Aplicación

Descripción generada automáticamente con confianza baja

**Universidad Técnica Federico Santa María**

**Sede Viña del Mar**

**Caso de Estudio – LENGUAJE ENSAMBLADOR**

ARQUITECTURA ORGANIZACIÓN DE COMPUTADORES

Nombres:

1. Sebastian Barra.
2. Josiel Videla.
3. Esteban Gallardo.

**ÍNDICE.**

1. **INTRODUCCIÓN.**

En el presente informe, realizaremos el desarrollo de un programa en lenguaje ensamblador, que corresponde a un lenguaje de bajo nivel que se caracteriza por ser una representación directa del código de máquina. Estos lenguajes poseen una cierta dificultad de ser entendido directamente por una persona, además de ser complejo estos lenguajes, son directamente programados sobre el Hardware de la computadora, donde suelen ser más rápidos de ejecutar y utilizan menos recursos del sistema en comparación a otros lenguajes.

El programa por realizar deberá permitir al usuario ingresar un valor hexadecimal por teclado que luego será convertido a un valor binario a través del algoritmo. Este programa será trabajado con el ensamblador Microsoft Macro Assembler (MASM) que corresponde a un ensamblador para la familia x86 de microprocesadores, además se le considera un ensamblador de alto nivel, es decir, es un ensamblador el cual contiene características encontradas en los modernos lenguajes de programación.

**1.1 ¿Por qué este ensamblador y no otro?**

Seleccionamos el ensamblador Microsoft Macro Assembler porque encontramos interesante de aprender sobre este ensamblador, ya que fue producido por Microsoft así asegurándonos que tuviera compatibilidad con nuestros equipos de trabajo aparte de que forma parte de uno de los ensambladores de alto nivel.

**DESARROLLO DEL PROGRAMA.**

A continuación, se encuentra la explicación del código del programa tomado de ejemplo para realizar la conversión de un carácter hexadecimal a binario.

Texto

Descripción generada automáticamente

En estas primeras líneas de programa, se declara la estandarización del programa para que funcione correctamente, aparte del tamaño de este.

Líneas de código.

Línea 1: En esta primera línea del código se define el nombre del programa a realizar.

1. En la sentencia DOSSEG
2. El .MODEL SMALL
3. .STACK

Campo de etiquetas:

Texto

Descripción generada automáticamente

En estos segmentos de código, corresponde al campo de etiquetas, que corresponde a la definición variables que serán utilizadas dentro del código principal. La sentencia db es utilizada definir los bytes de las variables creadas con el nombre seleccionado por el usuario, en este caso se le asignan bytes a una cadena de texto donde deberá estar entre comillas para definirlas como texto. La sentencia dw corresponde asigna una palabra de 2 bytes de almacenamiento para cada inicializador, donde en las líneas de código se definen estas variables como palabras para luego en las siguientes líneas de código ser restauradas para que el programa funcione correctamente dándoles un valor nulo.

Código principal:

Texto

Descripción generada automáticamente

En los inicios del programa principal se define **.code main** donde se encontrará todo el procedimiento del algoritmo y todo el funcionamiento de este. Seguidamente se define proc que sirve para marcar el inicio y el final de el procedimiento de todo el programa. Continuando la explicación de este segmento de código, en las líneas de código **17** y **18** se encuentran líneas de código importantes para el funcionamiento del programa, explicando detenidamente, las funciones de estas líneas de código son para mover(mov) @data en el registro AX(ax) para luego moverlo nuevamente hacia el registro DS(ds) ya que no se puede hacer directamente.

Para complementar lo dicho anteriormente, @data corresponde al numero de segmento de bytes que será utilizado para los datos del programa. AX correspondería a un registro de 16 bits y DS correspondería a la dirección del segmento de datos.

Imagen que contiene Texto

Descripción generada automáticamente

En estas líneas de código simplemente se reinician todos los valores/variables del programa para asegurar que la conversión del código funcione correctamente asignándoles un valor 0. Específicamente lo que se hacen en estas líneas, con la función mov se le asignan valores 0 a los registros de 16 bits: ax, bx, cx, dx, y a la variable print.



En este segmento de código ocurre lo mismo, se reinician todos los valores necesarios para cuando el programa se vuelva a iniciar debido al ciclo repetitivo existente en el código.

Texto

Descripción generada automáticamente

En las líneas de código desde la 29 a la 31, corresponde a el proceso de mostrar en pantalla la variable iniciada al inicio del programa llamada “msgstar”. El proceso inicia con un mov con la variable “msgstar”

**CONCLUSIÓN.**

**BIBLIOGRAFÍA.**