

Universidad del Valle de Guatemala  
Facultad de Ingeniería  
Departamento de Ciencias de la Computación  
CC-3005 Programación en Assembler  
Catedrática: Martha Ligia Naranjo F. De Ortiz  
Sección: 10

# Investigación Lenguaje Ensamblador Intel en S.O. GNU/Linux

Héctor Antonio Hurtarte Estrada

**Carné: 08119**

Juan Pablo Pinto Santana

**Carné: 08359**

Ernesto Rodríguez Pfister

**Carné: 08161**

Fecha: 21/09/2009

## RESUMEN

Esta investigación tiene por objetivo abarcar información referente al lenguaje ensamblador (x86) Intel en el S.O. GNU/Linux (GAS). Como el resto de los demás lenguajes ensambladores, usa una serie de nemónicos para efectuar operaciones fundamentales que el procesador con arquitectura x86 puede efectuar. El lenguaje ensamblador x86 tiene dos vertientes fundamentales de acuerdo a su sintaxis: la de Intel original y la de AT&T, la cual usa los mismos registros, instrucciones, etc, pero tiene diferente modalidad de programación y orden dentro de las instrucciones lo cual se puede comprobar por simple visualización al código fuente del programa. El lenguaje ensamblador tiene ventajas (acceso a registros y al I/O de forma directa) y desventajas (no portable para otras arquitecturas) de las cuales un programador se aprovecha para su uso en el desarrollo de aplicaciones críticas como sistemas de arranque, sistemas operativos, kernels, controladores de dispositivos o programas de uso general. En el caso del trabajo escrito, se hablará sobre cómo trabaja GAS para ser un ensamblador funcional dentro de la arquitectura x86.

En primera instancia es importante mencionar como la parte inicial del trabajo escrito las relaciones principales en la historia de la computación que dieron origen al sistema operativo UNIX, los diferentes sistemas operativos basados en este primero dando como ejemplo al sistema operativo GNU/Linux y como este se involucró con el proyecto GNU para que fuera software libre del sistema de copyright.

Como segunda parte del trabajo se explican los conceptos importantes que se deben tomar en cuenta a la hora de programar en lenguaje ensamblador Intel bajo el sistema operativo GNU/Linux tales como la sintaxis que soporta (AT&T e Intel), registros, instrucciones, etc. Es importante recalcar que la diferencia de sintaxis no es peor ni mejor que la de Intel, solo es diferente.

El último punto que se trata en este trabajo es la descripción del programa ejemplo para ilustrar un caso particular de cómo funciona el lenguaje ensamblador Intel bajo el S.O. GNU/Linux y de este modo que se pueda apreciar el código fuente y demás diferencias que puedan existir.

De forma general es importante mencionar que GAS fue inventado para soportar un compilador UNIX de 32 bit entonces usa la sintaxis AT&T (mas tarde con soporte de sintaxis Intel) que es estándar en el mundo UNIX.

Como puntos importantes en los conceptos a conocer de la forma en que se programa en GAS están los siguientes:

- Los nombres de los registros van antecidos por un % (Ej. %EAX).
- El orden de los operandos es: primero la fuente y el último el destino.

- El tamaño de los operandos se fija con un sufijo al final de la instrucción. B para byte, W para palabra y L para palabra doble (long). (Ej. MOVL, MOVB, MOVW)
- Operandos inmediatos son marcados con un prefijo \$ (Ej. %5)
- Para direccionamiento indirecto se usan los paréntesis para encasillar al registro índice o localidad de memoria.

Aunque este resumen da a conocer de forma no específica de lo que tratará el trabajo escrito, pretende demostrar lo esencial de lo que se puede encontrar de forma más específica y detallada junto con una descripción del programa ejemplo dentro del trabajo escrito, además de las conclusiones que resaltan puntos importantes del trabajo.