CSE 410: Systems Programming

Introduction

Ethan Blanton

Department of Computer Science and Engineering
University at Buffalo

Welcome to CSE 410

My name is Ethan Blanton.

Contacting me:

Email:	eblanton@buffalo.edu
Office:	Davis 334
Office Hours:	Monday 13:00-15:00 Wednesday 11:00-12:00 or by appointment

The syllabus is available on the course web page, at https://www.cse.buffalo.edu/~eblanton/course/cse410/.

So are these slides!

Expectations

For this course, I expect that you:

- Will be respectful to me, TAs, classmates
- Attend every lecture
- Attend every recitation
- Adhere strictly to the academic integrity policy
- Will seek assistance early if necessary
- Meet prereqs; among other things:
  - Have some experience programming
  - Understand linked lists and object references

Most of all, behave as adults and strive to maximize your and your classmates' learning experience in this course.

Academic Integrity

I take academic integrity very seriously.

Violators will

- fail this course, and
- be referred upward for further sanctions.

I and the TAs will watch for violations.

Automated tools will be used to identify shared code.

Online resources (*e.g.*, Stack Overflow, GitHub) will be monitored for copying.

Academic Integrity (*continued*)

You may:

- Seek help from instructors
- Discuss concepts with classmates
- Use anything from the text with clear attribution

You MAY NOT:

- Share code with classmates
- Use code from *anywhere else*
- Discuss implementation with classmates

Academic Integrity — Good Practices

To ensure a pleasant semester, please:

- Be careful with permissions on code on GitHub, Bitbucket, shared UB filesystems, *etc.*
- Don’t even look at each others’ code!
- Cite everything
- Review the department ↗ and University ↗ policies

If in doubt, ask!

Having a Pleasant Semester

I intend for this course to be fun and rewarding.

You’ll get out of it what you put in; no more, no less.

I do not take well to grade negotiation.

If you want a better grade, do better work.

If you’re willing to put in the time, I’m willing to help.

Other Policies

Entire submissions or exams will be re-graded only for grading errors.

No incompletes will be given.

No makeup exams will be given.

No grades will be changed for any reason other than grading error.

Systems Programming

This course is concerned with systems programming.

You will learn:

- More about the structure and properties of computer systems
- How architecture affects programs
- How to effectively write efficient and correct programs
- The C programming language and POSIX API

Programming in Context

Programming doesn’t occur in a vacuum.

Computer systems have greatly influenced our:

- Programming languages
- Development tools
- Preferred algorithms

This course will help you understand that context.

Platform

We will be learning about Linux on x86-64.

This is a very common and easily obtainable platform.

To help you, I have prepared a virtual machine image.

You should use this image if possible.
You are not required to use it, but your submissions are required to work on it!

Course Materials and Activities I

Materials to learn from:

- Lectures
- The text: *Computer Systems: A Programmer's Perspective (Third Edition)* by Bryant and O'Hallaron [2].
- The recommended text *The C Programming Language (Second Edition)* by Kernighan and Ritchie [3].
- Recitations
- Assigned (required) readings
- Suggested (optional) readings

Course Materials and Activities II

Activities to learn from:

- Projects
- Recitation activities
- Homeworks (assigned but not graded)
- Exams

Readings

Most lectures will have both required and optional readings.

Readings will appear at the end of the slides.

You must read the required readings even if I do not mention them in class.

You may wish to read the optional readings:

- to expand your understanding of related topics
- to help you understand the required material

Assistance

Our primary forum for assistance will be Piazza.

You should have been added to our course Piazza.

Please consult existing postings before asking a question!

- However, if you're not sure your question is answered, ask!
- If you wish to include code, solutions, *etc.*, send a private message to the instructors!

I will post important course announcements and materials to Piazza.

Programming Projects

A significant portion of your course grade will be projects.

- These are individual projects.
- Projects will be written in C.

Projects must run on the course VM image.

You should download and configure this image in a VM.

GitHub Classroom

We will use GitHub Classroom

- for assignment distribution
- for providing assistance

You must have (or create!) a GitHub account.

You are expected to use git and GitHub for development.

E.g., TAs won't look at code unless it's on GitHub!

Info:

- Git help: Git book, tutorial, Google

Project Assistance

Your TAs will be your primary source of help for projects.

To get the most out of your TAs, do:

- try the obvious things first,
- create minimal examples to show problems, and
- consult the documentation.

To avoid wasting TA time and failing to get help, don't:

- ask for help before you've tried to understand the problem, or
- start at the last minute.

Programming Tools

I will talk a lot about tools this semester.

You will be expected to use a few tools for this course:

- The C compiler
- `make`
- The `gdb` symbolic debugger
- A programmer's editor
- Others …

We will help you learn these tools.

Editors

I don’t care what editor you use, but it must be a programmer’s editor, and be capable of:

- Syntax highlighting
- Automatic indentation
- Brace/parenthesis/*etc.*matching
- Extensibility

Neither I nor the TAs will help you if you are not using an appropriate environment when you seek help!

My personal recommendation is Emacs.

Emacs is installed and configured on the VM image.

Project Submission

We will submit using Autograder.

Submission rules:

- Submitted w/in 24 hours of the deadline: -20%
  - Doesn’t count Saturday or Sunday
  - Doesn’t count University holidays
- Projects submitted after 24 hours will not be accepted

Example 1: Project is due Friday at 11:59 PM, turned in Monday at 3 PM — 20% penalty.

Example 2: Project is due Monday at 11:59 PM, turned in Wednesday at 12:15 AM — not accepted.

Today's Assignments

Immediately:

- Read the Syllabus.
- Join our Piazza instance.

By **Beginning of recitation, 2018-08-29**:

- Create a GitHub account if you don't already have one.
- Download and install the course VM.

By **Friday, 2018-09-07**:

- *Complete the Academic Integrity Quiz* at `https://www.cse.buffalo.edu/~eblanton/misc/academic_integrity/` and turn it in on Autograder.

Recitation

We have Recitation this Wednesday!

Next Time ...

- An overview of C
- An overview of the POSIX API
- A little bit about data types

References I

Required Readings

[1] Ethan Blanton. *Course Syllabus*. https://www.cse.buffalo.edu/ eblanton/course/cse410/materials/2018-1F/syllabus.pdf.

[2] Randal E. Bryant and David R. O'Hallaron. *Computer Science: A Programmer's Perspective*. Third Edition. Pearson, 2016.

Optional Readings

[3] Brian W. Kernighan and Dennis Ritchie. *The C Programming Language*. Second Edition. Prentice Hall, 1988.

License