ECE 466 Syllabus

Course: ECE 466
Credit Hours: 3
Course Title: Compiler Optimization and Scheduling

Course Description:

Provide insight into current compiler designs dealing with present and future generations of high performance processors and embedded systems. Investigate dataflow analysis and memory disambiguation, classical and parallelism enhancing optimizations, scheduling and speculative execution, and register allocation. Review of techniques used in current research compilers.

Prerequisite(s): ECE 209 or competence in any machine language programming, CSC 316 or competence with programming data structures and programming ability in C or C++/Java.

Textbook(s) and/or other required material:

2. Instructor course notes.

Course objectives. By the end of this course, the student should be able to (use demonstrative verbs):

1. describe the fundamental concepts of compiler functions
2. identify the fundamental elements of compiler optimization
3. discuss some of the technical challenges encountered in the areas of scalar optimization, dataflow analysis, scheduling and register allocation.
4. apply these fundamentals in compiler design.
5. practice these fundamentals by programming three parts of a working compiler.
6. calculate the performance of code before and after optimization.

Topics covered:

1. Introduction: Overview and history of compilers.
2. Survey of scanning and parsing.
3. Program representation.
4. Dataflow analysis.
5. Static single assignment form.
7. Register allocation.
8. Brief survey of microarchitecture.
9. Instruction scheduling

Class/laboratory schedule (sessions per week and duration of each session):

Two 75-minutes lectures per week.

Contribution of course to meeting the requirements of Criterion 5 - other:

Contribution of course to meeting the requirements of Criterion 5 - math and basic sciences:

Contribution of course to meeting the requirements of Criterion 5 - engineering topics:

3.0 Hours.
# ECE 466 Syllabus

## Contribution of course to meeting the requirements of Criterion 5 - general education:

## Relationship of this course to program learning outcomes:

<table>
<thead>
<tr>
<th>Learning Outcome</th>
<th>Level of Instruction</th>
<th>Related Course Content</th>
</tr>
</thead>
<tbody>
<tr>
<td>Outcome A</td>
<td>Major</td>
<td>Students learn about code performance and dataflow analysis and how to apply these principles for designing an optimizing compiler and for hand-optimizing code.</td>
</tr>
<tr>
<td>Outcome B</td>
<td>Major</td>
<td>Students implement three experimental projects in which they build a compiler front end, build a small code optimizer and build an instruction scheduler.</td>
</tr>
<tr>
<td>Outcome C</td>
<td>Major</td>
<td>Students design compiler components with the goal of achieving optimal or near-optimal performance and robust behavior across varying input programs considering cost/complexity/performance trade-offs.</td>
</tr>
<tr>
<td>Outcome D</td>
<td>Intermediate</td>
<td>Students build complex software systems in teams of two. They learn efficient task partitioning and continuous communication to ensure correct operation and timely project completion.</td>
</tr>
<tr>
<td>Outcome E</td>
<td>Major</td>
<td>Students analyze a simulator specification and apply their knowledge from the course to implement the simulator.</td>
</tr>
<tr>
<td>Outcome F</td>
<td>N/A</td>
<td>Students write comprehensive reports to explain their compiler</td>
</tr>
<tr>
<td>Outcome G</td>
<td>Intermediate</td>
<td></td>
</tr>
</tbody>
</table>
## Relationship of this course to program learning outcomes:

<table>
<thead>
<tr>
<th>Learning Outcome</th>
<th>Level of Instruction</th>
<th>Related Course Content</th>
</tr>
</thead>
<tbody>
<tr>
<td>Outcome H</td>
<td>Basic</td>
<td>component designs, describe their implementations and present/discuss their experimental findings. Students are exposed to software efficiency and its impact on society.</td>
</tr>
<tr>
<td>Outcome I</td>
<td>Basic</td>
<td>Students are exposed to rapid technology evolution to appreciate life-long education. Students learn how to keep up with latest technology through conference/journal papers.</td>
</tr>
<tr>
<td>Outcome J</td>
<td>Basic</td>
<td>Students study code generation techniques for state-of-the-art processors recently manufactured.</td>
</tr>
<tr>
<td>Outcome K</td>
<td>Major</td>
<td>Students write large programs and use debuggers in three course projects.</td>
</tr>
</tbody>
</table>

**Person who last prepared this description and date of preparation:**

- Tuck, James (jtuck) - Sep 21st, 2009 (12:36pm)