



ISEL – DEETC
Licenciatura em Engenharia Informática e de Computadores
Sistemas Operativos

1ª Época (23/01/2009)

Duração: 2:30 H

Justifique todas as suas respostas.

I

1. [1,5 Valores] Indique, justificando, em que consistem as instruções privilegiadas e de que modo contribuem para garantir o correcto funcionamento do Sistema Operativo.
2. [2 Valores] A sincronização de tarefas pode ser realizada através de esperas activas ou esperas passivas. Defina cada um dos tipos de espera e indique qual o estado, do ciclo de vida da tarefa, em que esta se encontra quando realiza cada uma das respectivas esperas.
3. [1,5 Valores] Descreva de forma detalhada a execução da função `void iniciar()`.

```
int y = 0;

DWORD WINAPI ImprimeThread(LPVOID args){

    ++y;

    return 0;

}
```

```
void iniciar() {

    HANDLE ht;
    ht=chBEGINTHREADEX(NULL,0,ImprimeThread,
                        NULL,NULL,NULL);

    ImprimeThread(NULL);
    _tprintf(TEXT("y = %d\n"), y);

}
```

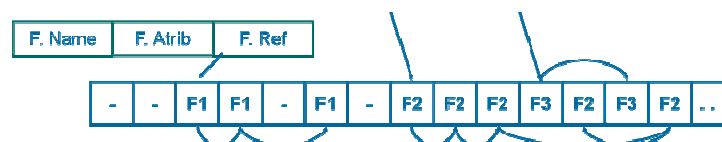
II

1. [2 Valores] Considere o seguinte código que utiliza as excepções estruturadas da API Win32 (SEH). Apresente a **ordem** e o **resultado** da execução do seguinte programa.

```
1 INT val = 0;
2
3 INT filtroSEH1() {
4     return EXCEPTION_CONTINUE_SEARCH;
5 }
6
7 INT filtroSEH2() {
8     val += 11;
9     return EXCEPTION_EXECUTE_HANDLER;
10 }
```

```
20 int _tmain(int argc, _TCHAR* argv[]){
21     __try {
22         __try {
23             int *p = 0x00000100;
24             *p = 10;
25             val+=66
26         } __finally {val+=99;}
27         __try {
28             val+=22;
29         } __finally {val+=66;}
30         __try {
31             val+=33;
32         } __except( filtroSEH1() ) {val+=55;}
33     } __except ( filtroSEH2() ) {
34         printf("Handler: val=%d\n", val);
35     }
36     printf("The Fat lady is singing\n");
37 }
```

2. [1 Valor] No contexto das aplicações gráficas na win32, explique a razão pela qual, é utilizado o envio de mensagens (`SendMessage(hEditCtrl, WM_GETTEXT, ...)`) para obter o texto contido num *EditControl* (campo de texto).
3. [2Valores] No contexto de um sistema de gestão de ficheiros de atribuição não contígua que utiliza a técnica de lista de blocos para manter a sequência de clusters que pertencem a um dado ficheiro (Como representado na figura). Discuta as características desta abordagem em relação a vantagens, desvantagens e do desempenho nos acessos aleatórios e sequenciais aos dados.



III

1. [2 Valores] Os sistemas operativos multi-processo têm de oferecer a garantia que cada processo nunca acede a memória de outros processos. Explique como o gestor de memória garante esta protecção.
2. [2 Valores] Considere uma arquitectura com suporte à gestão de memória virtual através de paginação com uma estrutura de dois níveis. A arquitectura suporta um espaço de endereçamento virtual de 4GBytes com páginas de 4 KBytes. A tabela de páginas do 1º nível tem 16 entradas. Cada entrada das tabelas de páginas (1º e 2º nível) ocupam 4 bytes. Apresente um esboço da organização deste sistema de gestão de memória, indicando o número de bits do endereçamento virtual, quantas páginas existem no espaço de endereçamento virtual e a dimensão mínima e máxima ocupada pelas estruturas de gestão de memória virtual associado a cada processo em execução.

IV

1. [2 Valores] Considere um cenário em que três tarefas (T_A , T_B e T_C) acedem a um objecto da classe *Acesso* que se apresenta de seguida. As tarefas T_A e T_B acedem ao método *passaTipo1* e a tarefa T_C acede ao método *passaTipo2*. Considere que: as tarefas se executam até ficarem bloqueadas ou enquanto houver instruções por executar; e o *scheduler* adiciona sempre as tarefas à cauda da lista *READY*.

| | | |
|---|---|---|
| <pre>class Acesso { private: Semaforo sem1, sem2, sem3; int i = 0; public Acesso() : sem1(1), sem2(1), sem3(1) { }</pre> | <pre>void passaTipo1() { a1 sem1.Wait(); a2 Sleep(0); a3 sem3.Wait(); a4 i++; a5 sem3.Signal(); a6 sem2.Signal(); }</pre> | <pre>void passaTipo2() { b1 sem2.Wait(); b2 sem3.Wait(); b3 i++; b4 sem3.Signal(); b5 sem1.Signal(); } }; // end class Acesso</pre> |
|---|---|---|

Pretende-se a sequência pela qual as tarefas executam as instruções dos referidos métodos, tendo em conta que já se encontram na lista de *Ready* pela seguinte ordem: T_A , T_B e T_C .

2. No decorrer do sucesso do livro “O código Da Vinci”, do escritor Dan Brown, o número de visitantes no museu do Louvre em Paris aumentou, obrigando a um reforço das medidas de segurança. Assim, existem seguranças que, periodicamente, fazem inspecções aos sistemas de vigilância em que nessas alturas não podem existir visitantes no museu. Por outro lado os seguranças esperam que todos os visitantes terminem a sua visita para entrarem e iniciarem a sua inspecção. De forma semelhante os visitantes têm de aguardar a saída de todos os seguranças.

```
class IGestorAcessoLouvre {
public:
    virtual void esperarParaVisitar() = 0;
    virtual void esperarParaInspeccionar() = 0;
    virtual void sairDeVisita() = 0;
    virtual void sairDeInspeccao() = 0;
};
```

- a) [3 Valores] Considerando que os visitantes e os seguranças são simulados por tarefas, realize uma implementação baseada em monitores do gestor de acesso ao museu que obedeça à interface apresentada.
- b) [1 Valor] Indique que alterações teriam de ser efectuadas, na sua solução, de modo a atribuir prioridade de acesso aos seguranças (Quando um segurança esperar para inspeccionar o museu, não entra mais nenhum visitante até este terminar a sua inspecção).

Nuno Oliveira e Diogo Remédios

Everywhere is within walking distance if you have the time.
--Steven Wright