

LABORATÓRIO DE LINGUAGEM DE MONTAGEM

PROGRAMAS RESIDENTES NA MEMÓRIA

Capítulo 15 do livro-texto, págs. 322 a 330

✍ **NORMALMENTE, QUANDO UM PROGRAMA TERMINA, O ESPAÇO DE MEMÓRIA OCUPADO É UTILIZADO PELO DOS PARA CARREGAR OUTROS PROGRAMAS.**

```
...  
MOV    AH,4Ch    ; função DOS para término normal  
INT    21h       ; usando a interrupção DOS 21h  
END
```

✍ **PROGRAMAS TSR – *TERMINATE AND STAY RESIDENT*: SÃO PROGRAMAS QUE AO TERMINAR NÃO LIBERAM A MEMÓRIA OCUPADA.**

```
...  
; preparação inicial:  
; DS:DX = endereço do byte a partir do qual se inicia  
; o programa que deve permanecer residente  
; DS = segmento  
; DX = offset  
;  
LEA    DX, <nome de um rótulo>  
INT    27h       ; função DOS para TSR  
END
```

- ✍ **PROGRAMAS RESIDENTES NORMALMENTE SÃO .COM.**
 - ✍ POSSUEM UM ÚNICO SEGMENTO;
 - ✍ TÊM TAMANHO REDUZIDO, ECONOMIZANDO MEMÓRIA.

- ✍ **UMA VEZ TERMINADOS, OS PROGRAMAS RESIDENTES NÃO ESTÃO ATIVOS. PODENDO SER ATIVADOS POR:**
 - ✍ UMA SEQUÊNCIA DE TECLAS;
 - ✍ PELO TIMER.

- ✍ **ALTERNATIVAS DE ATIVAÇÃO:**
 - ✍ INTERRUPÇÃO 09h - TECLADO;
 - ✍ INTERRUPÇÃO 1Ch - TIMER TICK, ASSOCIADA À INT 08h .

- ✍ **UMA VANTAGEM DOS PROGRAMAS RESIDENTES É QUE PODEM SER ATIVADOS ENQUANTO OUTRO PROGRAMA ESTIVER RODANDO.**

- ✍ **ESCREVER UM PROGRAMA RESIDENTE NÃO É UMA TAREFA SIMPLES.**
 - ? SE HOVER OUTROS TSR's , ELE PODE NÃO FUNCIONAR CORRETAMENTE.
 - ? A ORDEM DE **LINKAGEM** INFLUI NO FUNCIONAMENTO.
 - ? **TESTE SEUS TSR's EM MODO DOS.**

☞ ALGUMAS INTERRUPÇÕES

INT 27h *Terminate and stay resident*

Preparação: DS:DX = endereço do byte a partir do qual se inicia o programa que deve permanecer residente.

Retorno: nenhum

FUNÇÃO 31h INT 21h *Terminate process and remain resident*

Preparação: AH = 31h

 AL = código de retorno (pesquisar !)

 DX = tamanho de memória em parágrafos (pacotes de 16 bytes)

Retorno: nenhum

FUNÇÃO 02h INT 16h *Get keyboard flags - BIOS*

Preparação: AH = 02h

Retorno: AL = FLAGS de tecla

BIT	VALOR	SIGNIFICADO DE TECLA
0	1	Insert ON
1	1	Caps Lock ON
2	1	Num Lock ON
3	1	Scroll Lock ON
4	1	Tecla ALT pressionada
5	1	Tecla CTRL pressionada
6	1	Tecla LEFT SHIFT pressionada
7	1	Tecla RIGHT SHIFT pressionada

Obs: Se o valor for “0”, considere o significado oposto.

OS FLAGS DE TECLA (KEY FLAGS) ESTÃO NA ÁREA DE MEMÓRIA RESERVADA PARA BIOS, ENDEREÇO 0000:0417H

✍ **EXEMPLO: COLOCANDO O PROGRAMA COMO TSR**

```
TITLE SET_TSR
;
EXTRN  MAIN:NEAR,SETUP_INT:NEAR
EXTRN  NEW_VEC:WORD,OLD_VEC:DWORD
PUBLIC INITIALIZE
C_SEG  SEGMENT      PUBLIC
        ASSUME CS:C_SEG
;
INITIALIZE  PROC
;set up do vetor de interrupção
    MOV  NEW_VEC,OFFSET MAIN ; obtém offset do endereço
    MOV  NEW_VEC+2,CS        ; e segmento
    LEA  DI,OLD_VEC
    LEA  SI,NEW_VEC
    MOV  AL,09H              ; interrupção 09h de teclado
    CALL SETUP_INT          ; subrotina que troca vetores
;sai para o dos
    LEA  DX,INITIALIZE
    INT  27H                 ; termina e fica residente
INITIALIZE  ENDP
        END
;...
SETUP_INT  PROC
;salva vetor velho e seta novo vetor
    MOV  AH,35H
    INT  21H
    MOV  [DI],BX             ; salva offset
    MOV  [DI+2],ES          ; salva segmento
;setup vetor novo
    MOV  DX,[SI]            ; modifica offset
    PUSH DS                 ; não se pode perder o DS
    MOV  DS,[SI+2]          ; e segmento do vetor
    MOV  AH,25H
    INT  21H
    POP  DS                 ; restaura o DS
    RET
SETUP_INT  ENDP
```

Exercício - Fazer seu programa de mostrar o horário como residente.

☞ SUGESTÃO PARA ORGANIZAR UM PROGRAMA RESIDENTE

Programa no formato .COM

MÓDULO PRINCIPAL

Escrito como rotina de interrupção, para resolver um determinado problema, podendo utilizar outras subrotinas já existentes

```
Início do programa  
START: JMP INITIALIZE  
...  
DADOS  
...  
MAIN PROC  
salva registradores  
...  
produz os efeitos desejados (função)  
chama subrotinas  
...  
restaura registradores  
IRET  
END START
```

MÓDULO(S) DE SUBROTINAS

Composto por subrotinas novas ou disponíveis numa biblioteca de **.OBJ**. Deve obrigatoriamente conter a subrotina **SETUP_INT**

```
Definição de subrotina(s)  
...  
SETUP_INT PROC  
...  
usa alguma INT (09h ou 1Ch)  
e define troca de vetores  
...  
SETUP_INT ENDP  
END
```

MÓDULO DE INICIALIZAÇÃO

Deve ser o último módulo

```
INITIALIZE PROC  
...  
define MAIN como rotina de interrupção  
chama SETUP_INT  
realiza a troca de vetores  
termina e fica residente  
...  
INITIALIZE ENDP  
END
```

ANEXO – INTERRUPÇÕES DO BIOS E DOS - COMPLEMENTO

☞ SUGESTÃO PARA A MUDANÇA DO VETOR DE INTERRUPÇÃO

MÓDULO PRINCIPAL

Escrito para resolver um determinado problema, podendo utilizar subrotinas escritas especialmente para este programa ou já existentes

```
Início do programa
def STACK
...
def DADOS
...
def CODE
MAIN PROC
...
preparação inicial: define INT a ser usada
...
chama SETUP_INT para trocar vetor
...
executa a função do programa
chama subrotinas, se necessário
...
antes de terminar, chama SETUP_INT
para restaurar vetor velho
terminar programa
MAIN ENDP
END MAIN
```

MÓDULO(S) DE SUBROTINAS

Composto por subrotinas novas ou disponíveis numa biblioteca de **.OBJ**. Deve obrigatoriamente conter a subrotina **SETUP_INT**

```
Definição de subrotina(s)
...
SETUP_INT PROC
...
usa alguma INT (09h ou 1Ch)
e define troca de vetores
...
SETUP_INT ENDP
END
```