Course Description

The material I present in this course is an introduction to modern implementations of programming language features. We will focus on languages that execute via managed runtime systems (i.e., garbage collected and dynamically interpreted/compiled languages) for high-level languages (e.g. Java, Python, Javascript), including their similarities and differences. In addition, we will investigate the inner workings of the interpreter and just-in-time/dynamic compiler and how they can use information about the running program to optimize its execution.

Syllabus

- **Class participation, homeworks, quizzes** 50%
  Read the assigned papers, ask questions, respond to questions, participate in the discussion, perform well on assignments
  May also include presentation of research papers on related topics as part of class discussion (graduate version)
- **Class project or area survey** 50%
  Group size: 1-2
  Project Milestones and Ideas
  May also include presentation of project at end of class
  Projects only required for graduate version of the class

Class topics and material
Evolution of runtime systems for high-level languages (how we got here)
Structure and layout of the Java bytecode class file format (code and symbolic data), i.e. the Java Language and Virtual Machine specification
Bytecode verification (+ type-safety checking at load time) *if time*
Garbage collection (fundamentals and advanced extensions)
Interpretation and interpreter optimization (+python runtime)
Dynamic/Just-in-Time and adaptive compilation
Program performance profiling
Trace compilation (+javascript runtimes)
Cross-language interoperation *if time*

**Administrivia**

Class Times
- Monday/Wednesday 9-11:50am, Phelps 2510
Instructor:
  - [Chandra Krintz](mailto:ckrintz@gmail.com)
  - Office: Department of Computer Science, 2153 HFH
  - Office Hours by appointment
  - chat, skype on-demand (ckrintz)

© 2014-20 Chandra Krintz