The Missing CS Class

Where We Are & Where We Are Going

What We’ve Been Doing

Course Logistics

- 1 Unit P/NP
- Offered under ECS 98F - WQ & SQ
- Prereqs: ECS 32C, 36A, or consent
- 50 students WQ / More in SQ
- Instructor of Record: Joël Porquet
- Course materials open-sourced on GitHub

Course Content

Debugging

Strategies and tools required for successful debugging

UNIX

Motivate self-learning and further exposure to *nix systems.

Proposed Schedule

10 lectures

1. Course Introduction - Why learn this material?
2. Introduction to CLI
3. Putting Programs Together with Piping
4. Testing - Discover faults with scripting and unit tests
5. General debugging strategies
6. Text-interface debuggers
7. GUI debuggers
8. Shell Scripting
9. Applying Regular Expressions
10. The Unix Philosophy

Course Intro

Lecture 1

1. Why learn UNIX and debugging?
2. Course Logistics

Homework: SSH into the CSIF and clone the class repository

Unix Intro – Two Lectures

In the order of appearance

1. Introduction to *nix and REPL
2. Command line tools for developers

Introduction to CLI & *nix

- Identify discrepancies between *nix and other operating systems
- Understanding and navigating the UNIX file system
- Utilizing CLI text editors, i.e. vim
- Using man pages

Homework: TBD

CLI Expanded

- Understanding Exit Codes
- Unix program philosophy
- Piecing together commands with IO redirection

Homework: TBD

Debugging – Four Lectures

In the order of appearance

1. Testing
2. Strategies for troubleshooting
3. Text-interface debuggers
4. GUI debuggers

Testing

Fundamentals of software testing

1. Identify the need and use of different software techniques
2. Testing program output from the command line
3. Unit Testing in C with the *assert* macro

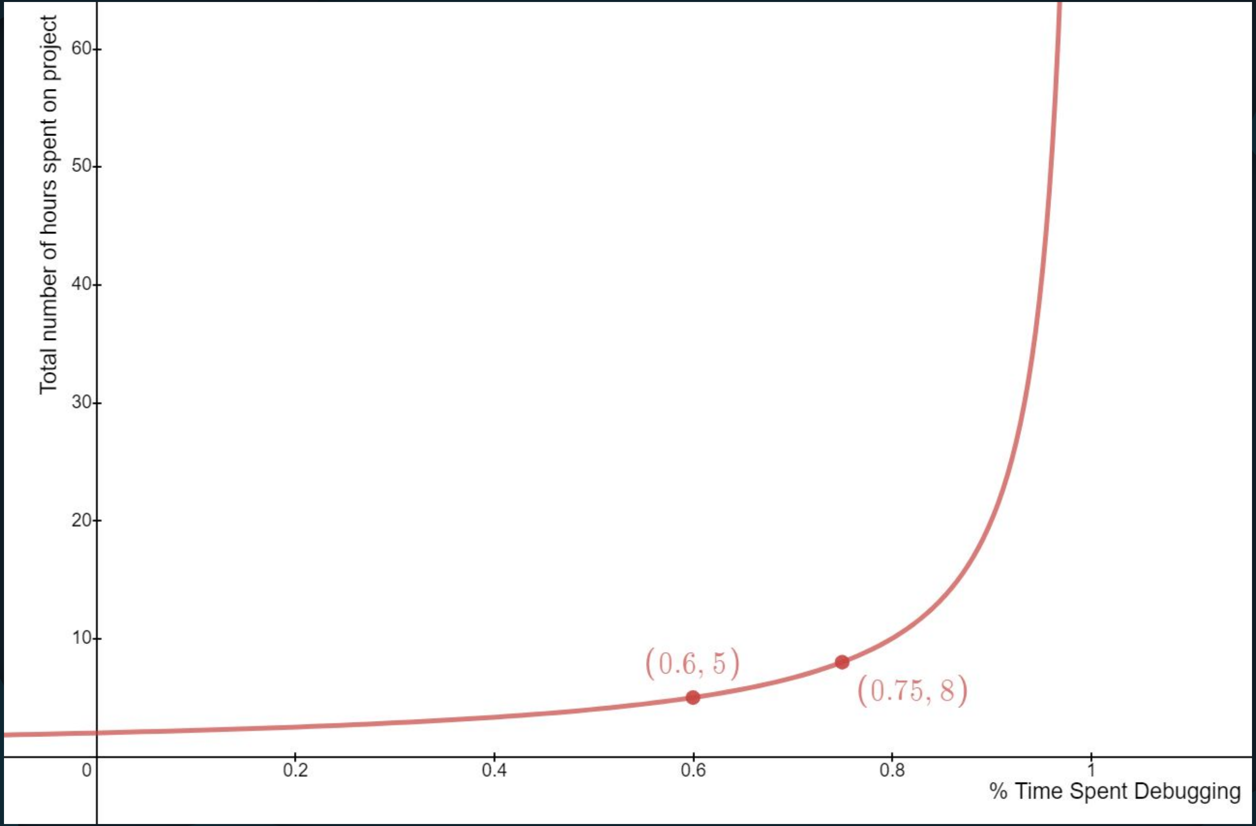
Homework: Given a set of erroneous compiled functions and a description of the intended behavior, write tests to discover bugs.

Debugging Strategies

Generalizable strategies independent of software

1. Preventing bugs. Proper planning & defensive programming
2. Classification of software bugs
3. Concrete strategies for localizing an error
4. Common difficulties in the debugging process

Homework: TDB



Total number of hours spent on project
60
50
40
30
20
10
0
0.2
0.4
0.6
0.8
1
% Time Spent Debugging
(0.6, 5)
(0.75, 8)

Text-Interface Debuggers

An overview of features

- Formalize the basic features of a debugger
- Debugging C code with GDB

Homework: Debug several small C programs using GDB,

submitting both fixed code and gdb log.

GUI Debuggers

- Introduce GUI debuggers and explain their advantage over text-based solutions
- Debugging C code using gdbgui

Homework: Harder practice debugging small C programs.

Further topics in UNIX

In the order of appearance

1. Shell Scripting
2. Regular Expressions

Shell Scripting

Automate everything

- Using Bash as a language
- Environment Variables and $PATH, .dotfiles, aliases
- Job and Process Control

Homework: Create a script that will generate a makefile with the rules to both compile and run unittests

Shell Scripting

Automate everything

- Using Bash as a language
- Environment Variables and $PATH, .dotfiles, aliases
- Job and Process Control
- Make?

Homework: Create a script that will generate a makefile with the rules to both compile and run unittests

Prereqs: ECS 32C, 36A, consent

Regex

Wrangling Data

- Why use Regular Expressions?
- Unix Wildcards
- POSIX Extended Regex
- Regex in shell scripting

Homework: Shell Script to trim and organize a large set of files based on file type

The Unix Philosophy

Rules of thumb for programming

- Introduce general programming tips from the founders of Unix
  - Modularity
  - Clarity
  - Generation
  - Diversity
  - Extensibility
  - Many more….

Homework: Write an autograder

Reference: https://homepage.cs.uri.edu/~thenry/resources/unix_art/ch01s06.html

What We Still Need to Do

Reproducible Demos

- Curriculum will be open source and should be accessible for educators
- Well documented and reproducible demonstrations are a requirement

Open Sourcing

- Can’t open source solutions to homework assignments
- Using a slide format that is accessible

Thank you for coming!

Questions, Comments, Concerns?