

Mais programas no TASM

Nesta seção forneceremos a você vários exemplos de programas a serem montados fazendo uso do TASM da Borland.

Para montá-los, siga os seguintes passos:

Primeiro passo

Montar o programa

Por exemplo:

```
C:\>tasm one.asm  
Turbo Assembler V2.0 Copyright (c) 1988, 1990 Borland International
```

```
Assembling file:   one.asm  
Error messages:   None  
Warning messages: None  
Passes:           1  
Remaining memory: 471k
```

```
C:\>
```

Isto criará um programa objeto com o mesmo nome do fonte, neste caso: one.obj

Segundo passo

Criar o programa executável

Por exemplo:

```
C:\>tlink one.obj  
Turbo Link Version 3.0 Copyright (c) 1987, 1990 Borland International
```

```
C:\>
```

Isto cria o programa executável com o mesmo nome do objeto e com extensão diferente, one.exe

Terceiro passo

Rodar o programa executável. Basta digitar o nome do programa criado.

Exemplos de Programas Assembly

Primeiro exemplo

```
;nome do programa: one.asm
;
.model small
.stack
.code
inicio:          ;Label p/ formar ender. de entrada do programa
    mov AH,1h    ;Função 1 do DOS
    int 21h      ;lê caracter e retorna código ASCII ao registrador AL
    mov DL,AL    ;move o código ASCII para o registrador DL
    sub DL,30h   ;subtrai de 30h para converter a um dígito de 0 a 9
    cmp DL,9h    ;compara se o dígito está entre 0 e 9
    jle digit1   ;se verdadeiro obtém o primeiro número (4 bits)
    sub DL,7h    ;se falso, subtrai de 7h para converter a uma letra A-F
digit1:
    mov CL,4h    ;prepara para multiplicar por 16
    shl DL,CL    ;multiplica para converter dentro dos 4 bits mais altos
    int 21h      ;obtem o próximo caracter
    sub AL,30h   ;repete a operação de conversão
    cmp AL,9h    ;compara o valor 9h com o conteúdo do registrador AL
    jle digit2   ;se verdadeiro, obtém o segundo dígito
    sub AL,7h    ;se falso, subtrai de 7h
digit2:
    add DL,AL    ;adiciona o segundo dígito
    mov AH,4Ch   ;função 4Ch do DOS (exit)
    int 21h      ;interrupção 21h
    End inicio   ;finaliza o programa
```

Este programa lê dois caracteres e os imprime na tela

Segundo exemplo

```
;nome do programa: two.asm
.model small
.stack
.code
PRINT_A_J PROC
    mov DL,'A'   ;move o character A para o registrador DL
    mov CX,10    ;move o valor decimal 10 para o registrador CX
                ;este valor é usado para fazer laço com 10 interações
PRINT_LOOP:
    call WRITE_CHAR ;imprime o caracter em DL
    inc DL         ;incrementa o valor do registrador DL
    loop PRINT_LOOP ;laço para imprimir 10 caracteres
    mov AH,4Ch    ;função 4Ch, para sair ao DOS
    int 21h       ;interrupção 21h
PRINT_A_J ENDP   ;finaliza o procedimento

WRITE_CHAR PROC
    mov AH,2h    ;função 2h, imprime caracter
    int 21h      ;imprime o caracter que está em DL
    ret         ;retorna o controle ao procedimento que chamou
WRITE_CHAR ENDP ;finaliza o procedimento
```

```
END PRINT_A_J ;finaliza o programa
Este programa mostra os caracteres ABCDEFGHIJ na tela.
```

Terceiro exemplo

```
;nome do programa: three.asm
.model small
.STACK
.code

TEST_WRITE_HEX PROC
    mov DL,3Fh ;move o valor 3Fh para o registrador DL
    call WRITE_HEX ;chama a sub-rotina
    mov AH,4Ch ;função 4Ch
    int 21h ;retorna o controle ao DOS
TEST_WRITE_HEX ENDP ;finaliza o procedimento

        PUBLIC WRITE_HEX
;.....;
;Este procedimento converte para hexadecimal o byte ;
;armazenado no registrador DL e mostra o dígito ;
;Use:WRITE_HEX_DIGIT ;
;.....;

WRITE_HEX PROC
    push CX ;coloca na pilha o valor do registrador CX
    push DX ;coloca na pilha o valor do registrador DX
    mov DH,DL ;move o valor de DL para o registrador DH
    mov CX,4 ;move o valor 4 para o registrador CX
    shr DL,CL
    call WRITE_HEX_DIGIT ;mostra na tela o primeiro número hexadecimal
    mov DL,DH ;move o valor de DH para o registrador DL
    and DL,0Fh
    call WRITE_HEX_DIGIT ;mostra na tela o segundo número hexadecimal
    pop DX ;retira da pilha o valor do registrador DX
    pop CX ;retira da pilha o valor do registrador CX
    ret ;retorna o controle ao procedimento que chamou
WRITE_HEX ENDP

        PUBLIC WRITE_HEX_DIGIT
;.....;
;Este procedimento converte os 4 bits mais baixos do registrador DL ;
;para um número hexadecimal e o mostra na tela do computador ;
;Use: WRITE_CHAR ;
;.....;

WRITE_HEX_DIGIT PROC
    push DX ;coloca na pilha o valor de DX
    cmp DL,10 ;compara se número de bits é menor que 10
    jae HEX_LETTER ;se não, salta para HEX_LETTER
    add DL,"0" ;se sim, converte para número
    jmp Short WRITE_DIGIT ;escreve o caracter
HEX_LETTER:
    add DL,"A"-10 ;converte um caracter para hexadecimal
```

```

WRITE_DIGIT:
    call WRITE_CHAR      ;imprime o caracter na tela
    pop  DX              ;Retorna o valor inicial do registrador DX
                        ;para o registrador DL
    ret                 ;Retorna o controle ao procedimento que chamou
WRITE_HEX_DIGIT  ENDP

```

```

        PUBLIC WRITE_CHAR
;.....;
;Este procedimento imprime um caracter na tela usando o D.O.S. ;
;.....;

```

```

WRITE_CHAR  PROC
    push  AX      ;coloca na pilha o valor do registrarador AX
    mov   AH,2    ;função 2h
    int  21h     ;interrupção 21h
    pop  AX      ;extrai da pilha o valor de AX
    ret         ;retorna o controle ao procedimento que chamou
WRITE_CHAR  ENDP

```

```

END TEST_WRITE_HEX ;Finaliza o programa

```

Quarto exemplo

```

;nome do programa: four.asm
.model small
.stack
.code

```

```

TEST_WRITE_DECIMAL  PROC
    mov  DX,12345      ;move valor decimal 12345 para registrador DX
    call WRITE_DECIMAL ;chama o procedimento
    mov  AH,4CH       ;função 4Ch
    int  21h          ;interrupção 21h
TEST_WRITE_DECIMAL ENDP ;finaliza o procedimento

```

```

        PUBLIC WRITE_DECIMAL
;.....;
;Este procedimento escreve um número de 16 bits como um número ;
;sem sinal em notação decimal ;
;Use: WRITE_HEX_DIGIT ;
;.....;

```

```

WRITE_DECIMAL  PROC
    PUSH AX      ;Põe na pilha o valor do registrador AX
    PUSH CX      ;Põe na pilha o valor do registrador CX
    PUSH DX      ;Põe na pilha o valor do registrador DX
    PUSH SI      ;Põe na pilha o valor do registrador SI
    MOV AX,DX    ;move o valor do registrador DX para AX
    MOV SI,10    ;move o valor 10 para o registrador SI
    XOR CX,CX    ;zera o registrador CX
NON_ZERO:
    XOR DX,DX    ;zera o registrador CX
    DIV SI       ;divisão entre SI

```

```

    PUSH DX    ;Põe na pilha o valor do registrador DX
    INC CX     ;incrementa CX
    OR AX,AX   ;não zero
    JNE NON_ZERO ;salta para NON_ZERO
WRITE_DIGIT_LOOP:
    POP DX     ;retorna o valor em modo reverso
    CALL WRITE_HEX_DIGIT ;chama o procedimento
    LOOP WRITE_DIGIT_LOOP ;loop
END_DECIMAL:
    POP SI     ;retira da pilha o valor do registrador SI
    POP DX     ;retira da pilha o valor do registrador DX
    POP CX     ;retira da pilha o valor do registrador CX
    POP AX     ;retira da pilha o valor do registrador AX
    RET       ;retorna o controle ao procedimento que chamou
WRITE_DECIMAL ENDP ;finaliza o procedimento

    PUBLIC WRITE_HEX_DIGIT
;.....;
;
;Este procedimento converte os 4 bits mais baixos do registrador DL ;
;num número hexadecimal e os imprime ;
;Use: WRITE_CHAR ;
;.....;

WRITE_HEX_DIGIT PROC
    PUSH DX     ;Põe na pilha o valor do registrador DX
    CMP DL,10   ;compara o valor 10 com o valor do registrador DL
    JAE HEX_LETTER ;se não, salta para HEX_LETTER
    ADD DL,"0"  ;se é, converte em dígito numérico
    JMP Short WRITE_DIGIT ;escreve o caracter
HEX_LETTER:
    ADD DL,"A"-10 ;converte um caracter para um número hexadecimal
WRITE_DIGIT:
    CALL WRITE_CHAR ;mostra o caracter na tela
    POP DX         ;retorna o valor inicial para o registrador DL
    RET           ;retorna o controle ao procedimento que chamou
WRITE_HEX_DIGIT ENDP

    PUBLIC WRITE_CHAR
;.....;
;Este procedimento imprime um caracter na tela usando uma função D.O.S.;
;.....;

WRITE_CHAR PROC
    PUSH AX     ;Põe na pilha o valor do registrador AX
    MOV AH,2h  ;Função 2h
    INT 21h    ;Interrupção 21h
    POP AX     ;Retira da pilha o valor inicial do registrador AX
    RET       ;Retorna o controle ao procedimento que chamou
WRITE_CHAR ENDP

    END TEST_WRITE_DECIMAL ;finaliza o programa

```

Este programa mostra na tela os números 12345

Quinto exemplo

```
;nome do programa: five.asm
.model small
.stack
.code

PRINT_ASCII      PROC
    MOV DL,00h    ;move o valor 00h para o registrador DL
    MOV CX,255    ;move o valor decimal 255 para o registrador CX
                  ;usado para fazer um laço com 255 interações

PRINT_LOOP:
    CALL WRITE_CHAR ;Chama o procedimento que imprime
    INC DL          ;Incrementa o valor do registrador DL
    LOOP PRINT_LOOP ;Loop para imprimir 10 caracteres
    MOV AH,4Ch      ;Função 4Ch
    INT 21h         ;Interrupção 21h
PRINT_ASCII      ENDP ;Finaliza o procedimento

WRITE_CHAR       PROC
    MOV AH,2h      ;Função 2h para imprimir um caracter
    INT 21h        ;Imprime o caracter que está em DL
    RET            ;Retorna o controle ao procedimento que chamou
WRITE_CHAR       ENDP ;Finaliza o procedimento

    END PRINT_ASCII ;Finaliza o programa
```

Este programa mostra na tela o valor dos 256 caracteres do código ASCII.

```

\!!!!!!/
( ã ã )
-----o000--( )-----
| Arquivo baixado da GEEK BRASIL      |
| O seu portal de informática e internet |
| http://www.geekbrasil.com.br        |
| Dúvidas ou Sugestões?                |
| webmaster@geekbrasil.com.br         |
-----o000-----
|_| |_|
||  ||
ooO  Ooo
```