

# UNIVERSIDAD AUTÓNOMA DE BAJA CALIFORNIA

FACULTAD DE CIENCIAS QUIMICAS E INGENIERIA



## Microcontroladores y microprocesadores

**Practica 2.** Introducción al intérprete 80x86 sobre la plataforma T-Juino

**Alumno:** Caudillo Sanchez Diego

**Grupo:** 561

**Docente:** Jesús García

**Fecha de entrega:** 13 de Febrero del 2020

# Práctica No. 2

## Introducción al intérprete 80x86 sobre la plataforma T-Juino

**Objetivo:** El alumno se familiarizara con el intérprete 80x86 (modo 16 bits) sobre la plataforma T-Juino. Esto con el fin de desarrollar programas en lenguaje X para dicha plataforma.

### Material:

- Computadora Personal (PC).
- Programa editor de texto (ASCII), TASM y TLINK.
- Tarjeta T-Juino (con interprete 80x86).
- Manejador DRSI instalado.

### Equipo:

- Computadora Personal
- Programa emulador de terminal.

### Teoria

#### Lenguaje de programación C

C es el lenguaje de programación de propósito general asociado, de modo universal, al sistema operativo UNIX. sin embargo, la popularidad, eficacia y potencia de C, se ha producido porque este lenguaje no está prácticamente asociado a ningún sistema operativo, ni a ninguna máquina en especial. Esta es la razón fundamental, por la cual C, es conocida como el lenguaje de programación de sistemas, por excelencia.

C es una evolución de los lenguajes BCPL y B para el primitivo UNIX de la computadora DEC PDP-7.

C nació realmente en 1978, con la publicación de *The C Programming Language*, por Brian Kernighan y Dennis Ritchie. Desde su nacimiento, C fue creciendo en popularidad y los sucesivos cambios en el lenguaje, a lo largo de los años, junto a la creación de compiladores por grupos no involucrados en su diseño, hicieron necesario pensar en la estandarización de la definición del lenguaje C.

C es un lenguaje de alto nivel, que permite programar con instrucción de lenguaje de propósito general. También, C se define como un lenguaje de programación estructurado de propósito general, aunque en su diseño primo el hecho de que fuera especificado como un lenguaje de programación de sistemas, lo que proporciona una enorme cantidad de potencia y flexibilidad.

El lenguaje C tiene una gran cantidad de ventaja sobre otros lenguaje y es, precisamente, la razón fundamental de que después de casi dos décadas de uso, C siga siendo uno de los lenguajes de más populares y utilizados en empresas, organizaciones y fábricas de software de todo el mundo.

## **Lenguaje ensamblador**

El único lenguaje que entienden los microcontroladores es el código máquina formado por ceros y unos del sistema binario. El lenguaje ensamblador expresa las instrucciones de una más natural al hombre a la vez que muy cercana al microcontrolador, ya que cada de esas instrucciones corresponde con otra en código máquina.

El lenguaje ensamblador trabaja con mnemónicos, que son grupos de caracteres alfanumericos que simbolizan las órdenes o tareas a realizar. La traducción de los mnemónicos a código máquina entendible por el microcontrolador la lleva a cabo un programa ensamblador. El programa escrito en lenguaje ensamblador se denomina *código fuente* (\*.asm). El programa ensamblador proporciona a partir de este fichero el correspondiente código máquina, que suele tener la extensión \*.hex

## **Enlace entre C y ensamblador**

Existe una relación muy estrecha entre la programación en C y la programación en ensamblador. C fue creado para evitarle a los programadores del sistema operativo UNIX seguir escribiéndolo en ensamblador. Las ventajas de programar en C lo mismo que antes se programaba en ensamblador son obvias: portabilidad y legibilidad a través de un código bien estructurado. Sin embargo las partes del sistema en las que es necesario trabajar con características específicas del hardware no pueden ser escritas en C.

En general hay dos estrategias para mezclar código en C con código en ensamblador. La primera consiste en compilar archivos escritos en C y ensamblar otros escritos en código de ensamblador para luego enlazarlos todos juntos. La segunda se trata de embeber código escrito en ensamblador dentro de código escrito en C a través de algún mecanismo que ofrezca el compilador usado. En este artículo se tratará sólo la primera estrategia.

Aunque las ideas que aquí se exponen son útiles independientemente de cualquier plataforma, toda la nota presupone un sistema tipo UNIX (GNU en particular) en una arquitectura de procesador 80386 o superior.

## **Conclusiones y comentarios**

El desarrollo de la práctica no fue complicado, pero al momento de correr mi código, la función *puts* no imprimía la cadena de manera correcta. Esto me llevo varios días solucionarlo, y al final el problema fue que en el momento de enlazar los archivos objeto, lo hacía en el orden incorrecto.

## **Bibliografía**

Zahonero, I. & Joyanes, L. (2005). *Programación en C. Metodología, algoritmos y estructura de datos*. Madrid:España. McGraw Hill.