



PROGRAMA DEL CURSO

Horario:	Grupo 2. L y J 7 – 9:00 AM, 304 IF
Consulta:	A convenir
Profesor:	Ricardo Gang Vincenzi
	Consultas: ricardogang@costarricense.cr
	Evaluaciones: ci1322g2@gmail.com
	Oficina 221 IF, Tel. 202-6166/894-6503

Objetivo general

Introducir los conceptos principales sobre el diseño e implementación de compiladores para lenguajes modernos.

Objetivos específicos

1. Aplicar la teoría de lenguajes formales en la construcción de compiladores para lenguajes de programación
2. Construir un compilador para un lenguaje estructurado sencillo

Contenido

1. Repaso de grafos y árboles
2. Autómatas de estado finito y expresiones regulares
3. Gramáticas libres de contexto

4. Introducción a la compilación: compiladores y traductores, fases y pasadas de un compilador, pre-procesadores
5. Análisis léxico: reconocimiento de componentes léxicos, detección y recuperación de errores
6. Análisis sintáctico: analizadores descendentes y ascendentes, analizadores LL y LR
7. Análisis semántico y chequeo de tipos
8. Traducción orientada a sintaxis, gramáticas con atributos, traducción descendente
9. Generación de código intermedio: lenguajes de tres direcciones
10. Ambiente de ejecución: manejo de memoria, acceso a variables, paso de parámetros
11. Generación de código objeto
12. Optimización de código
13. Uso de herramientas automatizadas para la construcción de compiladores

Metodología

La mayor parte del curso se llevará a cabo mediante el sistema de lecciones magistrales. Habrá un espacio para práctica dentro de las horas lectivas. Como material de lectura y estudio se usará un compendio de capítulos del libro de texto propuesto en la bibliografía. Este material se complementará con lecturas y ejemplos de otras fuentes.

Se espera que los estudiantes participen activamente en las lecciones mediante preguntas e intervenciones que enriquezcan la discusión. Se espera, además, que el estudiante ponga en práctica los conceptos del curso en las tareas programadas. En las tareas programadas el estudiante debe tomar iniciativa propia para solucionar los problemas propuestos.

Evaluación

La nota máxima de 10 puntos se distribuye así:

Primer examen parcial	15.0%
Segundo examen parcial	17.5%
Examen final	20.0%
Trabajo en clase	10.0%
Quices y tareas cortas	17.5%
Tareas programadas	20.0%

Para aprobar el curso el estudiante debe tener 6.75 puntos o más. Si la nota final está entre 5.75 y 6.74 tendrá derecho a realizar un examen de ampliación. En este examen el estudiante deberá obtener una nota mínima de 7.0 para aprobar el curso. En caso de que obtenga una nota menor a 5.75, o de presentar el examen de ampliación con una nota inferior a 7.0 reprobará el curso.

Observaciones

- Las **tareas cortas y programadas** deberán ser entregadas el día y hora propuestos en el enunciado de cada tarea al profesor por cualquier medio que se indique (no al asistente). Cualquier atraso en la entrega de las mismas será sancionada de acuerdo con la siguiente regla:

A la nota obtenida se le rebajará puntos de acuerdo a la siguiente fórmula (sea i el número de días de atraso):

$$e^{\frac{i}{P}} * 5$$

- El **trabajo en clase** normalmente consistirá en un cuestionario que se entregará al inicio de la clase. Durante el transcurso de la misma se darán las respuestas a dicho cuestionario, deberá entregarse completo al profesor al terminar la clase.
- Las pruebas cortas (**quices**) se realizarán al menos una vez a la semana y se efectuarán en cualquier momento durante el transcurso de la lección. En los quices se evaluará la materia vista en clase hasta una semana antes (la materia es acumulativa). Dichas pruebas no se repondrán a menos que exista una justificación válida de acuerdo con los reglamentos vigentes. Las pruebas cortas serán individuales.
- Una tarea corta consiste en la **exposición**, en grupo, de un tema que se asignará en la segunda semana de clases. Las exposiciones deberán ser lo suficientemente claras y completas para que los demás estudiantes puedan llenar un cuestionario de trabajo en clase respecto a cada tema. La nota de la exposición corresponde a 60% de la exposición y 40% del promedio de las notas del cuestionario del resto del grupo correspondiente a la exposición.

Fechas importantes

Fecha	Actividad
S 23 Setiembre 1:00 PM	I Examen Parcial
L 2 Octubre	Entrega de presentaciones
S 4 Noviembre 1:00 PM	II Examen Parcial
L 6 Noviembre	Exposiciones
J 9 Noviembre	Exposiciones
S 2 Diciembre 1:00 PM	Examen Final

Bibliografía

- Aho, Alfred V.; Sethi, Ravi y Ullman, Jeffrey D. Compiladores: Principios, técnicas y prácticas. Addison Wesley, México. 1ra Edición, 1990.
- Albahari, Ben; Drayton, Peter y Merrill, Brad. C# Essentials, 2da Edición, O'Reilly.
- Appel, A.W.; Modern Compiler Implementation in Java. Cambridge University Press, 1998.
- Cooper, Keith D.; Torczon, Linda. Engineering a compiler. Morgan Kaufmann Publishing. 1ra Edición. 2003.
- Fisher, C. N.; Leblanc, R. Crafting a Compiler. Benjamin Cummings, 1988.
- Hopcroft, John E.; Motwani, Rajeev y Ullman., Jeffrey D. Introducción a la Teoría de Autómatas, Lenguajes y Computación, 2da Edición, Pearson Educación S.A, Madrid, 2002.
- Hopcroft, John E. y Ullman, Jeffrey D. Introduction to Automata Theory, Languages and Computation, Addison Wesley. 1979.
- Lennan, B.J. Principles of Programming Languages. HRW editores. 1987.
- Levine, John R.; Mason, Tony y Brown, Doug. Lex & Yacc, 2da Edición, O'Reilly & Associates, Octubre 1992.
- Muchnick, Steve. Advanced Compiler Design and Implementation. Morgan Kaufmann Publishing, 1997.
- Pratt, Terrence W. y Zelkowitz, Marvin V. Lenguajes de Programación: Diseño e implementación, 3ra Edición, Prentice-Hall Hispanoamericana, 1998.
- Scott, Michael L.. Programming Language Pragmatics. Morgan Kaufman Publishers. 2000.
- Sethi., R. Lenguajes de Programación. Conceptos y constructores. 1992
- Teufel, Bernard; Schmidt, Stephanie y Teufel, Thomas. Compiladores Conceptos fundamentales. Addison-Wesley Iberoamericana, 1995.