Course Notes

Class Meetings  This class will meet from 4:00 PM to 5:15 PM in PKI 164 each Tuesday and Thursday from August 25 through December 10, 2015, with the exception of Tuesday, October 20 and Thursday, November 26.

Instructor  The instructor for this class is Professor Stanley Wileman. His office is in PKI 281E. His telephone number is 402-554-3583 (voice mail is never listened to or answered), and the department’s telephone number is 402-554-2834. Electronic mail for the instructor should be sent only to stanw@unomaha.edu; please use a subject that identifies the course number (4500). The e-mail address on loki.ist.unomaha.edu (the system to be used for programming assignments) should not be used. Scheduled office hours are 3:00 PM to 4:00 PM each Tuesday and Thursday on which the class meets. Other office hours are available by appointment. It is strongly recommended that you bookmark the instructor’s web site (see below) and regularly check for any exceptions to this schedule or other course information.

Prerequisites  This course assumes students have a good understanding of data structures and computer organization. Familiarity with assembler language programming for some contemporary processor is assumed, as is a strong background in program development using a contemporary programming language like C, C++, or Java; some of the programming assignments in the course and most of the examples will use the C programming language. Some assignments may allow the use of C++, Java, or Python. A good course in digital design may prove useful.

Course Organization  The course is organized into eleven modules, as follows:

Module 1: Introduction, Overview and History
Module 2: OS Concepts: Processes and File Systems
Module 3: The Process Model, Threads, and Interprocess Communication
Module 4: The Producer-Consumer Problem and Solution Methods
Module 5: Classic Interprocess Communication Problems
Module 6: Process Scheduling Methods
Module 7: Deadlock
Module 8: Input/Output
Module 9: Memory Management – Part 1
Module 10: Memory Management – Part 2
Module 11: File Systems and Security

Each of these modules has an associated set of lectures and a set of Power Point slides. These may be modified as the semester progresses, and the latest version of each will always be available in PDF on the class web page (see below). There are also related reading assignments in the textbook, a list of which is provided on the class web page (labeled “Textbook Readings”).
Class Meeting Schedule The class meeting schedule, indicating the course module(s) to be covered, dates of quizzes, issue/due dates for assignments, and the date of the comprehensive final examination appears on the class web site (see below). It will be updated as necessary to reflect any schedule changes.

Textbook The textbook for the course is *Operating Systems Design and Implementation* (third edition) by Andrew S. Tanenbaum and Albert S. Woodhull (Pearson/Prentice Hall publishers, copyright 2006, ISBN 0-13-142938-8). Most of the material in the first five chapters will be covered. Students are responsible for reading all of this material, even though some of it will not be covered in lectures; quizzes and the final examination may cover any this material (and other material covered in the lectures but not in the text). As noted above, the approximate correspondence between the class modules and the textbook sections appears in a document on the class web pages.

Web Sites cs3.ist.unomaha.edu/~stanw is the instructor’s web page. The web pages for the class are at cs3.ist.unomaha.edu/~stanw/153/csci4500/index.html. You will find it convenient to bookmark this in your browser for easy access. The pages should be checked regularly for announcements and other materials.

Somewhat dated Flash presentations of the slides with audio narration are available at cs2.unomaha.edu/~stanw/csci4500. There are also links to these from the class web page. You may use these if you wish, but make certain to also consider the current material and the textbook.

The university’s Blackboard site will be used only to communicate grades. The class web page has all other course materials.

Grading Grades will be based on three components:

- six 25-minute multiple choice quizzes (5 percent each, 30 percent total)
- three programming assignments (50 percent total, weighted equally)
- a comprehensive multiple choice final examination (20 percent).

The quizzes will be given during the last 25 minutes of class on the dates they are scheduled.

The final grade will be determined using the following scale, or one that is slightly more generous, as the numeric break points between the different letter grades may be lowered (based largely on overall class performance), but they will never be increased.

<table>
<thead>
<tr>
<th>Grade</th>
<th>Percentage</th>
</tr>
</thead>
<tbody>
<tr>
<td>A+</td>
<td>97 to 100 percent</td>
</tr>
<tr>
<td>A</td>
<td>93 to 97 percent</td>
</tr>
<tr>
<td>A–</td>
<td>90 to 93 percent</td>
</tr>
<tr>
<td>B+</td>
<td>87 to 90 percent</td>
</tr>
<tr>
<td>B</td>
<td>83 to 87 percent</td>
</tr>
<tr>
<td>B–</td>
<td>80 to 83 percent</td>
</tr>
<tr>
<td>C+</td>
<td>77 to 80 percent</td>
</tr>
<tr>
<td>C</td>
<td>73 to 77 percent</td>
</tr>
<tr>
<td>C–</td>
<td>70 to 73 percent</td>
</tr>
<tr>
<td>D+</td>
<td>67 to 70 percent</td>
</tr>
<tr>
<td>D</td>
<td>63 to 67 percent</td>
</tr>
<tr>
<td>D–</td>
<td>60 to 63 percent</td>
</tr>
<tr>
<td>F</td>
<td>Less than 60 percent</td>
</tr>
</tbody>
</table>

Graduate students (those taking CSCI 8506) will be required to write a short research paper on a topic to be approved by the instructor; the paper will add an additional 20 percent to the total weight, so the total possible is 120 percent. The weight of each programming assignment will be identified at the time the assignment is presented.
Programming Assignments  As noted above, there will be three programming assignments for this course. Solutions to the first programming assignment must be written in the C programming language. Solutions to the other assignments may be written in C, C++, Java, or Python, as specified in the problem statement. These solutions will be tested on a university Linux system (specifically, loki.ist.unomaha.edu) on which each student will have an account. Detailed information on the preparation, submission, and evaluation of solutions will be provided with the assignments on the class web page.

The first programming assignment emphasizes the use of concurrent programming and system calls, and the other assignments focus on various algorithms and methods utilized in operating systems. However, the basic skills used in moving from a problem statement to a solution design, implementation, and testing are still necessary. There are several important points to remember about the programming for this course:

1. The programming assignments are not trivial, nor are they incredibly complex. They do, however, require a reasonable amount of time to design, implement, and test a solution. To give you some idea about the amount of code that will likely be needed, the instructor’s solutions to the three programming assignments for a recent offering of this course were 303 lines, 916 lines, and 672 lines of code. These solutions were well-commented and the code included checks for input data violations (although you aren’t required to check for input data violations unless the problem explicitly states this). You should comment your code appropriately.

2. Significant time in the course schedule has been given to develop and test a solution for each of the programming assignments, but do not delay beginning work on the solutions after the problems are assigned. While some late assignments may be accepted (subject to suitable justification), in general, late assignments will not be considered.

3. Unfortunately, some individuals who attempt this course do not have appropriate programming skills. As the scores on the programming assignments will constitute 50 percent of the final grade, inability to produce reasonable solutions to the assignments will undoubtedly result in failure in the course. If you find that you’re having significant difficulty with the programming – but not the operating system concepts – then you are advised to drop the course and acquire better program development skills before attempting it again.

Of course, the instructor is always willing to assist students with problems related to the course, including programming problems. Do not, however, expect that you will be able to learn sufficient programming skills during the course if you do not already possess them.

Computing Resources  You are expected to be aware of, and abide by, the policy for responsible use of university computers and information systems. A copy of this policy can be found at www.nebraska.edu/about/exec_memo16.pdf. You may already have an account on the system to be used for completing programming assignments, loki.ist.unomaha.edu. You should immediately determine if your account works. If you do not have an account, or if your previous account is disabled, inform the instructor immediately. You may also find it useful to view the web site at http://loki.ist.unomaha.edu.
The Class Directory  The directory /home/stanw/csci4500 on loki.ist.unomaha.edu will contain files that will almost certainly be of significant value in developing your solutions to the programming assignments for this course. The text file MANIFEST in that directory will contain a list of these files and notes about them. Typically the directory includes small programs illustrating various system calls and concurrent programming, as well as the instructor’s solutions (executable form only!) and representative data that can be used to test your solutions. The directory and the files in it are protected from modification, but you should feel free to copy them to a directory where you have write permission to modify and/or compile them.

Department Policy on Cheating and Plagiarism  The general university policies on cheating and plagiarism apply within the department. Unless otherwise specified by an instructor, student work shall represent only the individual effort of that student, with portions of that work done by others given appropriate attribution. If a group effort is explicitly permitted or required by the instructor for one or more assignments, then the instructor shall indicate which part(s) of the assignment must be completed on an individual basis, if any.

If an instructor believes a student has plagiarized the work of another (regardless of whether the other person is a student in the same section/class or not), or represented as their own work that which another person produced (whether on a paid basis or not), then that instructor shall inform the student of the suspicion. The student shall be given an opportunity to explain, if they wish, why the work was not plagiarized. If after such student explanation the instructor still believes the work was plagiarized, the instructor has the responsibility for assigning an evaluation to the work that is substantially lower than if the work had not been completed at all. The department chair will be notified for the action. If the student whose work being copied or plagiarized knows the fact but does not take a proper action, the student will be held responsible the same as the copying or plagiarizing student.

If a second occurrence of plagiarism is evidenced for the same student, the instructor has the responsibility of assigning a grade of F to the student for the course and informing the registrar’s office that the student will not be permitted to withdraw from the course. Both the department chair and the college dean will be notified for the action.

Repeated occurrences of plagiarism (in multiple courses) by the same student shall result in notification of the Vice Chancellor for Academic Affairs and/or the Graduate Dean, as appropriate, from the department chair or the college dean, and possible dismissal of the student from the program.