



Universidad de Guadalajara

Centro Universitario de Ciencias Exactas e Ingenierías

DIVISIÓN DE TECNOLOGÍAS PARA LA INTEGRACIÓN CIBER-
HUMANA

DEPARTAMENTO DE CIENCIAS COMPUTACIONALES

Actividad 11 (practica 3)

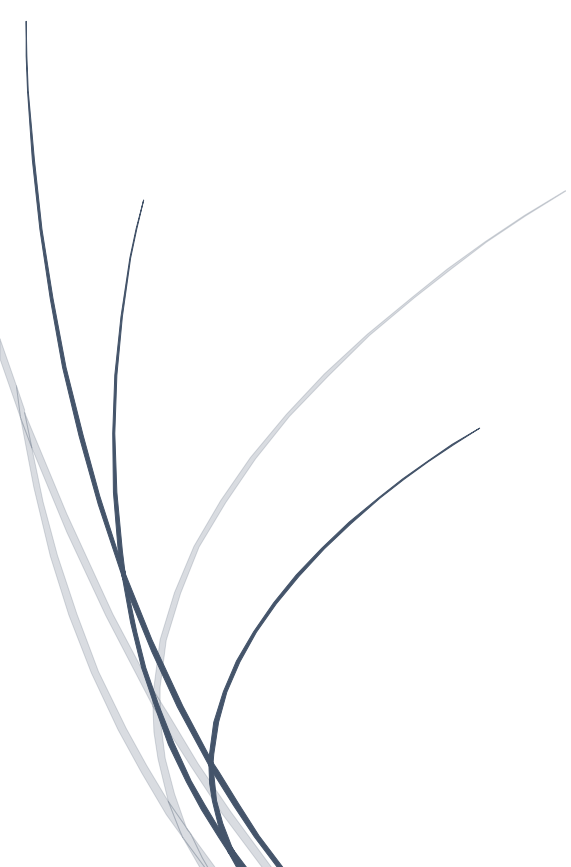
TEMA: Uso de la instrucción JUMP para Iterar (Función 4C00H
de la INT 21H)

NOMBRE DEL ESTUDIANTE: Padilla Perez Jorge Daray

NOMBRE DE LA MATERIA: Seminario de solución de
problemas de traductores de lenguaje

NOMBRE DEL PROFESOR: Roberto Patiño Ruiz

29/02/2023



Titulo

Uso de la instrucción JUMP para Iterar (Función 4C00H de la INT 21H)

BREVE RESUMEN (1/2 cuartilla)

Se elaboro un programa sencillo que trata de el procesamiento de un bucle llamado JMP el cual se pone a prueba en esta practica para saber como funciona, repitiendo en bucle una función llamada A20 la cual hace una suma a partir del registro AX, este resultado se suma al registro BX.

Se introduce también SHL la cual sirve para desplazar el operando a la izquierda en 1, esta instrucción se repite gracias al JMP que llama a la función A20, Al final es un bucle indefinido y no se detiene.

PLANTEAMIENTO DEL PROBLEMA A RESOLVER:

Codificar el programa de la práctica 3 a fin de evaluar el funcionamiento de la función 4C00H de la INT 21H, y el uso de la instrucción JUMP

RESULTADOS LOGRADOS (redacta un objetivo específico y los resultados alcanzados):

Se logro cumplir el objetivo principal ya que se pudo ejecutar el programa correctamente y entender cómo funciona la instrucción JUMP y en que se diferencia de LOOP

FUNDAMENTO TEÓRICO (Hipótesis)

Esta instrucción se utiliza para desviar el flujo de un programa sin tomar en cuenta las condiciones actuales de las banderas ni de los datos.

Controlar el flujo del programa es algo muy importante, aquí es donde su programa puede tomar decisiones de acuerdo con ciertas condiciones.

Salto incondicional (JMP):

La instrucción básica que transfiere el control a otro punto del programa. La sintaxis básica es :

JMP etiqueta

JMP et1 ;Indica a donde saltara et1

·
·
·

et1: ;realiza lo que se encuentra en ella et1

MOV AX, 1

OBJETIVO DEL PROGRAMA

Poder entender el funcionamiento de JUMP y la diferencia que tiene con el funcionamiento de la instrucción LOOP

DESARROLLO (Incluir Código comentado, imágenes y/o diagramas)

PAGE 60,132 ;Numero de filas y columnas

TITLE PROG3.EXE ;Titulo del programa

.MODEL SMALL ;Habilita dos segmentos de 64kb.

;

.CODE ;Inicio del codigo

ORG 100H ;Direccion que inicia el ensamblador

MAIN PROC NEAR ;Directiva de inicio NEAR

MOV AX,01 ;Mueve al registro AX el numero 1

MOV BX,01 ;Mueve al registro BX el numero 1

MOV CX,01 ;Mueve al registro CX el numero 1

;

A20: ;Declaracion de funcion llamada A20

ADD AX,01 ;Suma lo que contiene AX mas 1

ADD BX, AX ;Suma lo que contiene BX a AX

SHL CX,1 ;Sirve para desplazar el operando a la izquierda en 1, si es derecha es SHR

JMP A20 ;Hace saltos a la funcion A20 indefinidamente

MOV AX,4C00H ;Salida a DOS

INT 21H ;Interrupcion a funcion DOS

MAIN ENDP ;Fin del procedimiento

END MAIN ;Fin del programa

CONCLUSIONES (Breve descripción de los resultados obtenidos)

Para concluir los resultados obtenidos fueron satisfactorios ya que se elaboro el programa y funciono de la manera correcta al momento de su realización en clase, no se me hizo complicado y se le entiende a lo que hace el profesor en sus programas.

Además, se logró entender de manera correcta el programa y la diferencia que hay entre una instrucción y otra.

REFERENCIAS BIBLIOGRÁFICAS (Formato APA, en caso de consultar otras fuentes)

(s. f.). *Estructura de computadores 8086*. exaby. Recuperado 16 de febrero de 2023,
de

[https://www.exabyteinformatica.com/uoc/Informatica/Estructura_de_computadores/
Estructura_de_computadores_\(Modulo_6\).pdf](https://www.exabyteinformatica.com/uoc/Informatica/Estructura_de_computadores/Estructura_de_computadores_(Modulo_6).pdf)