

Este exame tem 7 questões, num total de 200 pontos.

**Fundamente todas as respostas.**

## Análise de código

1. Considere o exemplo seguinte, escrito na linguagem *assembly* da família IA-32 da Intel, constituído por um programa principal e duas sub-rotinas, *gkey* e *unic*. A sub-rotina *unic* processa o valor de AL, o valor de ECX e a cadeia de caracteres apontada por EDI, devolvendo o resultado desse processamento através da *flag* ZF; a sub-rotina *gkey* processa uma cadeia de caracteres, recorrendo a *unic* para o processamento de cada elemento.

```
1  include  masm32rt.inc           26      call    unic
2                                     27      jz      @b
3  gkey proto clk: ptr byte        28      stosb
4                                     29      inc     ecx
5  .data                            30      jmp     @b
6  KEY byte "Hello my World",0    31  done:
7                                     32      stosb
8  .code                            33      ret
9  start:                          34  gkey endp
10     print  offset KEY,13,10      35
11     invoke gkey, offset KEY      36
12     print  offset KEY,13,10      37  unic proc
13     inkey                               38      cmp     al,' '
14     exit                               39      jz      @f
15                                     40      push    ecx
16  gkey proc uses esi edi          41      push    edi
17      clk: ptr byte              42      mov     edi, offset KEY
18                                     43      repnz   scasb
19      mov     edi,clk             44      pop     edi
20      mov     esi,edi             45      pop     ecx
21      mov     ecx,0              46  @@: ret
22      cld                               47  unic endp
23  @@: lodsb                       48
24      cmp     al,0               49  end start
25      jz      done
```

- [10] (a) Qual é o objetivo da sub-rotina *unic*?

**Resposta:** A sub-rotina *unic* assinala na *flag* ZF (ZF=1) o facto de o carácter contido em AL ter o código ASCII do espaço (`cmp al,' '`) ou estar já presente nas primeiras ECX posições da *string* apontada por EDI (`repnz scasb`).

- [10] (b) Qual é o objetivo da sub-rotina **gkey**? Quantas vezes é **unic** chamada por **gkey**? O que fica escrito no monitor após a execução do programa?

**Resposta:** A sub-rotina **gkey** percorre todos os elementos da cadeia de caracteres KEY eliminando espaços e elementos repetidos. No fim move também o terminador (0) para a sua nova posição. A sub-rotina **unic** é chamada dentro de um ciclo que termina quando o elemento a ser analisado tiver o código 0. No caso presente, **unic** é chamada 14 vezes e no monitor irá aparecer:

```

Hello my World
HelomyWrd
Press any key to continue ...

```

- [10] (c) Apresente o prólogo da sub-rotina **gkey**.

**Resposta:** O prólogo gerado pelo assembler para a sub-rotina **gkey**, será:

```

push  ebp      ; prepara acesso aos
mov   ebp,esp  ; parâmetros na pilha
push  esi      ; preserva registos
push  edi      ; indicados com "uses"

```

- [10] (d) Imediatamente antes da execução da instrução **inc ecx** (linha 29) o valor de ESP é 0018FF78h e o estado da pilha é (valores em hexadecimal):

Endereço	Conteúdo
0018FF84	0040101E
0018FF80	0012FF94
0018FF7C	00000000
0018FF78	00000000

Indique o que representa cada uma das posições da pilha.

**Resposta:** No topo da pilha encontram-se, por esta ordem: os registos EDI e ESI (**uses ...**), o registo EBP e o endereço de retorno de **gkey**.

## Sistemas de entrada/saída

[Nota: Para simplificar os cálculos assuma kB=10<sup>3</sup> B, MB=10<sup>6</sup> B, GB=10<sup>9</sup> B e TB=10<sup>12</sup> B]

2. Um computador recebe mensagens que deve cifrar e guardar em disco. O CPU trabalha a 2 GHz e o disco magnético tem as seguintes características:

Tempo médio busca	Taxa de transferência	Vel. rotação	Controlador
4 ms	64 MB/s	15000 RPM	0 s

Cada mensagem cifrada tem 64 kB. Para cifrar uma mensagem, o CPU necessita de  $5 \times 10^6$  ciclos. A gravação de uma mensagem requer  $1 \times 10^6$  ciclos de CPU. Cada mensagem é gravada em setores sucessivos. Para cada mensagem, o sistema cifra-a e guarda-a em disco, após o que passa ao tratamento da mensagem seguinte.

- [10] (a) Determine o tempo necessário para tratar (cifrar e gravar) uma mensagem. Quantas mensagens podem ser tratadas por segundo? Qual a percentagem máxima de tempo que o CPU está ocupado com esta tarefa?

**Resposta:**

$$T_{\text{msg}} = T_{\text{CPU}} + T_{\text{disco}}$$

$$T_{\text{CPU}} = \frac{5 \times 10^6 + 1 \times 10^6}{2 \times 10^9} = 3 \text{ ms}$$

$$T_{\text{disco}} = 4 \text{ ms} + \frac{1}{2} \frac{60}{15000} + \frac{64 \text{ kB}}{64 \text{ MB/s}} = 4 \text{ ms} + 2 \text{ ms} + 1 \text{ ms} = 7 \text{ ms}$$

Portanto:

$$T_{\text{msg}} = 10 \text{ ms}$$

Número de mensagens por segundo (débito) D:

$$D = \frac{1000 \text{ ms}}{10 \text{ ms}} = 100$$

A taxa de utilização do CPU é:

$$\frac{T_{\text{CPU}}}{T_{\text{msg}}} = \frac{3 \text{ ms}}{10 \text{ ms}} = 30 \%$$

- [10] (b) Pretende-se implementar um sistema capaz de tratar 125 mensagens por segundo usando o mesmo CPU, mas com um disco magnético diferente. Determine o tempo máximo de gravação que o novo disco deve garantir. Caso não tenha resolvido a alínea anterior, assuma que são necessários 12 ms para tratar cada mensagem.

**Resposta:** Neste cenário, cada mensagem deve ser tratada em:

$$T_{\text{msg}} = \frac{1000 \text{ ms}}{125} = 8 \text{ ms}$$

Portanto, o tempo disponível para gravação é:

$$T_{\text{disco}} = 8 \text{ ms} - 3 \text{ ms} = 5 \text{ ms}$$

3. Considere um sistema RAID-5 com 2 grupos de proteção e 4 discos por grupo:

*Grupo A:* discos A1, A2, A3 e A4.      *Grupo B:* discos B1, B2, B3 e B4.

Cada disco tem 1,5 TB de capacidade.

[10] (a) Qual é a capacidade útil do sistema?

**Resposta:** Cada grupo tem 3 discos de capacidade útil. Como existem 2 grupos, a capacidade útil total é  $2 \times (3 \times 1,5 \text{ TB}) = 9 \text{ TB}$ .

[10] (b) Para cada um dos cenários indicados a seguir, determine se existe perda de dados.

- i) falha do disco A2;
- ii) falha dos discos A2 e A4;
- iii) falha dos discos A2 e B3;
- iv) falha dos discos A2 e B2.

**Resposta:** Num sistema RAID-5 apenas ocorre perda de dados quando falha mais que um disco do mesmo grupo de proteção, logo apenas ocorre perda de dados no cenário ii).

# Programação

4. O triângulo de Pascal é um triângulo numérico formado por binômios de Newton  $\binom{n}{k}$ , onde  $n$  representa a linha e  $k$  representa a coluna, iniciando-se esta contagem em 0. O binômio de Newton é definido por

$$\binom{n}{k} = \frac{n!}{k! \times (n-k)!}$$

Mostra-se a seguir as 6 primeiras linhas do triângulo de Pascal, obtidas com  $0 \leq n \leq 5$  e, para cada  $n$ ,  $0 \leq k \leq n$ .

$$\begin{array}{cccccccc}
n = 0 : & & & & & & & 1 \\
n = 1 : & & & & 1 & & 1 & \\
n = 2 : & & & 1 & & 2 & & 1 \\
n = 3 : & & 1 & & 3 & & 3 & 1 \\
n = 4 : & 1 & & 4 & & 6 & & 4 & 1 \\
n = 5 : & 1 & 5 & & 10 & & 10 & 5 & 1 \\
n = 6 : & & \cdots & & & & \cdots & & 
\end{array}$$

Pretende-se gerar uma sequência de números naturais correspondente às primeiras  $N$  linhas do triângulo de Pascal.

Por exemplo, as 5 primeiras originam a sequência  $\{1, 1, 1, 1, 2, 1, 1, 3, 3, 1, 1, 4, 6, 4, 1\}$ .

Considere a resolução do problema decomposta em três sub-rotinas, em que se assume que o resultado de cada operação é representável em 32 bits.

[20] (a) A função fatorial é definida por

$$\begin{aligned} n! &= n \times (n-1) \times \cdots \times 1, & n > 0 \\ 0! &= 1 \end{aligned}$$

Escreva a sub-rotina `fact` que calcula o fatorial de um número  $n$ , considerando o seguinte protótipo:

`fact` PROTO `n:DWORD`

**Resposta:**

```
fact PROC n:DWORD
    mov     eax, 1
    mov     ecx, n
    jecxz   fim
@@:  mul     ecx
    loop    @B
fim:  ret
fact ENDP
```

- [20] (b) Escreva a sub-rotina `tPascal` que gera a sequência de elementos das primeiras `nlin` linhas do triângulo de Pascal. O seu protótipo é:

`tPascal` PROTO `nlin:DWORD, seq:PTR DWORD`

Na resolução utilize a sub-rotina `bNewton`, já existente, que calcula o valor do binómio de Newton e tem o protótipo:

`bNewton` PROTO `n:DWORD, k:DWORD`

**Resposta:**

```
tPascal PROC uses ebx esi edi nlin:DWORD, seq:PTR DWORD
    mov     ebx, seq
    mov     esi, 0          ; Linha
lin:  mov     edi, 0          ; Coluna
    cmp     esi, nlin
    je      fim
col:  invoke  bNewton, esi, edi ; (n k)
    mov     [ebx], eax      ; Guarda elemento
    add     ebx, 4
    cmp     edi, esi        ; Última coluna?
    je      ultC
    inc     edi
    jmp     col
ultC: inc     esi
    jmp     lin
fim:  ret
tPascal ENDP
```

- [20] 5. A lei de Amdahl pode ser expressa pela função

$$M(p, s) = \frac{1}{1 - p + \frac{p}{s}}$$

Escreva a sub-rotina `amdahl` que calcula o valor de  $M(p, s)$  considerando o protótipo:

`amdahl` PROTO `p:REAL8, s:REAL8`

**Resposta:**

```

amdahl PROC p:REAL8, s:REAL8
    fld1
    fsub p
    fld p
    fdiv s
    fadd
    fld1
    fdivr
    ret
amdahl ENDP

```

- [20] 6. Suponha que necessita do valor de  $\pi$  e que não existe a instrução FLDPI. A expressão seguinte permite calcular uma aproximação do valor de  $\pi$ :

$$\pi = \sum_{i=0}^{+\infty} (-1)^i \frac{4}{2i+1}$$

Escreva a sub-rotina `valpi`, que calcula o valor aproximado de  $\pi$  considerando  $n$  parcelas ( $i = 0, \dots, n-1$ ). O protótipo da sub-rotina é:

`valpi PROTO n:DWORD`

**Resposta:**

```

        .data                                fld  quatro
quatro REAL8 4.0                             fdivr
        .code                                fld  soma
valpi PROC uses ebx n:DWORD                  xor  ebx, 1    ; toggle
    local i:real8                             jz   par      ; soma ou subtrai?
    local soma:real8                         fsubr
                                                jmp  sto
    mov  ecx, n                               par:fadd
    mov  ebx, 1                             sto:fstp soma
    fldz                                    fld1
    fst  i                                  fadd i
    fst  soma                              fst  i
@@: fadd st, st                             loop @B
    fld1                                    fld  soma
    fadd                                     ret
valpi ENDP

```

## Escolha múltipla

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

- [8] (a) Indique a instrução ilegal/inexistente.

A. `mov eax, esp`

- B. `cmps`
- C. `fld eax`
- D. `inc [edx+2*esi+640]`

**Resposta:** C. (Nota: Também foram aceites as respostas B ou D, devido a gralhas no enunciado.)

- [8] (b) Qual das seguintes afirmações sobre gestão de periféricos é falsa?
- A. Varrimento é uma técnica simples, mas que não deve ser aplicada a periféricos de alto débito.
  - B. A utilização de interrupções obriga o CPU a verificar periodicamente se um periférico está disponível.**
  - C. Acesso direto a memória requer a utilização de interrupções.
  - D. A utilização de DMA permite efetuar transferências enquanto o CPU executa outras tarefas.

**Resposta:** B.

- [8] (c) Em que posição prossegue a execução do fragmento seguinte?

```
.data
var1 BYTE 10
var2 BYTE 20
.code
mov esi,OFFSET var2
mov edi,OFFSET var1
cmpsb
ja L1
jb L2
je L3
L4:
```

A. L1  
B. L2  
C. L3  
D. L4

**Resposta:** A.

- [8] (d) Qual é o valor de ESI após execução do seguinte fragmento:

```
        .data
array   SWORD 8,2,3,5,-4,6,0,4
        .code
        mov esi,0
        mov ecx,LENGTHOF array
L1:     mov ax,array[esi]
        add esi,TYPE array
        cmp ax,0
        loopne L1
```

A. 6   B. 7   C. 12   **D. 14**

**Resposta: D.**

- [8] (e) Supondo que inicialmente AX=EB49h, qual dos seguintes fragmentos faz com que esse registo assumo o valor AX=FAD2h.

A.   shr ax,2      B.   sar ax, 1      C.   sar ax,1      D.   shl ax,1  
     shl ax,1      rcr ax, 1      ror ax,1      rcr ax,1

**Resposta: B.**

Fim do enunciado.