Syllabus

CSC311 Systems Software

23 November 2015

What we will learn.

Together we will learn what kinds of services an operating system provides. We will learn how to access these services through calls to functions that belong to the operating system. We will learn how an operating systems manages processes, memory, and files.

We will answer questions like these:

- What are measures of performance might inform the selection of an algorithm for the scheduling of processes?
- How can concurrent processes share a resource (such as a file, a cell in the computer's memory, or a network connection) without interfering with one another?
- Under what conditions can concurrent processes become deadlocked?
- How should an operating system allocate space in a computer's main memory or secondary storage?
- How can an operating system give programmers the illusion that they are working with a much larger main memory than the computer actually possesses?

Our principal tools and examples will be the Linux operating system, the Bash shell, and the C programming language.

Who, where, and when.

Instructor	**Professor Leon Tabak**	
Voice	319 895–4294	
E-mail	l.tabak@ieee.org	
Laboratory	Law 113	09:00 a.m. – 11:00 a.m.
Laboratory	Law 113	13:15 p.m. – 14:30 p.m.
Office	Law 206C	14:30 p.m. – 15:30 p.m.

Educational priorities & outcomes.

We will give special attention to three of Cornell College's educational priorities:

- Reasoning—Students will evaluate evidence; interpret data; and use logical, mathematical, and statistical problem-solving tools.
- Communication—Students will speak and write clearly, listen and read actively, and engage with others in productive dialogue.
- Ethical behavior—Students will recognize personal, academic, and professional standards and act with integrity.

Mathematics and logic are foundations of computer science. We will study algorithms and data structures. We will write programs that simulate functions of the operating system. We will write programs that exercise functions of the operating system. We will experiment. We will analyze. We will reason from first principles and examine the data that our simulations produce when we compare solutions to problems.

Success in a rapidly developing technological field requires collaboration. That in turn requires effective communication. You are part of a team and responsible for the success of our team.

Policies.

We will accommodate students with disabilities. If you need an accommodation, please inform me before the end of the third day of the term.

Much of our work will be collaborative. If you have questions about what kinds of collaboration are allowed, please speak with me.

I want you to seek help. I want you to acknowledge the help that you receive. I want you to be generous in helping classmates succeed.

Professional ethics means not only that we avoid dishonest acts but also that we are ready to use our special knowledge and skills for the advantage of clients and the advancement of colleagues.

So long as you are enrolled in the course, I expect you to complete all assigned reading and exercises and to contribute to discussions. Students learn from one another as well as from the instructor. I ask you to be with us each day for your own benefit and for the benefit of your classmates. Come to our meetings each day, prepared and on-time. If you need to miss a meeting of our class, please notify me by telephone or e-mail.

- Please review Cornell College's policy on Academic Honesty (http://www.cornellcollege.edu/registrar/pdf/Academic%20Honesty.pdf).

- Please review Cornell College's policies for Adding and Dropping Classes (http://www.cornellcollege.edu/registrar/gb-resources-student/add-drop.shtml).

- Please review Cornell College's Disability Services and Resources (http://www.cornellcollege.edu/academic-support-and-advising/disabilities/index.shtml).

Grading.

During each of the Wednesdays of the term, I will ask you to present a review of your reading or your experiments in the laboratory to me and your classmates. Part of your professional training is learning how to speak in front of your teammates. Another part of your professional training is learning how to be a good audience to a teammate. Please give your classmates your full attention when they are presenting. Ask questions. Make suggestions.

On the three Fridays of the term, I will ask you to write answers to 10 to 12 questions about our reading, discussion, and work in the laboratory.

You can earn the points for daily work by arriving on time each day, joining in our discussions, and treating everyone else with respect. These points ought to be free. I hope to award all of the available points for daily work to each student in the class.

	Daily work		10 points
	Presentation 0	(Wednesday, November 25)	10 points
	Presentation 1	(Wednesday, December 02)	10 points
	Quiz 0	(Friday, December 04)	10 points
	Presentation 2	(Wednesday, December 09)	10 points
	Quiz 1	(Friday, December 11)	10 points
	Presentation 3	(Wednesday, December 16)	10 points
+	Quiz 2	(Friday, December 18)	10 points
			80 points

Some resources to get us started.

You need not purchase a textbook for this course. We will use free online references and tutorials.

- Programming in the C language
  - Zed Shaw's "Learn C The Hard Way"
  - Daniel Weller's and Sharat Chikkerur's "6.087 Practical Programming in C" (MIT OpenCourseWare)
- Version control (the tool that we will use to share our code)
  - "Learn GIT" (Codecademy)
  - Udacity's "How to Use Git and GitHub (Version Control for Code)"
  - GitHub
  - "Writing READMEs: Because Code is for Humans, Too" (Udacity)
  - "Markdown" (Wikipedia)
- Bash, Emacs, Linux, and Screen (the tools that we will use to write our code)
  - Daniel Robbin's "Bash by Example"
  - "Emacs Reference Card" (PDF)
  - "Linux Quick Reference" (PDF)
  - "Screen Reference Card" (PDF)
- Systems programming
  - Lawrence Angrave's "Systems Programming"

  - Stephen Brennan's "Tutorial—Write a Shell in C"
  - Robert Morris' and Sam Madden's "6.033 Computer System Engineering" (MIT OpenCourseWare)

- Articles by Peter J. Denning
  - "Before Memory Was Virtual" (18 pages)
  - "The Locality Principle" (32 pages)
  - "Virtual Memory II" (5 pages)