

Este exame tem 6 questões, num total de 200 pontos.

**Responda às questões 1 e 2 numa folha simples e às restantes numa folha dupla.
Fundamente todas as respostas.**

Análise de código

1. Considere a execução do seguinte programa:

```
1  include \masm32\include\masm32rt.inc
2  cypher proto clt: ptr byte
3
4  .data
5  TXT byte "IVIN CBEGHTNY!",0
6  TAB byte "NOPQRSTUVWXYZ",
7          "ABCDEFGHIJKLM"
8
9  .code
10 start:
11     print    offset TXT,13,10
12     invoke   cypher, offset TXT
13     print    offset TXT,13,10,10
14     inkey
15     exit
16
17 cypher proc uses edi clt: ptr byte
18     mov      edi,clt
19 @@:  mov      al,[edi]
20     cmp      al,0
21     jz        done
22     call      rot13
23     stosb
24     jmp      @b
25 done: ret
26 cypher endp
27
28 rot13 proc
29     cmp al,'A'
30     jb  no
31     cmp al,'Z'
32     ja  no
33     mov esi, offset TAB
34     sub al,'A'
35     movzx eax,al
36     mov al,[eax+esi]
37 no:  ret
38 rot13 endp
39 end start
```

- [10] (a) Quantas vezes é invocada a sub-rotina `rot13`?

Resposta: A sub-rotina é executada tantas vezes quantas o número de caracteres do texto, pois o ciclo @@ só termina quando se encontra o terminador do texto (0). Neste caso, a sub-rotina é executada 14 vezes.

- [10] (b) O que acontece às letras minúsculas que possam aparecer no texto (variável `TXT`)?

Resposta: As letras minúsculas (de facto todo e qualquer carácter que não seja uma letra maiúscula) não são convertidas. Isso é garantido pelas instruções das linhas 29 a 32.

- [10] (c) Indique o que será escrito no monitor. Descreva a funcionalidade da rotina `cypher` e como ela é conseguida.

Resposta: No monitor aparecerá: IVIN CBEGHTNY!

VIVA PORTUGAL!

A sub-rotina percorre todo o texto substituindo cada letra maiúscula pela letra correspondente 13 caracteres mais à frente no alfabeto. Faz isso recorrendo à sub-rotina `rot13` que converte uma letra maiúscula de cada vez.

- (d) Assuma que imediatamente antes da execução da instrução da linha 29 o valor de `ESP` é `0018FF78h` e que o estado da pilha é o seguinte (valores em hexadecimal):

Endereço	Conteúdo
0018FF88	00403034
0018FF84	00401037
0018FF80	0012FF94
0018FF7C	00403000
0018FF78	0040107D

- [5] i. Indique o endereço de memória em que está codificada a instrução `stosb`.

Resposta: No topo da pilha está o endereço de retorno para onde o CPU irá quando executar o retorno da sub-rotina, logo o endereço pedido é `0040107D`.

- [5] ii. Indique a que corresponde o conteúdo do endereço de memória `0018FF7Ch`.

Resposta: O endereço referido corresponde ao endereço imediatamente a seguir ao topo da pilha. A última operação de escrita na pilha antes da chamada da sub-rotina `rot13` foi salvar o valor de `EDI` (...uses `EDI` ...) pelo que o conteúdo nesse endereço corresponde ao valor do registo `EDI`.

Sistemas de entrada/saída

[Nota: Para simplificar os cálculos assuma kB=10³ B, MB=10⁶ B, GB=10⁹ B e TB=10¹² B]

2. Um sistema equipado com um processador a 100 MHz recebe dados de uma rede sem fios a uma taxa *máxima* de 400 kB/s.

Considere duas alternativas para gerir a receção de dados:

Varrimento: Cada varrimento necessita de 200 ciclos.

Interrupções: A rotina de atendimento de interrupções necessita de 500 ciclos.

Qualquer dos métodos permite transferir 4 bytes/acesso. O sistema não deve perder dados em nenhuma situação.

- [10] (a) Para cada uma das alternativas, determine a percentagem máxima de tempo que o processador fica ocupado. Que método é melhor nestas circunstâncias?

Resposta: Número de acessos por segundo N :

$$N = \frac{400 \times 10^3}{4} = 10^5 \text{ acessos/s}$$

Número de ciclos gastos em varrimentos C_v :

$$C_v = 200 \times 10^5 = 2 \times 10^7$$

Percentagem de tempo (= percentagem de ciclos gastos em varrimentos):

$$P_v = \frac{2 \times 10^7}{100 \times 10^6} = \frac{2 \times 10^7}{10^8} = 0,2 = 20 \%$$

Para o atendimento por interrupções:

$$C_I = 500 \times 10^5 = 5 \times 10^7$$

$$P_I = \frac{5 \times 10^7}{100 \times 10^6} = \frac{5 \times 10^7}{10^8} = 0,5 = 50 \%$$

É melhor usar varrimento, porque leva a uma menor ocupação do processador.

- [10] (b) Suponha que o sistema *não* está permanentemente em comunicação, sendo a taxa *média* de transferência de dados 80 kB/s. Nesta situação, determine, para cada método, a percentagem de tempo que o processador fica ocupado. Qual o método a usar neste cenário?

Resposta: Varrimento: a situação não se altera, porque é preciso garantir o funcionamento correto à taxa máxima.

Interrupções: a sub-rotina de atendimento de interrupções só é executada se existirem dados a receber. Logo, interessa a ocupação média do CPU (desde que a ocupação máxima não ultrapasse 100%).

Assim: Número de acessos por segundo N_I :

$$N_I = \frac{80 \times 10^3}{4} = 0,2 \times 10^5 \text{ acessos/s}$$

$$C_I = 500 \times 0,2 \times 10^5 = 100 \times 10^5 = 10^7$$

$$P_I = \frac{10^7}{100 \times 10^6} = \frac{10^7}{10^8} = 0,1 = 10 \%$$

Neste cenário, é melhor utilizar o método de gestão por interrupções.

3. Os discos magnéticos Zspeed-Vx1 têm as seguintes características:

Capacidade	Preço	T. médio busca	Taxa de transferência	Vel. rotação
1 TB	100 €	4 ms	80 MB/s	10000 RPM

Pretende-se usar estes discos num sistema RAID com capacidade útil para 10 TB. As duas opções em consideração são (i) RAID-1 e (ii) RAID-5 com dois grupos de proteção. Em ambos os casos, os discos estão organizados em blocos de 4 kB e o controlador tem um tempo de *overhead* de 0,5 ms.

[10] (a) Determine o custo do sistema para as duas opções.

Resposta: RAID-1: Número de discos: $N = 2 \times \frac{10}{1} = 20$.

Custo inicial: $20 \times 100 = 2000\text{€}$

RAID-5: Número de discos: $N = 2 \times (5 + 1) = 12$.

Custo: $12 \times 100 = 1200\text{€}$

[10] (b) Para as duas opções, indique qual o cenário mais favorável para a escrita de um bloco. Determine o tempo correspondente.

Resposta: RAID-1: escrita simultânea em dois discos diferentes.

RAID-5: leitura simultânea de dois blocos (bloco a alterar e bloco de paridade) de dois discos diferentes, seguida de escrita simultânea de dois blocos. Tempo de acesso a um bloco:

$$t = 0,5 \text{ ms} + 4 \text{ ms} + \frac{1}{2} \frac{60}{10000} \text{ s} + \frac{4 \times 10^3}{80 \times 10^6} \text{ s}$$

$$t = (0,5 + 4 + 3 + \frac{4}{80}) \text{ ms} = (7,5 + 0,05) \text{ ms} = 7,55 \text{ ms}$$

Tempo para RAID-1: 7,55 ms; tempo para RAID-5: $2 \times 7,55 \text{ ms} = 15,1 \text{ ms}$.

Programação

4. Nas alíneas seguintes pretende-se modificar uma sequência com n elementos do tipo DWORD ($n > 2$).

- [20] (a) Escreva a sub-rotina **proc1** que altera a sequência através da seguinte transformação:
- cada elemento e_i é substituído pelo valor da expressão $(e_{i-1} + 2 \times e_i + e_{i+1})/4$, onde e_{i-1} e e_{i+1} são, respetivamente, o elemento anterior e o seguinte (assuma que o resultado de cada operação é representável em 32 bits);
 - o primeiro e último elementos, e_0 e e_{n-1} , mantêm-se inalterados.

Por exemplo, a sequência $\{2, 4, 2, 0, 15, 3, 3\}$ é transformada em $\{2, 3, 2, 4, 8, 6, 3\}$.

O protótipo da sub-rotina é:

proc1 PROTO seq:PTR DWORD, n:DWORD

Resposta:

```
proc1 PROC uses ebx esi seq:PTR DWORD,
                    n:DWORD

    mov     esi, seq
    mov     ecx, n
    sub     ecx, 2
    mov     edx, [esi]
    @@: add  esi, 4
    mov     ebx, [esi]
    shl     ebx, 1
    add     ebx, [esi-4]
    add     ebx, [esi+4]

    shr     ebx, 2
    mov     [esi-4], edx
    mov     edx, ebx
    loop    @B
    mov     [esi], edx
    ret

proc1 ENDP
```

- [20] (b) Escreva a sub-rotina **proc2** que substitui cada elemento e_i da sequência pela soma dos elementos seguintes (e_j , com $j = i + 1, \dots, n - 1$) maiores que o elemento atual, desde que essa soma seja superior a zero. Por exemplo, a sequência $\{3, 8, 1, 9, 1, 5\}$ é transformada em $\{22, 9, 14, 9, 5, 5\}$.

Considere o seguinte protótipo:

proc2 PROTO seq:PTR DWORD, n:DWORD

Resposta:

```
proc2 PROC uses ebx esi seq:PTR DWORD,
                    n:DWORD

    mov     esi, seq
    mov     ecx, n
    dec     ecx
    mov     eax, [esi]
    xor     edx, edx
    mov     ebx, 1
    c2: cmp  [esi+4*ebx], eax
    jbe     nao
    add     edx, [esi+4*ebx]
    nao: cmp ebx, ecx
    je      sai

    inc     ebx
    jmp     c2
    sai: cmp edx, 0
    je      ig0
    mov     [esi], edx
    ig0: add esi, 4
    loop    c1
    fim: ret

proc2 ENDP
```

5. Implemente duas sub-rotinas destinadas a calcular o valor de uma expressão de acordo com a especificação e protótipos seguintes.

[20] (a) A sub-rotina **pertence** determina se um número real **pertence** ao intervalo $[-k, k]$, com $k \in \mathbb{R}^+$. A função retorna 1 em caso afirmativo ou 0 no caso contrário.

O protótipo da sub-rotina é:

pertence PROT x:REAL8, k:REAL8

Resposta:

```
pertence PROC x:REAL8, k:REAL8
    xor edx, edx
    fld x
    fabs
    fcomp k
    fstsw ax
    sahf
    ja nao
    inc edx
nao: mov eax, edx
    ret
pertence endp
```

[20] (b) A sub-rotina **F_x** calcula a função $f(x)$, para qualquer valor de x , utilizando a sub-rotina **pertence** para determinar o intervalo a que x pertence.

$$f(x) = \begin{cases} -\sqrt{1 - \cos(x^2 + 1)} & \text{se } |x| > 1 \\ \frac{1}{\sqrt{1 - \cos(x^2 + 1)}} & \text{se } |x| \leq 1 \end{cases}$$

O protótipo da sub-rotina é:

F_x PROT x:REAL8

Resposta:

```
.data
lim REAL8 1.0

.code
F_x PROC x:REAL8
    fld x
    fmul x
    fld1
    fadd
    fcos
    fld1
    fsubr
    fsqrt
    invoke pertence, x, lim
    cmp eax, 1
    je inv
    fchs
    jmp sai
inv: fld1
    fdivr
sai: ret
F_x ENDP
```

Escolha múltipla

6. Para as alíneas seguintes, indique a única resposta correta.

[8] (a) Com que valores ficam os indicadores (*flags*) após a execução do fragmento seguinte:

```
mov    al, 6
cmp    al, 5
```

- A. CF=0, ZF=0, SF=0
- B. CF=1, ZF=0, SF=0
- C. CF=1, ZF=1, SF=0
- D. CF=1, ZF=1, SF=1

Resposta: A.

- [8] (b) Quais são os valores dos registos AX e DX após a execução da sequência de instruções seguinte:

```
mov    AX, 6B49h
mov    DX, 0095h
shl    AX, 1
rc1    DX, 1
```

- A. DX = 0148h, AX = C691h
- B. DX = 012Ah, AX = C9A2h
- C. DX = **012Ah**, AX = **D692h**
- D. DX = 024Bh, AX = D692h

Resposta: C

- [8] (c) Qual das seguintes afirmações sobre variáveis locais é verdadeira?
- A. **Fazem uso eficiente de memória porque o espaço que ocupam é libertado no fim da execução da função.**
 - B. Podem ser acedidas a partir de qualquer ponto do mesmo ficheiro de código.
 - C. São geralmente criadas no segmento de dados.
 - D. O nome da mesma variável local não pode ser usado em duas sub-rotinas diferentes.

Resposta: A.

- [8] (d) Que instrução faz com que os registos ESI e EDI sejam decrementados pela instrução MOVSB ?
- A. STC B. REP C. CLD **D. STD**

Resposta: D.

- [8] (e) Qual das seguintes afirmações sobre barramentos paralelos é verdadeira?
- A. Um barramento paralelo só pode ser usado para ligações ponto-a-ponto entre dois componentes.

- B. Um barramento paralelo é um canal de comunicação partilhado.**
- C. Quantos mais periféricos estão ligados a um barramento, mais eficiente é a comunicação entre eles.
- D. Periféricos ligados ao CPU por um barramento paralelo devem ser sempre geridos por varrimento (*polling*).

Resposta: B.

Fim do enunciado.