

Programa de Asignatura

Ingeniería en Computación - In.Co.

Nombre de la Asignatura	Diseño de Compiladores.
Créditos	12.
Objetivo de la Asignatura	<p>Introducir al estudiante a las distintas técnicas y herramientas a usar en el diseño y construcción de compiladores.</p> <p>Presentar los problemas que surgen al enfrentarse a las distintas características que ofrecen los lenguajes de programación (tipos de datos, pasaje de parámetros, anidamiento de procedimientos, etc.).</p> <p>Construcción de un pequeño compilador o traductor.</p>
Metodología de enseñanza	<p>El curso posee una duración de 14 semanas con 4 horas semanales de teórico durante las primeras 7 semanas. En la segunda mitad el estudiante debe dedicar un total de 40 horas de trabajo de laboratorio. Durante todo el curso hay clases prácticas de 2 horas semanales.</p>
Temario	<p>Introducción:</p> <ul style="list-style-type: none"> Contexto de un compilador Análisis lexicográfico, sintáctico y semántico Detección de errores Áreas de aplicación Herramientas para la construcción de compiladores <p>Análisis lexicográfico:</p> <ul style="list-style-type: none"> Función del analizador lexicográfico Definición de tokens, patterns, y lexemes Diagramas de transición y su implementación <p>Análisis sintáctico:</p> <ul style="list-style-type: none"> Función del analizador sintáctico Árboles de reconocimiento y derivaciones Ambigüedad, recursión por la izquierda, factorización Reconocimiento top down Gramáticas LL(1) Traducción dirigida por la sintaxis (gramáticas de atributos) <p>Análisis semántico:</p> <ul style="list-style-type: none"> Sistemas de tipos (expresiones de tipo, chequeo estático y dinámico, recuperación de errores). Especificación de un chequeador de tipos Equivalencia de tipos Sobrecarga de operadores

Ambiente en tiempo de ejecucion:
Organizacion de la memoria (registros de activacion, division de la memoria en tiempo de ejecucion, objetos locales).
Estrategias de asignacion de memoria
Acceso a los objetos globales

Generacion de código intermedio:
Codigo de 3 direcciones
Implementacion de sentencias en codigos intermedios
Alcance de las variables
Sentencia de asignacion
Expresiones booleanas

Generacion de codigo:
Manejo de la memoria
Seleccion del conjunto de instrucciones
Asignacion de registros
Generacion de codigo para las instrucciones
Manejo de la memoria en tiempo de ejecucion

Optimizacion

Bibliografía

Alfred V. Aho, Ravi Sethi and Jeffrey D. Ullman, Compilers Principles, techniques and tools. ISBN: 0-201-10088-6

Conocimientos previos exigidos y recomendados

Los adquiridos al cursar y aprobar las materias previas.

Anexo

1) Cronograma tentativo:

- Semana 1 - Introducción
- Semana 2 - Análisis Lexicográfico
- Semana 3 - Función del analizador sintáctico
- Semana 4 - Arboles de reconocimiento y derivaciones
- Semana 5 - Reconocimiento top-down y Gramáticas LL(1)
- Semana 6 - Análisis Semántico y Generación de código intermedio
- Semana 7 - Generación de código y Optimización.
- Semana 8 - Examen Parcial.
- Semana 9 - Supervisión del Trabajo Obligatorio
- Semana 10 - Supervisión del Trabajo Obligatorio
- Semana 11 - Supervisión del Trabajo Obligatorio
- Semana 12 - Supervisión del Trabajo Obligatorio
- Semana 13 - Supervisión del Trabajo Obligatorio
- Semana 14 - Supervisión del Trabajo Obligatorio
- Semana 15 - Presentaciones de los Obligatorios.

2) Modalidad del curso y procedimiento de evaluación

Se realizará una prueba escrita al terminar el dictado de la parte teórica del curso. Además, se ha de exigir el diseño e implementación de un traductor en grupos de 2 o 3 personas. Se deberá presentar la documentación completa del trabajo realizado y no se descarta la presentación oral del mismo.

6
Custro

Carga horaria: 6 horas semanales.

3)Materia

Programación.

4) Previaturas

Programación 3 y 4 (examen)

Taller de Programación

Teoría de Lenguajes (examen)

5) Cupo

No tiene.

Aprobado por Res. del Consejo de fecha 28.7.00 - Exp.93121