¿Porqué se sigue usando ensamblador?

Hoy en día, el lenguaje ensamblador se usa principalmente para la manipulación directa del hardware, el acceso a instrucciones especializadas del procesador o para abordar problemas críticos de rendimiento. Los usos típicos son controladores de dispositivos, sistemas integrados de bajo nivel y sistemas en tiempo real.

El lenguaje ensamblador es lo más cercano al procesador que puede obtener como programador, por lo que un algoritmo bien diseñado es excelente: el ensamblaje es ideal para la optimización de la velocidad. Se trata de rendimiento y eficiencia. El lenguaje ensamblador le brinda un control completo sobre los recursos del sistema. Al igual que una línea de ensamblaje, usted escribe código para insertar valores individuales en registros, maneja direcciones de memoria directamente para recuperar valores o punteros. (fuente: [codeproject.com](http://www.codeproject.com/Articles/89460/Why-Learn-Assembly-Language) )

Trabajos que solicitan programadores en ensamblador

Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamente

Interfaz de usuario gráfica, Texto, Aplicación, Chat o mensaje de texto

Descripción generada automáticamente

Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

Descripción generada automáticamente

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

1. Velocidad .- Como trabaja directamente con el microprocesador al ejecutar un programa, pues como este lenguaje es el mas cercano a la máquina la computadora lo procesa mas rápido.

2. Eficiencia de tamaño .- Un programa en ensamblador no ocupa mucho espacio en memoria porque no tiene que cargan librerías y demás como son los lenguajes de alto nivel

3. Flexibilidad .- Es flexible porque todo lo que puede hacerse con una máquina, puede hacerse en el lenguaje ensamblador de esta máquina; los lenguajes de alto nivel tienen en una u otra forma limitantes para explotar al máximo los recursos de la máquina. O sea que en lenguaje ensamblador se pueden hacer tareas especificas que en un lenguaje de alto nivel no se pueden llevar acabo porque tienen ciertas limitantes que no se lo permite

Desventajas

Tiempo de programación .- Como es un lenguaje de bajo nivel requiere más instrucciones para realizar el mismo proceso, en comparación con un lenguaje de alto nivel. Por otro lado, requiere de más cuidado por parte del programador, pues es propenso a que los errores de lógica se reflejen más fuertemente en la ejecución.

Programas fuente grandes .- Por las mismas razones que aumenta el tiempo, crecen los programas fuentes; simplemente requerimos más instrucciones primitivas para describir procesos equivalentes. Esto es una desventaja porque dificulta el mantenimiento de los programas, y nuevamente reduce la productividad de los programadores.

Peligro de afectar recursos inesperadamente .- Que todo error que podamos cometer, o todo riesgo que podamos tener, podemos afectar los recursos de la maquina, programar en este lenguaje lo más común que pueda pasar es que la máquina se bloquee o se reinicialize. Porque con este lenguaje es perfectamente posible (y sencillo) realizar secuencias de instrucciones inválidas, que normalmente no aparecen al usar un lenguaje de alto nivel.

Falta de portabilidad.- Porque para cada máquina existe un lenguaje ensamblador; por ello, evidentemente no es una selección apropiada de lenguaje cuando deseamos codificar en una máquina y luego llevar los programas a otros sistemas operativos o modelos de computadoras.