

Atividade de Fixação

1. Associe as afirmações a seguir aos seguintes modelos de threads: many-to-one (N:1); one-to-one (1:1); many-to-many (N:M):
(N:1) Tem a implementação mais simples, leve e eficiente.
(N:M) Multiplexa os threads de usuário em um pool de threads de núcleo.
(1:1) Se o número de threads de usuário aumenta, impõe uma carga muito pesada ao núcleo.
(N:1) Não permite explorar a presença de várias CPUs pelo mesmo processo.
(N:1) implementado por bibliotecas e não pelo sistema operacional.
(1:1) É o modelo implementado no Windows NT e seus sucessores.
(N:1) Se um thread bloquear, todos os demais têm de esperar por ele.
(1:1) Cada thread no nível do usuário tem sua correspondente dentro do núcleo.
(N:M) É o modelo com implementação mais complexa.

2. Em uma aplicação concorrente que controla saldo bancário em contas correntes, dois processos compartilham uma região de memória onde estão armazenados os saldos dos clientes A e B. Os processos executam, concorrentemente os seguintes passos, ou seja, o Processo 1 (cliente A) realiza um saque de 200 da conta de A e um depósito de 100 na conta de B, e o Processo 2 (cliente B) realiza um saque de 100 da conta de A e um depósito de 200 na conta de B.

Processo 1 (Cliente A)

Processo 2 (Cliente B)

```
/* saque em A */
```

```
1a. x = saldo_do_cliente_A;
```

```
1b. x = x - 200;
```

```
1c. saldo_do_cliente_A = x;
```

```
/*saque em A */
```

```
2a. y = saldo_do_cliente_A;
```

```
2b. y = y - 100;
```

```
2c. saldo_do_cliente_A = y;
```

```
/* deposito em B */
```

```
1d. x = saldo_do_cliente_B;
```

```
1e. x = x + 100;
```

```
1f. saldo_do_cliente_B = x;
```

```
/* deposito em B */
```

```
2d. y = saldo_do_cliente_B;
```

```
2e. y = y + 200;
```

```
2f. saldo_do_cliente_B = y;
```

Supondo que os valores dos saldos de A e B sejam, respectivamente, 500 e 900, antes de os processos executarem, pede-se:

- a) Quais os valores corretos esperados para os saldos dos clientes A e B após o término da execução dos processos?

200 e 1200, respectivamente.

- b) Quais os valores finais dos saldos dos clientes se a sequência temporal de execução das operações for: 1a, 2a, 1b, 2b, 1c, 2c, 1d, 2d, 1e, 2e, 1f, 2f?

Saldo cliente A = 400 e saldo cliente B = 1100

- c) Proponha uma solução que garanta a integridade dos saldos e permita o maior compartilhamento possível dos recursos entre os processos.

Solução de exclusão mutua primitiva Sleep/Wakeup.

3. Responda:

- (a) O que é condição de corrida ou de disputa? Dê um exemplo.

Inconsistências geradas por acessos concorrentes a dados compartilhados (pelos processos que competem pelos mesmos recursos de forma concorrente)

Exemplo: arquivos ou memória compartilhada.

- (b) O que é região crítica? Dê um exemplo.

Trecho do programa que acessa e altera o valor do recursos compartilhado, por exemplo quando dois processos diferentes querem usar a memória, o momento em que o primeiro está gravando dados, esse trecho torna-se uma região crítica, e o segundo processo é bloqueado de usá-la até que o primeiro processo seja finalizado.

4. Com relação a deadlock, responda:

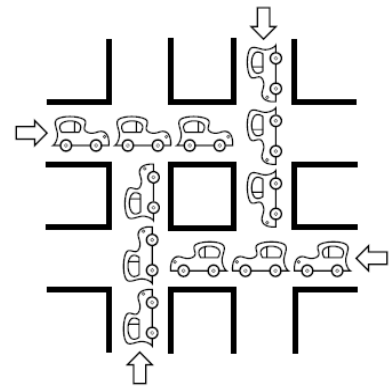
- (a) O que é deadlock?

Deadlock é a situação em que processos ficam parados sem possibilidade de continuar seu processamento.

- (b) Quais as condições para que ele ocorra?

Quando os processos desejam alocar recursos que não estão livres e um processo depende do outro liberar os recursos para poder usar.

- (c) A situação ao lado representa um deadlock no trânsito. Nessa situação de trânsito, um processo poderia ser comparado ao fluxo de carros em uma rua, e os recursos aos cruzamentos. Identifique nessa situação a ocorrência das quatro condições necessárias para que haja um deadlock.



- Exclusão mutua: um recurso ou está sendo utilizado ou está disponível;

- Posse e espera: processos podem solicitar acesso a outros recursos sem ter de liberar os recursos que já detém;

- Não-preempção: recursos já alocados não podem ser retirados dos processos que os alocou, somente o processo que alocou o recursos pode liberá-lo;

- Espera circular: existe um ciclo de espera pela liberação de recursos entre os processos envolvidos.

- (d) Proponha uma solução ou uma forma de evitar essa situação, explicando suas razões.

- Algoritmo do banqueiro: evitar dinamicamente o problema, alocando cuidadosamente os recursos.

Escolhi essa estratégia porque acredito que seja eficiente na prevenção desse problema, uma vez que os recursos serão alocados de forma cuidadosa para cada processo.