

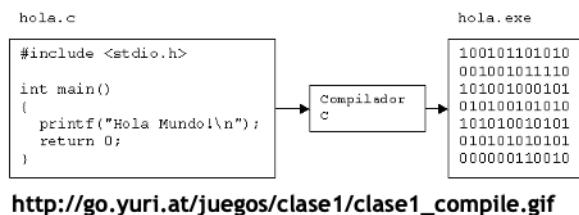
1

Consiste en los programas de instrucciones y datos que definen para el hardware los algoritmos necesarios para la resolución de problemas. Los aspectos más destacados en relación con los S. O. son los siguientes [7, Deitel]:

Programación lenguaje máquina

Lenguaje de máquina Cada tipo de CPU entiende su propio lenguaje de máquina. Las instrucciones en lenguaje de máquina son números almacenados como bytes en memoria.

(Paul A. Carter 2007)



http://go.yuri.at/juegos/clase1/clase1_compile.gif

2

Programación en lenguaje de máquina:

“Lenguaje de máquina”:

- Lenguaje de programación que un computador puede comprender directamente.
- Es “dependiente de la máquina”: un programa en lenguaje de máquina escrito en el computador de un fabricante, generalmente no puede ser ejecutado en el de otro, salvo que su lenguaje de máquina sea compatible.
- Muy poco usado actualmente.

Lenguaje de máquina

Cada tipo de CPU entiende su propio lenguaje de máquina.

Las instrucciones en lenguaje de máquina son números almacenados como bytes en memoria.

Cada instrucción tiene su propio y único código llamado código 5.

Sin embargo, ellas pueden usar una interfaz de bajo nivel (a nivel del kernel).

Las instrucciones del procesador 80x86 varían en tamaño.

El opcode (operation code o código de operación) está siempre al inicio de la instrucción.

Muchas instrucciones incluyen también datos (vgr. constantes o direcciones) usados por las instrucciones.

El lenguaje de máquina es muy difícil de programar directamente.

Descifrar el significado de las instrucciones codificadas numéricamente es tedioso para los humanos.

Por ejemplo, la instrucción que suma los registros EAX y EBX y almacena el resultado en EAX esta codificada por los siguientes códigos hexadecimales 03 C3

Esto no es obvio. Afortunadamente, un programa llamado ensamblador puede hacer este aburrido trabajo para el programador.

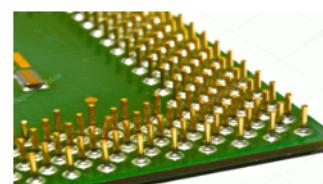
ENSAMBLADORES Y MACROPROCESADORES

Son los pioneros de los compiladores, ya que en los albores de la informática, los programas se escribían directamente en código máquina, y el primer paso hacia los lenguajes de alto nivel lo constituyen los ensambladores.

(Sergio Gálvez Rojas & Miguel Ángel Mora Mata 2005)



<http://jdavideb.blogspot.mx/2012/11/lenguajes-de-bajo-nivel.html>



<https://mx.depositphotos.com/27992317/stock-photo-macro-processor.html>

3

Los “*lenguajes ensambladores*” se desarrollaron para:

- Incrementar la velocidad de programación .
- Reducir los errores de codificación.

Los programas deben ser traducidos al “*lenguaje de máquina*” mediante un programa “*ensamblador*”:

- También es dependiente de la máquina.

Los “*macroprocesadores*”:

- Se desarrollaron para acelerar la codificación de un programa ensamblador.
- Se incorporaron en los ensambladores.
- Una “*macroinstrucción*” indica la ejecución de varias instrucciones en lenguaje ensamblador.
- El “*procesador de macroinstrucciones*” efectúa una “*macroexpansión*” cuando
- lee una macro durante la traducción de un programa:
 - Genera una serie de instrucciones en lenguaje ensamblador correspondientes a la macro.

Son los pioneros de los compiladores, ya que en los albores de la informática, los programas se escribían directamente en código máquina, y el primer paso hacia los lenguajes de alto nivel lo constituyen los ensambladores. En lenguaje ensamblador se establece una relación biunívoca entre cada instrucción y una palabra mnemotécnica, de manera que el usuario escribe los programas haciendo uso de los mnemotécnicos, y el ensamblador se encarga de traducirlo a código máquina puro. De esta manera, los ensambladores suelen producir directamente código ejecutable en lugar de producir ficheros objeto.

Un ensamblador es un compilador sencillo, en el que el lenguaje fuente tiene una estructura tan sencilla que permite la traducción de cada sentencia fuente a una única instrucción en código máquina. Al lenguaje que admite este compilador también se le llama lenguaje ensamblador.

En definitiva, existe una correspondencia uno a uno entre las instrucciones ensamblador y las instrucciones máquina. Ej: Instrucción ensamblador: LD HL, #0100 Código máquina generado: 65h.00h.01h Por otro lado, existen ensambladores avanzados que permiten definir macroinstrucciones que se pueden traducir a varias instrucciones máquina. A estos programas se les llama macroensambladores, y suponen el siguiente paso hacia los lenguajes de alto nivel. Desde un punto de vista formal, un macroensamblador puede entenderse como un ensamblador con un preprocesador previo.

Los “macro procesadores”: se desarrollaron para acelerar la codificación de un programa ensamblador. Se incorporaron en los ensambladores. Una “macroinstrucción” indica la ejecución de varias instrucciones en lenguaje ensamblador. El “procesador de macroinstrucciones” efectúa una “macro expansión” cuando lee una macro durante la traducción de un programa.

COMPILEADORES



<http://ingsistemascompilador.blogspot.mx/p/usos-y-funcion-de-los-compiladores.html>

4

- “*Lenguajes de alto nivel*”: se desarrollaron para resolver el problema de la dependencia respecto a la máquina.
- Permiten el desarrollo de programas “*independientes de la máquina*”.
- Se logra mayor velocidad de programación, programas transportables entre sistemas diferentes y menores requerimientos de conocimientos de hardware.
- “*Compiladores*”: traducen los lenguajes de alto nivel al lenguaje de máquina.
- “*Traductores*”: es la denominación para “*compiladores*” y “*ensambladores*”.
 - Entrada: “*programa fuente*” del programador.
 - Salida: “*programa objeto*” o “*programa resultante*”.

Es aquel traductor que tiene como entrada una sentencia en lenguaje formal y como salida tiene un fichero ejecutable, es decir, realiza una traducción de un código de alto nivel a código máquina (también se entiende por compilador aquel programa que proporciona un fichero objeto en lugar del ejecutable final).

Sistemas de control de entrada/salida

Sistema: es la combinación de componentes que actúan conjuntamente y cumplen un determinado objetivo.

Variable de entrada: es una variable del sistema tal que una modificación de su magnitud o condición puede alterar el estado del sistema.

Variable de salida: es una variable del sistema cuya magnitud o condición se mide.

(David Luis La Red Martínez 2001)

5

- EL IOCS: **input / output control system** libera al programador de aplicaciones de la complejidad de la administración de la e / s:
 - Programas de canal, coordinación de canales y procesadores, control de la e / s, etc.
- Es una manifestación de la tendencia a que los desarrolladores de aplicaciones se concentren en la producción de códigos orientados hacia las aplicaciones y no hacia los sistemas (hardware).

Utilización del SPOOL

Un mecanismo de entrada/salida que permitía que una computadora de propósito específico, mucho más económica y limitada, leyera las tarjetas y las fuera convirtiendo a cinta magnética, un medio mucho más rápido, teniéndola lista para que la computadora central la cargara cuando terminara con el trabajo anterior.

(Gunnar Wolf, Esteban Ruiz, Federico Bergero & Erwin Meza Vega Marzo 2014)

6

Spool: Un mecanismo de entrada/salida que permitía que una computadora de propósito específico, mucho más económica y limitada, leyera las tarjetas y las fuera convirtiendo a cinta magnética, un medio mucho más rápido, teniéndola lista para que la computadora central la cargara cuando terminara con el trabajo anterior. Del mismo modo, la computadora central guardaba sus resultados en cinta para que equipos especializados la leyeran e imprimieran para el usuario solicitante. La palabra spool (bobina) se tomó como acrónimo inverso hacia Simultaneous Peripheral Operations On-Line, operación simultánea de periféricos en línea.

- Un dispositivo de alta velocidad (ej.: disco) se interpone entre un programa en ejecución y un dispositivo de baja velocidad (ej.: impresora) relacionado con el programa en la e / s.
- Evita la demora en la ejecución de programas como consecuencia del uso de periféricos lentos.

Lenguajes orientados hacia el procedimiento versus lenguajes orientados hacia el problema:

► Orientado

- Hacia el procedimiento
- Hacia el problema



Logo de BASIC



11/03/21

7

7

- “O. hacia el procedimiento”: son de propósito general y aptos para resolver gran variedad de problemas:
 - Ej.: Pascal, Cobol, Fortran, Basic, PL/I.
- “O. hacia el problema”: son específicos para resolver determinados tipos de problemas:
 - Ej.: GPSS (simulación), SPSS (estadística).

Compiladores rápidos y sucios versus compiladores optimizadores

► “*rápidos y sucios*”

► “*optimizadores*”

Interpretadores

Cargadores absolutos y de relocalización

Cargadores de enlace y editores de enlace

11/03/21

8

8

- “*C. rápidos y sucios*”: producen rápidamente un programa objeto que puede ser ineficiente respecto de almacenamiento y velocidad de ejecución:
 - Utiles para el desarrollo y prueba de sistemas.
- “*C. optimizadores*”: producen con mayor lentitud un código de máquina altamente eficiente en almacenamiento y ejecución:
 - Utiles en etapa de producción de los sistemas.

Interpretadores:

- No producen un programa objeto.
- Ejecutan directamente un programa fuente.
- Son útiles en ambientes de desarrollo de programas.
- Son más lentos que los códigos compilados.

Cargadores absolutos y de relocalización:

- Los programas se ejecutan en el almacenamiento principal.
- “*Asignación*”: es la asociación de instrucciones y datos con localizaciones particulares de almacenamiento.
- “*Cargador*”: es un programa que coloca las instrucciones y datos de un programa dentro de localizaciones del almacenamiento principal.

- “*Cargador absoluto*”: coloca las instrucciones y datos en las localizaciones específicas indicadas en el programa de lenguaje de máquina.
- “*Cargador de relocalización*”: puede cargar un programa en varios lugares dentro del almacenamiento principal:
 - Depende de la disponibilidad de almacenamiento primario al momento de realizar la carga.
- “*Tiempo de carga*”: momento de realizar la carga.

Cargadores de enlace y editores de enlace:

- El programa en lenguaje de máquina producido por un traductor debe ser combinado con otros programas en lenguaje de máquina para formar una unidad ejecutable.
- La “*combinación de programas*” es realizada por “*cargadores de enlace*” y “*editores de enlace*” antes del tiempo de ejecución del programa.
- “*Cargador de enlace*”: en el momento de carga, combina cualesquiera programas requeridos y los carga directamente en el almacenamiento primario.
- “*Editor de enlace*”: ejecuta la combinación de programas mencionada y además crea una imagen de carga a memoria que preserva en el almacenamiento secundario (disco), para usos futuros:
 - Es muy útil en ambientes de producción, ya que la carga inmediata de la imagen de memoria previamente producida evita un nuevo proceso de combinación de programas previo a cada ejecución.

Preguntas

¿Qué es el lenguaje máquina?

El lenguaje de máquina o código máquina es el sistema de códigos directamente interpretable por un circuito microprogramable, como el microprocesador de una computadora o el microcontrolador de un autómata.

¿Para qué se desarrollaron los macro procesadores?

Para acelerar la codificación de un programa ensamblador.

¿Para qué sirven los compiladores?

Es aquel traductor que tiene como entrada una sentencia en lenguaje formal y como salida tiene un fichero ejecutable, es decir, realiza una traducción de un código de alto nivel a código máquina (también se entiende por compilador aquel programa que proporciona un fichero objeto en lugar del ejecutable final).

¿Qué es una variable de entrada y una variable de salida?

Variable de entrada: es una variable del sistema tal que una modificación de su magnitud o condición puede alterar el estado del sistema.

Variable de salida: es una variable del sistema cuya magnitud o condición se mide.

¿Cuál era la utilización del Spool?

En el modo spool, un dispositivo de alta velocidad como un disco duro se interpone entre un programa en ejecución y un dispositivo de baja velocidad relacionado con la entrada y salida del programa.