

Compiladors II

Codi	Tipus	Curs/Semestre	Crèdits
25002	Troncal Semestral	5è / 10è	4,5

Objectius

Competències específiques

Coneixements

- Optimización de código.
- Gestión de memoria.
- Compilación de lenguajes orientados a objeto.
- Compilación de lenguajes funcionales.
- Compilación de lenguajes lógicos.

Habilitats

- Un mayor conocimiento sobre el funcionamiento de los compiladores.
- Aprovechar mejor las capacidades de los compiladores.
- Escribir programas más óptimos y legibles.
- Entender los errores de compilación.
- Depuración más rápida de programas (especialmente en programación orientada a objeto).
- Programación más abstracta.

Competències genèriques

(Mirar el apartado anterior, habilitats)

Capacitats prèvies

Haber cursado la asignatura Compiladors I (conocimientos básicos sobre compiladores). Conocimientos sobre programación imperativa, funcional, lógica y orientada a objetos.

Continguts

0. Introducció de la assignatura	
Temario, forma de evaluación, prácticas, material docente, bibliografía.	

1. Tema 1. Código Intermedio	
Códigos intermedio: Notación polaca. n-tuplas. Árboles sintácticos abstractos. Código enhebrado. Código para máquinas abstracta.	
2. Tema 2. Optimización de Código	
<p>Introducción.</p> <p>Optimización local: Bloques básicos. Ensamblamiento. Propagación de constantes. Reducción de potencia.</p> <p>Eliminación de subexpresiones redundantes.</p> <p>Optimización dentro de bucle: Expansión de bucles. Reducción de frecuencia. Reducción de potencia.</p> <p>Optimización global: Construcción del grafo de flujo. Análisis del grafo de flujo. Aplicaciones de la optimización de programas.</p> <p>Optimización dependiente de máquina: Introducción. Asignación de Registros. Arquitectura máquina y Generación de Código Real</p>	
3. Tema 3. Generación de código para estructuras de control complejas	
Goto no local. Selección de casos. Tratamiento estructurado de excepciones	
4. Tema 4. Compilación de lenguajes orientados a objeto	
Representación de los objetos. Herencia. Sobrecarga. Llamada a métodos estática y dinámica. Construcción y destrucción de objetos. Excepciones y objetos. Templates.	
5. Tema 5. Gestión compleja de memoria	
<p>Heap.</p> <p>Liberación explícita. Algoritmos: Lista ordenadas de bloques libres. Bloques etiquetados en los extremos. Bloques compañeros.</p> <p>Liberación implícita: contador de referencias, garbage collection (marcar y barrer, recolección por copia), compactación</p>	
6. Tema 6. Compilación de lenguajes funcionales	
Diferencias entre lenguajes funcionales e imperativos. Evaluación estricta y diferida. Recursividad en cola. Ámbitos en el Heap. Funciones anónimas y clausuras. Tipos de llamadas.	

7. Tema 7. Compilación de lenguajes lógicos	
Máquina de Warren. No determinismo. Estructura. Construcción. Unificación. Backtracking.	
8. Tema 8. Especificación formal de lenguajes	
Semánticas interpretativas. Gramáticas de atributos.	

Metodologia docent

Teoría: Clases magistrales.

Problemas: no hay clases de problemas.

Prácticas: Implementación de los conceptos adquiridos en teoría

Avaluació

1a convocatòria (febrer/juny)		2a convocatòria (juliol/setembre)
Avaluació en grups	Avaluació individual	
-	- No hay evaluación continuada. - Hay un examen final obligatorio para todos.	- Está abierta a todos.

Bibliografia bàsica

- Advanced Compiler Design & Implementation. Steven S. Muchnick. Morgan Kaufmann Publishers, 1997.
- Building an Optimizing Compiler. Robert Morgan, Butterworth-Heinemann, 1998.
- Modern Compiler Design. Dick Grune et al. Wiley, 2000.
- [Warren's Abstract Machine. A TUTORIAL RECONSTRUCTION](#) (410 Kb). HASSAN AÏT-KACI
- [Intelligent Software Group](#). School of Computing Science. Simon Fraser University. Burnaby, British Columbia. V5A 1S6, Canada.
- The Theory and Practice of Compiler Writing, Jean-Paul Tremblay & Paul G. Sorenson, MCGRAW-HILL, 1985.
- Programming Language Concepts and Paradigms, David A. Watt, Prentice Hall, 1990.
- Théorie des programmes. Schémas, preuves, sémantique, C. Livery, Dunod, 1984.
- Formal Specification of Programming Languages: A Panoramic Primer, Frank G. Pagan, Prentice Hall Inc., 1981.

Bibliografia complementària

- Compiladores: Principios, técnicas y herramientas, Alfred V. Aho, Ravi Sethi & Jeffrey D. Ullman, Addison-Wesley, 1990.
- Programming Language Processors, David A. Watt, Prentice Hall, 1993.
- Functional Programming, Anthony J. Field & Peter G. Harrison, Addison-Wesley, 1988.
- Crafting a Compiler with C, Charles N. Fisher & Richard J. Leblanc jr., The Benjamin / Cummings Publishing Company inc., 1991.
- Programming Languages: Design and Implementation, Terrence W. Pratt, Prentice Hall International Editions, 1984.
- Compiler Construction. An Advanced Course, Edited by G. Goos and J. Hartmanis, Springer-Verlag, 1974.
- The Design of an Optimizing Compiler, William Wulf et al., North Holland, 1980.
- An Implementation Guide to Compiler Writing, Jean-Paul Tremblay & Paul G. Sorenson, MCGRAW-HILL, 1982.
- Writing Compilers & Interpreters: An Applied Approach, Ronald Mak, Wiley, 1991.
- Writing Interactive Compilers and Interpreters, J. P. Brown, Wiley, 1979.
- Compiler Design in C, Allen Y. Holub, Prentice Hall, 1990.
- Garbage Collection, Richard Jones y Rafael Lins, Wiley 1996.

Enllaços

[Enlace de la asignatura](http://www.cvc.uab.es/shared/teach/a20364/c20364.htm)

<http://www.cvc.uab.es/shared/teach/a20364/c20364.htm>