

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**. The systems languages to be emphasized here are: 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 a simple form of model checking to automatically model and check correctness of 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: Mon. and Wed., 4:30 – 5:30 (after class); and by appointment.

Textbook:

Computer Organization and Design MIPS Edition: The Hardware/Software Interface (6th edition), 2020, ISBN-13: 978-0128201091,
by Patterson and Hennessy, Elsevier Morgan Kaufmann Publishers
An earlier edition of this book can also be used as long as the earlier edition is also based on the MIPS CPU and assembly language.

ONLINE RESOURCES:

Operating Systems: Three Easy Pieces: <http://www.ostep.org> <https://greenteapress.com/wp/semaphore>

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.

ADDED FOR FALL, 2021: Since the university was unable to assign us a time slot for a final during finals week, our final exam will be on Monday, Nov. 29. The final exam will be 30% of the course grade. In addition, there will be a quiz on the last day of classes, Dec. 8, worth 10% of the course grade.

Syllabus:

<i>Week</i>	<i>Topics</i>	<i>Chapter</i>
Sept. 8	Introduction, Assembly	Ch. 1, Ch. 2.1–2.8
Sept. 13	Assembly/Machine Language	Ch. 2.1–2.8, 2.10, Appendix A.6, A.9, A.10 and green card
Sept. 20	Assembly/Machine Language (cont.), symbol table	(cont.)
Sept. 27	C pointers	class lectures
Oct. 4	UNIX syscalls; UNIX shell, fork/exec/wait, fd's	class lectures, ostep.org: Ch. 4, 5, 39.1-39.4
Oct. 13	Python; Cache (direct, set assoc., fully assoc.)	Ch. 5.3–5.4; ostep.org: Ch. 14
Oct. 18	Cache, Virtual Memory & MMU/TLB	Ch. 5.7; ostep.org: Ch. 15, 18, 19
Oct. 25	Cache, Virtual Memory (cont.); Mid-term	
Nov. 1	UNIX Process Table; virt. mem. page tables	xv6: proc.h, proc.c, vm.c
Nov. 8	Basics of POSIX threads	ostep.org: Ch. 26, 27.1, 27.2
Nov. 15	Process synchronization, Locks (mutex, semaphore)	ostep.org: Ch. 27.3, 28, 30, 31
Nov. 22	Model checking of multi-threaded programs	
Nov. 29	Condition variables, read-write locks	
Dec. 6	Finish model checking and threads; review for final (Wednesday is the last day of class.)	
Dec. 10	First possible day of Final Exams (see MYNEU for actual day)	