

Plan 254 Ing. en Informática

Asignatura 14009 PROCESADORES DE LENGUAJE

Grupo 1

Presentación

Lenguajes de programación. Compiladores. Intérpretes. Traductores. Análisis léxico. Análisis sintáctico. Semántica de lenguajes de programación. Generación y optimización de código.

Programa Básico

Objetivos

El objetivo global de la asignatura es presentar los conceptos, técnicas y herramientas útiles para la construcción de procesadores de lenguajes y, más en concreto, compiladores e intérpretes de lenguajes de programación. En este sentido, se trata de poner en práctica muchos de los conceptos ya adquiridos en otras asignaturas relativos a la teoría de autómatas y lenguajes formales y de relacionar la potencia de los modelos de cálculo más habituales con las tareas de análisis estructural y de significado relacionadas con el proceso de lenguajes. Como objetivos particulares, se proponen los siguientes:

- Conocer la arquitectura general y tipología de los procesadores de lenguajes.
- Comprender los fundamentos, metodología de construcción y herramientas necesarias para el diseño e implementación de sistemas capaces de obtener una representación sintáctica abstracta a partir de una entrada textual, tanto en su vertiente léxica como en su vertiente sintáctica.
- Estudiar los problemas relacionados con la generación de significado a partir de código estructural abstracto (AST) y, más en concreto, los derivados de una adecuada gestión del contexto:
 - Sistemas de tipos
 - Gestión de memoria
 - Entorno de ejecución
 - Generación de código intermedio
 - Introducir al alumno en los problemas relacionados con la generación de código final y la optimización de código, tanto local como global.

1. Introducción

1. Compiladores y Proceso de Compilación.
2. Lenguajes de Programación: Características.
3. Aspectos formales: lenguajes y máquinas.

2. Análisis léxico.

3. Análisis sintáctico determinista.

1. Análisis descendente LL
2. Análisis ascendente de precedencia y LR

4. Representación de la información en compilación

1. Tablas de símbolos
2. Árboles de sintaxis abstracta

5. Análisis semántico:

1. Gramáticas de atributos
2. Sistemas de tipos

6. Generación de código

7. Gestión de memoria y entornos de ejecución.

8. Introducción a la optimización de código.

Programa Práctico

Durante el curso, los alumnos deberán realizar, en grupos de 4 personas, un compilador y/o intérprete para un lenguaje de programación orientado a tarea. La actividad incluye tres fases, que deberán haberse completado antes de las vacaciones de navidad:

1. definición del lenguaje de acuerdo con el dominio elegido (se propone el de tratamiento digital de imagen, pero puede haber alternativas)
2. análisis y generación de AST
3. generador de código e intérprete

Todos los detalles sobre plazos y condiciones se consultan en la página web de la asignatura.

Evaluación

La parte teórica se evaluará en prueba escrita que incluirá ejercicios y cuestiones cortas, siempre relacionadas con los contenidos expuestos en el programa. El resultado de esta prueba representará el 60% de la nota final de la asignatura.

La evaluación de la parte práctica incluye la valoración del trabajo práctico y representará el 40% sobre la nota final.

Para poder promediar la nota de teoría y de prácticas, será preciso haber obtenido una calificación de al menos 4 puntos sobre 10 en teoría y de 5 puntos sobre 10 en la de prácticas.

La evaluación de la actividad de preparación y participación en los seminarios supondrá hasta un 10% de la parte de prácticas y, caso de resultar algún sobrante, podrá ser repercutida en la componente de teoría.

Las notas de teoría y de prácticas se conservarán por separado entre las convocatorias de Junio y Septiembre.

Bibliografía

1. AHO A., SETHI R., ULLMAN J. Compiladores: Principios, Técnicas y Herramientas. Addison Wesley.
2. APPEL, A. Modern Compiler Implementation in C. Cambridge University Press, 1998.
3. GRUNE D., BAL, H.E., Modern Compiler Design. Wiley & Sons, 2000.
4. TREMBLAY J.P. SORENSON P.G. The Theory and Practice of Compiler Writing. McGraw-Hill 1985.
5. WAITE W.M., GOOS G. Compiler Construction. Springer-Verlag 1985.