

## CSI 400 – Operating Systems – Fall 2004 – MWF 10:25-11:20AM, SS134

<http://www.cs.albany.edu/~sdc/CSI400> (Official location for announcements)

<b>Instructor:</b>	S. Chaiken LI 96H, 442-4282 sdc@cs.albany.edu	<b>Office Hours:</b>	M 11:30AM-12:30PM W, F: 11:30AM-1:00PM and by appointment
<b>Teaching Assistant:</b>	Xiaoyu Zheng LI 96P, 442-4285 xz7223@albany.edu	<b>Office Hours:</b>	To Be Announced and by appointment

**Prerequisites:** Albany CS courses through Programming at the Hardware/Software Interface (CSI 333) or equivalent<sup>1</sup>. [3 semesters of introductory computer science C/C++ and assembly language programming courses covering C++ types, statements, and functions (in C/C++ and assembly language) including arrays, linked lists, stacks, queues, trees, heaps (data structures), addresses, registers, machine and assembly language instructions, binary (and hexadecimal) number systems; Unix software building skills (separate compilation of modules with header files to define their interfaces, use of `make`, debugging, etc.) algorithms for sorting, searching and data structure traversal, recursion, function call linkage via the stack, etc.; skilled use of WWW.]

**Syllabus Summary:** This course will cover the at least the first 3 Parts in Silberschatz, Galvin and Gagne's *Operating System Concepts with Java* (OSC) book plus additional topics as time permits. The emphasis will be on what is entailed by **concurrency** (multiple communicating *processes running at the same time*, actually or apparently) and on **operating system organization** (process/thread, synchronization, virtual memory, input/output device, file system and other resource sharing and protection functionality, components and their implementation). The general principles from OSC will be supported by study of implementation details from Bovet and Cesati's book on Linux and from Linux kernel code, written homework questions and problems, and by systems programming project experience guided by some material from Haviland's book. Other projects will give you experience with concurrent programming with Java and POSIX threads and with simulation of some operating system components. Both required and supplemental readings will be assigned from other material published on the Web.

A supplemental goal is to cover some elements of Java both to provide experience with threads and concurrency, and prepare you for using this currently popular practical language.

**To maintain your registration in the course, send me sdc@cs.albany.edu an email message with the following information:**

1. **Your name.**
2. **Your Albany student ID number.**
3. **The user name assigned by ITS to you for using your UNIX account. If you don't have this because you are a transfer student, go to the ITS help office in LC-27 and request help for using UNIX for Computer Science major courses.**

---

<sup>1</sup>I will waive the CSI333 prerequisite this semester for those who got an A[4.0] in CSI310 and are co-registered in CSI333.

4. **The email address to which I can send you official notices, grade reports and other communication about the course.**
5. **A statement in your own words that you have read the syllabus and need no further clarification about it. If you need clarifications, see or email me and get them first!**

#### **Required Texts, assigned readings and Homework 1 for week 1:**

1. Daniel P. Bovet and Marco Cesati, "Understanding the Linux Kernel", O'Reilly Press, 2003, ISBN 0-596-00213-0. (ULK) **Pages to Read:** 8-10 (Concepts through Processes), 18-23 (Process overview), 72-75 (Processes), 34-35 (Basic Memory), 303-306 (Calling System Calls), 12-18 (Overview of Unix file support system calls).

**Homework:** Create a "glossary" of all the *italicized* words printed within the above assigned readings in ULK. For each glossary word, write an explanation of what it means and how that relates to the meanings of other glossary words and your prior experience. Use your own words to improve your explanations. It will be graded on clarity, completeness and understanding displayed by the writing. Due on Wednesday, Sept. 8.

2. Abraham Silberscahtz, Peter Galvin and Greg Gagne, "Operating System Concepts with Java, 6th edition" John Wiley, 2004, ISBN 0-471-48905-0. **No specific readings yet:** For now, please leaf through book and use the index look up further explanation of concepts from the current reading assignment.

3. Keith Haviland, Dina Gray and Ben Salama, "Unix System Programming, 2nd edition", Addison-Wesley, 1999, ISBN 0-201-87758-9. **Read:** Chapters 1, 2 and 9 through section (9.3.6). Programming project 1 will require the use of most of the functions covered. Do programming project 0: Write a C program that reads the entire standard input file and prints (i.e., writes to standard output) the following data:

Number of characters in the file: $n$

Number of words in the file: $m$

where a "word" is a maximal contiguous sequence of letters. So last phrase has 10 words. (The decimal numerals  $n$  and  $m$  are defined by the messages preceeding them.) The program must use the "read" system call to read the file, the C library function "sprintf" to format the two output strings above, the "strlen" function to determine its their length, and the "write" system call to write the output.

4. Other handouts and Web references that will be provided through the lectures throughout the semester, including Linux source code, Solaris and Posix specifications and references for Java and Posix threads, as assigned.

## Recommended Texts including those from prereq. courses, review for week 1:

1. \*Samuel P. Harbison and Guy L. Steele, jr, “C a Reference Manual”, Prentice-Hall 1995 or later. **Review:** ASCII code for project 1.
2. \*David A. Patterson and John L. Hennessy, “Computer Organization and Design, the Hardware/Software Interface”, Morgan Kauffmann. **Review:** Chapter 3 (roles of instructions, memory addresses, registers, the program counter, branches and jumps, memory access and arithmetic/logical operations in assembly language programs).
3. \*W. Richard Stevens, “Advanced Programming in the UNIX Environment”, Addison-Wesley, 1993; and the (new) Single Unix (POSIX) Specification from <http://www.UNIX-systems.org> are exhaustive but precise “everything you want to know but are afraid to ask” references on what Unix-like systems do, why they do it, and how to program them.
4. Your C++/C textbooks or other C++ textbooks you have found useful.
5. Materials from CSI333 on assembly language programming, the MIPS architecture, perhaps the SPIM simulator, etc. **Review:** See Patterson and Hennessy item above. Specific links will be put on the course web page as needed.

The books marked (\*) above are recommended for you to keep for use in more advanced courses, research and professional work.

### A. Grading:

Midterm 1	–	Oct. 8, 2004 (Fri.)	:	In class	–	25%
Final	–	Dec. 13, 2004 (Thu.)	:	1:00 PM to 3:00 PM	–	35%
Pencil and Paper Homework (6 or 7)	–				–	15%
Programming Projects (4 or 5)	–				–	25%

Final letter grades will be based on cutoffs applied to your score computed as above and modified according to the following policy:

40% on each of at least three of the 4-5 programming projects, a 40% total score AND a final or average exam grade of at least 40% is the minimum for passing the course (with a D-).

Details regarding the exams will be announced later.

**B. Programming Projects:** There will be four to five programming project assignments. You are expected to have a Unix account with U. Albany’s ITS. Projects may be done on ITS Unix (Sun Solaris) systems or other systems such as your own computer with Linux. However, they should be made portable whenever possible and will be submitted to and graded on the Albany’s ITS Unix systems. Hence, you must at test them there before you submit them.

For Academic Computing systems related problems or information, check “Computing” links from <http://www.albany.edu>, send email to [theshed@albany.edu](mailto:theshed@albany.edu), visit LC-27 or call 442-3700.

Although some of the programming work can be done on your own computers, especially if you install a Unix variant such as Linux or FreeBSD, problems (hardware, software, network access) with your system will not be accepted as excuses for late or missing programming projects.

Programming guidelines and submission information will appear in a separate handout. The projects will require the use of some assigned system interface functions and datatypes which are specified in C. So, some C programming will be required.

Depending on the progress we make toward advanced topics, we may provide access to certain CS Dept. computers for late semester projects. If so, usage of such a laboratory system will be strictly limited to activity assigned or specifically approved by Prof. Chaiken to be announced.

**Very Important:** If you do not turn in syntax error free and generally working programs for at least three of the programming projects assignments will, it will result an *E grade for the course*, regardless of homework and exam grades. We follow the same policy as other programming courses above the CSI201 level: **If it doesn't compile, it gets 0 points, no matter how much work you put into it!**

### C. Policy on Cheating:

1. Cheating in an exam will result in an E grade for the course. Further, the students involved will be referred to the University Judicial System.
2. The code and any written answers for programming assignments and lab exercises must be written by yourself. You are welcome to discuss the class material, the problems and ideas for solutions; but each person is expected to write the code and answers he or she submits independently, without copying.

Cheating in a programming assignment or lab exercise will result in a ZERO for that requirement for *all* the students involved. Students who cheat in two or more programming assignments/ lab exercises will receive an E grade for the course.

A report of the every cheating incident will also be made to the Office of Undergraduate Studies in accordance with the University regulations concerning "Penalties and Procedures for Violations of Academic Integrity" in the Undergraduate Bulletin.

We may do automated code comparisons between submissions of current students together with submissions from prior course offerings to detect copying.

3. Violations of Computer and Network Usage policy on University and Dept. laboratory systems are regarded as academic integrity violations, like cheating.

### D. Policies on Computer and Network Usage:

1. Attempts to use University computers or networks in violation of the regulations set forth in [http://www.albany.edu/policies/computer\\_usage](http://www.albany.edu/policies/computer_usage) and in [http://www.albany.edu/judicial\\_affairs/standardsofconduct.html#IXG](http://www.albany.edu/judicial_affairs/standardsofconduct.html#IXG) may result in immediate account suspension, course failure or referral to University disciplinary action. The implications of the responsible use requirements of shared computer and network infrastructure is an element of this course's curriculum and so course failure or grade reduction can be imposed by the instructor as a sanction against the violation of laboratory rules and procedures. Failure to comply with instructions given to you by system administration or course instructional team staff members in regard to Lab usage is disruptive and is grounds for sanction.

2. Ignorance of the `quota -v` command to monitor your account disk quota and the steps you must take to reduce disk space usage are likely to result in your account becoming unusable when you need it most. Lateness of programming projects will not be excused for this reason.

**E. Make-up Exams:** In the event of a valid and verifiable excuse (e.g. a major medical situation), the instructor will choose to either give a make-up exam or waive the exam and weigh the other exam more heavily. In such a case, it is your responsibility to contact the instructor *ahead of time* unless the situation made that impossible.

**F. Policy on I grades:** A grade of **I** will only be given for genuine extenuating circumstances that are beyond your control after the midterm point. Both of the following conditions must be met:

1. Your work must be in good standing as of October 10, 2002, the day of midterm exam; that is, you must have an average score of at least 50% on the programming assignments and at least 50% on the writing assignments due up to that point. Further, your midterm grade must also be equivalent to at least a **C**. Therefore, if you miss the midterm or have performed poorly on assignments, you are not eligible for an **I** grade.
2. Written documentation must, upon request, be supplied about the extenuating circumstance either by you or the University administration (see the Dean for Undergraduate Studies for help).

**G. Disabilities, etc:** Accommodations will be made for clients of the Office of Disabled Student Services upon adequate prior notice and according to that office's policies.

Students with genuine continuing hardship situations, or any disability related problems with computer usage should confer with the professor before October.

**H. Attendance:**

You are responsible for all material presented in the lectures. Some of that material will not be presented anywhere else. Make sure you have a trusted friend to lend you lecture notes if you are going to miss a lecture.

It will *not* be possible for your instructor to conduct makeup classes.

**J. Other Notes:**

1. During their office hours, the instructor and the teaching assistants for this class will be glad to help you with the course material and the programs.
2. In addition to the regular office hours, you can also set up an appointment to meet with your instructor and the teaching assistants. Please call at least a day in advance to set up an appointment.

\$Id: policies.tex,v 2.15 2004/08/31 20:19:10 sdc Exp \$