


Lab Overview

Mark Stanovich

Shuanglong Zhang

Contact

- Office
 - Love 105
 - Good place to find us
- Office hours
 - See course website
- Email 
 - cop4610t@cs.fsu.edu
- Ask questions!



Websites



- Lab
 - <http://www.cs.fsu.edu/~cop4610t>
- Lecture
 - http://www.cs.fsu.edu/~awang/courses/cop4610_f2014/
- Blackboard
 - campus.fsu.edu
 - myFSU Login

Lab Objectives

- Reinforce learning of OS concepts through implementation
- Understand practical considerations
 - Study real-world designs
- Become familiar with common development tools



Layout

- Friday lecture (HCB 215)
 - Present new material (e.g., Project description)
- Recitations (MCH 202)
 - Review material/tools useful to complete project
 - Workshops
 - Ask questions
- Computer Lab (MCH 202)
 - Unavailable
 - Tues, Wed, Thurs 3:35PM – 4:50PM (Python Course)
 - May use your own machine

Projects

- Total of 3 projects
- Teams
 - Exactly 2 student
 - Different partner for each project
- **C** is the only acceptable language



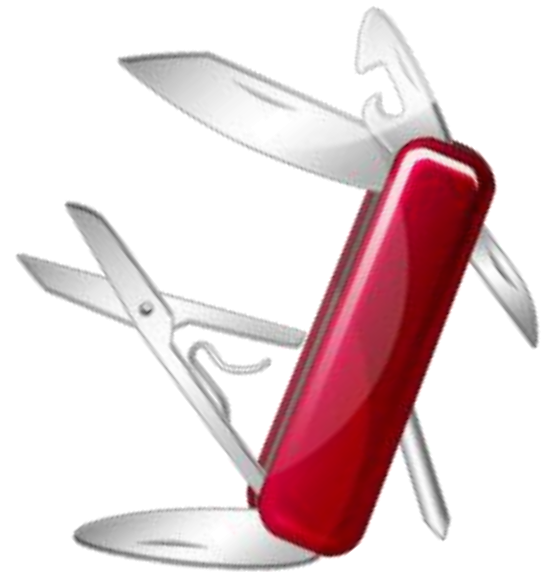
Expectations

- Prerequisites
 - Data structures
 - Algorithms
 - UNIX
- Ability to code in C programming language
- Use compiler and linker (i.e., gcc, ld)
- Efficiently create programs
 - Edit code (e.g., vim, emacs, eclipse, ...)
 - Testing and debugging
- Use command line interface (e.g., filesystem access)



Other Suggestions

- Version control
 - E.g., git
- Static code checks
 - Turn on and **heed** compiler warnings
- Dynamic code checks
 - E.g., valgrind
- Test cases
- Incremental development



Submission

- Deliverables
 - Only **one** member will turn in the project deliverables
 - Each project will describe the particular submission instructions
- Demo (if applicable)
 - **ALL** team members must be present
 - A sign-up sheet will be distributed

Submission Deliverables:

1: Code

- Implementation of the specified functionality
- Comments
- Header on each source code file
 - Developer names
 - Brief description of the file's contents

Submission Deliverables:

2: ASCII Readme

- Names of developers
- Date submitted
- Brief summary of code's function
- Precise instructions for building and running program
- List of source code files with a brief description of the files contents
- Deviation from nominal requirements
 - Known errors and/or missing features
 - Undocumented issues will generally be graded more harshly
 - Description of any bonus implemented

Submission Deliverables:

3: Report

- Design
 - General structure of the code (e.g., pseudo-code of portions of implementation)
 - Assumptions
- Journal of development process
 - Dates of meetings, decisions, accomplishments
- Contributions of each member
 - Division of work
- Reasons for missing functionality/errors (if applicable)
 - Examples
 - Describe attempts to fix error
 - Debugging process
- Any changes that you would make to project

Submission Deliverables:

4: Demo

- Live demonstration
- Convince us that your implementation is correct
- Questions will be asked related to your project
 - E.g., Give an explanation of how you implemented a given project requirement?

Project Grading Criteria

- Correct functionality
- Readability of code, readme, and report
- Demonstration (if applicable)
 - Answers to questions asked
 - Provides reasonable assurance that implementation functions correctly

Appeals

- Every team has the right to appeal their grade
 - The entire team should come to grader's office, or send an email, **no later than 1 week** after the grades are posted on Blackboard