Computer Systems — Fall, 2016
CS3650 — Professor G. Cooperman
http://www.ccs.neu.edu/course/cs3650/

Course Description:

Computer Systems discusses computers as an integrated whole, including: **hardware resources** (e.g., CPU cores, CPU cache, memory management unit (MMU), RAM); and **systems languages** (assembly language, C (the low-level high-level language), POSIX threads, the shell (the original UNIX scripting language), and Python (the de facto scripting language of today)). Tying all of this together is the **operating system**, which provides software abstractions (aka programmer's models) for the hardware resources.

At the heart of an operating system is a process table. It provides an abstraction for a process running on a CPU core. A process can be thought of as a running program: code plus data. The CPU core supports code. The CPU cache, MMU and RAM support data. Assembly language, C, the UNIX shell, and Python represent progressively higher levels of abstraction for manipulating these hardware resources.

Today's CPUs contain many cores. This motivates the **POSIX threads** programmer's model in which a program manipulates many CPU cores. Unfortunately, multi-threaded programs are not deterministic: A race condition may expose a bug during one run, but not on the next run. To counter this unfortunate state of affairs, it is also necessary to **model for correctness** any multi-threaded program. We will use the **PlusCal/TLA+** package for modeling C code.

Finally, on the homeworks, I encourage students to share ideas orally, and even to share *small* excerpts of code. (Students often learn best from other students.) But the final coding for the homework must be completely individual. Further, consulting the Internet for ideas is allowed only in the case of text-based articles (in English or another natural language), but *not* for code. Any violations will be considered as violations of academic integrity, and will be dealt with strictly.

Faculty Information:

Professor G. Cooperman
Office: 336 West Village H
e-mail: gene@ccs.neu.edu
Phone: (617) 373-8686
Office Hours: Tues. and Fri.: 11:30 - 12:30; and by appointment.

Textbook:

Computer Organization and Design: The Hardware/Software Interface
(5th edition), 2013, ISBN 978-0-12-407726-3,
by Patterson and Hennessy, Elsevier Morgan Kaufmann Publishers

ONLINE RESOURCES:

Operating Systems: Three Easy Pieces: `http://www.ostep.org` (version 0.80)
UNIX/XV6 SOURCE CODE: `http://pdos.csail.mit.edu/6.828/2014/xv6/xv6-rev8.pdf`

Exams and Grades:

There will be approximately eight homework assignments over the semester, plus a midterm and a final. They will be weighted 40% for the final, 30% for the midterm, and 30% for the homework. All homework assignments will be weighted equally. If sufficient grading resources are not available to the course, then the actual assignments graded may be a subset of those assigned, and the homework grade will be based on an equal weighting of those that are graded.

Syllabus:

Week	*Topics*	*Chapter*
Jan. 8	Introduction, Assembly	Ch. 1, Ch. 2.1–2.8
Jan. 15	Assembly/Machine Language	Ch. 2.1–2.8, Appendix A.6, A.9, A.10 and green card
Jan. 22	Assembly/Machine Language (cont.), symbol table	(cont.)
Jan. 29	C pointers	class lectures

Feb. 5	UNIX syscalls; UNIX shell, fork/exec/wait, fd's	class lectures, ostep.org: Ch. 4, 5, 39.1-39.4
Feb. 12	Python; Cache (direct, set assoc., fully assoc.)	Ch. 5.3–5.4; ostep.org: Ch. 14
Feb. 19	Cache, Virtual Memory & MMU/TLB	Ch. 5.7; ostep.org: Ch. 15, 18, 19
Feb. 26	Virtual Memory (cont.); Mid-term	
Mar. 12	UNIX Process Table; virt. mem. page tables	xv6: proc.h, proc.c, vm.c
Mar. 19	Basics of POSIX threads	ostep.org: Ch. 26, 27.1, 27.2
Mar. 26	Process synchronization, Locks (mutex, semaphore)	ostep.org: Ch. 27.3, 28, 31
Apr. 2	Modeling multi-threaded programs: PlusCal/TLA+	
Apr. 9	Continue PlusCal/TLA+	
Apr.. 16	Finish PlusCal/TLA+ and review for final. (Tuesday is the last day of class.)	
Apr. 23	First possible day of Final Exams (see MYNEU for actual day)	