

## CPS 444/544: Systems Programming I/Fall 2008

---

CPS 444/544 (3 sem hrs) provides an introduction to systems programming in UNIX and C. Topics include library and system calls, operating system structures, concurrency, and interprocess communication (pipes and signals). Students can also expect a survey of various software tools supporting systems programming, including gcc, gdb, make, sed and awk, and lex and yacc. The course does not aim to be comprehensive, but rather focuses on thematic issues. Assignments are designed to provide students with a pragmatic exposure to these tools as well as issues faced by modern practitioners. CPS 444/544 is a programming-intensive course and assumes no prior experience with UNIX or C.

- Pre-requisite(s):** CPS 350 (Data Structures and Algorithms) (with a minimum grade of C for students enrolled in CPS 544) and CPS 346 (Operating Systems)
- Meeting times:** T Th 4:30pm–5:45pm, MH 203
- Instructor:** Dr. S. Perugini, 229–4079, AN 145, last name at udayton dot edu
- Office hours:** T Th noon–1:00pm and 5:45pm–6:45pm, and by appointment.
- Teaching assistant:** John Cresencia, AN 152, cresenjv at notes dot udayton dot edu
- TA office hours:** M W 2:00pm–4:00pm, and by appointment.
- Required textbook:** *The UNIX Programming Environment* by B.W. Kernighan and R. Pike. Prentice Hall, Second edition, 1984. ISBN: 0–13–937681–X.
- Recmd'd textbook:** *The C Programming Language* by B.W. Kernighan and D.M. Ritchie. Prentice Hall, Second edition, 1988. ISBN: 0–13–110362–8.
- Final exam:** 12/16, 4:30pm–6:20pm, MH 203
- Course webpage:** <http://academic.udayton.edu/SaverioPerugini/courses/Fall2008/cps444/>

### Objectives:

- Develop a proficiency in UNIX and C as a systems programming language/environment.
- Survey various system-oriented software tools, including debuggers, and compilation and configuration managers.
- Establish an understanding of the design and development of systems software, such as command interpreters and compilers, through the study of system libraries, pattern matching and filters, interprocess communication, automatic program generation, and signals.
- Explore UNIX internals and establish an understanding of UNIX system calls.

### Evaluation:

Component	Qty	Pts per	Tot pts
Homeworks (lowest dropped)	8	71	497
CPS 544 Project	1	200	200
Exams (9/16, 10/14, 11/13)	3	100	300
Final exam (comprehensive)	1	203	203
<b>CPS 444 total:</b>			1,000
<b>CPS 544 total:</b>			1,200

Each homework requires a fair amount of critical thought and design, and approximately 100–500 lines of code. To prepare students for the realities of computer science problems in industry and

graduate school (and beyond) this course encourages (and rewards) self-reliance and independent, self-directed work. Handwritten assignments will not be accepted. Assignments are due at 4:30pm in class. Late assignments will not be accepted. All exams are in-class, closed-book, and closed-notes. Attendance is mandatory at all examinations; make-ups will not be given. Any missed exam will result in a zero. Make no assumptions about anything; always consult the instructor first. Final letter grades of A, A-, B+, B, B-, C+, C, C-, and D start at 93, 90, 87, 83, 80, 77, 73, 70, and 60, respectively.

**Workload:** CPS 444/544 is a challenging course and moves at a very fast pace. Spending a minimum of 9 hours outside of class each week programming is required. You are advised to see this instructor to discuss any problems you may have before you are evaluated.

**Classroom policies:** Students are expected to conduct themselves with professionalism and integrity. Keep cell phones or similar devices in a silent mode during class. The use of laptop computers or similar devices is not permitted in class.

**Academic integrity:** To achieve the course objectives, assignments must be a sole result of your individual work and must not be shared with classmates. Evidence indicating a violation of this policy will be handled according to the student handbook (Academic Information) and result in a doubly-weighted zero which will not be dropped. Make no assumptions about this policy; always consult the instructor first. No student should ever feel that they must resort to academic dishonesty. You are encouraged to consult the instructor if you are struggling with the course or an assignment. No grade is worth your integrity. Honesty in your academic work will develop into professional integrity. The faculty and students of the University of Dayton will not tolerate any form of academic dishonesty.

Course outline (subject to minor shifts):

Date	Topic(s)	Chapter(s)
	<i>Introduction to Systems Programming in UNIX and C</i>	
8/21	Introduction and the UNIX philosophy	1
8/26	System libraries; I/O	3 & 6
8/28	Topics in C (processes, memory management, error handling)	6
9/ 2	Files and directories (manipulation, management)	2 & 3
9/ 4	The UNIX shell	2 & 3
9/ 9	Process environment (variables, configuration, customization)	3 & 6
9/11	Topics in C (storage classes, thread-safe functions, macros)	3 & 6
9/16	<i>Exam I</i>	
9/18	Files and directories (inodes, hard and symbolic links)	2
	<i>Pattern Matching and Filters</i>	
9/23	(Extended) Regular expressions and (e)grep	4
9/25	Filters, interprocess communication, and sed	4
9/30	awk	4
	<i>Shell Programming</i>	
10/ 2	Command and control	3 & 5
10/ 7	Numbers and arrays	3 & 5
10/14	<i>Exam II</i>	
	<i>Automatic Program Generation</i>	
10/16	Scanning (finite state automata and lex)	8
10/21	Compilation (Makefiles) and configuration (RCS) management	8
10/23	Parsing (Backus-Naur form)	8
10/28	yacc	8
10/30	yacc ( <i>continued</i> )	8
11/ 4	yacc ( <i>continued</i> )	8
	<i>UNIX System Calls</i>	
11/ 6	I/O (open and close, and read and write)	7
11/11	Processes (fork)	7
11/13	Processes (wait and exec)	7
11/18	<i>Exam III</i>	
11/25	Special files (pipes and FIFOs)	7
11/28	Special files (pipes and FIFOs) ( <i>continued</i> )	7
12/ 2	Token ring of processes	7
12/ 4	Signals (job control)	5 & 7
12/ 9	Course reflection	–