



Organización de Lenguajes y Compiladores 1

Código:	777	Créditos:	4
Escuela:	Ciencias y Sistemas	Área a la que pertenece:	Ciencias de la Computación
Pre-requisito:	771 – Introducción a la programación y Computación 2 796 – Lenguajes Formales y de Programación 962 – Matemática para Computación 2	Post-requisito:	781 – Organización de Lenguajes y Compiladores 2
Categoría:	Obligatorio	Semestre:	Segundo 2014
Catedrático:	Ing. Mario Bautista	Auxiliar:	Diego Solis
Edificio:	T-3	Sección:	A
Salón del curso:	212	Salón del laboratorio:	216
Horas por semana del curso:	4	Horas por semana del laboratorio:	2
Días que se imparte el curso:	Martes y Jueves	Días que se imparte el laboratorio:	Viernes
Horario del curso:	07:10 – 08:50	Horario del laboratorio:	13:10 – 14:50

Descripción del curso:

Organización de Lenguajes y Compiladores 1 implementa los principios básicos de un compilador y/o intérprete, tomando como referencia diferentes herramientas de programación, describiendo las fases de cada proceso.

En el laboratorio se trata en detalle las primeras fases del proceso: análisis lexicográfico y análisis sintáctico. Así también el manejo de errores y de la tabla de símbolos.

Objetivo General:

Que el estudiante adquiriera experiencia en el uso de los diferentes tipos de herramientas de programación para el proceso de compilación.

Objetivos Específicos:

1. Que el estudiante aprenda en detalle las primeras fases del proceso de compilación, principalmente el análisis lexicográfico y el análisis sintáctico.
2. Capacitar y ejercitar al estudiante en los principios de análisis, diseño e implementación de compiladores.

Metodología:

Clase magistral una vez por semana.
Evaluaciones de rendimiento.
Tareas de investigación.
Prácticas y Proyectos de programación.

Evaluación:

El laboratorio tiene un valor de 40 puntos en la nota final del curso. La nota mínima de promoción es del 61% (24.4 puntos) y es OBLIGATORIO aprobar el laboratorio para aprobar el curso.

Tareas, Prácticas y Cortos	10 pts.
2 Proyectos	30 pts.

Total	40 pts.

Contenido:

Está comprendido en diez (10) unidades:

1. Procesador de Lenguaje

- 1.1. Diferencia entre intérprete y compilador
- 1.2. Tipos de compiladores
- 1.3. Fases de análisis
- 1.4. Fases de síntesis
- 1.5. Detección de errores

2. Análisis Léxico

- 2.1. Análisis lexicográfico
- 2.2. Función del analizador
- 2.3. Componentes léxicos, patrones y lexemas
- 2.4. Autómata

- 2.5. Expresiones regulares
- 2.6. Tratamiento de errores léxicos

3. Analizador Léxico (Jlex/JFlex)

- 3.1. Código de usuario
- 3.2. Directivas del analizador
- 3.3. Reglas de expresiones regulares
- 3.4. Acciones léxicas

4. Análisis Sintáctico

- 4.1. Análisis de sintaxis
- 4.2. Gramáticas independientes de contexto
- 4.3. Derivación
- 4.4. Ambigüedad
- 4.5. Árbol de análisis sintáctico

5. CUP – Constructor of Useful Parsers

- 5.1. Historia
- 5.2. Funcionamiento
- 5.3. Descripción de las clases generadas por CUP
- 5.4. Código de usuario
- 5.5. Declaración de símbolos gramaticales
- 5.6. Declaración de precedencia
- 5.7. Definición de gramáticas
- 5.8. Métodos de recuperación de errores

6. Gramáticas Ascendentes

- 6.1. Gramáticas
- 6.2. Gramáticas ascendentes
- 6.3. Funcionamiento
- 6.4. Implementar pila por desplazamiento y reducción
- 6.5. Uso de precedencia de operadores
- 6.6. Ejemplos de gramáticas

7. Gramáticas Descendentes

- 7.1. Funcionamiento
- 7.2. Reescritura de gramáticas
- 7.3. Supresión de la ambigüedad
- 7.4. Factorización por la izquierda
- 7.5. Eliminación de la recursividad por la izquierda

8. Tabla de Símbolos

- 8.1. Tabla de símbolos
- 8.2. Funcionamiento

- 8.3. Compilador de una pasada
- 8.4. Compilador de varias pasadas
- 8.5. Contenido de la tabla de símbolos
- 8.6. Operaciones de la tabla de símbolos

9. Traducción dirigida por la sintaxis

- 9.1. Definición dirigida por la sintaxis
- 9.2. Construcción del árbol de traducción dirigida por la sintaxis
- 9.3. Definición de atributos
- 9.4. Tipos de atributos
- 9.5. Atributos sintetizados
- 9.6. Ejemplo: gramática con atributos sintetizados
- 9.7. Atributos heredados
- 9.8. Ejemplo: gramática con atributos heredados
- 9.9. Esquema de traducción
- 9.10. Evaluación de atributos

10. Árbol de Sintaxis Abstracta (AST)

- 10.1. Construcción de AST
- 10.2. Sentencias básicas de programación
- 10.3. Ejecución de ciclos y sentencias de control
- 10.4. Manejo de la tabla de símbolos

Bibliografía:

- Compiladores. Principios, técnicas y herramientas. AHO, Alfred. Segunda Edición. Pearson Educación. México, 2008.