

Programación de ordenadores

- 1.- Conceptos de programa y lenguaje
- 2.- Código máquina
- 3.- Lenguaje ensamblador
- 4.- Ventajas e inconvenientes del leng. ensamblador y alto nivel
- 5.- Lenguajes macroensambladores
- 6.- " " alto nivel
- 7.- Entornos de programación
- 8.- Programas traductores
- 9.- Prog. intérpretes
- 10.- Compiladores
- 11.- Análisis lexicográfico
- 12.- " " sintáctico
- 13.- " " semántico
- 14.- Optimización del código
- 15.- Montador de enlaces
- 16.- Lenguajes alto nivel

1.- Conceptos de programa y lenguaje

Programa → secuencia de instrucciones que el ordenador ejecuta una tras otra.

Lenguaje → conjunto de reglas

↳ Tipos {

- Bajo nivel → máquina (binario)
- Alto nivel
- Ensamblador → nombres nemotécnicos

↳ entre bajo y alto nivel.

Ventajas de ensamblador frente al alto nivel {

- Más eficiente, rápido y ocupa menos espacio
- Bueno para tareas específicas

Inconvenientes {

- Conocer arquitectura del hardware
- Saber posiciones de las instrucciones y datos
- Tarea debe de ser descompuesta
- Difíciles de leer y entender y mantener

Leng. macroensambladores {

- Macroinstrucciones ⇒ ahorro de escritura

Problemas {

- Relación con el hardware
- Instrucciones poco potentes
- Funciona en un único ordenador
- Difícil de depurar
- Mayor tiempo de desarrollo

- Clasificación
Leng. a lto
nivel
- Aplicaciones científicas y técnicas
 - " gestión
 - software base y utilidades (S.O.)
 - Aplicaciones inteligencia artificial
 - Imperativos y procedurales
 - Orientados a objetos
 - Declarativos
 - Orientados al problema

- Generaciones
de lenguajes
- 1ª → 2. máquina
 - 2ª → " ensambladores y macroensambladores
 - 3ª → " imperativos
 - 4ª → " con facilidades para programación
 - 5ª → " orientados a inteligencia artificial

- Entorno de
programación
- Edición del programa → código fuente } Mecanografi. / t. real
 - Program. traductores ⇒ codi. fuente \xrightarrow{a} prog. ejecutable
 - Herramientas de depuración del prog. en ejecución

- Programas
traductores
- Intérpretes → convierten cada instrucción y la ejecutan ⇒
⇒ tiempo de ejecución alto
 - Compiladores → 1º traducen todo código fuente
2º ejecutan prog. ejecutable

Componentes
del compilador

- Análisis lexicográfico → Analiza
 - Nombres ilegales
 - Uso incorrecto de palabras
 - " " constantes
- Análisis sintáctico →
 - Frases no correctas
 - Símbolos
- Análisis semántico → incoherencias en el programa
- Optimización del código → elimina instruc. supérfluas
- Montador de enlaces → combina los módulos en un solo programa
 - ↳ tabla de símbolos

Lenguajes
de alto nivel

- Fortran
- Cobol
- Basic
- Pascal
- Lenguaje C