



PROGRAMA DEL CURSO: Compiladores

TIPO: Obligatoria

CÓDIGO: ISPCMP

TPLU: 4 0 2 5

PRELACIÓN: Teoría de la Computación

UBICACIÓN : 8^{vo} semestre

CICLO: Profesional

JUSTIFICACIÓN

Con este curso se pretende que el estudiante adquiera los conocimientos básicos del funcionamiento de un compilador y las técnicas que se emplean en la construcción de los mismos, con la finalidad de disponer de una herramienta más para el diseño de cualquier otro tipo de sistema automatizado.

OBJETIVOS

- Desarrollar habilidades en la construcción de analizadores léxicos de baja complejidad.
- Desarrollar habilidades en la construcción de analizadores sintácticos de baja complejidad.
- Desarrollar habilidades en la aplicación de los conocimientos adquiridos sobre gramáticas formales y autómatas al proceso de diseño de lenguajes y a su traducción.

CONTENIDO PROGRAMÁTICO

Unidad I: Introducción al proceso de compilación

- Tema 1. Lenguajes y tipos de traductores.
- Tema 2. Estructura de un compilador y fases de compilación.
- Tema 3. Herramientas para el desarrollo de un compilador.

Unidad II: Análisis léxico

- Tema 1. Concepto, funciones, especificación y reconocimiento de componentes léxicos.
- Tema 2. Lenguajes y gramáticas.
- Tema 3. Expresiones regulares.
- Tema 4. Autómatas finitos no deterministas.
- Tema 5. Autómatas finitos deterministas.
- Tema 6. Diseño e implementación de analizadores léxicos.

Unidad III: Análisis sintáctico

- Tema 1. Concepto. Funciones.
- Tema 2. Gramáticas independientes del contexto.
- Tema 3. Árboles de análisis sintáctico y derivación.
- Tema 4. Análisis Sintáctico Descendente: gramáticas LL, construcción de analizadores de tipo LL, tratamiento y recuperación de errores.

Tema 5. Análisis Sintáctico Ascendente: gramáticas LR, construcción de analizadores de tipo LR, tratamiento y recuperación de errores.

Tema 6. Generadores de analizadores sintácticos.

Unidad IV: Análisis semántico

Tema 1. Concepto y funciones.

Tema 2. Tablas de símbolos.

Tema 3. Gramáticas con atributos.

Tema 4. Semántica estática.

Tema 5. Declaraciones, tipos y restricciones contextuales.

Tema 6. Transformación de árboles.

Unidad V: Generación de código

Tema 1. Generación de código intermedio: Conceptos, ventajas y representaciones intermedias.

Tema 2. Generación de código: Conceptos y funciones.

Tema 3. Máquinas virtuales.

Tema 4. Organización de la memoria y representación de datos.

Tema 5. Representación de estructuras de control.

Tema 6. Optimización local de código

METODOLOGÍA DE ENSEÑANZA

La enseñanza de este curso se realizará a través clases teórico-prácticas y clases guiadas en el laboratorio.

RECURSOS

- Recursos multimedia: proyector multimedia, proyector de transparencias.
- Computadora portátil
- Guías disponibles en Publicaciones de la Facultad de Ingeniería.
- Laboratorio bien dotado de computadoras para realizar la parte práctica de la materia.
- Acceso a Internet

EVALUACIÓN

Serán evaluados los siguientes aspectos:

- Asistencia
- Participación en clase
- Evaluación del conocimiento teórico a través de pruebas parciales escritas
- Evaluación del conocimiento práctico a través de prácticas de laboratorio
- Evaluación del conocimiento práctico a través de una prueba en el laboratorio al final del semestre.

BIBLIOGRAFÍA

Aho, R. y Ullman, J. Compiladores: Principios, Técnicas y Herramientas, Addison-Wesley, 1990.

Aho, R. y Ullman, J. The Theory of Parsing, Translation and Compiling, Vol 1 y 2. Prentice Hall, 1972.

Appel, A. y Ginsburg, M. Modern compiler implementation in C. Cambridge University Press, 1998.

Muchnick, S. Advanced Compile Design and Implementation. Morgan Kaufmann Publishers, 1997.

Pittman, T. y Peters, J. Art of Compiler Design. The Theory and Practice. Prentice Hall, 1997.

Tremblay, J. y Sorenson, P. The Theory and Practice of Compiler Writing, McGraw-Hill, 1985.

Teufel, B.; Schmidt, S. y Teufel, T. Compiladores: Conceptos fundamentales. Addison-Wesley. 1995.