ISEL – DEETC – SES

Licenciatura em Engenharia Informática e de Computadores

Sistemas Operativos

1ª Época (27/06/2008) Duração: 2:30 H

Justifique todas as suas respostas.

I

1. **[1,5 Valores]** No contexto de um Sistema Operativo indique, justificando, em que consiste uma chamada ao sistema (*system call*) e a razão pela qual estas são necessárias.

2. **[1,5 Valores]** Explique porque motivo não se deve utilizar a primitiva *TerminateThread* , disponível na API da Win32, que permite terminar uma tarefa. Indique qual a forma correcta de terminar uma tarefa.

3. **[2 Valores]** Uma equipa de programação necessitou de desenvolver um troço de código para realizar um acesso exclusivo a um recurso. A solução encontrada foi a que se apresenta de seguida. Comente, justificando, a solução encontrada.

```
/* Tarefa 1 */
void funcTarefa1(){
  while(flag[1])
    ;
  flag[0] = true;
  /* Acesso ao recurso crítico */
  flag[0] = false;
  ...
}
```

```
/* Tarefa 2 */
void funcTarefa2(){
  while(flag[0])
    ;
  flag[1] = true;
  /* Acesso ao recurso crítico */
  flag[1] = false;
  ...
}
```

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       __try {
24         int *p = 0;
25         *p = 5;
26         val+=66
27         __try {
28           val+=22;
29         } __except( filtroSEH1() ) {val+=55;}
30       } __finally {val+=33;}
31       val+=44;
32     } __except(EXCEPTION_CONTINUE_SEARCH){val++;}
33   } __except ( filtroSEH2() ) {
34     printf("Handler: val=%d\n", val);
35   }
36   printf("The Fat lady is singing\n");
37 }
```

2. **[1 Valor]** Explique a razão pela qual a terminação de uma aplicação gráfica desenvolvida com recurso à API Win32 deve invocar a função `PostQuitMessage(0)` após a recepção da notificação de que a janela principal foi destruída.

3. **[2Valores]** No contexto do sistema de gestão de ficheiros `NTFS`, indique no diagrama apresentado em anexo o resultado das estruturas da *Master File Table* (*Root dir*, *freeBlock Bitmap*, entradas da `MFT` referentes aos ficheiros e directorias e *clusters* de dados ocupados) após a realização das seguintes operações:

   - Criação do ficheiro: `c:\a.txt` (com a *string* de dados "**`Olá Mundo`**")
   - Criação do ficheiro: `c:\dirX\dirY\b.txt` (1 *cluster* de dados)

III

1. **[2,5 Valores]** Os sistemas operativos com gestão de memória virtual paginada organizam as tabelas de páginas em vários níveis. Indique as razões que levam à adopção desta técnica bem como as suas consequências.

2. **[2,5 Valores]** Considere uma arquitectura com suporte à gestão de memória virtual através de paginação com uma estrutura de três níveis. A arquitectura suporta um espaço de endereçamento virtual com páginas de 128 *bytes*. Admita que as entradas da tabela de páginas do 1º nível ocupam 4 *bytes* e as do 2º e do 3º nível ocupam 2 *bytes*. Assuma que cada tabela de páginas (1º, 2º e 3º nível) ocupa o tamanho de uma página. 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, qual a dimensão do 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 código de sincronismo (classe Mecanismo) e assuma que existem 3 tarefas (T1, T2, e T3) a utilizar o recurso de acordo com a seguinte tabela (considere que a variável `gestor` é partilhada por todas as tarefas).

Tarefa 1 (T1)	Tarefa 2 (T2)	Tarefa 3 (T3)

`gestor->entrarA();`
`// acesso;`
`gestor->sair();`
	`gestor->entrarB();`
	`// acesso;`
	`gestor->sair();`
		`gestor->entrarA();`
		`// acesso;`
		`gestor->sair();`

```
1   class Mecanismo : public MonitorHoare
2   {
3     private:
4       int cnt = 1;
5       int f = 0;
6       Condition* c1;
7       Condition* c2;
8     public:
9       Mecanismo() {
10        c1 = new Condition(this);
11        c2 = new Condition(this);
12      }
13
14      void entrarA() {
15        Enter();
16        if ((f>0)||(cnt==0))
17          c1->Wait();
18        --cnt;
19        Leave();
20      }
```

```
21      void entrarB() {
22        Enter();
23        if ( cnt==0 ) {
24          ++f;
25          c2->Wait();
26          --f;
27        }
28        --cnt;
29        Leave();
30      }
31
32      void sair() {
33        Enter();
34        ++cnt;
35        if ( f>0 )
36          c2->Signal();
37        else
38          c1->Signal();
39        Leave();
40      }
41  };
```

Complete a seguinte ordem de execução até que todo o código das tarefas seja executado:

T1 – 15,16,18,19; **T2** – 22,23,24,25; **T3** – 15,16,17; **T1** – 33, …

2. **[3 Valores]** Considere uma aplicação que necessita de gerir o acesso a um conjunto de impressoras. Esta aplicação controla o estado da impressora (disponível/indisponível), bem como a atribuição (e respectiva libertação) de uma impressora a quem a requisitar. Apresente uma implementação do gestor de impressoras, obedecendo à interface em anexo (`IGestorImpressoras`). Serão valorizadas as implementações que privilegiarem a rotatividade da atribuição das impressoras. Utilize os mecanismos de sincronismo que achar mais adequados.

```
class IGestorImpressoras {
  public:
   virtual int pedirImpressora()=0;
   virtual void libertaImpressora(int nPrinter)=0;
   virtual void setOnline(int nPrinter)=0;
   virtual void setOffline(int nPrinter)=0;
};
```

Carlos Gonçalves e Diogo Remédios

Some days you're a bug, some days you're a windshield.
--Price Cobb

Anexo:

- Diagrama das estruturas do sistema de gestão de ficheiros `NTFS` para resposta da pergunta II.3
  Nota: O esquema assume que a *Master File Table* (`MFT`) ocupa os 3 primeiros clusters de dados.