

3. Assuma que existem 3 tarefas (do mesmo processo) a executar-se num determinado sistema operativo com escalonamento preemptivo, prioridades dinâmicas e quantum de 20ms (correspondendo a 4 quadrículas na tabela abaixo).

Por simplificação, assuma que a comutação de tarefas é instantânea.

Observou-se o estado de cada processo durante algum tempo, ilustrado na tabela seguinte.

Legenda: E – em execução, W – executável, B – bloqueada, T – terminada.

	0ms	5ms	10ms	15ms	20ms	25ms	30ms	35ms	40ms	45ms	50ms	55ms	60ms	65ms	70ms	75ms	80ms	85ms	90ms	95ms	100ms
T1	W	E	E	E	E	W	W	W	W	E	E	E	E	W	E	W	E	E	E	E	W
T2	E	B	B	B	B	B	B	E	E	B	B	B	B	B	B	E	T	T	T	T	T
T3	W	W	W	W	W	E	E	W	W	W	W	W	W	E	B	B	W	W	W	W	E

- a. [0,5v] Como classifica a tarefa T1, E/S-intensiva ou CPU-intensiva? E a tarefa T2? Justifique.

- b. [0,5v] Indique um momento em que ocorreu preempção. Justifique.

- c. [0,5v] Uma das tarefas acima chamou a função pthread_join durante a execução apresentada. Qual foi essa tarefa, em que momento foi feita a chamada à função?

- d. [0,5v] No instante 5ms, a tarefa T2 chamou pthread_mutex_lock sobre um trinco lógico trancado e bloqueou-se. Qual a tarefa que estava nesse momento na posse do trinco? Justifique.

Grupo II [4v]

Considere um serviço municipal que disponibiliza bicicletas gratuitamente aos munícipes. As bicicletas existem num estacionamento do qual podem ser retiradas, uma por cada pessoa, e devolvidas. As bicicletas estão organizadas numa fila sendo retiradas e devolvidas sempre na mesma extremidade da fila (similar aos carrinhos de compras ns supermercados). Quando uma pessoa pretende retirar uma bicicleta, pode fazê-lo caso exista pelo menos uma; caso não existam bicicletas, a pessoa deve aguardar que seja devolvida pelo menos uma. Havendo mais que uma pessoa à espera de bicicletas, deve ser respeitada a ordem de chegada das pessoas.

Considere o pseudo-código dos procedimentos que se apresentam de seguida, que implementam o cenário acima descrito:

<pre>#define MAX 100 char bicicletas[MAX]; int index = MAX-1;</pre>	
<pre>bicicleta_t* retiraBicicleta() { bicicleta_t b; b = bicicletas[index]; index--; return b; }</pre>	<pre>devolveBicicleta(bicicleta_t *b) { index++; bicicletas[index]=b; }</pre>

1. [1v] Descreva a situação errónea que pode suceder se duas pessoas tentarem retirar uma bicicleta ao mesmo tempo (havendo pelo menos 2 bicicletas disponíveis).

2. [1v] Como poderia evitar a situação descrita na sua resposta anterior? Diga que linhas de pseudo-código é que inseria e onde.

3. [1v] Altere o pseudo-código de modo que, caso não existam bicicletas, a pessoa aguarde que seja devolvida pelo menos uma.

--

4. [0,5v] Qual a característica da entidade de sincronização que criou na sua resposta anterior deve ter para que, havendo mais que uma pessoa à espera de bicicletas, seja respeitada a ordem de chegada?

5. [0,5v] Escreva o pseudo-código de um monitor que implemente a solução para o problema apresentado.

--

Grupo III [4v]

Considere um sistema de ficheiros do tipo Unix.

1. [1v] Apresente as tabelas de ficheiros abertos do processo, a tabela global de ficheiros abertos e a cache de inodes, assim como a sua interligação, ao longo de cada um dos passos seguintes:
- Processo P1 inicia a sua execução e cria um ficheiro de nome F1 para escrita e leitura.
 - Processo P1 abre ficheiro já existente, denominado F2, apenas para leitura.
 - Processo P1 cria processo filho P2.
 - Processo P2 abre ficheiro F1 para leitura.
 - Processo P2 abre ficheiro F2 apenas para leitura.

--

2. [1v] Dado o nome de um ficheiro /dir1/dir2/ficheiro3, como é que o sistema operativo obtém o inode correspondente ? Descreva passo por passo a tradução do nome.

3. [1v] Assuma que um bloco de um dado ficheiro é modificado por um processo. Quando é que esse bloco é realmente escrito em disco? Indique porque razão é assim.

4. O sistema de ficheiros do CD-ROM obriga a que todos os blocos de um mesmo ficheiro estejam de forma consecutiva e ordenada na partição.
- a. [0,5v] Porque razão o sistema de ficheiros do CD-ROM não é adequado a sistemas de ficheiros cujo conteúdo pode mudar dinamicamente?

- b. [0,5v] Em contraste, o ext2 permite que os blocos de dados de um determinado ficheiro possam estar dispersos pelo disco. Explique como é que as estruturas de dados do ext2 tornam isto possível. Ilustre com um exemplo.

Grupo IV [4v]

1. [0,5v] Considere o seguinte programa em Linux (a notação "%p" imprime o endereço contido num ponteiro):

```
int *ponteiro = ...
printf("%p", ponteiro);
```

O que será impresso no ecrã, um endereço virtual ou um endereço real? Justifique.

2. Considere um sistema com uma arquitectura paginada de memória virtual de 32 bits. Neste sistema, cada endereço virtual é composto em 22 bits (mais significativos) que indicam o nº de página e 10 bits (menos significativos) que indicam o deslocamento. Assuma que não existe TLB.

- a. [0,5v] Qual a dimensão das páginas deste sistema? Justifique.

- b. [0,5v] Qual a dimensão da tabela de páginas de um dado processo? Justifique.

- c. [0,8v] Sob a forma de uma função em pseudo-código, apresente o algoritmo que a Unidade de Gestão de Memória executa para traduzir um endereço virtual num endereço real.

No seu pseudo-código use a seguinte notação:

- para lançar uma excepção, escreva "lança excepção;"
- para acesso ao campo X da linha Y da tabela de páginas do processo em execução, escreva "tabelaPaginas[Y].X"

Assuma que o argumento acesso pode ter 1 de 3 valores: leitura, escrita, execução.

```

endereço traduz(num_pagina, deslocamento, acesso) {

return _____;
}

```

- d. [0,6v] Apresente um exemplo em que ocorrem 2 acessos a endereços virtuais distintos e em que a existência de TLB seria vantajosa. Justifique.

1º acesso: nº página: _____ deslocamento: _____
 2º acesso: nº página: _____ deslocamento: _____

Justificação:

3. [0,6v] Num sistema paginado, assuma que, num dado momento, o conteúdo da tabela de páginas é o seguinte para as páginas A e B.

	P	Prot	Base
Página A	1	rw	00 0111 0010
Página B	0	rw	00 0010 1001

Assuma que o processo em execução tenta ler da página B, que não está presente em RAM. Em consequência, a página B é transferida para memória RAM em substituição da página A. Indique o novo conteúdo das linhas relativas às páginas A e B na tabela de página.

	P	Prot	Base
Página A			
Página B			

4. [0,5v] Em Linux, um programa tenta escrever numa determinada página. Em consequência, ocorre uma exceção de acesso não permitido e o gestor de memória do núcleo é executado para tratar a exceção. O bit "Copy-on-Write" da página em causa está activo. Explique o acontecimento anterior ao acesso de escrita que originou esta situação.

Grupo V [4v]

1. [0,5v] Considere um canal de comunicação (através do núcleo) do ponto de vista da sua capacidade de armazenamento de informação. Esta capacidade permite o quê no que diz respeito à capacidade de recepção? Qual a vantagem?

2. [0,5v] Considere a sincronização no envio de mensagens denominada síncrona (*rendez-vous*). Diga em que consiste.

3. Considere um canal com ligação (modelo de diálogo).

a. [1v] Represente através de uma figura este modelo, indicando os processos cliente, servidor e servidor dedicado.

b. [1v] Considere que são usados *sockets* para implementar o modelo em causa. Diga qual a relação entre o processo servidor e processo servidor dedicado (não se esqueça de justificar).

c. [1v] Considere que são usados *sockets* para implementar o modelo em causa e que os processos servidor e servidores dedicados são implementados como tarefas (servidor e servidores dedicados são tarefas no mesmo processo). Se assim for, o sistema funciona como pretendido? Responda tendo em conta as implicações de serem tarefas reais ou pseudo-tarefas.
