



ISEL – DEETC – SES
Engenharia de Informática e de Computadores
Sistemas Operativos

1ª Época (12/01/2007)

Duração: 2:30 H

Justifique todas as suas respostas.

I

1. [2 Valores] Caracterize, relacionando-os, cada um dos seguintes conceitos: processo, ficheiro e programa.
2. [2 Valores] Nos sistemas operativos multiprogramados e multiutilizador uma das acções realizadas, pelo núcleo, é a comutação de contexto (*context switch*). Diga em que consiste e em que circunstâncias é realizada.

II

1. [1,5 Valores] Digas as acções a realizar sobre a estrutura de ficheiros FAT quando se pretende mover o ficheiro `f.txt` da directoria `c:/dirOrig/` para `c:/dirDest/`.
2. [1,5 Valores] Apresente as vantagens e desvantagens dos sistemas de gestão de ficheiros de atribuição contígua. Indique uma circunstância onde este tipo de sistemas poderia ser adoptado.

III

1. [1,5 Valores] Qual o papel da TLB no processo de transformação de endereços virtuais em endereços físicos.
2. [2 Valores] Considere um processador que utiliza uma estrutura de paginação de dois níveis e uma dimensão de página de 16K. Sabendo que a directoria de página e as tabelas de página ocupam uma página, a entrada da directoria de página e das tabelas é de 4 bytes presente, de forma justificada, a dimensão do espaço de endereçamento virtual e quantas páginas existem nesse espaço de endereçamento.
3. [1,5 Valores] Considere o troço de código apresentado ao lado que utiliza o mecanismo SEH. Comente-o quanto à robustez do código face a anomalias de execução. No caso de existir uma forma mais adequada de estruturar o código apresente a sua solução.

```
...
__try {
    func();
    WaitForSingleObject(hsem, INFINITE);
    usarRecurso();
} __finally {
    ReleaseSemaphore(hsem, 1, NULL);
}
...
```

IV

1. [3 Valores] Considere que se pretende determinar o número total de ficheiros de música *mp3* existentes em 4 directorias (“dir1”, “dir2”, “dir3” e “dir4”). Para o efeito pretende-se um programa que utilize uma tarefa para pesquisar em cada uma das directorias, apresentando no final o valor total de ficheiros encontrados. Assuma que existe a função `[int pesquisa(char *dir, char *ext)]`, que devolve o número de ficheiros com extensão **ext** na directoria **dir**. Apresente o código da aplicação.

2. [2 Valores] Considere um cenário em que quatro tarefas (T_a , T_b , T_c e T_d) acedem a um objecto da classe seguidamente apresentada. As tarefas T_a e T_b acedem ao método `passaTipo0` e as tarefas T_c e T_d acedem ao método `passaTipo1`. Pretende-se a sequência pela qual as tarefas executam as instruções dos referidos métodos, tendo em conta que a ordem de início de execução é: $T_c - b1$, $T_d - b1$, $T_a - a1$, e $T_b - a1$. Em que a representação $T_x - l_i$ representa que a tarefa T_x executa a instrução da linha l_i . Considere que: depois do início, as tarefas executam as instruções até serem bloqueadas ou enquanto houver instruções por executar; e que o *scheduler* adiciona sempre as tarefas à cauda da lista *READY*.

<pre>class Acesso { private: Semaforo sem1, sem2, sem3; public Acesso() : sem1(1), sem2(0), sem3(0) { }</pre>	<pre>void passaTipo0() { a1 sem1.Wait(); a2 sem2.Signal(); a3 sem2.Signal(); a4 sem3.Wait(); a5 sem1.Signal(); }</pre>	<pre>void passaTipo1() { b1 sem2.Wait(); b2 sem3.Signal(); } } // end class Acesso</pre>
--	--	--

3. Pretende-se uma implementação de um mecanismo de sincronismo, baseado no monitor de Lampson, para controlar o jantar dos cavaleiros da Tábua Redonda*. Esta ordem consiste em 10 cavaleiros dos quais um deles é o Rei do reino. Todos eles alternam entre comer e conviver. No entanto, quando o rei convive nenhum dos outros cavaleiros pode começar a comer, de modo a não mostrar indiferença perante o rei. Por questões de segurança nunca podem estar todos os cavaleiros a comer, pois deve haver sempre um de vigília.



```
class IMesaTabuaRedonda {
    public :
        virtual void rei_comer()=0;
        virtual void rei_conviver()=0;
        virtual void cavaleiro_comer()=0;
        virtual void cavaleiro_conviver()=0;
};
```

- a) [2 Valores] Implemente a classe para este controlador que respeite a interface `IMesaTabuaRedonda` que se apresenta ao lado.
- b) [1 Valor] Considerando que as variáveis de condição do monitor de Lampson garantem ordenação (por exemplo, o primeiro cavaleiro a esperar é o primeiro a ser libertado). Indique justificando, se apesar dessa garantia o mecanismo que implementou garante a ordenação entre os cavaleiros.

Note que o único método bloqueante é o `cavaleiro_comer()`;

* Versão alternativa da Ordem da Távola Redonda do Rei Artur.

Diogo Remédios e Nuno Oliveira

Determine never to be idle...It is wonderful how much may be done if we are always doing.
-- Thomas Jefferson