

Disciplina DCE540 - Computação Paralela e Distribuída	Método de entrega Moodle da disciplina	Data de entrega 11/02/2022 às 23h59
Professor Iago Augusto de Carvalho (iago.carvalho@unifal-mg.edu.br)		

## Prova 02

Cada aluno deverá submeter um único arquivo .pdf com a resolução da prova.

A prova pode ser realizada de duas maneiras:

- Com papel e caneta, sendo posteriormente escaneada e enviada
- Digitada em algum editor de texto, e.g., Word ou LaTeX

A prova deverá ser entregue no Moodle da disciplina até a data limite.

- Atrasos não serão tolerados

### Exercício 1 (15%)

Qual é a principal ideia por trás da utilização do *Network Time Protocol* (NTP)? Explique, com suas próprias palavras, as principais vantagens e desvantagens da utilização deste método de sincronização.

### Exercício 2 (15%)

Apresente os principais conceitos, vantagens e desvantagens dos seguintes algoritmos de exclusão mútua. Além disso, apresente uma aplicação prática onde cada algoritmo pode (ou deve) ser utilizado

- a) Centralizado
- b) Distribuído
- c) *Token-ring*

### Exercício 3 (20%)

Algoritmos de exclusão mútua coordenam o acesso a recursos compartilhados. Estes algoritmos devem garantir que não ocorram *deadlocks*, que não exista *starvation* e que o acesso aos recursos compartilhados sejam realizados com equidade e justiça. Explique, com suas próprias palavras, como os três algoritmos de exclusão mútua apresentados em aula (centralizado, distribuído e *token-ring*) garantem estas três propriedades.

### Exercício 4 (10 %)

Quando estamos trabalhando com algoritmos de sincronização de *clock*, devemos garantir uma acurácia e uma precisão mínima para todos os *clocks* de um sistema distribuído. Neste contexto, explique o que é

- a) Acurácia
- b) Precisão

## Exercício 5 (20%)

Algoritmos de eleições são utilizados para eleger processos coordenadores em sistemas distribuídos, isto é, aqueles processos que são responsáveis por gerenciar o acesso aos recursos compartilhados. Em sistemas relativamente pequenos, podemos utilizar o algoritmo de bully e o algoritmo do anel para eleger os processos coordenadores. Apresente ambos os algoritmos, descrevendo as vantagens e desvantagens de cada um.

## Exercício 6 (20 %)

*Clocks* físicos são baseados em cristais de quartzo, sendo que o passar do tempo é registrado através da captura das oscilações deste cristal. Sabe-se que estes cristais possuem uma frequência de oscilação ideal  $F$ . Entretanto, é muito comum que a oscilação destes cristais se desvie por, até mesmo,  $\rho$  unidades da frequência ideal  $F$  devido a fatores externos, como desgaste do cristal, alterações na bateria de alimentação do cristal, mudanças extremamente bruscas de temperatura e/ou pressão do ambiente, dentre outros fatores

Assuma que estes fatores externos não sejam totalmente aleatórios, sendo que o ambiente local faz com que ocorra uma diminuição na frequência das oscilações do cristal. Deste modo, qual é o desvio de *clock* esperado para uma máquina qualquer do seu sistema distribuído após um intervalo de tempo  $t$  qualquer? Justifique sua resposta e apresente um histograma (montado a partir da observação do desvio de *clock* de  $n$  dispositivos do seu sistema distribuído em um mesmo instante de tempo  $t$ ), indicando o valor mais provável para o desvio de *clock* de um dispositivo qualquer e a maior diferença que deverá ser observada entre o desvio de *clock* de dois diferentes dispositivos. Assuma que  $n$  é um valor grande o suficiente e que todos os dispositivos do seu sistema distribuído são iguais.

# Gabarito

## Exercício 1

O NTP é um protocolo para sincronização de *clocks* de dispositivos computacionais baseado no protocolo UDP. É utilizado para sincronização do *clock* de um conjunto de computadores e dispositivos em rede com latência variável. O NTP permite manter o *clock* de um computador sincronizado com grande exatidão.

**Vantagens:** Simples implementação, rápida execução, boa acurácia e precisão.

**Desvantagens:** Em algumas situações onde os servidores estejam lentos, pode ser que a hora obtida não reflita exatamente o horário real. Desta forma, existe um custo adicional de computar o *delay* das mensagens de rede quando estamos atualizando o *clock* de um dispositivo utilizando o protocolo NTP. Também existem preocupações com segurança, pois é necessário enviar e receber mensagens de um servidor externo ao sistema distribuído.

## Exercício 2

- a) O algoritmo centralizado tem como principal característica a existência de um único processo coordenador que gerencia o acesso a todos os recursos compartilhados. A principal vantagem deste algoritmo é sua simples implementação e eficiência, pois poucas trocas de mensagens são necessárias e o *delay* para acesso a recursos compartilhados geralmente é pequeno. Entretanto, este algoritmo tem como principal desvantagem o fato de que o processo coordenador não pode nunca falhar. Este algoritmo deve ser utilizado quando existe um nó da rede dedicado para a tarefa de coordenação, sendo que este nó tem fortes mecanismos de prevenção e recuperação a falhas.
- b) O algoritmo distribuído baseia-se na ideia de *clocks* lógicos, sendo que o acesso a recursos compartilhados é realizado de acordo com a ordem lógica dos pedidos de acesso. Entre as vantagens deste método, podemos apontar o reaproveitamento do protocolo de *clocks* lógicos para realizar, também, o gerenciamento de acesso a recursos compartilhados. Entretanto, ele possui diversas desvantagens, como o elevado número necessário de troca de mensagens e o grande número de pontos de falhas (pois se qualquer dispositivo/aplicação do sistema distribuído falhar, o protocolo deixa de funcionar). Este algoritmo pode ser aplicado em um sistema distribuído de pequena escala que já utilize algoritmos de sincronização de *clock* lógicos e deseja-se implementar o gerenciamento de acesso a recursos compartilhados com pouco esforço de desenvolvimento.
- c) O algoritmo *token-ring* constrói uma estrutura de rede virtual conectando todos os processos/aplicações/dispositivos do sistema distribuído na forma de um ciclo. Para cada recurso compartilhado, ele cria um *token* e faz este *token* circular na rede virtual, sendo que quem tem a posse do *token* pode realizar o acesso ao recurso. As vantagens deste algoritmo são sua simples implementação e a desnecessidade de termos um processo coordenador, além do fato de que o gerenciamento e acesso a diferentes recursos compartilhados é realizado separadamente (cada um com seu *token*). Já em relação as desvantagens, podemos citar a perda de *tokens* na rede: quanto ocorre uma falha em uma aplicação que está em posse de um *token*, é difícil para o *middleware* e o sistema distribuído descobrirem que alguém caiu e o *token* está perdido. Este algoritmo de exclusão mútua é indicado para sistemas distribuídos onde não há a possibilidade de criarmos um processo coordenador e não é necessário centralizar as ações de coordenação de acesso em um único dispositivo.

## Exercício 3

Primeiro, vamos apresentar aqui os três conceitos.

- *Deadlock*: O processo coordenador consegue interpretar quando ocorre um *deadlock* observando as mensagens de requisição de acesso a recursos compartilhados. A partir daí, ele consegue tratar da maneira que achar mais eficiente
- *Starvation*: Sempre que um processo realiza uma requisição, ele é enfileirado para que possa acessar o recurso compartilhado. *Starvation* nunca ocorre pois a fila sempre anda.

- Igualdade e justiça: Todos os processos tem a mesma chance de acessar o recurso compartilhado, basta que eles enviem uma requisição. Além disso, a justiça vem do fato de que, quem pediu primeiro, consegue acessar primeiro.

Agora, conseguimos dizer como cada algoritmo implementa estes conceitos:

#### **Algoritmo distribuído**

- *Deadlock*: *Deadlocks* não ocorrem devido a utilização dos tempos de *clock* nas requisições para decidir quais processos tem acesso a quais recursos compartilhados. Requisições com valores de *clock* mais baixos tem prioridade sobre as de valores mais altos.
- *Starvation*: Sempre que um processo requer acesso a um recurso compartilhado, o tempo de seu *clock* é armazenado. Esta requisição é inserida na fila de todos os outros processos interessados neste recurso compartilhado e, quando o recurso for liberado, o primeiro processo é notificado. Assim, garante-se que não exista *starvation*
- Igualdade e justiça: Todos os processos tem a mesma chance de acessar o recurso compartilhado, basta que eles enviem uma requisição. Além disso, a justiça vem do fato de que, quem pediu primeiro, consegue acessar primeiro.

#### **Algoritmo *token-ring***

- *Deadlock*: É garantido pois somente um processo faz acesso a cada recurso compartilhado de uma vez, devido ao sistema de *tokens*
- *Starvation*: Os *tokens* circulam entre todos os processos do sistema distribuído. Desta forma, em algum momento, um processo que queira acessar um recurso compartilhado (simbolizado por um *token*), vai tomar posse deste *token* e poderá realizar o acesso.
- Igualdade e justiça: Os *tokens* circulam por toda a rede e passam por todos os processos com a mesma prioridade.

### **Exercício 4**

- a) Acurácia refere-se a diferença entre o *clock* de quaisquer dois dispositivos pertencentes a um mesmo sistema distribuído. Normalmente, deseja-se uma acurácia mínima  $\alpha$
- b) Precisão refere-se a diferença entre o *clock* de um dispositivo de um sistema distribuído e um *clock* de acreditação global, como o UTC. Normalmente, deseja-se uma precisão mínima  $\pi$

### **Exercício 5**

**Algoritmo de bully**: Neste algoritmo, quando algum processo percebe que o coordenador falhou, todos os processos (ativos e inativos) recebem um identificador numérico e é dado início as eleições. O processo de eleições consiste na troca de mensagens entre os diversos processos do sistema distribuído. Estas mensagens são sempre enviadas dos processos com menor valor de identificador para aqueles com maior valor. Assim, temos duas opções:

1. Se um processo recebe uma mensagem e ele está ativo, então ele responde com um *OK* e retira o emissor da mensagem das eleições.
2. Se o processo está inativo e não consegue responder a mensagem, nada ocorre.

Este processo ocorre de forma iterativa até que um processo de identificador alto não receba mensagens de nenhum outro com valor de identificador acima dele. Logo após, ele envia uma mensagem para todos os outros processos do sistema informando que ele é o novo processo coordenador. **Vantagens**: Um único

coordenador é eleito em todas as situações. **Desvantagens:** É necessário realizar um grande número de troca de mensagens. Pode ser um pouco lento devido a demora na resposta de processos inativos

**Algoritmo do anel (token-ring):** Neste algoritmo, assume-se que os processos são organizados na forma de um ciclo (ou anel) e que cada processo conhece quem é seu sucessor. Quando é identificado que processo coordenador falhou, faz-se necessário eleger um novo coordenador. A eleição é inicializado pelo(s) processo(s) que perceber(am) que o coordenador falhou. A eleição consiste em circular uma mensagem na rede duas vezes: a primeira para verificar quais processos estão ativos no momento e a segunda para informar a todos os processos qual é o novo coordenador, que será aquele que iniciou a eleição. **Vantagens:** Simples implementação. **Desvantagens:** Um número muito grande de mensagens deve ser trocada, o que pode sobrecarregar a rede; Não funciona para sistemas distribuídos muito grandes; Caso dois ou mais processos iniciem as eleições, é necessário um mecanismo adicional para decidir qual deles será o coordenador.

## Exercício 6

Levando em consideração que a frequência não seja totalmente aleatória, mas sim que ocorra uma tendência na diminuição das oscilações deste cristal de quartzo, pode-se afirmar que os *clocks* dos computadores tendem a representar um horário **atrasado** em relação ao horário real.

Podemos representar estes desvios através de um histograma, onde os desvios de *clock* são representados como uma curva gaussiana (distribuição normal) e a maior diferença entre dois desvios quaisquer seria igual a  $2\rho\Delta t$ , onde  $\Delta t$  representa o intervalo de tempo decorrido desde a sincronização anterior dos clocks. Entretanto, a média desta curva está deslocada a esquerda, de forma a representar o atraso nos clocks, sendo assim uma distribuição normal enviesada positivamente. Note que o valor esperado é relativamente menor do que aquele que seria obtido caso a frequência ideal de oscilação ( $F$ ) fosse alcançada.

