

- Entor d'un sistema de bases de dades

- Concurrencia
- Recuperacions
- Control d'integritats
- Sistemes distribuïts
- Bases de dades orientades a objectes
- Bases de dades documentals

PRACTIQUES

Les pràctiques es realitzaran utilitzant els sistemes RDB i ORACLE sobre MS-DOS.

BIBLIOGRAFIA

DATE, C.J., *Introduction to database Systems*, vol. I 5ª ed., Addison Wesley, 1986.

DATE, C.J., *Introduction to database Systems*, vol II Addison Wesley, 1986.

MARTIN, J. Organización de las bases de datos, Prentice-Hall.

ULLMAN, J.D., *Principles of Databases Systems*, Computer Science pRes, 1980.

1289 - COMPILADORES

1. - Introducció.

- 1.1 Fases de la compilació.
- 1.2 Estructura de un compilador.
- 1.3 Estratègies para la construcció de un compilador.

2. - Anàlisi lexicogràfic y Scanners.

- 2.1. Disseny de un scanner por autòmats finitos.
- 2.2. Disseny pràctic de un scanner.

3. - Anàlisi sintàctic y parsers.

- 3.1. Gramàtiques lliures del context.
- 3.2. Anàlisi Top-down.
- 3.3. Gramàtiques LL1.
- 3.4. Parser Top-down.
- 3.5. Anàlisi botton-up.

4. - Anàlisi semàntic y tabla de símbols.

- 4.1. ¿Que es una tabla de símbols?.
- 4.2. Interfase con una tabla de símbols.
- 4.3. Técnicas básicas de implementación: árboles, tablas hash, etc.
- 4.4. Tablas de símbols estructurados en bloques.
- 4.5. Anàlisi semàntic de declaraciones.
- 4.6. Verificación de tipos.

5. - Organización de memoria en tiempo de ejecución.

- 5.1. Asignación de memoria estática.
- 5.2. Asignación de memoria en la pila.
- 5.3. asignación de memoria dinámica.

6. - Generación de código.

- 6.1. Intrucciones de control de flujo.
- 6.2. Expresiones.

PRACTICAS.

Compilador de un lenguaje reducido.

BIBLIOGRAFIA.

Jean-Paul Tremblay and Paul G. Sorenson, **The Theory and Practiqce of Compiler Writing**, McGraw-Hill.

Charles N.Fisher, Richard J. LeBlanc, jr., **Crafting a Compiler with C**, The Benjamin/Comunming Publishing Company, Inc.

Alfred V. Aho, Ravi Sethi, Jeffrey D. Ullman, **Compiladores: principios, técnicas y herramientas**, Addison-Wesley.

1290 - INGENIERÍA DEL SOFTWARE I

1. Visión general de la Ingenieria del Software.

- 1.1. Software: importancia, características, aplicaciones.
- 1.2. Crisis del software: Problemas y causas.
- 1.3. Paradigmas de la ingenieria del software: ciclo de vida clásico del software, modelo espiral, prototipos, lenguajes de la cuarta generación.

2. Anàlisi de requerimientos y especificación del software.

- 2.1. Fundamentos del análisis de requerimientos.
- 2.2. Especificación.
- 2.3. Anàlisi estructurado.

- 2.4. Análisis orientado al objeto.
- 2.5. Métodos de análisis alternativos y métodos formales.

3. Diseño del software

- 3.1. Fundamentos del diseño del software.
- 3.2. diseño modular efectivo
- 3.3. Diseño de datos.
- 3.4. Diseño arquitectónico.
- 3.5. Diseño procedural.
- 3.6. Documentación.

4. Métodos de diseño.

- 4.1. Diseño orientado al flujo de datos.
- 4.2. Diseño orientado al objeto.
- 4.3. Diseño orientado a las estructuras de datos.

5. Diseño de la interfase de usuario.

- 5.1. Factores humanos.
- 5.2. Estilos de interacción hombre-máquina.
- 5.3. Diseño de interfases.
- 5.5. Interfases standar.

6. Diseño de sistemas de tiempo real.

- 6.1. Consideraciones del sistema.
- 6.2. Sistemas en tiempo real.
- 6.3. Análisis y simulación de sistemas de tiempo real.
- 6.4. Métodos de diseño.
- 6.5. Un método orientado al flujo de datos: DARTS.

PRACTICAS

Análisis de requerimientos y diseño de una aplicación.

BIBLIOGRAFIA

Roger S.Pressman, **Software Engineering, a Practitioner's Approach**, McGraw-Hill (tercera edición).

Roger S. Pressman, **Ingeniería del Software, un enfoque práctico**, McGraw-Hill.

Richard Fairley, **Ingeniería de software**, McGraw-Hill.

1291 - INGENIERÍA DEL SOFTWARE II

1. Codificación de un diseño.

- 1.1. El proceso de traducción.
- 1.2. Características de los lenguajes de programación.
- 1.3. Fundamentos de los lenguajes de programación.
- 1.4. Clases de los lenguajes.
- 1.5. Estilo de codificación.
- 1.6. Eficiencia.

2. Técnicas de prueba del software.

- 2.1. Calidad del software.
- 2.2. Técnicas de prueba del software.

3. Mantenimiento y reutilización del software.

- 3.1. Una definición del mantenimiento del software.
- 3.2. Características del mantenimiento.
- 3.3. Tareas del mantenimiento.
- 3.4. Efectos secundarios del mantenimiento.
- 3.5. Ingeniería inversa.
- 3.6. Administración de configuraciones.

4. Ingeniería de sistemas basados en ordenadores.

- 4.1. Ingeniería de sistemas basados en ordenadores.
- 4.2. Análisis de sistemas.
- 4.3. Modelado de la arquitectura del sistema.
- 4.4 Especificación del sistema.

5. Gestión de un proyecto de software.

- 5.1. Métricas del software.
- 5.2. Estimación.
- 5.3. Planificación.

6. Automatización: CASE.

PRACTICAS

Continuación de la práctica de Ingeniería del software I: codificación y prueba de la aplicación.

BIBLIOGRAFIA

Roger S. Pressman, **Software Engineering, a Practitioner's Approach**, McGraw-Hill (tercera edición).