

**UNIVERSIDADE DE SÃO PAULO**  
**SEGUNDO SEMESTRE LETIVO DE 2014**  
**RECUPERAÇÃO**

Escola	EACH		TURMA		Nota do aluno na PROVA
Curso	Sistemas de Informação				
Disciplina	Sistemas Operacionais - ACH2044	Data da Prova	12/02/15		
Professor	Clodoaldo Aparecido de Moraes Lima				
Aluno					
No. USP					

<b>QUESTÃO 01</b>	<b>Valor da Questão:</b>	2,5
-------------------	--------------------------	-----

Cinco processos em batch, A a E, chegam em um centro de computação quase que ao mesmo tempo. Eles têm tempos de execução estimados de 12, 10, 4, 8 e 6. Suas prioridades, definidas externamente, são 2, 3, 1, 5 e 4, com 5 sendo a mais alta. Para cada um dos seguintes algoritmos, determine o tempo médio de execução completa (mean turnaround time) desses processos. Ignore o tempo gasto com a troca de processos.

- (a) Round Robin
- (b) Prioridade
- (c) First-come, First-served (na ordem 6, 12, 8, 4, 10)
- (d) Shortest Remaining Time Next

Pede-se

- a) (0.5) Para i), assuma que o sistema aceita multiprogramação, e que cada processo recebe uma fatia de 3 da CPU. Para (ii) a (iv) assuma que somente um processo pode rodar por vez, rodando até o fim. Todos os processos são CPU bound (sem E/S)..

(a) Round Robin

E										
D										X
C							X	X	X	
B				X	X	X				
A	X	X	X							
0	1	2	3	4	5	6	7	8	9	10

E			X	X	X					
D	X	X								
C										
B									X	X
A						X	X	X		
10	11	12	13	14	15	16	17	18	19	20

E						X	X	X		
D			X	X	X					
C		X								
B	X									
A									X	X
20	21	22	23	24	25	26	27	28	29	30

E										
D					X	X				
C										
B		X	X	X						
A	X						X	X	X	
30	31	32	33	34	35	36	37	38	39	40

$$T_m = (39+40+22+36+28)/5$$

Prioridade

E										
D										
C										
B	X									
A										
40	41	22	23	24	25	26	27	28	29	30

- b) (2,0) Para i), assuma que o sistema aceita multiprogramação, e que cada processo recebe uma fatia de 3 da CPU. Para (ii) a (iv) assuma que somente um processo pode rodar por vez, rodando até o fim. Para todos os itens assuma que somente o processo B realiza E/S. Assuma que somente o processo B tenha um surto de CPU de 2 e que a E/S tenha duração de 4.

<b>QUESTÃO 02</b>	<b>Valor da Questão:</b>	1,0
-------------------	--------------------------	-----

Os pedidos para acesso a um disco chegam em seu driver para os cilindros 10, 22, 20, 2, 40, 6, 38 nesta ordem. O disco demora 4 ms para movimentar o braço de leitura/escrita entre dois cilindros consecutivos. Considerando que o braço está inicialmente no cilindro 16 calcule o tempo gasto em seek para cada um dos algoritmos de escalonamento do braço do disco:

- a) (0,3 ponto) FCFS (Primeiro-a-Chegar-Primeiro-a-Ser-Servido);

Fila [16 10 22 20 2 40 6 38] 0,1

tempo = ((16-10) + (10-22) + (22-20) + (20-2) + (2-40) + (40-6) + (6-38)) \* 4ms 0,1

tempo = (6 + 12 + 2 + 18 + 38 + 34 + 32) \* 4 = 568 ms 0,1

- b) (0,3 ponto) SSF (Menor Seek Primeiro);

Fila [16 20 22 10 6 2 38 40] 0,1

tempo = ((16-20) + (20-22) + (22-10) + (10-6) + (6-2) + (2-38) + (38-40)) \* 4ms 0,1

tempo = (4 + 2 + 12 + 4 + 4 + 36 + 2) \* 4 = 256 ms 0,1

cada erro 0,05

- c) (0,4 ponto) O algoritmo do elevador, suponha que braço está se movendo inicialmente para cima, ou seja, dos cilindros de número menor para os de número maior.

Fila [16 20 22 38 40 10 6 2] 0,1

tempo = ((16-20) + (20-22) + (22-38) + (38-40) + (40-10) + (10-6) + (6-2)) \* 4ms 0,1

tempo = (4 + 2 + 16 + 2 + 30 + 4 + 4) \* 4 0,1

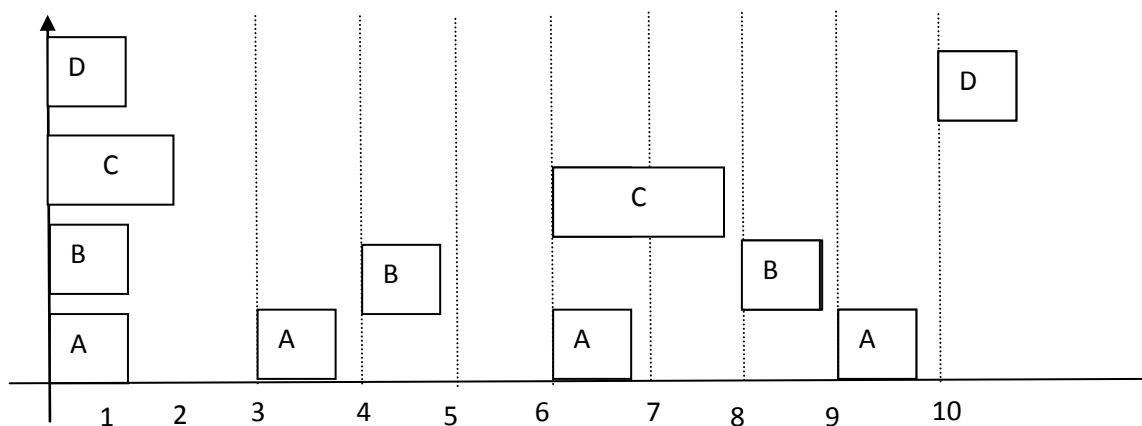
tempo = 248 ms 0,1

erro calculo 0,05

**QUESTÃO 03** Valor da Questão: 1,0

Um sistema de tempo real tem quatro eventos periódicos com períodos de 3, 4, 6 e 10 ms cada. Suponha que os quatro eventos requeiram 1, 1, 2 e 1 ms de tempo de CPU, respectivamente. Ilustre o escalonamento dos processos segundo (durante 20 ms)

- a) (1,0) Rate Monotonic Scheduling  
b) (1,0) Earliest Deadline First



a)

D											D										D
C							C					C							C		
B					B				B			B				B				B	
A				A			A			A		A			A			A			
	A	B	C	A	B	C	A	C	B	A	C	A	B	C	A	B	C	A	C	B	
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21

Cada item errado -0.075

Não parou em 10 - 0,30

b)

D											D										D
C							C					C							C		
B					B				B			B				B				B	
A				A			A			A		A			A			A			
	A	B	C	A	C	B	A	D	B	A	C	C	A	B	C	A	C	B	D	C	
	A	B	C	C	A	B	A	D	B	A	C	C	A	B	C	C	A	B	D	A	
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21

Cada item 0.0375

**QUESTÃO 04** Valor da Questão: 1,5

Considere um arquivo de nome prova.pdf com tamanho  $y$  armazenado em uma partição de  $z$  Mbytes, cujo endereçamento é de 16bits com blocos de 4K.

- a) (0,2 ponto) Qual o tamanho máximo da partição em Mbytes?

**Tamanho da Partição =  $2^{16} \cdot 4K / (1024)^2 = 256$  Mbytes**

- b) (0,5 ponto) Suponha que a FAT que mapeia este arquivo possui os seguintes valores:  $x, x, 8, 7, -1, -1, 3, 2, 5, 0, 0$ . Sendo  $x$  não relevante para os nossos propósitos, o 0 indica uma posição livre e  $-1$  indica fim de arquivo. Se a entrada do diretório para este arquivo tem 6 como o bloco inicial, quantos blocos possui este arquivo e qual o tamanho deste arquivo?

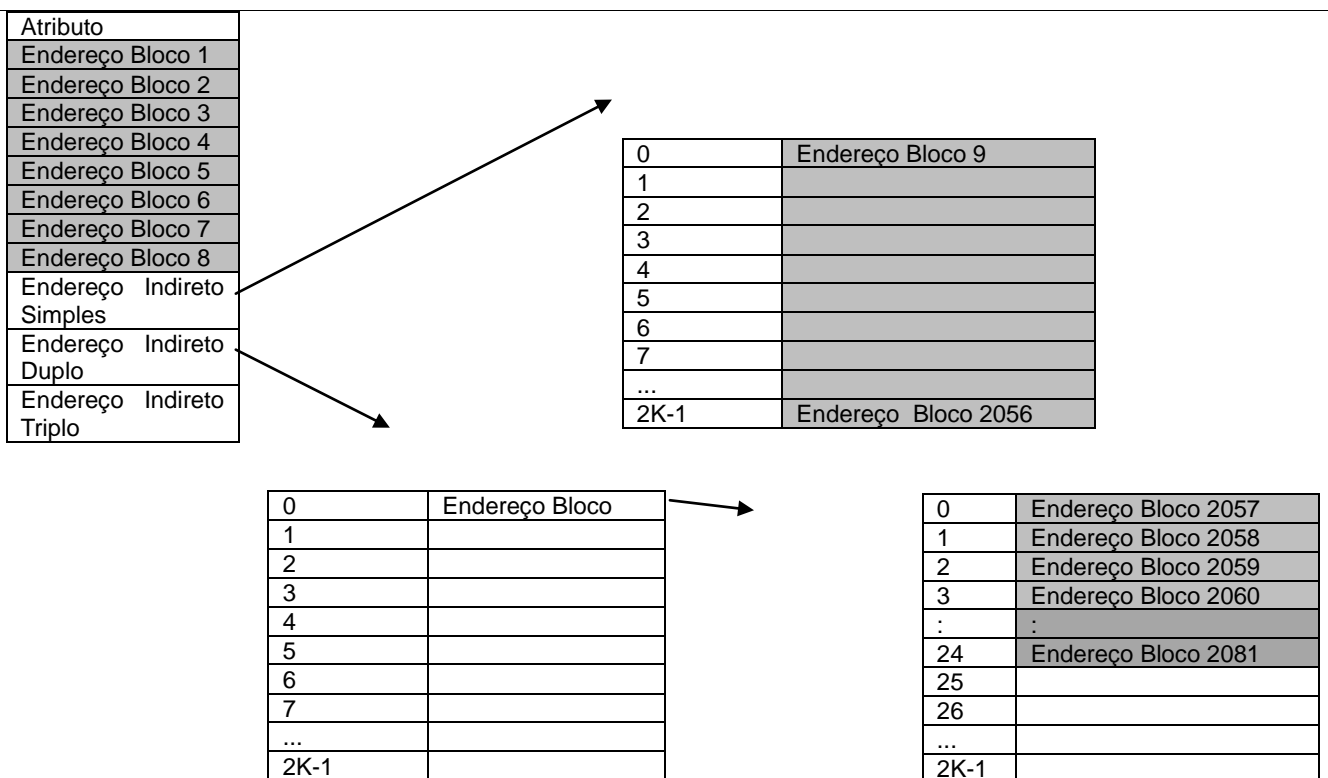
0	X
1	x
2	8
3	7
4	-1
5	-1
6	3
7	2
8	5
9	0
10	0
...	...

**Tamanho do arquivo 0,3**

**6-> 3-> 7 --> 2--->8--->5, logo o tamanho máximo do arquivo é  $6 \times 4 \text{Kbyte} = 24 \text{Kbyte}$**

**Faltou bloco 6, -0.15**

c) (0,4 ponto) Considere que o sistema de arquivo utilizado seja baseado em nó-i (i-node), o qual possui 8 endereços de disco para blocos de dados, 1 endereço de bloco para endereçamento indireto simples, 1 para endereçamento indireto duplo e 1 para endereçamento indireto triplo. Ilustre o mapeamento dos blocos deste arquivo nesta partição (faça um desenho ilustrando), considerando que o arquivo possuía 8322 Kbytes.



**Endereço Indireto Simples = 4 KByte / 2Byte = 2K endereço**

**Numero de Blocos = 8322 K / 4 K = 2081 (0,1)**

Desenho ilustrando 0.1

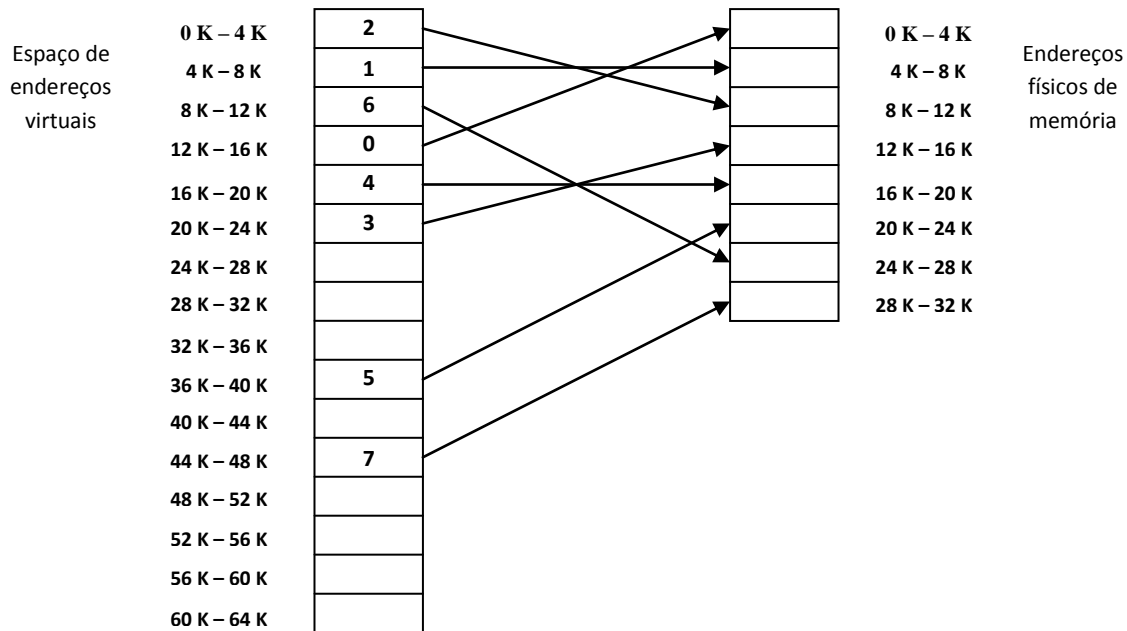
d) (0,4 ponto) Para o item b), qual é o tamanho do maior arquivo que pode ser manipulado? Mostre todos os cálculos.

**Tamanho máximo = (8 + 2K + 2K \* 2K + 2K\*2K\*2K) \* 4 Kbyte**

**Esquecer de multiplicar por 4 Kbyte**

<b>QUESTÃO 05</b>	<b>Valor da Questão:</b>	1,0
-------------------	--------------------------	-----

Usando a tabela de páginas abaixo, pede-se



a) (0,2 ponto) Quantos bits são necessários para endereçar todos os endereços virtuais e reais?

**Tamanho da Pagina = 4 K, logo 12 bits**

**Número de páginas = 16, logo 4 bits**

**Número de frames = 8, logo 3 bits**

**Endereçamento virtual = 4 bits + 12 bits = 16 bits 0,1**

**Endereçamento real = 3 bits + 12 bits = 15 bits 0,1**

b) (0,3 ponto) Quantos bits são usados para **número de páginas, frames** e o **deslocamento**?

**Número de páginas = 16, 4 bits 0,1**

**Número de frames = 8, 3 bits 0,1**

**Deslocamento = 12 bits 0,1**

c) (0,6 ponto) Apresente o endereço físico (em binário) correspondente a cada um dos seguintes endereços virtuais: 250; 12300; 37200, 1011 1111 1011 0000

250 --> **0000** 0000 1111 1010 será mapeado para **010** 0000 1111 1010

12300 --> **0010** 1111 0100 0100 será mapeado para **110** 1111 0100 0100

37200 --> **1001** 0001 0101 0000 --> será mapeado para **101** 0001 0101 0000

**1011** 1101 1011 0000 --> **não esta mapeado em memória**

c) (0,4 ponto) Supondo uma tabela de paginas de 2 níveis, os endereços são quebrados em um campo de x bits para a tabela de paginas de nível 1, um campo de y bits para a tabela de pagina de nível 2 com 8 entradas. Considerando um processo de 18 Kbytes, quantas tabelas, no mínimo, deverão estar em memória física.

**Solução**

Tabela Nível 2 - 8 entradas --> 3 bits 0.1

Tabela Nível 1 - ---> 1 bits 0.1

Cada tabela Nível 2 mapeia =  $5 \times 4 \text{ K byte} = 20 \text{ K Byte}$  0.1

Logo duas tabelas, tabela de nível 1 e uma tabela de nível 2 0.1

QUESTÃO 06	Valor da Questão:	1.5
------------	-------------------	-----

Considere o problema dos leitores/escritores, onde existem processos que lêem o valor das variáveis compartilhadas, chamados leitores e processos que escrevem na região compartilhada, chamados escritores. Os leitores podem ler de modo concorrente, enquanto os escritores só podem executar em exclusão mútua. O seguinte código resolve o problema dos leitores/escritores, desde que sejam colocados adequadamente os semáforos. Nesta implementação, foi feita a suposição de que, enquanto houver um leitor acessando a base de dados, o escritor é suspenso. Havendo pelo menos um leitor ativo, leitores subsequentes serão admitidos e o escritor permanecerá suspenso até que nenhum leitor esteja presente.

```
typedef int semaphore;
semaphore mutex = 1; /* Controla o acesso a rc
semaphore db = 1;    /* Controla o acesso a base de dados
int rc = 0;          /*Número de processos lendo ou querendo ler
```

cada erro 0,2

```
void leitor (void)
{
    while (TRUE) {
        __a) down(&mutex)_____;
        rc++; /* Um leitor a mais agora
        if (rc ==1) __c) down(&db)_____;
        __b) up(&mutex)_____;
        read_data_base(); /* Acessa os dados
        a) down(&mutex)_____;
        rc--; /* Um leitor a menos agora
        if (rc == 0) __d) up(&db)_____;
        __b) up(&mutex)_____;
        use_data(); /*Região não critica
    }
}

void escritor (void)
{
    while (TRUE) {
        prepare_data(); /* Região não critica
        __c) down(&db)_____;
        write_data_base(); /* Atualiza os dados
        __d) up(&db)_____:
    }
}
```

Considerando a) down(&mutex), b) up(&mutex), c) down(&db), d) up(&db) preencha adequadamente os 8 espaços no programa acima com a opção adequada.

<b>QUESTÃO 07</b>	<b>Valor da Questão:</b>	1.5
-------------------	--------------------------	-----

Em um computador, o endereço virtual da memória virtual por paginação é de 16 bits e as páginas têm tamanho de 2 K endereços. O limite de páginas reais de um processo qualquer é de quatro páginas. Ilustre o mapeamento e calcule quantas interrupções por falta de página ocorrerão na seguinte sequência de referências à memória: 1,2,3,4,2,1,5,6,2,1,2,3,7,6,2,3,1 para os algoritmos de substituição de página. Considere que os 4 quadros (frames) disponíveis para o processo estão inicialmente vazios.

a) Ótimo

Página virtual	Páginas na memória	Page fault	Página a ser substituída
2	-----	S	-----
1	2	S	-----
3	1,2	S	-----
4	1,2,3	S	-----
2	1,2,3,4	N	-----
1	1,2,3,4	N	-----
5	1,2,3,4	S	4
6	1,2,3,5	S	5
2	1,2,3,6	N	-----
1	1,2,3,6	N	-----
2	1,2,3,6	N	-----
3	1,2,3,6	N	-----
7	1,2,3,6	S	1
6	2,3,6,7	N	-----
2	2,3,6,7	N	-----
3	2,3,6,7	N	-----
1	2,3,6,7	S	2 ou 3, ou 6 ou 7
	1,3,6,7 ou 2,1,6,7 ou 2,3,1,7 ou 2,3,6,1		

Numero de faltas de paginas \_\_\_\_\_ 8 \_\_\_\_\_

Cada 0,125

Pagina adicional ser substituída -0,05

Pagina em memória -0,05

b) Segunda Chance

Página virtual	Páginas na memória	Page fault	Página a ser substituída
2	-----	S	-----
1	2 [1]	S	-----
3	2,1 [1,1]	S	-----
4	2,1, 3 [1,1,1]	S	-----
2	2,1,3,4 [1,1,1,1]	N	-----
1	2,1,3,4 [1,1,1,1]	N	-----
5	2,1,3,4 [1,1,1,1]	S	2
6	1,3,4, 5 [0,0,0,1]	S	1
2	3,4, 5,6 [0,0,1,1]	S	3
1	4, 5,6,2 [0,1,1,1]	S	4
2	5,6,2,1 [1,1,1,1]	N	-----
3	5,6,2,1 [1,1,1,1]	S	5
7	6,2,1,3 [0,0,0,1]	S	6
6	2,1,3,7 [0,0,1,1]	S	2



**UNIVERSIDADE DE SÃO PAULO**  
**SEGUNDO SEMESTRE LETIVO DE 2014**  
**RECUPERAÇÃO**

2	1,3,7, 6 [0,1,1,1]	S	1
3	3,7, 6, 1 [1,1,1,1]	N	-----
1	3,7, 6, 1 [1,1,1,1]	N	

Numero de faltas de paginas \_\_\_\_\_ 12 \_\_\_\_\_

Cada erro 0,05

Numero de pagina 0,05

c) LRU

Página virtual	Páginas na memória	Page fault	Página a ser substituída
2	-----	S	-----
1	2	S	-----
3	2,1	S	-----
4	2, 1, 3	S	-----
2	2, 1, 3, 4	N	-----
1	1, 3, 4, 2	N	-----
5	3, 4, 2, 1	S	3
6	4, 2, 1, 5	S	4
2	2, 1, 5 6	N	-----
1	1, 5, 6, 2	N	-----
2	5, 6, 2 1	N	-----
3	5, 6, 1, 2	S	5
7	6, 1, 2, 3	S	6
6	1, 2, 3, 7	S	1
2	2, 3, 7, 6	N	-----
3	3, 7, 6, 2	N	-----
1	7, 6, 2, 3	S	7
	6, 2, 3, 1		

Numero de faltas de paginas \_\_\_\_\_ 10 \_\_\_\_\_