Ejemplo 1

|  |  |
| --- | --- |
| Programa Fuente | Código intermedio |
| Inicio  Imprime (“Hola Mundo !”)  Fin | Imprime (“Hola Mundo !”) |

Código de tres direcciones

|  |  |  |  |
| --- | --- | --- | --- |
| Operando 1 | Operando 2 | Operación | Resultado |
|  |  | Imprime | " Hola Mundo ! " |

Código en ensamblador

Mismo Encabezado para todos los programas.

TITLE Imprime un letrero

.MODEL SMALL ; El programa Imprime Hola Mundo

.STACK 64

Se localizan del archivo de cuádruples los mensajes de salida y la declaración de variable

.DATA

msg: db " Hola Mundo ! ", 0Dh,0Ah, 24h

.CODE

inicio PROC FAR

MOV AX,@data ;se direcciona al segmento de datos

MOV DS,AX

Se recorre cada cuádruple

MOV AH,09h ;función 09h de la interrupción 21h para mostrar msg

LEA DX,msg

INT 21h

MOV AH,4ch ;devuelve el control para seguir trabajando en consola

INT 21h

inicio ENDP

END

Ejemplo 2

|  |  |
| --- | --- |
| Programa Fuente | Código intermedio |
| Inicio  N: Entero;  Imprime (“ cuantas veces quieres imprimir hola Mundo !!!”)  Lee(n)  For (i:= 1, i>=n, i++)  Imprime (“Hola Mundo !”)  Fin | Imprime (“ cuantas veces quieres imprimir hola Mundo !!!”)  Lee(n)  i:= 1  10:  if i<=n go to 20  goto 30  20:  Imprime (“Hola Mundo !”)  I=I+1  Goto 10  30:  Fin |

Código de tres direcciones

|  |  |  |  |
| --- | --- | --- | --- |
| Operando 1 | Operando 2 | Operación | Resultado |
|  |  | Imprime | " Hola Mundo ! " |
|  |  | Lee | N |
| I |  | = | 1 |
|  |  | 10 | : |
| I | n | <= | 20 |
|  |  | 30 |  |
|  |  | 20 | : |
|  |  | Imprime | “Hola Mundo !” |
| I | 1 | + | i |
|  |  | 10 |  |
|  |  | 30 | : |

Código Ensamblador

|  |  |
| --- | --- |
| .model small  .stack 64  .data  msgc db 10,13,"Cuantas veces quieres imprimir un hola mundo: ",'$'  msg1 db 10,13,"hola mundo !",'$'  n db ?  i db 1  .code  inicio proc far  mov ax,@data  mov ds,ax  mov ah,09  lea dx,msgc  int 21h  ;leemos el carácter numérico  mov ah,01h  int 21h  sub al,30h  mov n,al | a10:    mov cl,n  cmp cl,i  jae a20  jmp a30  a20:    mov ah,09h  lea dx,msg1  int 21h  mov cl,i  add cl,1  mov i,cl  jmp a10    a30:  ;salida al ms-dos  mov ah,4ch  int 21h  inicio endp  end |

Ejemplo 3

|  |  |
| --- | --- |
| Programa fuente | Código de tres direcciones |
| Inicio  n1: entero;  Imprime ( “Dame tu calificación ”);  Lee (n1);  If (n1 >= 7 );  Imprime (“Reprobaste la materia ”)  Else  Imprime(“Aprobaste la materia ”)  Fin | n1:entero  Imprime ( “Dame tu calificación ” );  Lee (n1);  If n1 >= 7 go to 10  Goto 20  10:  Imprime (“Reprobaste la materia ”)  Goto 30  20:  Imprime(“Aprobaste la materia ”)  30 |

Código en cuádruples

Recuerda que cuando se genera el análisis semántico se genera una tabla de símbolos, con los datos que se declaran del programa fuente.

|  |  |  |  |
| --- | --- | --- | --- |
| Operando 1 | Operando 2 | Operación | Resultado |
|  |  |  |  |
|  |  | Imprime | “Dame tu calificación ” |
|  |  | Lee | N1 |
| n1 | 7 | >= | 10 |
|  |  | 20 |  |
|  |  | 10: | : |
|  |  | imprime | “Reprobaste la materia “ |
|  |  | 30 |  |
|  |  | 20: |  |
|  |  | imprime | “ Aprobaste la materia “ |
|  |  | 30: |  |

Código en ensamblador

TITLE compara dos números

.MODEL SMALL ;el programa va ser pequeño

.STACK 64

;-----------------Declaración de variables-----------------------

.DATA

n1 DB 0 ;DB dato de 8bits inicializado en cero

msgn1 DB 10,13,"Dame tu calificación",'$'

msgn2 DB 10,13,"Aprobaste la materia ",'$'

msgn3 DB 10,13,"Reprobaste la materia ",'$'

;-------------------------------------------------------------------------------------

.CODE

inicio PROC FAR

MOV AX,@data ;se direcciona al segmento de datos

MOV DS,AX

;-----------------Mostrar mensaje y habilitar teclado--------------------------------

MOV AH,09h ;función 09h de la interrupción 21h para mostrar msgn1

LEA DX,msgn1

INT 21h

MOV AH,01h ;habilita teclado para ingresar número

INT 21h

SUB AL,30h ;se resta 30 para convertirlo a número

MOV n1,AL

;-------------------Mostrar Resultados-----------------------------------------

mov cl,n1

|  |
| --- |
| LEA DX,msgn3  INT 21h  ;---------------------------------------------------------------------------  a30:  MOV AH,4ch ;devuelve el control para seguir trabajando en consola  INT 21h  inicio ENDP  END |

cmp cl,7

jae a10

jmp a20

a10:

MOV AH,09h

LEA DX,msgn2

INT 21h

jmp a30

a20:

MOV AH,09h

Ejemplo 4

|  |  |
| --- | --- |
| Programa Fuente | Código intermedio |
| class ejm{  writeLine(“Ejemplo Codigo”);  int a=8;  int b=3;  int z;  Int q;  Compara (){  If (a>b){  z=a-b;  writeLine(“El valor de la resta es:” + z)  }  else  q=a+b;  writeLine(“El valor de la suma es:” + q)  }  } | writeLine(“Ejemplo Codigo”)  int a=8;  int b=3;  int z;  int q;  If(a>b) goto 10  goto 20  10:  z=a-b;  writeLine(“El valor de la resta es:” );  writeLine(z);  goto 30  20:  q=a+b;  writeLine(“El valor de la suma es:” );  writeLine(q);  30: |

Código de cuatro direcciones

|  |  |  |  |
| --- | --- | --- | --- |
| Operando 1 | Operando 2 | Operación | Resultado |
|  |  | writeLine | “Ejemplo Codigo” |
| A | B | > | 10 |
|  |  | 20 |  |
|  |  | 10: | : |
| A | B | - | Z |
|  |  | writeLine | “El valor de la resta es:” |
|  |  | writeLine | Z |
|  |  | 30 |  |
|  |  | 20: |  |
| A | B | ´+ | Q |
|  |  | writeLine | “El valor de la suma es:” |
|  |  | writeLine | Q |
|  |  | 30 | : |

Código en ensamblador

TITLE compara dos números

.MODEL SMALL ; compara dos valores y realiza suma o resta según los valores de entrada

.STACK 64

.data

titulo db 'Ejemplo Codigo',10,13,'$'

resta1 db 'El valor de la resta es:',10,13,'$'

suma1 db 'El valor de la suma es:',10,13,'$'

a db 2

b db 5

z db ?

q db ?

.code

mov ax, @data

mov ds,ax

inicio:

mov ah,09h

lea dx,titulo

int 21h

;compara los valores de entrada

mov cl,a

cmp cl,b

jae a10

jmp a20

a10:

mov al,a

sub al,b

mov q,al

mov ah,09h

lea dx,resta1

int 21h

mov dl,q

add dl,30h

mov ah,02h

int 21h

jmp a30

a20:

mov al,a

add al,b

mov z,al

mov ah,09h

lea dx,suma1

int 21h

mov dl,z

add dl,30h

mov ah,02h

int 21h

a30:

.exit

end

Ejemplo 5

|  |  |
| --- | --- |
| Código fuente | Código intermedio |
| Inicio  Entero n1,n2,suma,resta;  suma=0;  resta:=0;  imprime (“Dame el primer valor ”);  Lee (n1);  imprime (“Dame el segundo valor ”)  Lee (n2)  suma =n1+n2;  resta=n1-n2;  Imprime( “Suma =: “);  Imprime (suma);  Imprime( “Resta =: “);  Imprime (resta);  Fin | suma=0;  resta:=0;  imprime (“Dame el primer valor ”);  Lee (n1);  imprime (“Dame el segundo valor ”)  Lee (n2)  suma =n1+n2;  resta=n1-n2;  Imprime( “Suma =: “);  Imprime (suma);  Imprime( “Resta =: “);  Imprime (resta); |

Código de cuatro direcciones

|  |  |  |  |
| --- | --- | --- | --- |
| Operando 1 | Operando 2 | Operación | Resultado |
| Suma |  | = | 0 |
| Resta |  | = | 0 |
|  |  | imprime | “Dame el primer valor “ |
|  |  | lee | N1 |
|  |  | imprime | “Dame el segundo valor “ |
|  |  | lee | N2 |
| N1 | N2 | + | suma |
| N1 | N2 | - | resta |
|  |  | imprime | “Suma =: “ |
|  |  | imprime | suma |
|  |  | imprime | “Resta =: “ |
|  |  | imprime | resta |
|  |  |  |  |

Código en ensamblador

TITLE Operaciones de Suma y resta

.MODEL SMALL ;el programa pide dos valores los suma y los resta

.STACK 64

;-----------------Declaración de variables-----------------------

.DATA

n1 DB 0 ;DB dato de 8bits inicializado en cero

n2 DB 0

suma DB 0 ;donde se va a guardar el resultado

resta DB 0

msgn1 DB 10,13,"Dame el primer valor:",'$'

msgn2 DB 10,13,"Dame el segundo valor:",'$'

msg3 DB 10,13,"Suma = ",'$'

msg4 DB 10,13,"Resta = ",'$'

;-------------------------------------------------------------------------------------

.CODE

inicio PROC FAR

MOV AX,@data ;se direcciona al segmento de datos

MOV DS,AX

;-----------------Mostrar mensaje y habilitar teclado--------------------------------

MOV AH,09h ;función 09h de la interrupción 21h para mostrar msgn1

LEA DX,msgn1

INT 21h

MOV AH,01h ;habilita teclado para ingresar número

INT 21h

SUB AL,30h ;se resta 30 para convertirlo a número

MOV n1,AL

;----------------Mostrar mensaje y habilitar teclado----------------------------

MOV AH,09h

LEA DX,msgn2

INT 21h

MOV AH,01h

INT 21h

SUB AL,30h

MOV n2,AL

;----------------Realiza operaciones---------------------------------------------

MOV AL,n1

ADD Al,n2

MOV suma,AL

MOV AL,n1

SUB AL,n2

MOV resta,AL

;-------------------Mostrar Resultados-----------------------------------------

MOV AH,09h

LEA DX,msg3

INT 21h

MOV DL,suma

ADD DL,30h ;ahora se suma para visualizar

MOV AH,02 ;para mostrar mensaje en pantalla

INT 21h

MOV AH,09h

LEA DX,msg4

INT 21h

MOV DL,resta

ADD DL,30h

MOV AH,02

INT 21h

;---------------------------------------------------------------------------

MOV AH,4ch ;devuelve el control para seguir trabajando en consola

INT 21h

inicio ENDP

END

Ejemplo 6

|  |  |
| --- | --- |
| Código fuente | Código intermedio |
| Inicio  Entero n1,n2,multi, divi;  multi=0;  divi:=0;  imprime (“Dame el primer valor ”);  Lee (n1);  imprime (“Dame el segundo valor ”)  Lee (n2)  multi =n1\*n2;  divi=n1/n2;  Imprime( “Multiplicación =: “);  Imprime (multi);  Imprime( “División =: “);  Imprime (divi);  Fin | multi=0;  divi:=0;  imprime (“Dame el primer valor ”);  Lee (n1);  imprime (“Dame el segundo valor ”)  Lee (n2)  multi =n1\*n2;  divi=n1/n2;  Imprime( “Multiplicación =: “);  Imprime (multi);  Imprime( “División =: “);  Imprime (divi); |

Código de cuatro direcciones

|  |  |  |  |
| --- | --- | --- | --- |
| Operando 1 | Operando 2 | Operación | Resultado |
| Multi |  | = | 0 |
| Divi |  | = | 0 |
|  |  | imprime | “Dame el primer valor “ |
|  |  | lee | N1 |
|  |  | imprime | “Dame el segundo valor “ |
|  |  | lee | N2 |
| N1 | N2 | + | multi |
| N1 | N2 | - | divi |
|  |  | imprime | “Multiplicación =: “ |
|  |  | imprime | divi |
|  |  | imprime | “División =: “ |
|  |  | imprime | resta |

Código en ensamblador

TITLE Operaciones Multiplicación y División

.MODEL SMALL ;el programa pide dos valores los multiplica y los divide

.STACK 64

;-----------------Declaración de variables-----------------------

.DATA

n1 DB 0 ;DB dato de 8bits inicializado en cero

n2 DB 0

multi DB 0 ;donde se va a guardar el resultado

divi DB 0

msgn1 DB 10,13,"Dame el primer valor:",'$'

msgn2 DB 10,13,"Dame el segundo valor:",'$'

msg3 DB 10,13,"Multiplicacion = ",'$'

msg4 DB 10,13,"Division = ",'$'

;-------------------------------------------------------------------------------------

.CODE

inicio PROC FAR

MOV AX,@data ;se direcciona al segmento de datos

MOV DS,AX

;-----------------Mostrar mensaje y habilitar teclado--------------------------------

MOV AH,09h ;funcion 09h de la interrupción 21h para mostrar msgn1

LEA DX,msgn1

INT 21h

MOV AH,01h ;habilita teclado para ingresar número

INT 21h

SUB AL,30h ;se resta 30 para convertirlo a número

MOV n1,AL

;----------------Mostrar mensaje y habilitar teclado---------------------------

MOV AH,09h

LEA DX,msgn2

INT 21h

MOV AH,01h

INT 21h

SUB AL,30h

MOV n2,AL

;----------------Realiza operaciones---------------------------------------------

MOV AL,n1

MUL n2

MOV multi,AL

MOV AL,n1

DIV n2

MOV divi,AL

;-------------------Mostrar Resultados-----------------------------------------

MOV AH,09h

LEA DX,msg3

INT 21h

MOV DL,multi

ADD DL,30h ;ahora se suma para visualizar

MOV AH,02 ;para mostrar mensaje en pantalla

INT 21h

MOV AH,09h

LEA DX,msg4

INT 21h

MOV DL,divi

ADD DL,30h

MOV AH,02

INT 21h

;---------------------------------------------------------------------------

MOV AH,4ch ;devuelve el control para seguir trabajando en consola

INT 21h

inicio ENDP

END

Ejemplo 7

org 100h

include "emu8086.inc"

DEFINE\_PRINT\_STRING

DEFINE\_PRINT\_NUM

DEFINE\_PRINT\_NUM\_UNS

DEFINE\_SCAN\_NUM

TITLE Operaciones

.MODEL lange ;el programa va ser pequeño

.STACK 64

;-----------------Declaración de variables-----------------------

.DATA

saltoLN db 0D,0AH,"$"

msg1 DB 10,13,"Ingresa el valor de a ",'$'

msg3 DB 10,13,"Ingresa el valor de b ",'$'

msg4 DB 10,13,"la suma es: ",'$'

a dw ?

b dw ?

c dw ?

;-------------------------------------------------------------------------------------

.CODE

mov ah,09h

lea dx,msg1

int 21h

call SCAN\_NUM

mov a,cx

lea dx, saltoLN

mov ah,09h

int 21h

mov ah,09h

lea dx,msg3

int 21h

call SCAN\_NUM

mov b,cx

lea dx, saltoLN

mov ah,09h

int 21h

mov ax,a

mov cx,b

add ax,cx

mov c, ax

mov ah,09h

lea dx,msg4

int 21h

mov ax,c

call PRINT\_NUM

lea dx,saltoLN

mov ah,09h

int 21h

;mov ax,@data

;mov ds,ax

;mov ax,13

;mov a,ax

.exit