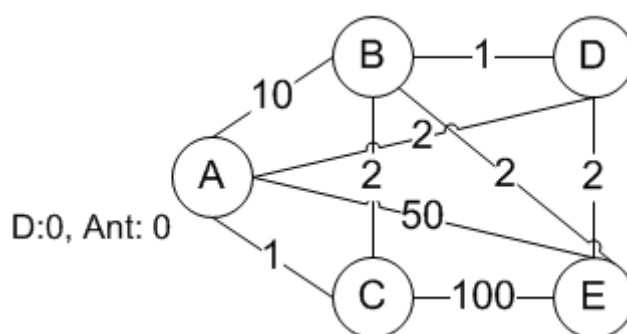
(2) Os nós de índices mais altos respondem e pedem para o nó que enviou ELEIÇÃO parar de enviar mensagens.

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

1) Algoritmo de Estado de Enlace

Acesse o link: <https://codeboard.io/projects/387220>

a) Dado o grafo a seguir, complete a tabela de ligações:



	A	B	C	D	E
Vértice A	0	10	1	2	50
Vértice B	10	0	2	1	2
Vértice C	1	2	0	0	100
Vértice D	2	1	0	0	2
Vértice E	50	2	100	2	0

b) Apresente a sequência de passos para obter o caminho mais curto até a raiz.

VerticeAnt. Distancia da Raiz

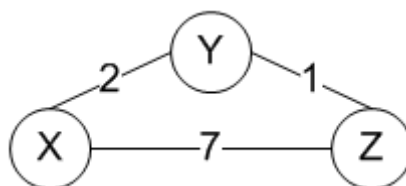
c) Complete a tabela de acordo com o algoritmo de Estado de Enlace:

Vértice	Anterior	Distância até a Raiz
A	A	0
B		
C		
D		
E		

6) Algoritmo de Vetor de Distância

Acesse o link: [📄 SDCO8A - Algoritmo de Vetor de Distância](#)

a) Dado o grafo a seguir, complete a tabela de acordo com o algoritmo de Vetor de Distância:



X recebe de Y	Antiga X	Next	Custo Y	Tabela Y	Resultado	Nova X	Next
X	0	-	2	2	4	0	-
Y	2	-	2	0	2	2	-
Z	7	-	2	1	3	3	Y
X recebe de Z	Antiga X	Next	Custo Z	Tabela Z	Resultado	Nova X	Next
X	0	-	7	7	14		-
Y	2	-	7	1			-
Z	3		7	0			Y
Y recebe de X	Antiga Y	Next	Custo X	Tabela X	Resultado	Nova Y	Next
X	2	-	2	0	2		-
Y	0	-	2	2			-
Z	1	-	2	3			-

Y recebe de Z	Antiga Y	Next	Custo Z	Tabela Z	Resultado	Nova Y	Next
X	2	-	1	7	8		-
Y	0	-	1	1			-
Z	1	-	1	0			-
Z recebe de X	Antiga Z	Next	Custo X	Tabela X	Resultado	Nova Z	Next
X	7	-	3	0	3		-
Y	1	-	3	2	5		-
Z	0	-	3	3	6		-
Z recebe de Y	Antiga Z	Next	Custo Y	Tabela Y	Resultado	Nova Z	Next
X	7	-	1	2	3	3	Y
Y	1	-	1	0	1	1	-
Z	0	-	1	1	2	0	-