

Instituto Tecnológico de Costa Rica
Campus Tecnológico Central de Cartago

Escuela de ingeniería en computación
IC-1400 Fundamentos de organización de computadoras
Profesor Ing. Esteban Arias Méndez, M.Sc

Reporte #7 – Semana 11
Lectura semana 11

Realizado por: Ledvin Manuel Leiva Mata
Carné: 2023071280

Semestre 1, año 2023
Fecha de entrega 5 de mayo 2023

Resumen Capítulo 1 del Libro Sivarama P. Dandamudi Guide to Assembly Language Programming in Linux.

Capítulo 1: Lenguaje ensamblador

Introducción:

La programación en lenguaje ensamblador es considerada de bajo nivel ya que cada instrucción de lenguaje ensamblador realiza una tarea mucho más detallada en comparación con una instrucción de lenguaje de alto nivel. Es nativa del procesador utilizado en el sistema, lo que significa que un programa escrito en lenguaje ensamblador de Intel no puede ser ejecutado en un procesador PowerPC. La programación en lenguaje ensamblador requiere conocimientos detallados sobre la arquitectura del procesador, la organización de la memoria, etc. A pesar de ser considerado un lenguaje de bajo nivel, la programación en lenguaje ensamblador no expone todos los detalles de bajo nivel del sistema gracias a que el sistema operativo oculta algunos de ellos. El hardware del sistema, que consiste en circuitos lógicos digitales y la electrónica de soporte asociada, es el que finalmente ejecuta las instrucciones de lenguaje de máquina.

¿Qué es el lenguaje ensamblador?

El lenguaje ensamblador es un lenguaje de bajo nivel que se ve influido por la arquitectura del procesador y el conjunto de instrucciones. Es procesado por un programa ensamblador para generar lenguaje máquina. Para los procesadores Intel se utilizan ensambladores como NASM, MASM y TASM, siendo NASM una opción popular debido a su compatibilidad con varios formatos y a que es gratuito. El lenguaje ensamblador utiliza instrucciones crípticas expresadas mediante mnemónicos, lo que dificulta su lectura. Sin embargo, es más fácil de leer que las instrucciones equivalentes en lenguaje máquina. Aunque existe una correspondencia uno a uno entre el lenguaje ensamblador y el lenguaje máquina, programar en lenguaje máquina es casi imposible, y los primeros programadores tuvieron que recurrir a él cuando se introdujeron los microprocesadores.

Ventajas de los lenguajes de alto nivel.

Los lenguajes de alto nivel se prefieren a los lenguajes ensambladores para programar aplicaciones porque ofrecen una abstracción más conveniente del sistema, lo que facilita la resolución de problemas. A continuación, se enumeran algunas ventajas de programar en lenguajes de alto nivel:

Desarrollo de programas más rápido: Los lenguajes de alto nivel proporcionan estructuras que hacen que el desarrollo de programas sea más rápido, más pequeño y fácil de codificar y depurar, en comparación con los lenguajes ensambladores.

Mantenimiento más sencillo: Los programas en lenguajes de alto nivel son más fáciles de entender, mantener y modificar que los programas en lenguaje ensamblador, que son más largos y requieren más tiempo de codificación y depuración.

Portabilidad: los programas en lenguaje de alto nivel contienen menos detalles específicos del procesador, lo que permite utilizarlos sin apenas modificaciones en distintos sistemas informáticos. Los programas en lenguaje ensamblador son específicos del procesador y no son portables.

¿Por qué programar en lenguaje ensamblador?

Aunque se prefieren los lenguajes de alto nivel para programar aplicaciones, hay dos razones principales por las que se sigue programando en lenguaje ensamblador: la eficiencia y la accesibilidad al hardware del sistema.

Los programas en lenguaje ensamblador tienden a ser más compactos y a ejecutarse más rápido que sus homólogos en lenguaje de alto nivel, lo que los hace idóneos para controladores integrados y dispositivos portátiles en los que la eficiencia espacial es importante.

Además, el lenguaje ensamblador permite controlar directamente el hardware del sistema, lo que lo hace necesario para tareas como escribir controladores de dispositivos para hardware nuevo. Sin embargo, con las mejoras en la tecnología de los compiladores, los lenguajes de alto nivel son cada vez más eficientes, lo que acorta la distancia con el lenguaje ensamblador.

Aplicaciones típicas de la programación en lenguaje ensamblador

La programación en lenguaje ensamblador ofrece tres ventajas principales: **eficiencia temporal, accesibilidad al hardware y eficiencia espacial**. Los programas en lenguaje ensamblador eficientes en el tiempo son útiles en dos tipos de aplicaciones: las que requieren comodidad de tiempo para mejorar el rendimiento y las que son críticas en el tiempo y deben completar las tareas dentro de un plazo determinado. Algunos ejemplos de aplicaciones en las que el tiempo es un factor crítico son los sistemas de navegación de aeronaves, los sistemas de control de procesos y el software de control de robots.

La programación en lenguaje ensamblador también proporciona un control directo sobre el hardware del sistema, que a menudo es necesario para el software del sistema, como los sistemas operativos, los controladores de dispositivos y las interfaces de red.

La accesibilidad al hardware también es importante en aplicaciones como los videojuegos. Por último, la programación en lenguaje ensamblador puede producir código más eficiente en términos de espacio, algo esencial en dispositivos portátiles y de mano, así como en sistemas de control de naves espaciales.

Referencias:

[1] S.P. Dandamundi, <Assembly Language Programming in Linux>, Springer, Ottawa.