

INSTITUTO TECNOLOGICO DE CANCUN.

TÉCNICAS DE DISEÑO.

INDICE

INTRODUCCION.....	3
CLASIFICACION DE LOS LENGUAJES.....	4
BREVE DESCRIPCION DE LOS LENGUAJES.....	
¿QUE SON LOS TRADUCTORES ?.....	
TIPOS DE TRADUCTORES CON EJEMPLOS.....	
¿QUE ES CODIGO FUENTE ?.....	
¿QUE ES CODIGO OBJETO ?.....	
TIEMPO DE COMPILACION.....	
TIEMPO DE EJECUCION.....	

LASIFICACION DE LOS LENGUAJES

Un lenguaje de programación consiste en todos los símbolos, caracteres y regla de uso que permiten a las personas comunicarse con las computadoras.

Los lenguajes de programación deben tener instrucciones que pertenecen a las categorías ya familiares de *entrada/salida, calculo/manipulación de textos, lógica/comparación y almacenamiento/recuperación.*

LENGUAJES MAQUINA

El lenguaje maquina de una computadora consta de cadenas de números binarios (ceros y unos) y es el único que entienden directamente los procesadores. Todas las instrucciones preparadas en cualquier lenguaje maquina tienen por lo menos dos partes. La primera es el comando u operación, que dice a las computadoras cual es la función que va a realizar. Todas las computadoras tienen un código de operación para cada una de las funciones. La segunda parte de la instrucción es el operando, que indica a la computadora donde hallar o almacenar los datos y otras instrucciones que se van a manipular, el numero de operandos de una instrucción varia en distintas computadoras.

LENGUAJES ENSAMBLADORES

A principios de la década de los 50 y con el fin de facilitar la labor de los programadores, se desarrollaron códigos mnemotécnicos para las operaciones y direcciones simbólicas. Los códigos mnemotécnicos son los símbolos alfabéticos del lenguaje maquina.

La computadora sigue utilizando el lenguaje maquina para procesar los datos, pero los programas ensambladores traducen antes los símbolos de código de operación especificados a sus equivalentes en el

lenguaje maquina.

En la actualidad los programadores no asignan números de dirección reales a los datos simbólicos, simplemente especifican donde quieren que se coloque la primera localidad del programa y el programa ensamblador se encarga de lo demás, asigna localidades tanto para las instrucciones como los datos.

Estos programas de ensamble o ensambladores también permiten a la computadora convertir las instrucciones en lenguaje ensamblador del programador en su propio código maquina. Un programa de instrucciones escrito en lenguaje ensamblador por un programador se llama *programa fuente*. Después de que el ensamblador convierte el programa fuente en código maquina a este se le denomina *programa objeto*. Para los programadores es mas fácil escribir instrucciones en un lenguaje ensamblador que en código de lenguaje maquina pero es posible que se requieran dos corridas de computadora antes de que se puedan utilizar las instrucciones del programa fuente para producir las salidas deseadas.

Los lenguajes ensambladores tienen ventajas sobre los lenguajes maquina ; ahorran tiempo y requieren menos atención a detalles. Se incurren en menos errores y los que se cometen son mas fáciles de localizar. Además, los programas en lenguaje ensamblador son mas fáciles de modificar que los programas en lenguaje maquina. Pero existen limitaciones ; la codificación en lenguaje ensamblador es todavía un proceso lento. Una desventaja es que tienen una orientación a la maquina, es decir, están diseñados para la marca y modelo específico del procesador que se utiliza y es probable que para que una maquina diferente se tengan que volver a codificar los programas.

LENGUAJES DE ALTO NIVEL

Los primeros programas ensambladores producían solo una instrucción en lenguaje maquina por cada instrucción del programa fuente. Para agilizar la codificación se desarrollaron programas ensambladores que podían producir una cantidad variable de instrucciones en lenguaje maquina por cada instrucción del programa fuente. Una sola *macroinstruccion* podía producir varias líneas de código en lenguaje maquina.

El desarrollo de las técnicas nemotécnicas y las macroinstrucciones condujo, a su vez, al desarrollo de lenguajes de alto nivel que a menudo están orientados hacia una clase determinada de problemas de proceso.

A diferencia de los programas de ensamble, los programas en lenguaje de alto nivel se pueden utilizar con diferentes marcas de computadoras sin tener que hacer modificaciones considerables. Otras ventajas de los lenguajes de alto nivel son :

- Son mas fáciles de aprender que los lenguajes ensambladores.
- Se pueden escribir rápidamente.
- Permiten tener mejor documentación.
- Son mas fáciles de mantener

Un programador que sepa escribir programas en uno de estos lenguajes no esta limitado a utilizar un solo tipo de maquina.

BREVE DESCRIPCION DE LOS LENGUAJES

BASIC

Beginner's All-purpose Symbolic Instruction Code es un lenguaje interactivo muy popular que tiene una aceptación debido a la facilidad de su uso. Un *lenguaje interactivo* permite la comunicación directa entre el usuario y el sistema de computo durante la preparación y uso de los programas. Aquella persona que tenga que resolver un problema usando una computadora y con poco o ningún conocimiento de las computadoras o

de la programación puede aprender muy pronto a escribir programas en BASIC en una terminal remota o en un teclado o microcomputadora.

FORTTRAN

Formula Transitar es notorio por la facilidad con que permite expresar una ecuación. Muchas de sus características fueron incorporadas más tarde en el primer lenguaje BASIC.

El FORTRAN tiene la ventaja de ser un lenguaje compacto que sirve muy bien para satisfacer las necesidades de los científicos y los estadísticos de los negocios. El lenguaje es también ampliamente utilizado para aplicaciones de negocios que no requieren manejo de grandes archivos de datos.

COBOL

Common Business Oriented Language fue diseñado específicamente para el procesamiento de datos de tipo comercial. Actualmente es el lenguaje más ampliamente utilizado para grandes aplicaciones de negocios.

Una ventaja del COBOL es que puede escribirse en forma muy parecida al idioma inglés y puede emplear términos comúnmente utilizados en los negocios.

PL/I

Programming Language I fue puesto en marcha en la década de los 60 para resolver todo tipo de problemas tanto de negocios como científicos. PL/I es un lenguaje flexible y sofisticado. El elemento básico de este programa es el *enunciado* que termina en *punto y coma*. Los enunciados se combinan en *procedimientos*. Un procedimiento puede representar por completo a un programa pequeño o un bloque de construcción o módulo de un programa más complejo.

RPG

Report Program Operator fue introducido en 1960 como un lenguaje para duplicar rápidamente el enfoque de proceso utilizado con un equipo de tarjeta perforada. Su uso está aún limitado sobre todo para las aplicaciones de negocios que son procesadas en pequeñas computadoras. Como su nombre lo sugiere, el RPG está diseñado para generar los reportes de salida que resultan del proceso de aplicaciones de negocios.

A pesar de las capacidades de actualización de archivos, el RPG es un *lenguaje de propósito limitado* porque los programas objeto generados por el compilador de RPG siguen sin desviación, un ciclo de procesamiento básico.

Una ventaja del RPG es la relativa facilidad para aprenderlo y usarlo. Dado que la lógica de la programación es fija, existen menos reglas formales que en otros lenguajes.

ALGOL, Pascal Y Ada

El ALGOL (ALGORitmic Language) fue presentado en 1958. Fue orientado al uso de quienes participan en proyectos científicos y matemáticos.

Un descendiente del ALGOL es el *Pascal*. Su nombre se le ha dado en honor de Blaise Pascal. Este lenguaje fue elaborado a finales de la década de los 60 y a principios de los 70 por el profesor Nicklaus Wirth. El Pascal fue el primer gran lenguaje creado después de haber sido ampliamente diseminados los conceptos asociados con la programación estructurada.

Otro lenguaje en la línea del ALGOL y el Pascal es el *Ada*. Este lenguaje se llama así en honor a la hija de Lord Byron, Ada Augusta, la condesa de Lovelace. El lenguaje Ada esta patrocinado por el Departamento de Defensa de Estados Unidos para su uso en servicios militares. Este lenguaje fue presentado por su equipo de diseño a finales de los 80. Los críticos lo llamaron inflexible e ineficiente, en tanto que sus favorecedores lo consideraban un gran avance en la tecnología del *software*.

¿QUE SON LOS TRADUCTORES ?

A fines de la década de los 40, cada programa tenia que estar escrito en lenguaje maquina, el único lenguaje que una computadora puede entender directamente. Todos los usuarios tenían que escribir programas compuestos de largas cadenas de ceros y unos para especificar la dirección de información.

Mas tarde, se desarrollaron programas llamados *traductores*, los cuales aceptaban como entrada cierto lenguaje simbólico o mnemotécnico para luego convertirlo automáticamente en lenguaje maquina. Estos traductores se conocen como ensambladores. Los sistemas de ensamblador, aunque ahorran al usuario mucho trabajo, no eran atractivos para la mayoría de los usuarios. Era muy molesto tener que especificar simbólicamente , direcciones y códigos de operaciones. Uno tenía que programar todavía en un lenguaje parecido al lenguaje maquina.

TIPOS DE TRADUCTORES

INTERPRETADORES

La diferencia principal entre un interpretador y un compilador es que un interpretador determina la estructura sintáctica de las oraciones originales una a una, y cuando se produce un pseudo – código de cada oración, este es inmediatamente ejecutado en la pseudo – computadora asociada.

Los interpretadores son traductores populares que poseen las siguientes ventajas sobre los compiladores :

- Son mas fáciles de mecanizar ; y
- Los errores son mas fáciles de encontrar ya que las oraciones se traducen y ejecutan de una en una.

Y las siguientes desventajas :

- No permiten labores, durante la traducción, que requieren examinación como una entidad, del programa que se esta traduciendo ; y
- Son mas lentos en la ejecución.

¿QUE ES CODIGO FUENTE ?

Esta creado en algún lenguaje de alto nivel, por lo que es entendido 100% por el ser humano. Este debe estar complementado por su documentación o manuales donde se indica el desarrollo lógico del mismo.

¿QUE ES CODIGO OBJETO ?

Es creado por los compiladores y nos sirve como enlace entre el programa fuente y el ejecutable.