



**Justifique todas as suas respostas.**

**I**

1. **[1,5 Valores]** No contexto de um sistema operativo multi-utilizador, discuta quais as consequências, de fazer variar a dimensão das fatias temporais (*time quantum*) para dimensões demasiado pequenas e demasiado longas.
2. **[1,5 Valores]** No decorrer da disciplina estudou a criação de processos e tarefas no sistema Windows. Refira as principais diferenças entre um processo e uma tarefa.
3. **[1,5 Valores]** A sincronização de tarefas pode ser realizada através de esperas activas ou esperas passivas. Defina cada um dos tipos de espera e apresente um pequeno exemplo ilustrativo de cada uma delas.

**II**

1. **[1,5 Valores]** No contexto das aplicações gráficas multi-tarefa no Windows, comente a seguinte afirmação: “Todo o código da WndProc, associada ao tratamento de eventos recebidos da MessageQueue, é sempre executado apenas por uma tarefa, pelo que nunca é preciso utilizar mecanismos de sincronismo”.
2. **[2 Valores]** No contexto das bibliotecas de carregamento dinâmico do Windows (DLL's), indique qual o objectivo da função opcional `dllMain(...)`, em que circunstâncias é invocada e que tipo de operações tipicamente são lá colocadas.
3. **[1,5 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. Identifique potenciais problemas indicando a respectiva solução.

```
1 INT filtroSEH1() {  
2     printf("Teste 1\n");  
3     return EXCEPTION_CONTINUE_SEARCH;  
4 }
```

```
20 int _tmain(int argc, _TCHAR* argv[]){  
21     int *i = 0;  
22     __try {  
23         while(true){  
24             __try {  
25                 *i++;  
26             } __except (EXCEPTION_EXECUTE_HANDLER) {  
27                 printf("Erro 1...\n");  
28             }  
29         }  
30     } __except ( filtroSEH1() ) {  
31         printf("Erro 2...\n");  
32     }  
33     printf("The Fat lady is singing\n");  
34 }
```

## III

1. [2 Valores] Considere uma arquitectura de suporte à gestão de memória paginada utilizando um endereçamento virtual de 32 bits e páginas de 4 KBytes. Um processo P, em execução nesse sistema, possui uma estrutura de suporte à paginação com a tabela de páginas apresentada na figura 1. Cada tabela ocupa a dimensão de uma página e cada entrada nas tabelas ocupa a dimensão de 4Bytes.

Para cada um dos seguintes **endereços físicos**, acedidos pelo processo, indique o respectivo **endereço virtual justificando a sua resposta**.

a.1) Leitura 0x0E1E0135

a.2) Escrita 0x001AEF53

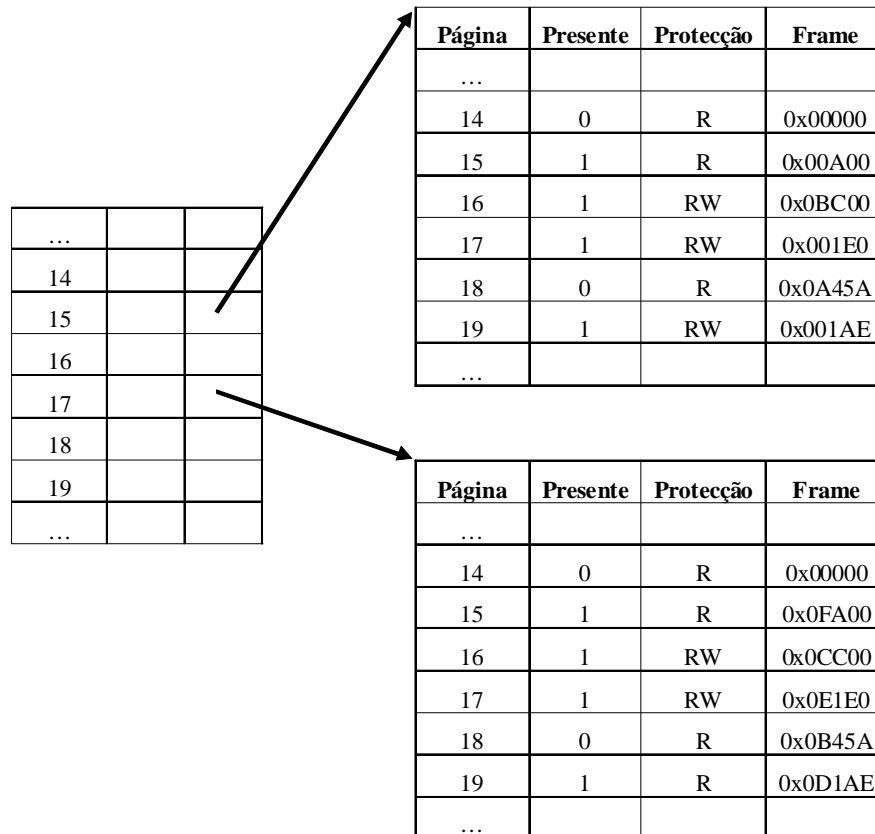


Figura 1

2. [1,5 Valores] Faça uma análise, que pondere as implicações, em relação ao número de níveis que deve ser utilizado no esquema de tradução de endereços na memória virtual utilizando tabelas de páginas. Indique vantagens, desvantagens e exemplos concretos que o demonstrem.
3. [1,5 Valores] Uma forma de utilizar a memória virtual no Windows é através da função VirtualAlloc que possui as opções de MEM\_RESERVE e MEM\_COMMIT. Descreva em que consiste estes dois modos de operação e qual as vantagens da sua utilização.

## IV

1. [2,5 Valores] Implemente uma função, que recebe um argumento com o caminho e o nome de um ficheiro executável e que tem de lançar um processo para o executar. Após a terminação, do processo, tem de ser escrito no *stdout* a indicação que o processo com o nome respectivo terminou a sua execução. Tenha em atenção que a função não pode ser bloqueante, para permitir lançar vários processos em simultâneo.
2. [3 Valores] Pretende-se implementar o mecanismo de caixa de correio, a ser utilizado num ambiente de multitarefa, onde é possível depositar três tipos diferentes de mensagens e receber as mensagens de forma selectiva. Realize uma classe que implementa a interface **ICaixaCorreio** que possui os métodos **enviar** e **receber**. No método **enviar** além dos dados, passados como um vector bytes, é também indicado o tipo de mensagem. No método **receber** é indicado o tipo de mensagem que se pretende receber. No caso de não existir nenhuma mensagem do tipo indicado a tarefa que invocou o método deve bloquear até outra tarefa enviar uma mensagem desse tipo.

```
class ICaixaCorreio {
public:
    virtual void enviar(BYTE *msg, int tipo)= 0;
    virtual BYTE *receber (int tipoMsg) = 0;
};
```

*Nuno Oliveira e Diogo Remédios*

"The difference between school and life? In school, you're taught a lesson and then given a test. In life, you're given a test that teaches you a lesson."  
-- Tom Bodett

## ANEXO

```

BOOL WINAPI CreateProcess(
    __in LPCTSTR lpApplicationName,
    __in_out LPTSTR lpCommandLine,
    __in LPSECURITY_ATTRIBUTES lpProcessAttributes,
    __in LPSECURITY_ATTRIBUTES lpThreadAttributes,
    __in BOOL bInheritHandles,
    __in DWORD dwCreationFlags,
    __in LPVOID lpEnvironment,
    __in LPCTSTR lpCurrentDirectory,
    __in LPSTARTUPINFO lpStartupInfo,
    __out LPPROCESS_INFORMATION lpProcessInformation
);

VOID WINAPI ExitProcess(
    __in UINT uExitCode
);

BOOL WINAPI GetExitCodeProcess(
    __in HANDLE hProcess,
    __out LPDWORD lpExitCode
);

HANDLE WINAPI CreateThread(
    __in LPSECURITY_ATTRIBUTES lpThreadAttributes,
    __in SIZE_T dwStackSize,
    __in LPTHREAD_START_ROUTINE lpStartAddress,
    __in LPVOID lpParameter,
    __in DWORD dwCreationFlags,
    __out LPDWORD lpThreadId
);

VOID WINAPI ExitThread(
    __in DWORD dwExitCode
);

BOOL WINAPI GetExitCodeThread(
    __in HANDLE hThread,
    __out LPDWORD lpExitCode
);

DWORD WINAPI WaitForSingleObject(
    __in HANDLE hHandle,
    __in DWORD dwMilliseconds
);

DWORD WINAPI WaitForMultipleObjects(
    __in DWORD nCount,
    __in const HANDLE* lpHandles,
    __in BOOL bWaitAll,
    __in DWORD dwMilliseconds
);

void WINAPI InitializeCriticalSection(
    __out LPCRITICAL_SECTION lpCriticalSection
);

void WINAPI EnterCriticalSection(
    __in_out LPCRITICAL_SECTION lpCriticalSection
);

void WINAPI LeaveCriticalSection(
    __in_out LPCRITICAL_SECTION lpCriticalSection
);

void WINAPI DeleteCriticalSection(
    __in_out LPCRITICAL_SECTION lpCriticalSection
);

HANDLE WINAPI CreateMutex(
    __in LPSECURITY_ATTRIBUTES lpMutexAttributes,
    __in BOOL bInitialOwner,
    __in LPCTSTR lpName
);

BOOL WINAPI ReleaseMutex(
    __in HANDLE hMutex
);

HANDLE WINAPI CreateSemaphore(
    __in LPSECURITY_ATTRIBUTES lpSemaphoreAttributes,
    __in LONG lInitialCount,
    __in LONG lMaximumCount,
    __in LPCTSTR lpName
);

BOOL WINAPI ReleaseSemaphore(
    __in HANDLE hSemaphore,
    __in LONG lReleaseCount,
    __out LPLONG lpPreviousCount
);

HANDLE WINAPI CreateEvent(
    __in LPSECURITY_ATTRIBUTES lpEventAttributes,
    __in BOOL bManualReset,
    __in BOOL bInitialState,
    __in LPCTSTR lpName
);

BOOL WINAPI SetEvent(
    __in HANDLE hEvent
);

BOOL WINAPI ResetEvent(
    __in HANDLE hEvent
);

```