

Codi: 21289

Compiladors

6è Semestre

Enginyeria Tècnica Informàtica
Escola Universitària de Sabadell (UAB)

Curso 2004 - 2005

	Créditos	Horarios
Teoría y Problemas	3	Teoría Martes Aula 23 11:00 13:00h
		Problemas Viernes Aula 23 12:00 13:00h
Prácticas	3	(ver sección prácticas)

	Profesor	Consultas	e-mail
Teoría, problemas y prácticas	Xavier Sánchez	EUIS (P) S/247 Martes 13:00-14:00h	javier@cvc.uab.es
		EUIS (P) S/247 Viernes 13:00-14:00h	
Teoría, problemas y prácticas	Xavier Sánchez	ETSE (P) QC/1024 Jueves 11:00-12:00h	javier@cvc.uab.es
		ETSE (P) QC/1024 Jueves 15:00-16:00h	

(I): semestre impar, (P) semestre par.)

[\[Objetivos \]](#) [\[Temario \]](#) [\[Apuntes y transparencias \]](#) [\[Bibliografía \]](#) [\[Prácticas \]](#) [\[Notas \]](#) [\[Método de evaluación \]](#)

AVISO

La clase del 12-04-2005 de 11:00 a 13:00 de la asignatura de Compiladores queda anulada.

Objetivos de la asignatura

Comprender cómo funcionan los intérpretes y compiladores.

Temario

- **TEMA 1. INTRODUCCION. CONCEPTOS BASICOS**
 - [Lecciones 1 y 2.] **GENERALIDADES.** Código fuente. Código objeto. Gramáticas y lenguajes de programación. Etapas de compilación: análisis lexicográfico, análisis sintáctico, análisis semántico y generación de código. Tabla de símbolos. Compiladores y Recursividad.

- **TEMA 2. ANALISIS LEXICOGRAFICO: SCANNER**
 - [Lección 3.] **ANALISIS LEXICOGRAFICO**. Simbolo. Scanner. Expresiones regulares. Autómatas finitos y scanners.
 - [Lección 4.] **SCANNER**. Consideraciones prácticas. Palabras reservadas. Final del código fuente. Analisis de multiples caracteres hacia delante. Recuperación de errores léxicos. Tabla de símbolos y scanner
- **TEMA 3. GRAMATICAS Y ANALISIS SINTACTICO: PARSER.**
 - [Lección 5.] **GRAMÁTICAS**. Gramáticas Libres del Contexto. Definición de gramática. Clasificación: regulares, libres del contexto, dependientes del contexto, recursivas. Notación BNF extendida
 - [Lecciones 6 y 7.] **NOTACIONES**. Notación. Diagramas Sintácticos. Traducción de BNF a diagramas sintácticos.
 - [Lección 8]. **GRAMÁTICAS Y LENGUAJES DE PROGRAMACIÓN**. Características de las Gramáticas y Lenguajes de Programación. Árbol sintáctico. Ambigüedad gramatical. Estructura de un lenguaje y su gramática
 - [Lección 9]. **PARSERS**. Gramáticas y Parsers LL(1): Análisis un símbolo en adelante sin vuelta a tras. Primeros y siguientes. Definición de gramática LL(1).
 - [Lección] **PARSERS**. Construcción de un parser LL(1): traducción de las reglas BNF en un parser, traducción de los diagramas sintácticos en un parser
 - [Lección 15]. **ERRORES**. Recuperación de Errores.
- **TEMA 4. ANÁLISIS SEMÁNTICO**
 - [Lección 16]. **TRADUCCIÓN DIRIGIDA POR LA SINTAXIS**. Rutinas semánticas, símbolos de acción, atributos.
 - [Lección 17]. **TABLA DE SÍMBOLOS I**. Contenido: símbolos y atributos. Técnicas básicas de búsqueda: secuencial, árbol binario, tabla hash. Estructura de bloques de la tabla de símbolos para: lenguajes no estructurados, lenguajes estrucados por bloques anidados, etc.
 - [Lección 18]. **TABLA DE SÍMBOLOS II**. Representación de atributos recursivos (tipos). Referencias hacia delante. Consideraciones de orden práctico: Creación y destrucción en la tabla de símbolos.
 - [Lección 19]. **ANÁLISIS SEMÁNTICO DE DECLARACIONES**.
 - [Lección 20]. **ANÁLISIS SEMÁNTICO DE EXPRESIONES**.
- **TEMA 5. ORGANIZACIÓN DE MEMORIA EN TIEMPO DE EJECUCION.**
 - [Lección 21]. **AMBIENTE DE EJECUCIÓN**. Organización de memoria de un programa. Asignación estática de memoria: Arrays. Registros. Conjuntos.
 - [Lección 22 y 23]. **PILA DE EJECUCIÓN**. Asignación de la Pila. Registro de activación. Enlaces estáticos y dinámicos. Displays. Llamada y retorno de una función. Resultados intermedios.
 - [Lección 24]. **HEAP**. Asignación Dinámica de Memoria. Liberación explícita. Liberación Implícita: contador de referencias, garbage collection, compactación.
- **TEMA 6. GENERACION DE CODIGO.**
 - [Lección 25]. **GENERACIÓN DE CÓDIGO**. Procesamiento de declaraciones y tabla de símbolos. Compilación de Estructuras de Control.
 - [Lección 26 y 27]. **GENERACIÓN DE CÓDIGO DE EXPRESIONES**. Notación postfija. Pila para los resultados intermedios. Casos: expresiones PASCAL y C.
 - [Lección 28 y 29]. **DIRECCIONES**. Procesar referencias a Estructuras de Datos. Selectores como operadores. Cálculo de la dirección. Casos: estructuras de datos PASCAL y C.

Apuntes y transparencias

Los apuntes de la asignatura se pueden bajar en formato PowerPoint de los siguientes enlaces

- [Tema1.ppt](#)
- [Tema2.ppt](#)
- [Tema3.ppt](#)
- [Tema4.ppt](#)
- [Tema5.ppt](#)
- [Tema6.ppt](#)

- [Tema7.ppt](#)
- [Heap.ppt](#)

Las transparencias con los apuntes del curso se dejarán en fotocopias.

Bibliografía

- **BIBLIOGRAFÍA BÁSICA**
 - **Crafting a Compiler with C**, Charles N. Fisher & Richard J. Leblanc jr., The Benjamin / Cummings Publishing Company inc., 1991.
 - **Modern Compiler Design**. Dick Grune et al. Wiley, 2000.
 - **The Theory and Practice of Compiler Writing**, Jean-Paul Tremblay & Paul G. Sorenson, MCGRAW-HILL, 1985.
 - **Compiladores: Principios, técnicas y herramientas**, Alfred V. Aho, Ravi Sethi & Jeffrey D. Ullman, Addison-Wesley, 1990.
 - **BIBLIOGRAFÍA DE CONSULTA**
 - **Programming Language Concepts and Paradigms**, David A. Watt, Prentice Hall, 1990.
 - **Programming Language Processors**, David A. Watt, Prentice Hall, 1993.
 - **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 all., 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.
-

Prácticas

Enunciado

El objetivo de la práctica es realizar un interprete de un lenguaje especificado en el [enunciado de la práctica](#), al que el alumno puede realizar las ampliaciones que considere interesantes.

Entregas

Para poder realizar un seguimiento de la práctica hay una serie de entregas que se realizarán antes de finalizar la práctica. Estas son:

1. Scanner.
2. Parser.
3. Análisis Semántico
4. Generación de código y entrega final.

Las Entregas se realizarán una semana después de la última sesión de prácticas dedicada a la entrega:

- Primera entrega: 12 de marzo
- Segunda entrega: 16 de abril
- Tercera entrega: 14 de mayo
- Cuarta entrega: antes de la fecha del examen de teoría

Asistencia a Prácticas

La asistencia a las sesiones es **opcional**, pero se pasará **lista** en las sesiones de prácticas para poder realizar un seguimiento sobre el trabajo realizado por el alumno. En caso de duda sobre la nota de la práctica se tendrá en cuenta la asistencia. En las sesiones de prácticas se explicarán las entregas de la práctica, por ello se recomienda su asistencia.

Para los que repiten convocatoria

Pueden entregar la práctica del año pasado adaptándola al enunciado de este año.

Prácticas copiadas

Si se demuestra que algún(os) grupo(s) se ha(n) copiado la práctica de otro(s), todos los grupos implicados automáticamente tendrán suspendida la convocatoria. **suspendida**

Horarios de Problemas y Prácticas

El horario de las sesiones de prácticas es

- Viernes de 10:00 a 12:00
- Viernes de 14:00 a 16:00.

El horario de la clases de problemas es

- Viernes de 12:00 a 13:00

El calendario de prácticas es

- 25-02-2005 Prácticas 01 Entorno CrossVisions+Scanner
- 11-03-2005 Prácticas 02 Scanner
- 18-03-2005 Prácticas 03 Parser
- 01-04-2005 Prácticas 04 Parser
- 15-04-2005 Prácticas 05 Parser
- 22-04-2005 Prácticas 06 Semántico
- 06-05-2005 Prácticas 07 Generación de código
- 13-05-2005 Prácticas 08 Generación de código
- 20-05-2005 Prácticas 09 Generación de código
- 27-05-2005 Prácticas 10 Generación de código

El calendario de problemas es

- 25-02-2005 Problemas 01
- 11-03-2005 Problemas 02
- 18-03-2005 Problemas 03
- 01-04-2005 Problemas 04
- 15-04-2005 Problemas 05
- 22-04-2005 Problemas 06
- 06-05-2005 Problemas 07
- 13-05-2005 Problemas 08
- 20-05-2005 Problemas 09
- 27-05-2005 Problemas 10

Notas

Método de evaluación y normativas

- Para aprobar la asignatura se tienen que presentar las prácticas. En el caso de no presentar la práctica antes de la fecha del examen del junio, el alumno se examinará de teoría.
- Los grupos de prácticas son de dos personas.
- Hay varias entregas de la práctica que se especificarán en el enunciado de prácticas.
- En la entrega final de la práctica se realizará un examen oral sobre la práctica. Este se utilizará para puntuar a los componentes del grupo. Por lo tanto pueden haber notes diferentes.
- No se guardaran notas después de la última convocatoria de la asignatura (septiembre). Sino se ha aprobado la asignatura an junio o septiembre se ha de volver a hacer la asignatura entera (teoría + práctica). Si la práctica del año siguiente es la misma se puede volver a presentar.
- Se suspende la asignatura si se ha suspendido prácticas o el examen.

Dept. Informàtica

Universitat Autònoma de Barcelona
Edifici Q
08193 Bellaterra (Barcelona)
Catalunya - Spain

Javier Sánchez javier@cvc.uab.es
Ultima Modificación: 21-02-2005
Caduca el 30-09-2005