



UNIVERSIDAD DE LAS PALMAS
DE GRAN CANARIA

GUÍA DOCENTE

CURSO: 2015/16

12695 - PROCESADORES DE LENGUAJES

ASIGNATURA: 12695 - PROCESADORES DE LENGUAJES

CENTRO: Escuela de Ingeniería Informática

TITULACIÓN: Ingeniero en Informática

DEPARTAMENTO: INFORMÁTICA Y SISTEMAS

ÁREA: Lenguajes Y Sistemas Informáticos

PLAN: 10 - Año 199 **ESPECIALIDAD:**

CURSO: Quinto curso **IMPARTIDA:** Primer semestre **TIPO:** Troncal

CRÉDITOS: 4,5

TEÓRICOS: 1,5

PRÁCTICOS: 3

Descriptores B.O.E.

Fase de Compilación.

Optimización de Código. Macroprocesadores.

Temario

Nota: El número de horas se mantiene a efectos informativos para el estudiante para la estimación del peso de cada módulo dentro de la asignatura.

I. INTRODUCCIÓN (~1h)

II. GENERACIÓN DE ANALIZADORES LÉXICOS (~1h) [Grune, Levine]

III. GENERADOR DE ANALIZADORES SINTÁCTICOS (~2h) [Aho, Levine]

1. Especificación y generación

2. Ambigüedad

3. Tratamiento de errores

IV. ANÁLISIS SEMÁNTICO (~2h) [Tremblay, Grune]

1. Tabla de símbolos

2. Chequeo de tipos

3. Representaciones internas

V. GENERACIÓN DE CÓDIGO (~7h) [Aho, Fischer]

1. Expresiones y asignaciones

2. Estructuras de control

3. Estructuras de datos

4. Rutinas y registros de activación

VI. INTRODUCCIÓN A LA OPTIMIZACIÓN (~2h) [Morgan, Cooper]

Requisitos Previos

Conocimientos básicos en lenguajes de programación, compiladores e intérpretes. Experiencia en programación con lenguajes de alto y de bajo nivel.

Objetivos

El objetivo principal es el de complementar la formación adquirida en el conocimiento y diseño de procesadores y lenguajes, principalmente en la asignatura de Compiladores. Se pretende que el alumno consolide y amplíe dichos conocimientos, mediante la exposición teórica de técnicas y elementos nuevos, y mediante una actividad práctica que incorpore diseño e implementación. El énfasis es principalmente práctico, junto con la presentación de los elementos teóricos fundamentales, mediante el uso de herramientas actuales, que posibiliten una continuidad en la autoformación en este campo del futuro titulado. Consideraremos asimismo importante fomentar la propia iniciativa, el espíritu crítico y la capacidad de exposición y debate de las ideas.

Metodología

Si bien se desea dotar al alumno de unos fundamentos teóricos sólidos, el enfoque de la asignatura es práctico y realista, mediante la realización de un único trabajo de un procesador completo de lenguaje. Dentro de este enfoque realista, se pone énfasis en la planificación, el seguimiento de la realización, la documentación, y su defensa. Pretendemos motivar al alumno en este aspecto central de la informática ofreciéndole flexibilidad (supervisada) en la especificación de los objetivos y el diseño, y con el fomento y valoración de la iniciativa, novedad y originalidad.

Criterios de Evaluación

Para la superación del curso existirá una única prueba teórica recuperable y una evaluación del trabajo realizado, ambos necesarios y con igual ponderación final. En los trabajos se valorará el esfuerzo individual, la iniciativa, la aplicación de las técnicas y herramientas apropiadas, la orientación práctica y la regularidad en su desarrollo temporal.

Descripción de las Prácticas

Se desarrollará a lo largo del cuatrimestre un procesador de lenguaje, en el que normalmente habrán de aplicarse las técnicas expuestas en clase. Habrán de diseñarse primeramente los aspectos léxico, sintáctico y semántico, en su caso, del lenguaje a implementar (~4h), y, tras una primera toma de contacto con el entorno de generación, se procederá fundamentalmente al desarrollo de las fases de análisis léxico (~4h), sintáctico (~6h), semántico (~8h) y de generación de código (~8h). Como resultado se obtendrá finalmente un procesador funcional.

La documentación final describe el lenguaje, la metodología empleada, los procedimientos para la generación y utilización del código generado, ejemplos de los mismos, así como los códigos fuente y objeto en formato informático.

Bibliografía

[1 Básico] Compilers: principles, techniques and tools /

Alfred V. Aho ...[et al.].
Addison Wesley,, Boston : (2007) - (2nd ed.)
0321491696 (ed. int.)

[2 Básico] Modern compiler design /

Dick Grune, Henri E. Bal, Criel J.H. Jacobs, Koen G. Langendoen.
John Wiley & Sons,, Chichester : (2000)
0-471-97697-0

[3 Recomendado] Crafting a compiler with C /

Charles N. Fischer, Richard J. LeBlanc.
Benjamin/Cummings,, Redwood City (California) : (1991)
0805321667

[4 Recomendado] Theory and practice of compiler writing /

Jean-Paul Tremblay, Paul G. Sorenson.
McGraw-Hill,, New York : (1985)
0070651612

[5 Recomendado] Engineering a Compiler /

Keith Cooper a Linda Torczon.
Morgan Kaufmann,, San Francisco : (2003)
1-55860-699-8(paperback)

[6 Recomendado] Lex & yacc.

Levine, John
O'Reilly and Associates,, Sebastopol, CA : (1995) - (2nd. ed.)
1565920007

[7 Recomendado] Programming language pragmatics /

Michael L. Scott.
Morgan Kaufmann,, San Francisco : (2000)
1-55860-578-9

[8 Recomendado] Compiler design /

Reinhard Wilhelm, Dieter Maurer.
Addison-Wesley,, Wokingham (England) : (1995)
0201422905

[9 Recomendado] Building an optimizing compiler /

Robert Morgan.
Butterworth - Heinemann,, Boston [etc.] : (1998)
155558179X

Equipo Docente

JOSÉ FORTES GÁLVEZ

(COORDINADOR)

Categoría: TITULAR DE UNIVERSIDAD

Departamento: INFORMÁTICA Y SISTEMAS

Teléfono: 928458724 **Correo Electrónico:** jose.fortes@ulpgc.es

Resumen en Inglés

Subjects covered are lexical analysis (Flex), syntactic analysis (Bison), symbol tables and type checking, code generation, and introduction to optimization.

It is required previous knowledge of high-level imperative programming language concepts and programming, as well as assembly-level architecture.

The course is practice-oriented. The student (usually within a small team, with the assistance of the teacher) has to progressively fully implement and document a simple non-optimizing, although useful, compiler for a language to be defined by the team itself.

Additionally, a theory exam needs to be passed. The final grade obtained is calculated as the mean between the theory exam and a practical work.