CSCE 4930 Syllabus

Instructor: Philip Sweany

Office: NTDP F262

Office Hours: Monday and Wednesday, 1:30 – 3:15; Tuesday, 1:45 – 3:45

Email: sweany@cs.unt.edu

Textbook:

Aho, Lam, Sethi, Ullman

Compilers: Principles, Techniques & Tools, Second Edition

Addison-Wesley, 2007

Course Objectives

1. Students will develop a good understanding of basic compiler analysis and optimization techniques.
2. Students will sharpen their skills in comparing and evaluating different compiler techniques with the intent of choosing among several techniques for inclusion into a production compiler.
3. Students will develop their skill in adding to existing software.
4. Students will be able to design, implement and write about experimental compiler research in a professional manner.
5. Students will develop skills in professional oral presentation of their work.

Course Outcomes: By the end of CSCE 4930, each student should be able to:

1. Given a hypothetical 3-address assembly language form of a program, build a control flow graph by hand.
2. Given a control flow graph for a function, complete the following list of analyses by hand; 1) dependence analysis, 2) live variable analysis, 3) static single assignment, 4) Tarjan symbolic covers.
3. Given a control flow graph for a function, and using standard analysis techniques, build both symbolic execution DAGs for each basic block and a control and dependence graph for the entire function.
4. Given both a hypothetical 3-address assembly language form of a program and some description of the details of the target architecture, perform the following optimizations by hand; 1) common subexpression elimination, 2) copy propagation, 3) dead code elimination, partial redundancy elimination, 4) local instruction scheduling, 5) global instruction scheduling, 6) software pipelining, and 7) graph coloring register assignment.
5. Describe how iterative dataflow problems used in compiler analysis map to formal lattice theory.
6. Given a simple compiler which generates a control flow graph of a hypothetical 3 address assembly language statements for each basic block, implement an optimizer including a significant subset of the analysis and optimizations listed above and complete an

experimental report describing the effects of that optimizer on the code generated for a significant benchmark suite of input programs.

Course Topics

Control Flow and Data Flow Analysis
"Traditional" Program Transformations
Register Assignment
Data Dependence Analysis
Instruction Scheduling Optimizations
Re-targetable compilers
Current research papers

Course Organization

1. The early part of the course will investigate traditional compiler optimization techniques.
2. A portion of the course will include discussion of current research papers in the field of compilers.
3. A major portion of each student's efforts will be directed towards experimental work based upon implementation of optimization technique(s)

Grading

1. Experimental compiler study. This will entail a proposal due no later than (date), implementation, gathering of experimental data, and a "conference" paper. – 30% of the grade
2. Presentation of experimental compiler study – 10% of the grade. (Presentations to take place during the regularly scheduled finals time)
3. Midterm Exam – 30% of grade
4. Participation in class discussion – 10% of the grade
5. Homework – 20% of the grade.

Collaboration and Cheating:

On programs, do your own work. Do NOT work with other students on shared program solutions. Do NOT get help coding from anyone other than Dr. Sweany, who won't be much help anyway.

SETE

The Student Evaluation of Teaching Effectiveness (SETE) is a requirement for all organized classes at UNT. This short survey will be made available to you at the end of the semester, providing you a chance to comment on how this class is taught. I am very interested in the feedback I get from students, as I work to continually improve my teaching. I consider the SETE to be an important part of your participation in this class.

ADA

UNT complies with all federal and state laws and regulations regarding discrimination, including the Americans with Disability Act of 1990 (ADA). If you have a disability and need a reasonable accommodation for equal access to education or services please contact the Office of Disability Accommodation.

TENTATIVE Schedule and Topics

Week	Topic	Reading	Assignments
9/10/12	Control Flow Graphs	Section 8.4	
9/17/12	Dataflow Analysis	Sections 9.2-3,9.6	
9/24/12	Static Single Assignment	Sections 3.1-3.3	Opt. Paper, 9/28
10/1/12	Tarjan Symbolic Covers	Section 3.4	
10/8/12	'Traditional" Optimizations	Sections 4.1-4.2	
10/15/12	Data Dependence Analysis	Sections 4.3-4.4	Homework 1, 10/19
10/22/12	Midterm Exam		
10/29/12	Instruction-Level Parallelism	Sections 5.1-5.5	
11/5/12	ILP – part 2	Sections 6.1-6.2	Homework 2, 11/9
11/12/12	Retargetable Compilers	Sections 6.1, 6.3	
11/19/12	Research Papers	Online Material	
11/26/12	Research Papers	Online Material	Homework 3, 11/30
12/3/12	Research Papers	Online Material	Optimizer 12/7/12
12/10/12	Project Presentations		Project Paper, at Final