Course Goals

Understand two big picture problems
- program optimization
- security or program analysis for debugging

In depth knowledge of program analysis ranging
- from entirely at compile-time, to run-time, to post run-time, to barely any analysis at all.

How these program analyses are applied to program optimization and security

Various Analysis Issues
- Representing and analyzing programs
- Efficiency versus precision tradeoff
- Precision when analyzing array usage in loops

Approach
- read and critique recent and influential papers
- implement a program analysis idea

Other Important Course Goals

Research Exam Preparation
- paper reviewing skills will be critical for the research exam
- presenting the crucial points in a paper
- synthesizing ideas between papers
- suggesting future research questions

Thesis Project Preparation
- how to break projects into manageable pieces
- how to leverage existing research tools
- presenting research in a succinct fashion

Compile-time Program Analysis

Discover properties of programs by looking at its source
- local (a few lines of straight-line code)
- global or intraprocedural (full procedure)
- interprocedural (several procedures)
  - special case: whole program analysis

Example of Compile-time Program Analysis

```
2 = 7
if x < 0 then
  y = z + 1
else
  y = z + 5

b = y > 12
```

What are the possible values of y?

Does answering this question require local, global, or interprocedural analysis?
Run-time Program Analysis

Discovers properties of programs by examining its runs

Source → Instrument → Instrumented source → Link and run

Example of Run-time Program Analysis

What is the most common target of o.m()?
What can we do with that information?

Tips for Reading a Research Paper (by Tia Newhall)

Read the paper three times
- First read the abstract, introduction, related work, and conclusion
- Next read the entire paper writing down questions and vocabulary words you need to look up
- Finally re-read the paper critically, answer the paper review questions

Questions to answer in a paper review
- What problem did the paper address?
- Is it important/interesting? What was the context for the paper?
- What is the approach used to solve the problem?
- How does the paper support or otherwise justify the conclusions it reaches?
- How does it evaluate the approach?
- What problems are explicitly or implicitly left as future research questions?

Selecting a Paper to Present

Read the abstract, introduction, and conclusion for all papers you are seriously considering

Consider selecting a paper related to your course project

Schedule around your deadlines in this course and other courses

Paper selection will be first-come-first-serve
- I will start posting assignments as soon as they are sent
- You must indicate which paper you would like to present by January 31
<table>
<thead>
<tr>
<th><strong>Project</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Five deliverables</strong></td>
</tr>
<tr>
<td>- proposed tool example with preliminary project proposal</td>
</tr>
<tr>
<td>- proposal and tool example</td>
</tr>
<tr>
<td>- verbal status report (15 minutes)</td>
</tr>
<tr>
<td>- intermediate report</td>
</tr>
<tr>
<td>- final report and elevator speech</td>
</tr>
</tbody>
</table>

**Possible projects accessible from colostate.edu**

<table>
<thead>
<tr>
<th><strong>Tool Example</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>C-breeze</strong></td>
</tr>
<tr>
<td>- in CS553 Fall 2005 wrote a walker that counted operations in the program</td>
</tr>
</tbody>
</table>

**LLVM compiler infrastructure**
- [http://llvm.cs.uiuc.edu/docs/Projects.html](http://llvm.cs.uiuc.edu/docs/Projects.html), describes how to create a project that uses LLVM
- could count operations or memory references or if statements or ...

**TAU: Tuning and Analysis Utilities**
- [http://www.cs.uoregon.edu/research/tau/home.php](http://www.cs.uoregon.edu/research/tau/home.php)
- could profile a benchmark program from last semester

**OpenAnalysis**
- could count the different types of memory reference expressions