

Unidad III: Introducción a la Programación

3.1 Características del lenguaje de programación

Un lenguaje de programación es un idioma artificial diseñado para expresar procesos que pueden ser llevadas a cabo por máquinas como las computadoras. Pueden usarse para crear programas que controlen el comportamiento físico y lógico de una máquina, para expresar algoritmos con precisión, o como modo de comunicación humana. Está formado por un conjunto de símbolos y reglas sintácticas y semánticas que definen su estructura y el significado de sus elementos y expresiones. Al proceso por el cual se escribe, se prueba, se depura, se compila y se mantiene el código fuente de un programa informático se le llama programación.

También la palabra programación se define como el proceso de creación de un programa de computadora, mediante la aplicación de procedimientos lógicos, a través de los siguientes pasos:

El desarrollo lógico del programa para resolver un problema en particular.

Escritura de la lógica del programa empleando un lenguaje de programación específico (codificación del programa).

Ensamblaje o compilación del programa hasta convertirlo en lenguaje de máquina.

Prueba y depuración del programa.

Desarrollo de la documentación.

3.2 Estructura básica de un programa.

Como ya se ha visto, cuando se escribe un programa lo que se trata de hacer es indicarle a la computadora cómo ha de resolver un problema. Estos programas siguen determinada estructuras, que dependen del tipo de lenguaje que usemos y del entorno en el cual lo creemos.

Principios generales:

Un cargador - todo programa necesita ser cargado en la memoria por el sistema operativo. De esto se encarga el intérprete.

Definición de los datos - la mayoría de los programas operan con datos y por lo tanto en el código fuente debemos definir qué tipo de datos vamos a utilizar en el programa. Esto se realiza de manera diferente en los distintos lenguajes. Todos los lenguajes que usaremos tienen la posibilidad de crear una nueva definición de datos simplemente al utilizar los datos. Veremos esto en la próxima sección.

Instrucciones - son la parte central del programa. Las instrucciones manipulan los datos que hemos definido, realizan cálculos, muestran los resultados, etc.

3.3 Traducción de un programa: compilación, enlace de un programa, errores en tiempo de compilación

Un compilador es un programa informático que traduce un programa escrito en un lenguaje de programación a otro lenguaje de programación, generando un programa equivalente que la máquina será capaz de interpretar. Usualmente el segundo lenguaje es lenguaje de máquina, pero también puede ser un código intermedio (bytecode), o simplemente texto. Este proceso de traducción se conoce como compilación.]

Un compilador es un programa que permite traducir el código fuente de un programa en lenguaje de alto nivel, a otro lenguaje de nivel inferior (típicamente lenguaje de máquina). De esta manera un programador puede diseñar un programa en un lenguaje mucho más cercano a cómo piensa un ser

humano, para luego compilarlo a un programa más manejable por una computadora.

3.4 Ejecución de un programa

A un programa en ejecución se le suele llamar también proceso.

El proceso de ejecución de un programa escrito en un lenguaje de programación y mediante un compilador tiene los siguientes pasos:

1. Escritura del programa fuente con un editor (programa que permite a una computadora actuar de modo similar a una máquina de escribir electrónica) y guardarlo en un dispositivo de almacenamiento.
2. Introducir el programa fuente en memoria.
3. Compilar el programa con el compilador.
4. Verificar y corregir errores de compilación.
5. Obtención del programa objeto
6. El enlazador (linker) obtiene el programa ejecutable.
7. Se ejecuta el programa y, si no existen errores, se tendrá la salida del programa.

3.5 Elementos del lenguaje: datos, literales y constantes, identificadores, variables, parámetros, operadores, entrada y salida de datos

TIPOS DE DATOS

Todos los lenguajes de programación soportan algún conjunto de diferentes tipos de datos. Un tipo de dato es un conjunto de valores y un conjunto de operaciones definidas para esos valores. Un tipo de dato describe un conjunto de objetos con la misma representación. Existe un número de operaciones asociadas con cada tipo, por ejemplo: es posible realizar aritmética sobre tipos de datos enteros y reales, concatenar con los datos de tipo cadena o recuperar o modificar el valor de un elemento de un arreglo. Un tipo de dato proporciona una descripción de sus instancias que indican al compilador cosas como cuanta memoria se debe asignar para una instancia, cómo representar los datos en memoria y qué operaciones son permisibles sobre esos datos.

Los tipos de datos que se construyen en el compilador, se conocen como tipos de datos fundamentales o predefinidos. Cada lenguaje de programación incorpora una colección de tipos de datos fundamentales, que incluyen normalmente enteros, reales, carácter etc. Los lenguajes soportan también un número de constructores de tipos de datos estructurados o compuestos, que son tipos más complejos

SIMPLES O PRIMITIVOS (Básicos, Fundamentales o predefinidos): Son aquellos que reconoce directamente el lenguaje, sirven de base para otros tipos de datos (complejos), ocupan solo una localidad de memoria, para el almacenamiento de un valor simple, no de objetos complejos.

3.6 Errores en tiempo de ejecución

Los errores en tiempo de ejecución son los que se producen durante la ejecución del programa. Son los más difíciles de encontrar, no son detectados por el compilador, ya que son errores de lógica, no de sintaxis. Aunque al compilar un programa no de errores, el programa puede funcionar incorrectamente y/o dar errores durante su ejecución. Por ejemplo:

- Un programa puede producir resultados erróneos, al equivocarnos (errores lógicos) al programar el algoritmo (sumar en vez de restar, etc.).
- Un programa puede interrumpirse bruscamente, por ejemplo si tenemos que hacer una división y el divisor es cero,etc.

Los errores que se pueden producir en la fase de compilación son:

- Errores fatales: Son raros. Indican errores internos del compilador. Cuando ocurren la compilación se detiene inmediatamente.
- Errores de sintaxis: Son los errores típicos de sintaxis. No detienen la compilación sino que al finalizar ésta se mostrará la lista con todos los errores encontrados. Algunos errores suelen ser consecuencia de otros cometidos con anterioridad. Con este tipo de errores no se puede obtener un programa objeto y por lo tanto tampoco el ejecutable.
- Advertencias o avisos (warnings): Indican que hay líneas de código sospechosas que a pesar de no infringir ninguna regla sintáctica, el compilador las encuentra susceptibles de provocar un error. Cuando se detecta un warning la compilación no se detiene.