

Departamento de Ciencias e Ingeniería de la Computación
Curso de Compiladores
Periodo 2008-1

Objetivos:

- Estudiar los fundamentos de la teoría de lenguajes formales y aplicarla en el desarrollo de un compilador para un lenguaje de programación de capacidades acordes con la extensión del curso.
- Conocer las etapas que componen el proceso de compilación, entender sus características fundamentales y las de las herramientas que se utilicen para implementarlas en un compilador.
- Aprender técnicas de diseño e implementación útiles para el desarrollo de compiladores.

Motivación:

Entender cómo funciona un compilador es un conocimiento necesario para poder ser un programador verdaderamente profesional. Por ejemplo, en tareas de optimización de código, el conocimiento del proceso de compilación permite aprovechar hasta el límite las capacidades de la máquina y mejorar así el tiempo o el espacio que se necesitan para ejecutar un programa. Es claro que si un ingeniero de sistemas se ve ante la tarea de desarrollar un lenguaje de programación le hará falta conocer en detalle los temas de este curso. Sin embargo, también debe ser claro que aún si ese ingeniero nunca se viera en la tarea de desarrollar un nuevo lenguaje, también encontrará de gran utilidad conocer las técnicas de diseño e implementación que aquí se presentan, ellas le permitirán ejercer una programación de un nivel más profesional y también le serán de utilidad, por ejemplo, cuando necesite procesar lenguaje natural o cualquier sublenguaje necesario dentro de una aplicación dada. Dado que el curso parte de la teoría y llega hasta la práctica, en este curso el estudiante que se involucre activamente verá fortalecido su conocimiento sobre los fundamentos teóricos de las ciencias de la computación y también finalizará conociendo con suficiencia la técnica que permite construir programas eficientes y correctos para compilar programas en un lenguaje dado.

Metodología:

El curso es presencial y se impartirá mediante clases magistrales. Antes de iniciar cada tema, se recomendará la lectura de la bibliografía relacionada, de modo que el estudiante tenga una base de conocimiento que le permita participar en el desarrollo del tema durante la clase.

Temas del curso:

Semana	Fecha	Descripción	Bibliografía
1	enero 21	Introducción	[1] Cap 1,2.
2	enero 28	Análisis léxico - expresiones regulares	[1] Cap 3.
3	febrero 4	Análisis léxico – flex	
4	febrero 11	PRIMERA EVALUACION – Análisis sintáctico	[1] Cap. 4.
5	febrero 18	Análisis sintáctico – gramáticas libres de contexto	[1] Cap. 4.
6	febrero 25	Análisis sintáctico - tipos de análisis	
7	marzo 3	Análisis sintáctico – traducción dirigida por sintaxis	[1] Cap. 5.
8	marzo 10	Análisis sintáctico – BISON	
9	marzo 17	SEMANA SANTA	
10	marzo 24	SEGUNDA EVALUACION - Análisis semántico	[1] Cap 6.

11	marzo 31	Análisis semántico	
12	abril 7	Entornos de ejecución	[1] Cap 7.
13	abril 14	Entornos e ejecución	
14	abril 21	Generación de código intermedio	[1] Cap 8.
15	abril 28	Generación de código intermedio	
16	mayo 5	Generación de código	[1] Cap 9.

Evaluación:

La evaluación se compone de evaluaciones escritas y de la evaluación del proyecto, los porcentajes propuestos son:

Primer parcial

Examen escrito: 20%

Informe y código del proyecto + sustentación 10%

Segundo parcial

Examen escrito: 20%

Informe y código del proyecto + sustentación 10%

Evaluación final

Examen escrito 20%

Informe + código del proyecto + sustentación 10%

Talleres, quices y tareas 10%

Precisiones sobre la evaluación:

- Los exámenes escritos serán siempre individuales y no se permitirá el uso de apuntes durante ellos.
- Cada entrega del proyecto constará del código desarrollado y de un informe breve que contenga: las indicaciones para ejecutar el/los programas y explicaciones técnicas según lo que se haya solicitado en el enunciado del proyecto para cada entrega.
- Las sustentaciones serán siempre individuales. En ellas se evaluará el conocimiento del proyecto por parte del estudiante, su capacidad para explicar las decisiones tomadas y para argumentar acerca de ellas y el dominio y fluidez en el uso de las herramientas que se utilizan en el proyecto.

Bibliografía:

[1] Aho A., Sethi R., Ullman J. Compilers: principles, techniques and tools. Tercera Edición. Addison-Wesley. 2007.

[2] Wilhelm R., Dieter M. Compiler Design. Addison-Wesley. 1995.

[3] Manual de Flex. http://atlas.puj.edu.co/ftp/cursos/estandares/flex-es-2_5.tar.gz

[4] Manual de Bison. http://atlas.puj.edu.co/ftp/cursos/estandares/bison-es-1_27.tar.gz