

Syllabus

COEN 259 Compilers Department of Computer Engineering Santa Clara University

Dr. Ming-Hwa Wang
Phone: (408) 526-4844
Course website:
Office Hours:

Winter Quarter 2016
Email address: m1wang@scu.edu
<http://www.cse.scu.edu/~mwang2/compiler/>
Tuesday & Thursday 9:00-9:30pm

Course Description

Principles and practice of the design and implementation of a compiler, focusing on the application of theory and trade-offs in design. Lexical analysis, syntactic analysis, semantic analysis, symbol tables, type checking, run-time organization, code generation, optimization and data-flow analysis.

Prerequisites

Prerequisites: Principles of Programming Languages (COEN 256) or Operating Systems (COEN 283) or Computer Architecture (COEN 210).

Required Textbooks

1. "Compilers, Principles, Techniques, and Tools, 2nd Edition" by Alfred V. Aho, Monica S. Lam, Ravi Sethi, Jeffrey D. Ullman, ISBN: 978-0321486813, Addison-Wesley 2007
2. "Flex & Bison" by John Levine, ISBN: 978-0596155971, O'Reilly & Associates, 2009

Recommended Textbooks

1. "Engineering A Compiler, 2nd Edition" by Keith D. Cooper, Linda Torczon, ISBN: 978-0120884780, Morgan Kaufmann 2011
2. "Crafting a Compiler" by Charles N. Fischer, Ronald K. Cytron, Richard J. LeBlanc, Jr., ISBN: 978-0136067054, Benjamin-Cummings 2009
3. "Writing Compilers and Interpreters: A Software Engineering Approach" by Ronald Mak, ISBN: 978-0470177075, Willey 2009
4. "Modern Compiler Implementation in Java, 2nd Edition" by Andrew W. Appel, ISBN: 978-0521820608, Cambridge University Press 2002
5. "The Art of Compiler Design, Theory and Practice" by Thomas Pittman and James Peters, ISBN: 978-0130481900, Prentice-Hall 1991
6. "Lex & Yacc" by Doug Brown, John R. Levine, Tony Mason, ISBN: 978-1565920002, O'Reilly & Associates, 1995

References

1. "Advanced Compiler Design & Implementation" by Steve S. Muchnick, ISBN: 978-1558603202, Morgan Kaufmann, 1997
2. "Embedded Computing, A VLIW Approach to Architecture, Compilers, and Tools" by Joseph A. Fisher, Paolo Faraboschi, Cliff Young, ISBN: 978-1558607668, Morgan Kaufmann, 2005
3. "High Performance Compilers for Parallel Computing" by Michael Wolfe, ISBN: 978-0805327304, Addison-Wesley 1995
4. "Optimizing Compilers for Modern Architectures, A Dependence-Based Approach" by Randy Allen, ISBN: 978-1558602861, Morgan Kaufmann, 2001
5. "Introduction to Automata Theory, Languages, and Computation, 3rd Edition" by John E. Hopcroft, Rajeev Motwani, Jeffrey D. Ullman, ISBN: 978-0321455369, Addison-Wesley Publishing Company, 2006
6. "Introduction to Formal Language Theory" by Michael A. Harrison, ISBN: 978-0201029550, Addison-Wesley 1978
7. "Programming Languages, Design and Implementation, 4th Edition" by Terrence W. Pratt, ISBN: 978-0130276780, Prentice-Hall, Inc., 2000
8. "Programming Language Concepts" by Carlo Ghezzi, Mehdi Jazayeri, ISBN: 978-0471104261, 1997
9. "Systems Programming" by John J. Donovan, ISBN: 978-0070176034, McGraw-Hill Book Company, 1972

Course Objectives

1. To learn advanced and cutting edge state-of-the-art knowledge and implementation in compilers.
2. To read and understand research publications in the technical area of compilers, beyond that of the traditional textbook level.
3. To conduct independent project and to equip for scholarly research in Compilers.
4. Explore parallel/distributed language compiler design or other advanced topics if time permit.

Expected Learning Outcomes

1. Demonstrate the knowledge of compiler front-end theory including lexical analysis, syntactic analysis, semantic analysis, type checking, symbol tables, etc.
2. Demonstrate the knowledge of compiler back-end theory including intermediate code generation, run-time organization, code generation, optimization, data-flow analysis, etc.
3. Demonstrate the ability to implement a subset of a high-level language compiler.
4. Demonstrate the ability to do research in compiler: able to read/understand current research papers and implement research group

project in compiler optimization, parallelization, vectorization, virtualization, or VLIW.

Grading Policy

Course grade is determined based on the total score (maximum 1100 points + up to 200 optional bonus points for extra work) from:

1. Mid-term and final exams of 200 points each (close book with one A4 note, no sitting together, no wireless connection.) Makeup exams (must have a very good reason) are much difficult than normal exams.
2. Two programming assignments of 200 points each (late penalty: 40 points/day.) Makeups are more difficult too. You can call Design Center at 408/554-4909 for setup account or IT support, and ssh linux.scudc.scu.edu to work remotely.
3. A group (2-3 people in a team) programming term project of 300 points (late penalty: 60 points/day.) No makeup is allowed.
4. Bonus assignments will be assigned at each week with 20 points each. Due before next lecture begin by email to me (in plain text or PDF) with title "coen259 bN" (where N can be 2, 3,..., 10) and cc to the grader. The solution for bonus assignments will be posted on my protected web page. Please read solutions of bonus assignments before asking questions. No late work accepted for bonus assignments. 75-80% of exam questions are similar to bonus assignments.
5. Class average targeted at **A-**.

Table 1: Grade-score table

1000	950	900	850	800	750	700	650	0
-	-	-	-	-	-	-	-	-
1300	999	949	899	849	799	749	699	649
A	A-	B+	B	B-	C+	C	C-	F

Course Schedule (Tuesday & Thursday 7:10pm-9:00pm)

Table 2: Course Schedule

#	week	Readings	Remarks
1	1/5 1/7	introduction	submit due 1/7
2	1/12 1/14	reg exp & scanner	
3	1/19 1/21	CFG & parser	
4	1/26 1/28	IR	program #1 due 1/24
5	2/2 2/4	run-time	mid-term exam 2/4
6	2/9 2/11	code gen	problem due 2/9 group & topic due 2/12
7	2/16 2/18	optimization	program #2 due 2/14 paper presentation 2/16 2/18

8	2/23 2/25	ILP	last withdraw date 2/19
9	3/1 3/3	parallelism & locality	proposal due 2/23
10	3/8 3/10	interprocedural	final exam 3/10
11	3/15	review/evaluation	project defense 3/15

Reminder

- No cheating, and no register complaint without talking to me first.
- No incomplete. No sit-in or audit the class except formally registered.
- Read files under /home/mwang2/tips for help.
- Handouts, assignments, and solutions will be posted on the web. You should check the class web site at least once a week. You are responsible for printing and bring the handout to the class.
- Office hours: Tuesday & Thursday 9:00pm-9:30pm.

Honor Code

All students taking course in the school of engineering agree, individually and collectively, they will neither give nor receive unpermitted aid in examinations or other course work that is to be used by the instructor as a basis of grading.

Disability Accommodation Policy:

To request academic accommodations for a disability, students must contact Disability Resources located in The Drahmman Center in Benson, room 214, (408) 554-4111; TTY (408) 554-5445. Students must provide documentation of a disability to Disability Resources prior to receiving accommodations.