Universidad Nacional Autónoma de México

Facultad de ingeniería

FUNDAMENTOS DE PROGRAMACIÓN

Actividad asíncrona 1

Historia de la programación

Sánchez García Rocío

02/10/2020

HISTORIA DE LA PROGRAMACIÓN

Breve historia del cómputo

Uno de los primeros dispositivos mecánicos fue el ábaco, se trataba de un dispositivo que poseía cuentas sobre varillas, de acuerdo con la posición de estas es como se almacenaban los datos. Mas tarde apareció la Pascalina, inventada por Blaise Pascal (1623-1662) y Gottfried Wilhelm Von Leibniz, los datos eran introducidos manualmente y estaban representados mediante las posiciones de los engranajes.

Charles Babbage creó la primera computadora, se trataba de un dispositivo mecánico para efectuar sumas de manera repetida.

Charles Jacquard fue un fabricante de tejidos quien creo un telar que se encargaba de producir modelos de tejidos de manera automática leyendo la información a través patrones de agujeros perforados en tarjetas de papel rígido.

En la Universidad de Harvard se diseño la Mark I en 1944, la cual fue creada por un grupo de personas que fueron dirigidas por Howard H. Aiken, no fue considerada computadora electrónica debido a que su funcionamiento estaba basado en dispositivos llamados relevadores.

Durante 1947 en la Universidad de Pennsylvania la ENIAC por sus siglas en inglés (Electronic Numerical Integrator And Calculator), fue considerada la primera computadora electrónica su diseño fue dirigido por John Mauchly y John Eckert, esta abarcaba todo el sótano de la universidad y requería de un sistema de aire acondicionado, pero a pesar de todo podía realizar cinco mil operaciones aritméticas en un segundo. Gracias a las ideas de Von Neumann, un ingeniero y matemático, posteriormente se avanzó en su desarrollo y es considerado el padre de las computadoras.

Una de las ideas fundamentales de Von Neumann era que en la memoria coexistieran datos con instrucciones, para que la computadora fuera programada con un lenguaje

Historia de la programación

El algoritmo considerado primer programa fue creado por Ada Lovelance entre 1842 y 1843, quien colaboro con Charles Babagge quien es considerado como el "padre de la informática"

ENIAC → Su programación estaba basada en componentes físicos, se cambiaban cables de si sitio para conseguir la programación de la máquina, la entrada y salida de datos era realizada mediante tarjetas perforadas.

Primeros lenguajes de programación, orientadas por las maquinas:

- Algebraic Interpreter: MIT

- A-2: Univac

Primera generación

*Abarco la década de los cincuenta y eran programadas en lenguaje de máquina.

*Apareció la UNIVAC, fue la primera computadora comercial y podía leer cintas magnéticas.

Segunda generación

*Estaban construidas con circuitos de transistores.

*Se programaban en nuevos lenguajes llamados lenguajes de alto nivel.

*Se reduce su tamaño y son de menor costo.

*Algunas computadoras eran programadas con cintas perforadas y otras por medio de cableado en un tablero.

* Los programas eran hechos por un grupo de expertos que se encargaban de resolver los problemas y cálculos solicitados por la administración.

*Aparecen los programas procesadores de palabras como Word Star y se ponen al alcance menús que se encargan de orientar al usuario.

Tercera generación

*Surge en la década de los 1960.

*Su fabricación electrónica estaba basada en circuitos integrados y su manejo es por medio de los lenguajes de control de los sistemas operativos.

*Las computadoras que surgieron durante esta generación se caracterizaban por ser muy potentes y veloces.

A mediados de la década de 1970, aparecieron en el mercado computadoras de un tamaño mediano o minicomputadoras que eran tan costosas como las grandes, pero estás tenían una mayor capacidad de procesamiento.

Cuarta generación

*Aparecen los microprocesadores

*Nacen las computadoras personales y que han influido en la llamada “revolución informática”.

*Surgen aplicaciones como los procesadores de palabras, las hojas electrónicas de cálculo, paquetes gráficos, etc.

Quinta generación

Los objetivos para lograr son:

*Proceso en paralelo mediante arquitecturas y diseños especiales, y circuitos de gran velocidad.

*Manejo de lenguaje natural y sistemas de inteligencia artificial.

Lista de los lenguajes de la programación

- PHP
- Java
- Python
- C/C++
- Javascript
- C#
- Visual Basic. NET
- Objective-C
- Ruby
- Swift
- SQL
- Delphi
- Perl
- Lenguaje de programación R
- Rust
- TypeScript
- Lenguaje de programación Go
- Kotlin
- Scheme
- Erlang
- Elixir
- Pascal
- Postscript
- Haskell
- Scala
- Lava

Clasificación de los lenguajes de programación

Lenguajes de bajo nivel

+Están a un nivel muy cercano a la máquina.

+Las instrucciones del lenguaje son las del microprocesador del ordenador.

+Es difícil de programar

- **Lenguaje maquina:**
  -Se codifican en binario.
  -Los datos se referencian por su posición de memoria.

- **Lenguaje ensamblador:**
  -Codificación mnemotécnica del lenguaje máquina.

-Necesita un traductor.
-Se pueden utilizar etiquetas en vez de posiciones de memoria.

Lenguajes de alto nivel:

+Están basados en máquinas abstractas que facilitan la comprensión del usuario.

+Instrucciones más flexibles.

+Necesitan un traductor para convertir el programa a lenguaje de máquina.

+No depende del procesador.

+Al tener que traducirlo, es mas lento e ineficiente que el lenguaje de bajo nivel.

Clasificación de lenguajes según la administración de memoria

Estáticos: los requisitos de memoria del programa se pueden calcular antes de ejecutar el programa.

Basados en pila: se calculan los requisitos de memoria generales del programa antes de ejecutarlos, el resto de la memoria se utiliza en forma de pila.

Dinámicos: no se puede saber la cantidad de memoria que utilizara el programa. El programa puede crear y destruir estructuras de datos en cualquier lugar del programa.

Por la forma en que se pasa a lenguaje maquina

- Lenguajes compilados
- Lenguajes interpretados

Por el objetivo principal de los programas escritos en el lenguaje

- Lenguajes de propósito general
- Lenguajes para la enseñanza
- Lenguajes para calculo científico
- Lenguajes para gestión
- Lenguajes para la gestión de bases de datos
- Lenguajes de inteligencia artificial
- Programación multiplataforma e internet

Tipos de paradigmas de programación

- **Imperativos**

- ★ Ofrece al programador conceptos que son traducidos de manera natural al modelo de la máquina.
- ★ Su origen se basa en la arquitectura de Von Neumann, consta de una secuencia de celdas en las que se pueden guardar datos e instrucciones.
- ★ Su procesador es capaz de ejecutar de manera secuencial una serie de operaciones, aritméticas y booleanas.
- ★ El programador tiene que traducir la solución abstracta del programa por lo que los programas son más comprensibles para la máquina que para el hombre.

Ventaja: la eficiencia en la ejecución es muy elevada.
Desventaja: resulta complicado escribir el código en lenguaje imperativo.

- **Funcionales**

  - ★ Ofrece conceptos entendibles y fáciles de manejar, en este lenguaje es necesario construir funciones a partir de las ya existentes.
  - ★ Importante conocer y comprender las funciones que conforman la base del lenguaje.

  Desventaja: la eficiencia de ejecución de los intérpretes de lenguajes funcionales es peor que la ejecución de los programas imperativos.

- **Lógicos**

  - ★ También conocidos como lenguajes declarativos ya que al solucionar un problema el programador solo tiene que describirlo con axiomas y reglas de deducción.
  - ★ El conocimiento sobre el problema se expresa en forma de predicados (axiomas) que establecen relaciones sobre los símbolos que representan los datos del dominio del problema.

Ventaja: la formulación del programa es más sencilla y natural.

Desventaja: la eficiencia de la ejecución es inferior a la de un programa equivalente en lenguaje imperativo.

- **Orientados a Objetos**
  - ★ Facilitan la construcción de sistemas o programas en forma modular.
  - ★ Los objetos ayudan a expresar programas en términos de abstracciones del mundo real, lo que aumenta su comprensión.

Desventaja: cuando es interpretado en la arquitectura de Von Neumann conlleva a un excesivo manejo dinámico de memoria debido a la constante creación de objetos, por lo que son

ineficientes, en tiempo y memoria contra los programas equivalentes en lenguajes imperativos, aunque les ganan en comprensión de código.

BIBLIOGRAFÍA Y PÁGINAS DE CONSULTA:

Hernández G. I., Historia de las computadoras
Dirección web: https://www.uv.mx/personal/gerhernandez/files/2011/04/historia-compuesta.pdf

http://informatica.uv.es/iiguia/AED/oldwww/2004_05/AED.Tema.02.pdf

(2011). Historia, Evolución y Futuro de la Programación
Dirección web del trabajo: https://es.slideshare.net/tacubomx/historia-de-la-programacion