CSC484 Compiler Theory

General

Syllabus

Required Text: *Compiler Construction: Principles and Practice* by Kenneth C. Louden, PWS Publishing Company, 1997, ISBN #0-534-93972-4

Course Description: An introduction to the basic structures of compilers and their design. Course topics include computer language structure, translation/ recognition techniques of lexical analysis, parsing and syntax-directed translation. The course will also consider the impact of run-time environments on the design of computer languages and the constraints of code optimization on code generation. A small compiler will be developed.

Prerequisite: A grade of C or better in CSC384.

Objectives:

After completing this course, a student should be able to:

- represent language tokens using regular expressions and finite automata
- write a lexical analyzer for a language, and generate a lexical analyzer using Flex
- develop a top-down recursive-descent parser, and generate a bottom-up parser using Yacc, which parse syntax expressed by a context-free grammar and produce a syntax-tree representation of the input
- implement type checking for a block-structured language using a symbol table
- generate target code for a stack-based runtime environment
- understand how source programs are mapped to target platforms via the compilation process, and write better source programs based on this understanding

Covered Topics:

- Scanning / lexical analysis
  - Hand-coded scanner
  - Generated scanner using Flex
  - Use of regular expressions and finite automata
- Parsing / syntax analysis
  - Hand-coded recursive-descent parser
  - Generated parser using Yacc
  - Use of context-free grammars and syntax trees
- Semantic analysis
  - Symbol table
  - Type checking
- Code generation
  - Instruction generation and memory layout
  - Stack-based runtime environment