

## CC3007 – Construcción de Compiladores

### I. Descripción:

Este curso expone los temas fundamentales de las últimas fases en la construcción de un compilador, el análisis semántico, generación de código intermedio y generación de código, además una introducción sobre la optimización de código intermedio. Para estas partes se estudian y analizan los algoritmos respectivos que son básicos en la construcción de estas herramientas además de la interacción que poseen estas fases en el proceso de compilación.

### II. Requisitos: Algoritmos y Estructuras de Datos. Matemática Discreta, Diseño de Lenguajes de Programación.

### III. Objetivos:

Qué el estudiante:

- Desarrolle su capacidad de abstracción e implementación en la elaboración de componentes de software.
- Aprenda de las ideas y técnicas de diseño de software que se utilizan en la construcción de un compilador.
- Elabore un compilador de un lenguaje subconjunto de C complementando un analizador léxico y un analizador sintáctico con el análisis semántico, generación de código intermedio y generación de código.
- Identifique los principales aspectos de optimización de código intermedio y la implementación de los mismos.
- Aprenda a utilizar herramientas de software que faciliten la elaboración de la construcción del compilador.

### IV. Contenido del Curso:

- Comprobación de Tipos**
  - Sistema de tipos
  - Comprobador de tipos
  - Equivalencia de expresiones de tipos.
- Ambientes para la ejecución**
  - Aspectos del lenguaje fuente
  - Organización de la memoria
  - Estrategias de asignación de memoria
  - Acceso a nombres no locales
  - Paso de parámetros
  - Tablas de símbolos
- Generación de código intermedio**
  - Lenguajes intermedios
  - Declaraciones
  - Propositiones de asignación
  - Expresiones booleanas
  - Llamadas a procedimientos
- Generación de Código**
  - Aspectos de diseño
  - Máquina objeto
  - Administración de la memoria
  - Bloques básicos y grafos de flujo
  - Generador de código simple
  - Distribución y asignación de registros
- Optimización de Código**
  - Introducción
  - Principales fuentes para optimización
  - Optimización de bloques básicos

## V. Metodología:

El curso tendrá una duración de 20 semanas. Consta de 3 periodos semanales magistrales. Se evaluará por tareas, exámenes cortos, proyectos y exámenes parciales.

## VI. Fechas Importantes

### a) Parciales

- Parcial #1 – Comprobación de Tipos, Ambientes de Ejecución
- Parcial #2 – Generación de Código Intermedio, Generación de Código

### b) Entrega de Proyectos

- Proyecto #1 Análisis semántico (Comprobación estática)
- Proyecto #2 Generación de Código Intermedio
- Proyecto #3 Generación de Código (Extra: Optimización de código intermedio)

## VII. Bibliografía

- Aho, Alfred V.; Sethi, Ravi y Ullman, Jeffrey. **Compiladores Principios, técnicas y herramientas**. Editorial Addison-Wesley Iberoamericana. Delaware, México. 1990. ISBN 968-444-333-1
- Aho, Alfred V.; Sethi, Ravi y Ullman, Jeffrey. **Compilers: Principles, Techniques and Tools**. Addison-Wesley, 1986. ISBN 0-20110-1947
- Appel, Andrew W. **Modern Compiler Implementation in Java**. Cambridge University Press, 1998. ISBN 0-52158-388-8
- Muchnick, Steven. **Advanced Compiler Design and Implementation**. Morgan Kaufman Publishers, 1997. ISBN 1-55860-320-4
- Cooper, Keith D. & Torczon Linda. **Engineering a Compiler**. Morgan Kaufman Publishers, 2003. ISBN 1-55860-698-X

## VIII. Evaluación

Actividad	Total
Tareas y cortos	20%
Parciales	20%
Proyectos	60%
<b>Total</b>	<b>100%</b>

Para aprobar el curso es necesario que tenga una nota mínima de 61 puntos en total. Debido a que no existe examen final, este curso no posee examen de recuperación.

## IX. Catedráticos

Sección	Nombre	Correo Electrónico
10	Ing. Bidkar A. Pojoy	bidkarpc@gmail.com