

Procesadores de Lenguajes II

Profesorado

- ❑ José Antonio Troyano Jiménez (Coordinador)
- ❑ Rafael Corchuelo Gil
- ❑ Vicente Carrillo Montero

Objetivos y requisitos

Con esta asignatura se pretende que el alumno adquiriera los conocimientos básicos para la construcción metódica de procesadores de lenguajes y que adquiriera algunos conceptos fundamentales relacionados con el procesamiento del lenguaje natural.

Los objetivos generales de la asignatura se pueden resumir en los siguientes puntos:

- ❑ Construcción metódica del comprobador de tipos de un compilador.
- ❑ Construcción metódica de generadores de código portables.
- ❑ Estudio de algunos conocimientos generales sobre las técnicas de procesamiento de lenguaje natural.

Evidentemente, ninguno de estos objetivos se puede conseguir si el alumno no tiene una buena base en los temas tratados en la asignatura Procesadores de Lenguajes I, en los conceptos básicos de programación basada en tipos abstractos de datos o en conceptos generales acerca de los lenguajes de programación.

Todo el trabajo práctico de esta asignatura se desarrolla en lenguaje C, por lo que el alumno, evidentemente, debe contar con unos conocimientos sólidos de este lenguaje y con habilidad suficiente como para desarrollar una aplicación de moderada envergadura.

Temario

1. El problema del contexto
2. Comprobación de tipos
3. Generación de código
4. Introducción a la optimación de código
5. Introducción al procesamiento del lenguaje natural
6. Formalismos gramaticales
7. Esquemas de parsing
8. Estrategias de parsing
9. Rasgos. Algoritmos de unificación

Metodología

La asignatura se organizará en torno a clases teóricas y de laboratorio. En las clases teóricas se expondrán los contenidos de la asignatura, desarrollando el temario anteriormente propuesto. Se presentarán los conceptos teóricos necesarios para entender las distintas técnicas y herramientas, así como ejemplos y casos prácticos que ayuden a asimilar dichos conceptos. En las clases de laboratorio se utilizarán dichas herramientas, aplicándose en la realización de pequeños problemas.

Durante el curso se propondrá la realización de una práctica que afianzará los conceptos adquiridos por el alumno a lo largo de la asignatura. El enunciado de la práctica constará de una descripción de un lenguaje de programación para el que se construirá un compilador.

La práctica de la asignatura se podrá desarrollar en grupos de trabajo que incorporen cualquier número de personas. No obstante, a partir de tres miembros en el equipo, la nota obtenida se multiplicará por el factor de corrección $3/N$, siendo N el número de personas que presentan la práctica de forma conjunta.

Evaluación

A la hora de evaluar y calificar a los alumnos atenderemos a dos factores: por un lado el trabajo realizado a lo largo del curso en el desarrollo de la práctica y por otro lado la calificación obtenida en el examen final. La nota definitiva de la asignatura se calculará con la siguiente fórmula:

$$Nota\ final = 0'3 * Nota\ Practica + 0'7 * Nota\ Examen$$

Aquellos alumnos que lo deseen, podrán realizar trabajos prácticos que complementarán su nota. Antes de realizar estos trabajos, los alumnos interesados deberán ponerse en contacto con cualquiera de los profesores de la asignatura para que éste le informe sobre los trabajos a desarrollar y la valoración de los mismos. Es muy importante tener en cuenta que estos trabajos podrán entregarse, como máximo, un mes antes de la realización del examen de la asignatura. Bajo ningún concepto se admitirán trabajos después de esa fecha.

Bibliografía

- A.V. Aho, R. Sethi, J.D. Ullman, Compiladores. Principios, Técnicas y Herramientas, Compiladores. Principios, Técnicas y Herramientas.
- C.N. Fischer, R.J. Leblanc. Crafting a Compiler with C, Benjamin/Cummings Publishing Company, Inc. Redwood City, California, 1991.
- A.I. Holub, Compiler Design in C, Prentice-Hall, Inc. Englewood Cliffs, New Jersey, 1990.
- B.W. Kernighan, P.J. Plauger, Software Tools in Pascal, Addison-Wesley Publishing Company, Reading, Massachusetts, 1981.
- B. Meyer, Introduction to the Theory of Programming Languages, Prentice-Hall International Series in Computer
- R. Sethi, Lenguajes de programación. Conceptos y constructores, Addison-Wesley Iberoamericana, Wilmington, Delaware, 1992.

- ❑ J.P. Tremblay, P.G. Sorenson, The Theory and Practice of Compiler Writing, McGraw-Hill International Editions, 1985.
- ❑ D.A. Watt, Programming Language Processors, Prentice-Hall International Series in Computer Science, 1993.
- ❑ K. Sikkel, Parsing Schemata, Ed. Springer
- ❑ J. Allen, Natural Language Understanding, Ed. Benjamin/Cummings