

Este exame tem 10 questões, num total de 200 pontos. Fundamente todas as respostas.

Sistemas de entrada/saída

1. Uma empresa pretende usar um sistema RAID-5 com 5 grupos de proteção para armazenar a sua base de dados, que ocupa 40 TB. Existem duas opções: discos do modelo D1 com capacidade para 1 TB ou discos do modelo D2 com capacidade para 2 TB.

[10] (a) Determinar o número de discos que é necessário comprar para cada opção.

Resposta: Com existem cinco grupos de proteção, cada grupo deve ter capacidade útil de $40 \text{ TB} / 5 = 8 \text{ TB}$. Além disso, cada grupo deve ter um disco adicional que corresponde à informação de paridade. Logo:

D1 Número de discos: $5 \times (8 + 1) = 45$,

D2 Número de discos: $5 \times (4 + 1) = 25$.

[10] (b) Determinar qual das opções leva a um sistema mais fiável.

Resposta: O sistema D1 suporta 1 falha em cada grupo, i.e., 1 falha em cada 9 discos. O sistema D2 também suporta 1 falha em cada grupo, i.e., 1 falha em cada 5 discos. Logo, o sistema D2 é mais fiável.

2. Um recetor GPRS (usado para comunicação de dados via rede de telemóvel) suporta uma taxa máxima de transferência de 16000 bit/s. Este periférico agrupa os dados recebidos em blocos de 400 bit. Sempre que um bloco de dados está completo, o processador deve transferi-lo do periférico para a memória. O recetor só recebe mais dados quando o bloco já completado tiver sido transferido (tempo de transferência desprezável).

[10] (a) Assumindo que a gestão do periférico é feita por varrimento, determinar de quanto em quanto tempo é necessário fazer um varrimento de forma a garantir que não se perdem dados.

Resposta: Primeiro é preciso determinar quanto blocos devem ser transferidos por segundo. Como o computador pode receber no máximo 16000 bit/s, o número máximo de blocos é $16000 / 400 = 40$.

É preciso fazer um varrimento para transferir um bloco, logo é necessário garantir que são feitos 40 varrimentos por segundo, i.e., um varrimento cada 25 ms.

- [10] (b) Assumir agora que o atendimento do periférico é feito com uso de interrupções e que que o computador está, em média, a receber dados durante 10% do tempo. Determinar quantas vezes por segundo, em média, é que a sub-rotina de atendimento é executada.

Resposta: Em média, num segundo, o computador recebe 10% de 16000 bit, ou seja, 1600 bit. A sub-rotina de atendimento de interrupções é chamada sempre que um bloco de 400 bit está completo. Num segundo, a sub-rotina é chamada $1600/400 = 4$ vezes.

Análise de código

3. Considere o seguinte programa em *assembly* IA-32.

```
1  include mpcp.inc
2
3  SBR  proto p1:ptr byte, p2:ptr byte, n:dword
4
5      .data
6  cad1 byte "As armas e os baroes assinalados", 0
7  cad2 byte lengthof cad1 dup(?)
8  res  byte "%s",0
9
10     .code
11  main:
12      invoke SBR, offset cad1, offset cad2, lengthof cad1
13      invoke printf, offset res, offset cad2
14      invoke _getch
15      invoke ExitProcess, 0
16
17  TAM  proc uses esi p:ptr byte
18      mov esi, p
19      .while (byte ptr [esi] != ' ' && byte ptr [esi] != 0)
20          inc esi
21      .endw
22      mov eax, esi
23      sub eax, p
24      ret
25  TAM  endp
26
27  SBR  proc uses esi edi p1:ptr byte, p2:ptr byte, n:dword
28      mov esi, p1
29      mov edi, p2
30      add edi, n
31      .repeat
32          invoke TAM, esi
33          inc  eax
34          mov  ecx, eax
35          sub  edi, eax
36      @@: movsb
37          loop @B
38          sub  edi, eax
39      .until (byte ptr [esi-1] == 0)
```

```

40     mov edi, p2
41     mov byte ptr [edi+eax-1], ' '
42     add edi, n
43     mov byte ptr [edi-1], 0
44     ret
45 SBR endp
46 end main

```

- [10] (a) Indique para que efeito é utilizada a sub-rotina **TAM** e determine quantas vezes é invocada durante a execução do programa.

Resposta:

A sub-rotina **TAM** é usada para determinar o tamanho de cada palavra da cadeia de caracteres definida por **cad1**. Portanto, é executada 6 vezes.

- [10] (b) Indique o que é apresentado no monitor quando o programa é executado e descreva a função realizada pela sub-rotina **SBR**.

Resposta:

No monitor aparece: assinalados baroes os e armas As

A sub-rotina **SBR** recebe o endereço de uma cadeia de caracteres e cria uma nova cadeia composta pelas mesmas palavras da cadeia original por ordem inversa.

- [10] (c) Considere a execução da primeira iteração do ciclo **.repeat**. Determine o número de acessos a memória durante a execução das instruções entre as linhas 36 e 39, inclusive.

Resposta: Na primeira iteração a sub-rotina **TAM** devolve **EAX=2**, correspondente ao tamanho da palavra “As”, pelo que a instrução **movsb** é executada três vezes. Em cada uma delas é feita uma leitura e uma escrita, originando 6 acessos a memória. Na avaliação da condição na diretiva **.until** é feita uma leitura de memória.

Conclusão: o número de acessos a memória é 7.

- [10] (d) A tabela ao lado apresenta o estado da pilha (valores em hexadecimal) imediatamente após a execução do prólogo da sub-rotina **TAM** na primeira vez em que é invocada.
O valor **011F5770H** aparece em três posições da pilha. Indique a que se refere este valor e determine o endereço das posições de memória ocupadas pelos dados declarados no programa.

Endereço	Conteúdo
001AFBBC	011F5770
001AFBB8	011F1B21
001AFBB4	001AFBD0
001AFBB0	00000000
001AFBAC	00000000
001AFBA8	011F5770
001AFBA4	011F1B71
001AFBA0	001AFBB4
001AFB9C	011F5770

Resposta:

A execução do prólogo da sub-rotina **TAM** foi antecedida da invocação da sub-rotina **SBR**, execução do respetivo prólogo e invocação de **TAM**. Atendendo às instruções **push** executadas nestas operações pode deduzir-se o significado de cada conteúdo da pilha.

A tabela seguinte apresenta o significado do conteúdo em cada endereço da pilha e a operação que o originou.

Endereço	Conteúdo	Significado	Operação
001AFBBC	011F5770	Endereço de cad1	Invocação de SBR
001AFBB8	011F1B21	Endereço de retorno a main	Invocação de SBR
001AFBB4	001AFBD0	Valor de EBP	Prólogo de SBR
001AFBB0	00000000	Valor de ESI a preservar (uses)	Prólogo de SBR
001AFBAC	00000000	Valor de EDI a preservar (uses)	Prólogo de SBR
001AFBA8	011F5770	Valor de ESI (endereço de cad1)	Invocação de TAM
001AFBA4	011F1B71	Endereço de retorno a SBR	Invocação de TAM
001AFBA0	001AFBB4	Valor de EBP	Prólogo de TAM
001AFB9C	011F5770	Valor de ESI a preservar (uses)	Prólogo de TAM

O valor **011F5770H** que ocorre nas posições **001AFBBCH**, **001AFBA8H** e **001AFB9CH** é o endereço da cadeia de caracteres definida em memória por **cad1**.

Os dados definidos pela diretiva **.data** são três sequências de bytes, traduzindo cadeias de caracteres. **cad1** tem 33 (=21H) caracteres (incluindo o delimitador), tal como **cad2**. A última cadeia, **res**, possui apenas 3 caracteres. O total de caracteres (bytes) é 69 = 45H. Portanto, os dados em memória ocupam os endereços de **011F5770H** a **011F57B4H**.

Programação

4. Considere duas sequências **S1** e **S2**, com **N1** e **N2** inteiros positivos (**N1**>0, **N2**>0), respetivamente. Cada elemento **ELEM** de **S1** é índice de um elemento de **S2**. Para cada elemento de **S1**, pretende-se obter a soma dos elementos de **S2** a partir desse índice **ELEM**. Este índice é válido se for menor que **N2**. Pretende-se criar uma sequência **SEQRES**, formada pelas somas assim obtidas e determinar o seu tamanho.

Exemplo: se **S1**={2, 4, 0, 9, 1} (**N1**=5) e **S2**={8, 5, 1, 0, 3, 2} (**N2**=6)

então **SEQRES**={6, 5, 19, 11} e o tamanho é 4. O elemento 9 não é um índice válido de **S2**, não originando qualquer termo em **SEQRES**.

- [20] (a) Implementar a sub-rotina **SUBSOMA** que determina a soma dos elementos de uma sequência **SEQ**, com **N** elementos, a partir do índice dado por **ELEM**. Caso esse índice seja válido (**ELEM** < **N**) a sub-rotina deve retornar a soma, senão deve retornar o valor -1. O protótipo desta sub-rotina é:

SUBSOMA proto **ELEM:dword, SEQ:ptr dword, N:dword**

Resposta:

```

SUBSOMA proc uses esi ebx ELEM:dword, SEQ:ptr dword, N:dword
    mov     esi, SEQ
    mov     ecx, N
    sub     ecx, ELEM
    .if (sdword ptr ecx>0)
        mov     eax, 0
        mov     ebx, ELEM
        shl     ebx, 2
        add     esi, ebx
    @@:     add     eax, [esi]
            add     esi, 4
            loop   @B
    .else
        mov     eax, -1
    .endif
    ret
SUBSOMA endp

```

- [20] (b) Utilizar a sub-rotina anterior para implementar a sub-rotina **SEQSOMAS** que executa a tarefa descrita inicialmente (preenche **SEQRES** e retorna o respetivo tamanho). O protótipo da sub-rotina é:

```

SEQSOMAS proto S1:ptr dword,N1:dword,S2:ptr dword,N2:dword,
            SEQRES:ptr dword

```

Resposta:

```

SEQSOMAS proc uses esi edi ebx S1:ptr dword, N1:dword,
            S2:ptr dword, N2:dword, SEQRES:ptr dword
    mov     ecx, N1
    mov     esi, S1
    mov     edi, SEQRES
    xor     ebx, ebx      ; Tamanho de SEQRES
    @@:     push    ecx
            invoke  SUBSOMA, dword ptr [esi], S2, N2
            pop     ecx
            .if (sdword ptr eax>=0)
                mov     [edi], eax
                add     edi, 4
                inc     ebx
            .endif
            add     esi, 4
            loop   @B
            mov     eax, ebx
            ret
SEQSOMAS endp

```

5. Considerar a seguinte série de potências de um valor real x em que todos os coeficientes são números inteiros:

$$f(x) = a_0 + a_1x + a_2x^2 + \dots, \quad a_i \in \mathbb{Z}, x \in \mathbb{R}$$

- [20] (a) Escrever a sub-rotina **serie**, que recebe uma sequência de N+1 coeficientes inteiros e um valor real X e calcula a função real $f(x)$. O protótipo da sub-rotina é:

```
serie proto A:ptr sdword, N:dword, X:real8
```

Resposta:

```
serie PROC a:ptr SDWORD, n: DWORD, x: REAL8
    mov     edx, a
    fild    SDWORD PTR [edx]
    add     edx, type SDWORD
    mov     ecx, n
    fld1
ciclo:  fmul    x
        fild    SDWORD PTR [edx]
        fmul    st(0), st(1)
        faddp   st(2), st(0)
        add     edx, type sdword
        loop    ciclo
        fstp    st(0)
        ret
serie ENDP
```

- [20] (b) A função exponencial pode ser aproximada por:

$$e^x \approx \frac{360 + 120x + 12x^2}{360 - 240x + 72x^2 - 12x^3 + x^4}$$

Usando a sub-rotina da alínea anterior, escrever um programa que apresenta no monitor o valor aproximado de e^x para $x = 0, 0.05, 0.1, 0.15, \dots, 2.0$ (21 valores). Cada linha deve incluir o valor de x e o valor aproximado de e^x correspondente.

Resposta:

```
NPONTOS = 21
serie PROTO a: ptr SDWORD, n: DWORD, x: REAL8
    .data
output    BYTE    "exp(%f) = %f", 13, 10, 0
contador  dword 0
coef_a    sdword 360, 120, 12
coef_b    sdword 360, -240, 72, -12, 1
xval      real8 0.0
res       real8 ?
c5        real8 0.05
    .code
main:
    .WHILE contador < NPONTOS
        invoke serie, offset coef_a, lengthof coef_a - 1, xval
```

```
fstp    res
invoke  serie, offset coef_b, lengthof coef_b - 1, xval
fdivr   res
fstp    res
invoke  printf, offset output, xval, res
fld     xval
fadd    c5
fstp    xval
inc     contador
.ENDW
invoke  _getch
invoke  ExitProcess, 0
end main
```

Escolha múltipla

[8] 6. Supondo que AL=54h, indique qual das instruções tem como resultado AL=45h.

- A. ror AL,3 B. rol AL,4 C. rcr AL,3 D. rcr AL,4

Resposta: B.

[8] 7. Supondo que inicialmente ECX=2 e EDI=200000h, o fragmento de código

```
std
xor  eax,eax
rep stosd
```

preenche com zero a zona de memória compreendida entre:

- A. 200000h e 200007h B. 200002h e 200009h
C. 1FFFFCh e 200003h D. 1FFFF8h e 1FFFFFh

Resposta: C.

[8] 8. A declaração em C++ da rotina `calc` é:

```
extern "C" double calc(int n, double f);
```

Indique o cabeçalho da correspondente sub-rotina em linguagem *assembly*.

- A. `calc proc C uses edi esi n:dword, f:real8`
B. `calc proc C uses ebx n:sdword, f:real8`
C. `calc proc C n:sword, f:real10`
D. `calc proc C n:sdword, f:real4`

Resposta: B.

- [8] 9. O modelo de disco magnético M1 roda a 7500 RPM. O modelo M2 é idêntico, mas tem uma velocidade de rotação de 15000 RPM.

Para ler um setor, o disco M1 tem um tempo médio de acesso t_1 e o disco M2 um tempo médio t_2 . Indique a afirmação verdadeira.

A. $t_2 = \frac{t_1}{2}$ B. $t_1 = t_2$ C. $t_2 < \frac{t_1}{2}$ D. $\frac{t_1}{2} < t_2 < t_1$

Resposta: D.

- [8] 10. Considere o seguinte fragmento:

```
mov edi, 07004000H
mov esi, 07005000H
mov ecx, 16
rep movsw
```

Quantos bytes de dados são mudados de posição durante a execução do fragmento?

A. 32 B. 16 C. 0 D. 64

Resposta: A.

Fim do enunciado.