



Nome:

Nº aluno:

- Qualquer tentativa de fraude conduzirá à anulação da prova para todos os intervenientes.
- Consulta apenas em papel. Durante o exame todos os dispositivos eletrónicos têm que permanecer desligados, com exceção de calculadoras simples.
- Todas as respostas devem ser diretas, objetivas e obrigatoriamente efetuadas na folha fornecida.

1. Diga o que entende por espera ativa quando usada numa solução para o problema da secção crítica. Esta solução pode resolver o problema? Se sim, deve ser usada?

2. Explique em que consiste o *copy-on-write* e de que forma pode otimizar a criação de processos. No Linux, a vantagem do *copy-on-write* mantém-se mesmo que após o `fork()` o utilizador faz logo um `exec()`?

3. Que vantagens e desvantagens existem no uso de uma tabela de páginas invertida?

4. Acha que é possível existir *Thrashing* num sistema operativo que tenha poucos processos a executar? Justifique.

5. Considere que tem um sistema de memória paginada hierárquico, com 2 níveis, com um espaço de endereçamento virtual de 4GByte ( $2^{32}$  bytes). Sabendo que cada página de memória ocupa 8KBytes e que cada PTE ocupa 64 bits, responda as seguintes questões:

- 5.1. Se cada tabela de páginas de 2.o nível tiver de caber numa única página de memória, qual a divisão de bits no endereço lógico? Apresente os cálculos que realizar.

- 5.2. Qual o tamanho ocupado pela tabela de páginas de 1.o nível?

- 5.3. No total, quantas páginas de 8KB são ocupadas por todas as tabelas de páginas (1.o e 2.o nível)?

6. Considere um sistema com 5 processos em execução (P1 a P5) e 3 tipos de recursos (R1 a R3). O número máximo de recursos de cada tipo existentes no sistema é: **R1=9, R2=2, R3=9** (**NOTA: este não é o número dos que estão disponíveis nesse momento, mas sim os que o sistema tem!**). Num determinado momento o sistema está no seguinte estado:

	Recursos em uso Pelos processos			Recursos máximos que os processos precisam		
	R1	R2	R3	R1	R2	R3
P1	0	1	0	2	2	2
P2	2	0	0	3	2	6
P3	3	0	3	9	0	4
P4	2	1	1	2	1	2
P5	2	0	3	4	2	4

**6.1.** Prove que o sistema está num estado seguro (*safe state*).

**6.2.** O processo P4 fez o seguinte pedido (R1=0, R2=0, R3=1). Qual será a decisão do gestor de recursos se aplicar o algoritmo do banqueiro (*Banker's Algorithm*)? Apresente o processo que usou para dar a resposta.

**7.** Considere um sistema onde existem 8 páginas de processos (1 a 8) e 4 *page frames* em RAM. Vão ser feitos acessos à memória usando a *string* de referência seguinte:

**W(1) ; W(3) ; R(2) ; R(3) ; R(8) ; W(5) ; R(3) ; R(8) ; R(1) ; W(6)**

Supondo que inicialmente todas as *frames* estão vazias, que R(...) é uma operação de leitura, que W(...) é uma operação de escrita e que o sistema faz uso do *modify-bit*, preencha a tabela abaixo considerando os algoritmos pedidos:

	FIFO	LRU	CLOCK
<b>Page-faults</b>			
<b>Swap-outs</b>			
<b>Estado final das <i>frames</i></b> (indique a <i>frame</i> , a página que cada <i>frame</i> contém e o <u>estado do <i>modify bit</i></u> )			

**Nota:** no início, com as *frames* vazias, as páginas ocupam primeiro a *frame* 0, depois a 1 e assim sucessivamente.

8. Suponha que temos um sistema Linux com blocos de disco de 4KB e ponteiros com 32 bits. Nesse sistema os inodes têm 8 ponteiros diretos, 1 ponteiro indireto e 1 ponteiro duplamente indireto para blocos. O sistema acabou de fazer um reboot pelo que todas as caches se encontram limpas e não existe nenhum inode em memória.

8.1. Qual o tamanho máximo de um ficheiro neste sistema? Pode enunciar apenas os cálculos.

8.2. Qual o número mínimo de blocos que é preciso ler do disco para ler o inode do ficheiro /usr/teste.pdf? Justifique.

8.3. Depois de ler o inode do ficheiro, quantos mais blocos é necessário ler para aceder ao byte 32770 desse mesmo ficheiro? Justifique.

**NOTA:** Esta folha não é para entregar

**Potência de 2**

$2^6$	$2^7$	$2^8$	$2^{10}$	$2^{12}$	$2^{14}$	$2^{16}$	$2^{18}$	$2^{20}$	$2^{30}$
64	128	256	1024	4096	16384	65536	262144	1048576	1073741824

**Grelha para algoritmos de substituição de páginas.**


Page Faults														
Swap-outs														


Page Faults														
Swap-outs														


Page Faults														
Swap-outs														