SISTEMAS OPERACIONAIS – 2018.2

Engenharia da Computação

PROF. FERNANDO PARENTE GARCIA

EXERCÍCIO DE SEMÁFOROS

ALUNO: Francisco Luciano Lima da Silva

Problema do Bar Pequeno: Suponha um bar com três cadeiras. Se você chega ao bar e existe alguma cadeira vazia, você pode sentar-se imediatamente. Mas se você chega e todas as três cadeiras estão ocupadas, significa que todos estes clientes estão jantando juntos e você terá que esperar (bloqueado) que eles desocupem todas as cadeiras para só então se sentar. Os clientes que estão aguardando podem ser selecionados aleatoriamente, ou seja, não há nenhum mecanismo de fila. Existe algum erro nos algoritmos? Se existir, explique detalhadamente os problemas ocasionados por este(s) erro(s) e em seguida faça as devidas correções.

```
INT N;                          INT EATING =0;          INT WAITING = 0;        INT MUST_WAIT = 0;
SEMAPHORE MUTEX = 1;            SEMAPHORE BLOCK = 3; 0
```

THREAD CLIENTE:

```
DOWN(MUTEX)
IF (MUST_WAIT == 1)  {
        WAITING ++;
        UP(MUTEX);
        DOWN(BLOCK);
}
ELSE {
        EATING ++;
        IF (EATING == 3) MUST_WAIT = 1 ELSE MUST_WAIT = 0;
        UP(MUTEX);
}
COME_E_CONVERSA_NO_BAR();
DOWN(MUTEX);
EATING--;
IF (EATING == 0) {
        IF (WAITING >= 3) N = 3 ELSE N = WAITING;
        WAITING = WAITING - N;
        EATING = EATING + N;
        IF (EATING == 3) MUST_WAIT = 1 ELSE MUST_WAIT = 0;
        UP(BLOCK); up(block); up(block); up(block, N);
}
UP(MUTEX);
```

1) O erro está em se instanciar o semáforo block = 3, pois quando todas as cadeiras estiverem ocupadas e se chegar um novo cliente, ele não será bloqueado e prosseguirá para a função come_e_conversa(). Para corrigir, o semáforo block deve ser instanciado para 0.

2) O erro está na linha up(block). Ao executar a linha, sem block vai para 1 e acordará todas as threads dormidas pelo semáforo, mas ao menos uma das threads acordadas, ela executará o down novamente, pondo o sem. block para 0, logo as outras que iniciaram não ficarão bloqueadas. Para corrigir, deve-se adicionar mais dois up(block), para que sejam liberadas mais threads.

INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA CEARÁ

ENGENHARIA DA COMPUTAÇÃO
SISTEMAS OPERACIONAIS
SEMESTRE 2018.2 – Prova 1ª Etapa
PROF. FERNANDO PARENTE GARCIA

ALUNO: Francisco Lucas Lima da Silva

2,0

1) (2,0 Pontos) Sobre conceitos básicos de sistemas operacionais, responda:
   a) Cite três tipos de sistemas operacionais e cite um exemplo de cada um deles.
   b) O que é *system call* e qual sua importância para a segurança do sistema?

4,4 / 7,0

2,6

2) (2,0 Pontos) Sobre processos e threads, responda:
   a) Defina contexto de software e contexto de hardware. Qual deles pode ser compartilhado pelos threads de um mesmo processo?
   b) Que problema ocorre se for definido um *timeslice* muito grande para os processos? Justifique.

0,8

3) (2,0 Pontos) Suponha que os seguintes processos chegaram para execução nos tempos indicados. Cada processo rodará a quantidade de tempo listada na tabela.

Processo	Tempo de Chegada (ms)	Tempo de Execução (ms)	Prioridade
A	5	20	1
B	10	12	2
C	12	10	2
D	14	5	3
E	30	13	1

Qual o tempo médio de espera para estes processos quando são utilizados os algoritmos de escalonamento abaixo. Considere que o sistema operacional gasta 1 ms para realizar a troca de contexto.
   a) PMTR (Próximo de Menor Tempo Restante).
   b) Escalonamento circular com prioridade estática e *timeslice* de 4 ms.

X

4) (2,0 Pontos) Um sistema tem quatro processos e cinco recursos alocáveis. A alocação atual e as necessidades máximas de cada processo são mostradas na tabela abaixo. O vetor de recursos existentes é (5, 2, 6, **X**, 2). Qual o menor valor de **X** para que esse estado seja seguro?

Processo	Alocado					Máximo				
A	1	0	2	1	1	1	1	2	2	2
B	2	0	1	2	0	2	2	2	3	0
C	1	1	0	1	0	2	1	3	2	0
D	1	1	1	2	0	1	1	2	4	1

2,0

5) (2,0 Pontos) *Problema da Montanha Russa*: Suponha que há **n** processos *passageiros* e um processo *vagão*. Os passageiros repetidamente esperam para andar no vagão, que comporta v passageiros, onde v<n. O vagão só pode percorrer a montanha russa quando está cheio. **Os passageiros devem dormir durante a viagem do vagão**. Verifique se os algoritmos propostos abaixo estão corretos, e caso contrário faça as devidas correções.

```
SEMAPHORE FULL = 0;
SEMAPHORE FIM_VIAGEM = 1;
SEMAPHORE EMPTY = V;      SEMAPHORE MUTEX = 1;
INT PASSAGEIROS = 0;
```

PROCESSO PASSAGEIRO:

```
WHILE (TRUE) {
        DOWN(EMPTY);
        DOWN(MUTEX);
        ENTRA_VAGAO();
        PASSAGEIROS++;
        SENTAR_VAGAO();
        IF (PASSAGEIROS == V) UP(FULL);
        UP(MUTEX);
        DOWN(FIM_VIAGEM);
        SAIR_DO_VAGAO();
        DOWN(MUTEX);
        PASSAGEIROS--;
        IF (PASSAGEIROS == 0)
          FOR (I=1;I<V;I++) UP(EMPTY);
        UP(MUTEX);
}
```

PROCESSO VAGÃO:

```
WHILE (TRUE) {
        DOWN(FULL);
        PERCORRER_MONTANHA();
        UP(FIM_VIAGEM);
}
```

BOA PROVA!

a) 1. SO para computadores pessoais. Ex: Windows, [illegible]
2. SO para sistemas embarcados. Ex: Android, IOS
3. SO para sistemas de servidores. Ex: Windows Server

1,0

b) System calls são instruções do sistema chamadas pelo SO que fazem a conexão entre a aplicação e o hardware. É importante para a segurança pois o usuário não precisaria usá-los para sua aplicação, já que o próprio SO faz a chamada.

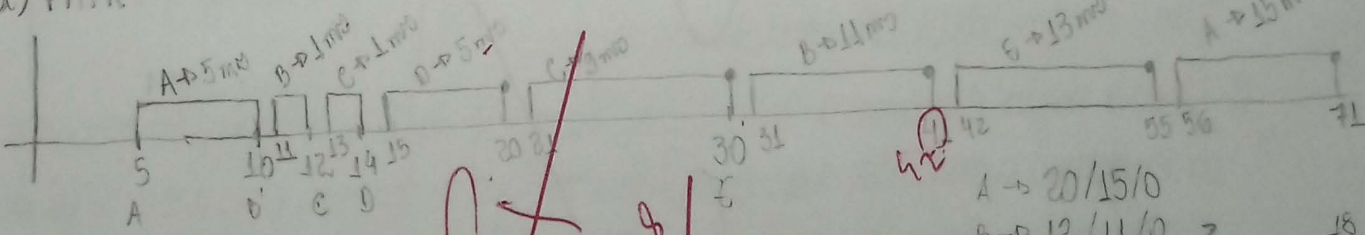
1,0

02) a) Contexto de hardware: estrutura que guarda informações dos registradores gerais de uma CPU e registradores específicos como PC, SP e registrador de status. É lá onde ocorre a troca de contexto entre processos.
Contexto de software: estrutura que armazena as propriedades de um processo, como identificador e endereço de memória armazenado.
~~Threads de um mesmo processo podem compartilhar contexto de hardware.~~

0,6

~~b) Se for definido um timeslice muito grande pode prejudicar o desempenho da máquina, pois o tempo de resposta ao usuário será grande e afetará outros processos que irão demorar a executar.~~

03)
a) PMTR



A → 20/15/0
B → 12/11/0
C → 10/9/0
D → 5/0
E → 13/0

$$M = \frac{66 + 31 + 18 + 6 + 25}{5} = 29,2 \text{ ms}$$

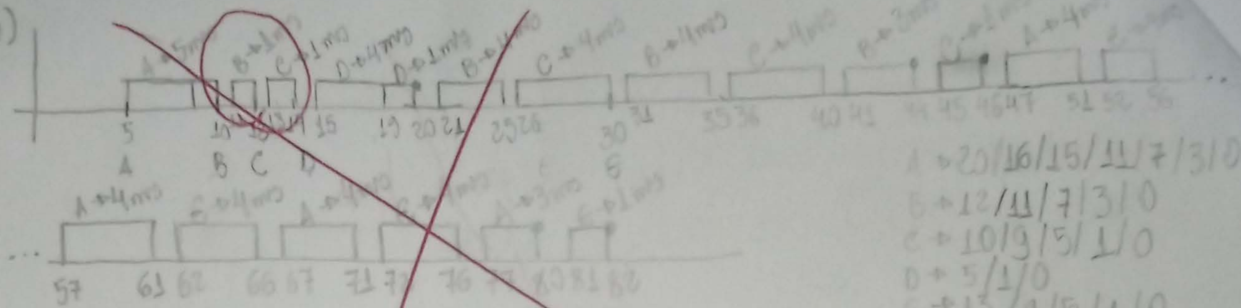
0,8

66
+ 31
+ 18
+ 6
+ 25
146 | 5
46 29,2
10

18
+ 8

Teo Lucas.

b)



A → 20/16/15/11/7/3/0
B → 12/11/7/3/0
C → 10/9/5/1/0
D → 5/1/0
E → 13/9/5/1/0

$M = \frac{75 + 34 + 34 + 6 + 52}{5} = 40,2ms$

04)

$$R = \begin{bmatrix} 0 & 1 & 0 & 1 & 1 \\ 0 & 2 & 1 & 1 & 0 \\ 1 & 0 & 3 & 1 & 0 \\ 0 & 0 & 1 & 2 & 1 \end{bmatrix}$$

A = (0 0 2 x-6 1)
D+A = (1 1 3 x-4 1)
B+A = (3 1 4 x-2 1)
A+A = (4 1 6 x-1 1)
C+A = (5 2 6 x 1)

R - X deverá ser maior que 6 para que o estado seja seguro.

QUAL O MENOR VALOR?

05)

1. O erro está no semáforo fim_viagem = 1, pois quando um processo dá um DOWN, ele imediatamente sai do vagão e não dorme na viagem. Para corrigir, o semáforo fim_viagem deve ser iniciado em 0.

2. Após o vagão percorrer a montanha, dando UP no fim_viagem, os processos passageiros acordarão, mas um deles voltará a dormir, pois a outra já deu DOWN no semáforo. Para corrigir, deve-se dar UP na quantidade de passageiros que embarcaram, ou seja, deve-se colocar UP(FIM_VIAGEM, V).

2,0