

Número: Nome: **LEIC/LETI – 2015/16 - 2º Teste de Sistemas Operativos****21 de Novembro de 2015****Responda no enunciado, apenas no espaço fornecido. Identifique todas as folhas.****Duração: 1h****Grupo I [12 Val]**

Nas alíneas seguintes considere um processo em ambiente Linux com 2 tarefas reais (também denominadas tarefas núcleo), em que cada uma executa um dos programas seguintes. O trinco lógico (*mutex*) e a variável de condição no código seguinte são mantidos pelo núcleo.

```
int numPedidos = 0;
lista pedidos = nova_lista();
mutex_t m = mutex_init();
condvar_t c = condvar_init();
```

Tarefa X:	Tarefa Y:
<pre>1 pedido p; 2 while (true) { 3 lock(m); 4 while (numPedidos == 0) 5 wait(c, m); 6 p = pedidos.removeProximo(); 7 numPedidos--; 8 unlock(m); 9 funcaoX(p); }</pre>	<pre>10 pedido p; 11 while (true) { 12 pedido p = funcaoY(); 13 lock(m); 14 numPedidos++; 15 pedidos.adiciona(p); 16 signal(c); 17 unlock(m); }</pre>

1. [2 Val] Descreva resumidamente os passos que ocorrem entre o momento em que a tarefa X chama a função `wait(c,m)` e o momento em que essa função retorna. Considere fundamentalmente os aspectos directamente relacionados com os objectos de sincronização.

2. [2 Val] Considere que a tarefa X está a executar a linha 7. Considerando agora a tarefa Y e as linhas 12-17 do código por ela executado, quais dessas linhas é que a tarefa Y pode nesse momento estar a executar? Justifique (as linhas que indicou e aquelas que deixou de fora da sua resposta).

3. [1,5 Val] As chamadas às funções `lock`, `unlock`, `wait` e `signal` são mais demoradas que as chamadas às funções `funcaoX` e `funcaoY` (se ignorarmos o tempo dispendido na execução das respetivas rotinas). Justifique, apontando 2 ações que acontecem quando as primeiras são chamadas e que não acontecem com as últimas.

4. [1,5 Val] Considere que a tarefa X chamou `wait(c, m)`, bloqueando-se e perdendo o processador para a tarefa Y. Indique qual a pilha (*stack*) está em uso pelo processador nos seguintes momentos:

- a. [0,5 Val] Imediatamente antes da tarefa X chamar `wait(c, m)`.

- b. [0,5 Val] Durante a execução da rotina associada ao `wait(c, m)` no núcleo.

- c. [0,5 Val] Após o retorno da interrupção (trap) causada pela chamada a `wait(c, m)`.

5. Assuma que funcaoX é ES-intensiva e inclui uma chamada sistema para ler do teclado; enquanto que funcaoY é CPU-intensiva e nunca chama funções sistema.
- a. [1,5 Val] Assuma também que enquanto a tarefa Y se executa, o utilizador insere dados através do teclado. Qual o mecanismo do Linux que permite que a tarefa X reaja rapidamente à ação do utilizador? Justifique descrevendo o mecanismo em causa.

- b. [1,5 Val] Em Linux, qual tarefa (X ou Y) tenderá a ser mais prioritária? Justifique referindo objetivamente a forma como o algoritmo de escalonamento do Linux calcula as prioridades. Assuma que as tarefas X e Y começaram simultaneamente e com prioridades iguais.

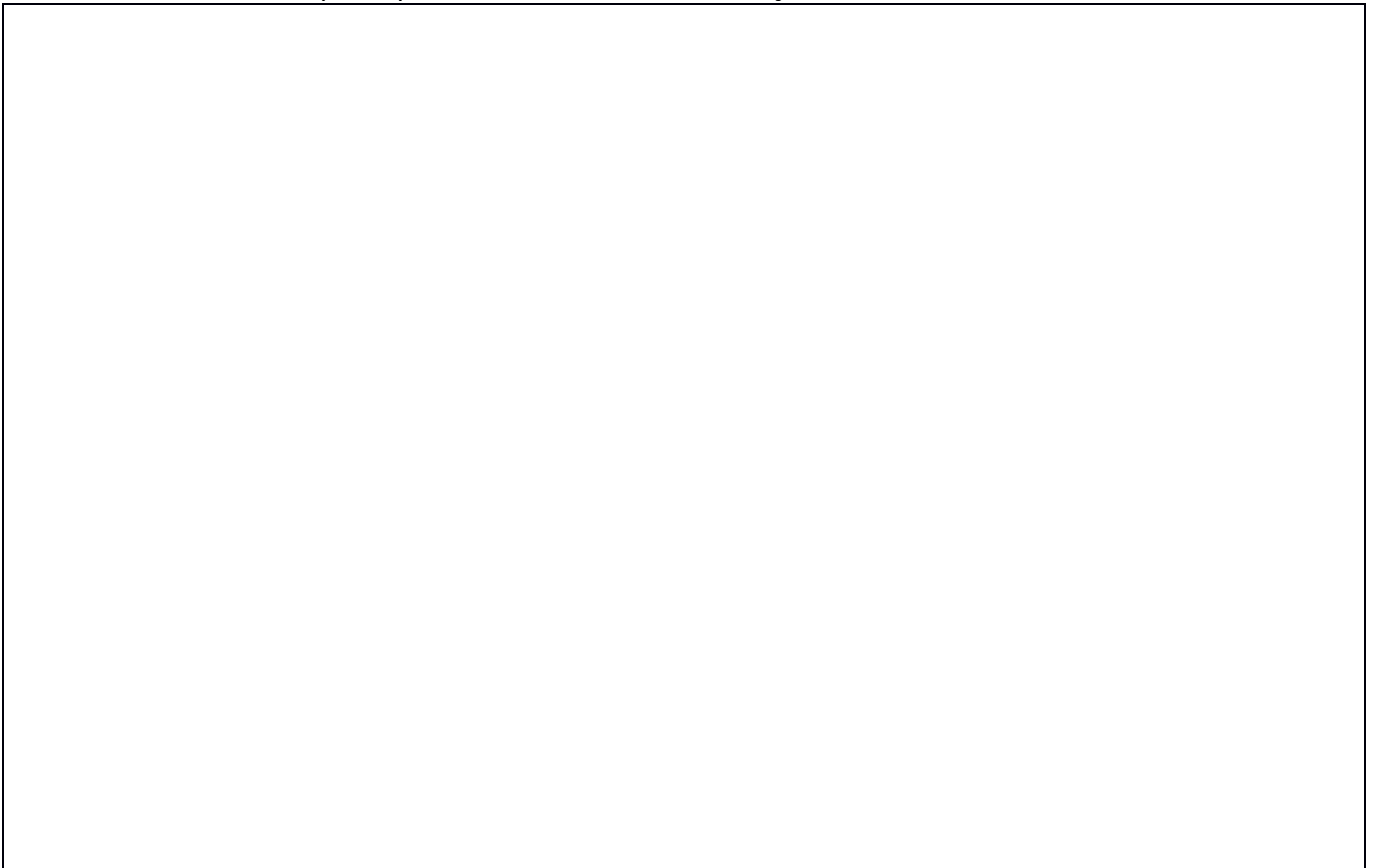
- c. [2 Val] O temporizador é uma componente *hardware* fundamental em sistemas de tempo partilhado (e.g. Linux). Assuma que, num dado gestor de processos, não existiam interrupções periódicas lançadas pelo temporizador. Partindo dos dados acima, apresente sucintamente um exemplo que ilustre o problema deste hipotético gestor de processos.

Grupo II [8 Val]

1. Considere um sistema de ficheiros do tipo EXT (Linux) no qual cada inode tem as seguintes características:

- 64 bytes usados para guardar metadados sobre o respectivo ficheiro,
- 5 referências directas para blocos,
- 1 referência indirecta de 1º nível,
- 1 referência indirecta de 2º nível,
- cada referência para um bloco em disco ocupa 4 bytes (dimensão R),
- cada bloco em disco tem 1024 bytes (dimensão B).

a. [1 Val] Qual a dimensão máxima de um ficheiro? Justifique a sua resposta apresentando uma figura que ilustre o inode e blocos endereçados assim como a fórmula que explicita essa dimensão em função de B e R.



b. [1 Val] Quantos blocos são ocupados por um ficheiro com a dimensão de 5130 bytes? Justifique a sua resposta.

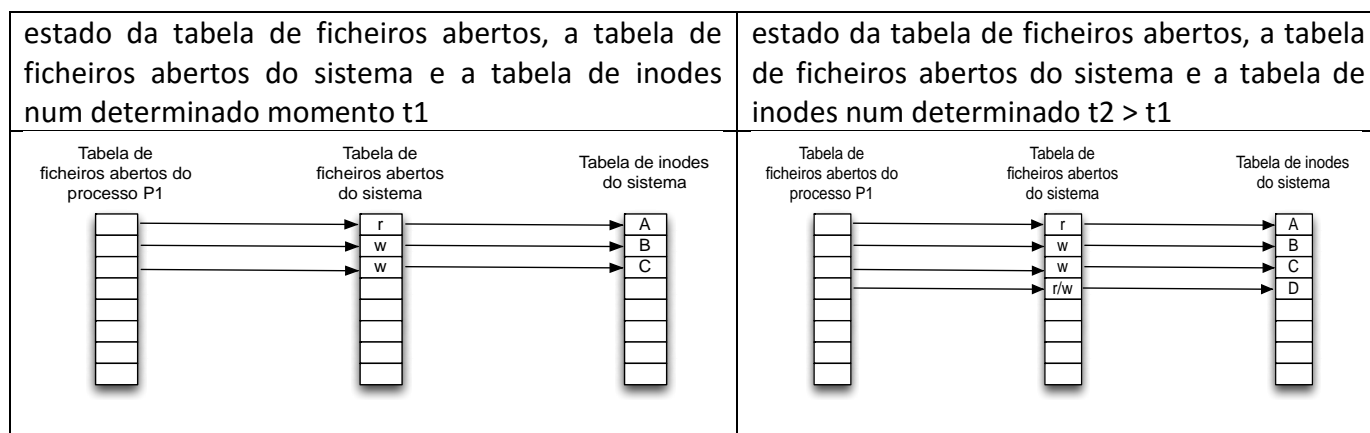


- c. [1 Val] Assuma que existe apenas uma partição no disco. Qual o número máximo de ficheiros que este sistema suporta sabendo que a tabela de *inodes* em disco ocupa, no máximo, X bytes ? Justifique a sua resposta.

2. [1 Val] Considere agora que tem um sistema de ficheiros do tipo FAT no qual cada referência para bloco têm 4 bytes e cada bloco tem 1024 bytes (tal como na questão anterior), e no qual está guardado um ficheiro com $5 \cdot 1024$ bytes (ou seja 5120 bytes). Conhecendo o índice do 1º bloco do ficheiro, quantas leituras da FAT tem de efetuar até localizar o bloco que contém a posição 5000? Justifique a sua resposta apresentando uma figura que ilustre a sua resposta.

3. [1 Val] Como compara, em termos de rapidez, o acesso (em leitura ou escrita) ao byte 5000 de um ficheiro que está guardado no EXT (Linux) e na FAT? Justifique a sua resposta.

4. Considere a figura seguinte que ilustra, à esquerda, o estado da tabela de ficheiros abertos, a tabela de ficheiros abertos do sistema e a tabela de *inodes* num determinado momento, e à direita ilustra o estado das mesmas tabelas depois do processo P1 executar uma chamada ao sistema *syscall1*.



- a. [1 Val] Diga qual foi a chamada sistema correspondente a *syscall1* e quais os valores dos argumentos (aqueles que a figura acima lhe permite conhecer).

- b. [1 Val] Considere que a partir do estado anterior, o processo P1 faz *fork* criando o processo P2. Complete a figura para representar o estado das tabelas após o *fork*.

- c. [1 Val] Considere que a partir do estado anterior, o processo P2 faz *exec*. Complete a figura para representar o estado das tabelas após o *exec*.