



Programación de Sistemas

Mtro. en IA José Rafael Rojano Cáceres
rrojano@gmail.com
<http://www.uv.mx/rojoano>

Organización del curso

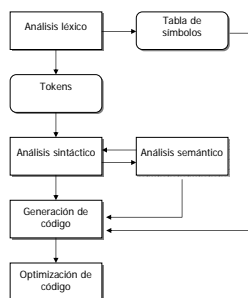
- INTRODUCCIÓN A LA PROGRAMACIÓN DE SISTEMAS
- EVOLUCIÓN DE LOS COMPONENTES DE LA PS
- EDITORES
- ENSAMBLADORES
 - Funcionamiento de un ensamblador
 - La reubicación del código
- Diseño general de un ensamblador
- IA386 PROCESADORES
- LIGADORES Y CARGADORES
 - Clases de cargadores
 - Cargador compile y ejecuta
 - Cargador dinámico
 - Cargador relocable
 - Cargador de ligado directo
 - Cargador dinámico
- COMPILADORES
 - El proceso de compilación
- INTERPRETES
- ORGANIZACIÓN DE UNA COMPUTADORA
- TAXONOMÍA DE LOS SISTEMAS DE WINDOWS
 - Arquitectura de Windows 95
 - Arquitectura de Windows NT
- CARACTERÍSTICAS DE LAS VERSIONES DE WINDOWS
- ARQUITECTURA DE UN SISTEMA UNIX
- LOS SABORES DE WINDOWS Y LOS CONTROLADORES
 - MODO REAL DE WINDOWS
 - MODO ESTÁNDAR DE WINDOWS
 - MODO MEJORADO DE WINDOWS
- MÁQUINAS VIRTUALES
- HARDWARE VIRTUAL
- CONTROLADORES DE DISPOSITIVOS VIRTUALES
- PROGRAMACIÓN DE SISTEMAS EN WINDOWS
 - PROGRAMACIÓN DE 16 BITS
 - PROGRAMACIÓN DE 32 BITS
 - PROGRAMACIÓN DE 64 BITS
- INTERFACES PARA PROGRAMACIÓN DE SISTEMAS
- BIBLIOTECAS DE ENLACE DINÁMICO (DLL)
- PROGRAMACIÓN DE DLL PARA DIFERENTES LENGUAJES
- PROGRAMACIÓN EN C#
 - EJEMPLO DE ACCESO A DISPOSITIVO
- PROGRAMACIÓN DE APLICACIONES EN JAVA
 - EJEMPLO DE ACCESO A DISPOSITIVO
- BIBLIOGRAFÍA
- INFORMACIÓN ACERCA DEL PIC 8259 Y SUS IRQ
- PROGRAMACIÓN, PROGRAMACION Y MAS PROGRAMACION DE USTEDES

Compiladores e Intérpretes

Compilador

- Requiere mayor esfuerzo a diferencia de un traductor, recibe como entrada código fuente de un lenguaje de alto nivel. Entre las tareas que realiza se encuentran:
 1. el código de alto nivel debe ser traducido a más de una instrucción de lenguaje ensamblador, a diferencia del ensamblador a código máquina.
 2. Cada declaración en alto nivel supone innumerables formas sintácticas, la cantidad de operaciones y tipos de datos incluidos.
 3. en los lenguajes de alto nivel existe gran ambigüedad, debido a los diferentes tipos de datos que pueden intervenir en una misma instrucción [Englander 02, Pág. 688].

Proceso de compilación



[Englander 02, Pág. 703]

Análisis léxico

- La etapa del **análisis léxico** se encarga de dividir o explorar cada cadena de entrada que conforma al código con el propósito de agrupar términos significativos (tokens)

Análisis sintáctico

- El **análisis sintáctico** tomará como entrada cada uno de los tokens y verá la relación existente entre ellos para ver si son estructuras sintacticamente bien construidas.

Análisis semántico

- El **análisis semántico** es una fase asociada a la etapa sintáctica, ya que le corresponde determinar si los tokens tienen sentido, por ejemplo si una variable x declarada como entero, se emplea como tal.

Generación de código

- La **generación de código** se considera algunas veces como la etapa final, en ella se determinan localidades y registros de memoria asociado con los objetos del programa. También se genera el código acorde a cada producción del programa. Dentro de esta etapa se incluyen también **las librerías de soporte** para la ejecución del código.

Optimización de código

- La última etapa de **optimización de código** consiste en la verificación del código generado con el fin de ver si las estructuras generadas pueden ser reducidas en tamaño.

Bibliografía

1. [Donovan 72] John Donovan, Systems Programming, McGraw Hill, 1972
2. [Englander 02] Irv Englander, Arquitectura computacional 2da Edición, CECSA, 2002
3. [Jurgens 91] David Jurgens, Help PC 2.10 software de referencia, 1991.
4. [Oney 96] Michael Oney, Systems Programming for Windows 95, Microsoft Press, 1996
5. [Powell 01] Robert Powell, C# and the .NET Framework The C++ perspective, Sams, 2001
6. [Tanenbaum 00] Andrew Tanenbaum, Organización de computadoras un enfoque estructurado, Pearson Education, 2000
7. [Tischer 96] Michael Tischer, PC Interno 5, Marcombo, 1996.
8. [Duran 07] Luis Duran Rodriguez, El gran libro del PC Interno, AlfaOmega, 2007
9. [Conger 92] Conger, James L. , Windows API bible : the definitive programmer's reference, Waite Group Pr