

Curso académico: 2008-2009

Asignatura: Procesadores de Lenguaje

Departamento: Ingeniería del Software e Inteligencia Artificial

Código: UCM-360-98-432 Tipo: Troncal

Nivel: Segundo ciclo

Curso: Cuarto curso

Cuatrimestre: Curso completo

Créditos BOE: 9

Créditos ECTS: 10,1

Horas/semana primer cuatrimestre: 3 horas en aula

Horas/semana segundo cuatrimestre: 3 horas en aula

Profesor: Grupo A: Alfredo Fernández-Valmayor Crespo (desp. 444).

Tutorías: Miércoles de 12:00 a 14:00, Jueves, Viernes de 16:00 a 18:00

Campus Virtual UCM: <http://www.ucm.es/campusvirtual>

Objetivos:

El objetivo de esta asignatura es introducir al alumno en los principios y técnicas utilizadas en la definición de lenguajes de programación y en la construcción de sus procesadores (compiladores o intérpretes).

- El instrumento fundamental para alcanzar este objetivo es la realización de dos trabajos prácticos.
- En el primero se define formalmente un lenguaje de programación elemental (con dos tipos de variables, expresiones e instrucciones de asignación) y se construye manualmente su procesador, incluyendo la máquina virtual que debe ejecutar el lenguaje objeto.
- En el segundo se corrigen los defectos del primer trabajo y se amplía el lenguaje con nuevos tipos de instrucciones, definición de tipos compuestos, procedimientos y funciones. En la implementación del procesador de este lenguaje ampliado pueden utilizarse herramientas de construcción de procesadores (que aunque no se utilicen hay que conocer).

Conocimientos y destrezas que se requieren:

Teoría de Autómatas y Lenguajes Formales: nivel medio; Estructuras de datos: nivel medio; Lenguajes: Prolog, Pascal, C , C++, Java: nivel medio;

Contenidos:

1. Especificación de un lenguaje de programación mínimo y construcción de su procesador.

- Introducción a los lenguajes de programación y a los procesadores de lenguajes: el modelo análisis/síntesis.

- Elementos de la teoría de lenguajes formales. Introducción a las gramáticas de atributos. Definiciones dirigidas por la sintaxis y semántica de los lenguajes de programación.
- Definición del lenguaje fuente:
 - aspectos léxicos
 - aspectos sintácticos (incontextuales)
 - tabla de símbolos
 - restricciones contextuales.
- Definición del lenguaje objeto y máquina virtual
- Especificación formal de la traducción, semántica operacional
- Construcción del procesador. Analizadores recursivos descendentes y procesadores de un solo paso. Transformaciones en las gramáticas de atributos. Esquemas de traducción.
- Revisión:
 - Fases de un compilador.
 - Reingeniería de dos ejemplos de procesadores de lenguajes sencillos.

2. Ampliación del lenguaje mínimo y su procesador.

- Ampliación del lenguaje y ampliación de la máquina virtual
- instrucción compuesta, instrucciones de control y subrutinas,
- definición y construcción de tipos,
- procedimientos y funciones.

3. Herramientas de prototipado y construcción de procesadores de lenguaje.

- Programación lógica: DCGs y procesadores de lenguaje.
- Gramáticas regulares y autómatas finitos: introducción al lenguaje Lex.
- Análisis ascendente: introducción al lenguaje Yacc.

4. Traductores descendentes.

- Estudio de la fase de análisis del compilador y su sincronización con la traducción.
- Gramáticas independientes de contexto y analizadores sintácticos.
- Gramáticas LL(k) y analizadores descendentes.
- Analizadores no recursivos descendentes.
- Traducción en los analizadores no recursivos descendentes.
- Tratamiento de errores durante el análisis.

5. Traductores ascendentes.

- Revisión de la fase de análisis. Analizadores ascendentes.
- Gramáticas LR(k) y analizadores LR.
- Técnicas para construir un analizador LR.
- Tratamiento de errores en el analizador LR.
- Revisión de la traducción dirigida por sintaxis. Evaluación ascendente de las definiciones S-atribuidas.
- Definiciones L-atribuidas y evaluación ascendente de atributos heredados.

Conocimientos y destrezas que se adquieren:

Definir y especificar lenguajes de programación y sus procesadores: nivel medio;
Construir procesadores mediante lenguajes de propósito general: nivel medio; Utilizar herramientas de prototipado y construcción de procesadores: nivel medio; Conocer los fundamentos de las herramientas de construcción de procesadores: nivel medio;

Método docente:

Enseñanza presencial teórica. Enseñanza presencial de ejercicios y supuestos prácticos. Realización en grupo de prácticas no tutorizadas. Exposición y discusión, en clase y mediante las herramientas del campus virtual (foros, correo e intercambio de archivos), de los ejercicios propuestos y de los trabajos de prácticas.

Método de evaluación:

En junio:

1. Participación del alumno en el aula (necesaria): resolución de problemas, preguntas y exposición total o parcial del trabajo de practicas. Se valorará hasta un 20% de la nota final.
2. Realización de prácticas (necesaria): los alumnos deberán realizar 2 prácticas en equipo para entregar al final del 1er. y 2º cuatrimestre, se valorarán hasta un 30% de la nota final.
3. Examen final (obligatorio), 50 % de la nota final. En junio se exige una nota en el examen final de al menos 4 puntos para poder sumar con la nota de prácticas y participación en clase.

En septiembre:

Examen final: 100% de la nota final

Bibliografía:

Aho, A.V. , Sheti, R., Ullman, J.D.; *Compilers. Principles, Techniques and Tools.*; Addison-Wesley. 1988 (existe edición en español);

Wilhen, R. , Maurer, D.; *Compiler Design*; Addison-Wesley. 1995;

Fischer, C.N. , LeBlanc, R.J. Jr.; *Crafting a Compiler with C.* ; Addison-Wesley. 1991;

Complementaria

Sethi, R; *Programming Languages, Concepts and Constructs*; 2nd. Edition. Addison-Wesley. 1996;