

Universidad Autónoma de Baja California
Facultad de Ciencias Químicas e Ingeniería



ORGANIZACIÓN DE LAS COMPUTADORAS Y LENGUAJE ENSAMBLADOR

Practica 11

**Interfaz entre el lenguaje C89 y el lenguaje ensamblador
procesador 8086**

Docente: Sanchez Herrera Mauricio Alonso

Alumno: Gómez Cárdenas Emmanuel Alberto

Matricula: 1261509

Contenido

TEORIA.....	2
DESARROLLO.....	3
Parte 1	3
Parte 2	3
CONCLUSIONES.....	3

TEORIA

Funciones del lenguaje de ensamblador pueden ser enlazadas con un programa C. Aunque se tienen que tomar muchas precauciones al usar registros, obtener parámetros, llamar a otras funciones y a la hora de entrar o salir de funciones.

Cuando los registros son utilizados en lenguaje ensamblador, deben seguirse las reglas al obtener o modificarlos. Especialmente cuando se utilizan registros de propósitos especiales.

El compilador hace asunciones sobre como las funciones tratan los registros.

Unas de las tantas reglas que existen al momento de guardar o recuperar valores en los registros son:

- Los registros deben ser salvados haciéndoles push al stack

- Al inicio de una función de lenguaje ensamblador, todos los registros que han de ser usados en la función deben ser guardados. Al final de la función, estos han de ser recuperados.

- Antes de una llamada a función, todos los registros que contienen datos importantes deben de ser guardados y recuperados después de la función.

En cuanto a C, los argumentos son pasados a las funciones poniéndolos en registros o el stack, siguiendo las reglas:

- Hasta 3 argumentos pueden ser pasados en registros

- Una vez que un argumento sea pasado al stack, todos los demás (a la derecha) están en el stack.

- Todos los valores mayores a 32 bits son pasados por el stack.

Cuando argumentos son pasados usando el stack, se hace push de derecha a izquierda, por ej. `test(int a, char b, float c);`

Primero se mandan el parámetro c, luego el b y por último el a.

DESARROLLO

Parte 1

```
Hola mundo
```

Parte 2

```
Matriz A:
  1      2      3
  4      5      6
  7      8      9

Matriz B:
  9      8      7
  6      5      4
  3      2      1

Matriz Resultante:
 30      24      18
 84      69      54
138     114      90
```

CONCLUSIONES

Esta práctica nos abrió la posibilidad de poder implementar un código en ensamblador para posteriormente ser ejecutado en código C, lo cual es muy útil si se quiere intentar implementar un procedimiento o función lo más eficiente posible, gracias a que ensamblador nos ofrece un control total (o casi total) de los procedimientos que ejecuta el procesador.

REFERENCIA

Signal.uu.se. (2020). Retrieved 14 December 2020, from <http://www.signal.uu.se/Staff/pd/DSP/Doc/ctools/chap4.pdf>.