Goal of the Project

- Lecture and homework can’t go into depth with any one language
- I want to give you a more intimate experience with at least one of the languages
- You’ll have to explore the literature and do independent research
- Lecture and homework more theoretical; project will apply that theory
- I’m hoping to promote your research
- I’m hoping to promote my research

More About the Project

Prof. Stephen A. Edwards

Project Rules

- One-paragraph project proposal due September 24
- Literature survey presentations October 29
- Literature survey due October 31
- Project presentations December 3 & 5
- Project writeup due December 10
- Multiple teams may choose the same project

Project Ideas

- Hierarchy browser for the Verilog language
  - Verilog models hardware
  - Systems contain modules with instances of others
  - Project: Use an existing parser, extract connectivity information, display it attractively
  - Prerequisites: Understand the Verilog language (use the text), understand a freely-available parser
- Compiled event-driven simulator for Esterel
  - Event-driven simulation divides behavior into events (e.g., change this wire’s value) and schedules them in an event queue
  - Project: Apply this to (part of) Esterel
  - Prerequisites: Understand Esterel, understand my compiler for it
- Compare Verilog and SystemC
  - Both are able to model hardware, but which is better
  - Project: Pick some example (e.g., a processor cache controller) and implement it in both
  - Prerequisites: Some hardware design knowledge, understand both languages
- An Environment for Kahn Process Networks
  - Kahn proposed a C-like language with “send” and “receive” statements
  - Project: Create a library that allows you to run these systems (Java? C?) Main challenge: scheduler.
  - Prerequisites: Understand the Kahn model, decent low-level programming skills

Project Ideas

- Develop a simulator for an assembly language
  - Compare the many approaches (interpreter-based, object-to-object translation); pick one to implement
  - Project: Build the simulator in C or Java
  - Prerequisites: Intimate knowledge of one assembly language
- A survey of language concurrency models
  - Compare how Java, POSIX threads, Ada, Verilog, etc. handle concurrency
  - Prerequisites: understand the concurrency models of each of these languages. Read some language reference manuals
Project Ideas

- Create a simplified Verilog simulator
  - Take an existing front-end and create a new back-end
  - Try to make it faster
  - Prerequisites: some knowledge of digital design, in-depth understanding of Verilog (read text, papers)
- Compare performance of Linux and an RTOS
  - Figuring out how to measure this is the challenge
  - Read some of the OS literature to figure this out
  - Prerequisites: detailed OS knowledge

Project Ideas (My Favorites)

- Propose a language for device drivers
  - Start from French group’s work on video drivers
  - Look for patterns in existing, handwritten drivers
  - Propose a simple language capturing these patterns
  - Write a simple compiler (perhaps using m4)
- Propose a language for communication protocols
  - Use some of the others as starting points
  - Discuss their advantages and disadvantages
  - Propose extensions, simplifications, others
  - Consider different compilation techniques

Example Project

- Implementing Process Networks in Java
- Arnab Basu and Hampapur P. Vijay Kishen, 2000
- Done at UT Austin in Brian Evans’ class
- Used Parks’ scheduling algorithm to resolve deadlocks
- Writeup (from Brian’s class site):
  - Survey of different process networks
  - Description of other, similar projects
  - Description of their implementation
  - Experiments compare various scheduling policies