CS3411 Systems Programming

Course Syllabus

Instructor: Soner Onder
Office: Rekhi 303
Telephone: 487-2123
E-mail address: soner@mtu.edu
Office Hours: Thursday 2pm-3pm
Other times by appointment

TA: Tino Moore
Office: Rekhi 104
E-mail address: tinom@mtu.edu
Office Hours: T,W,TH 11 - 11:59am
Other times by appointment

Prerequisites:

CS1142 - Programming at the Hardware Software Interface
CS3421 - Computer Organization

General Description:

This course will cover basics of systems programming in a UNIX environment. We will cover operation of fundamental elements of the UNIX operating system interface, as well as certain scripting tools commonly used to manage UNIX system software resources. Systems programmers develop programs that often run in a privileged mode, with unrestricted access to system resources, and that are fundamental to correct operation of the programming platform presented to users. A primary goal of the course then is to reinforce, in a systems programming context, the skills you have already acquired to develop code that is robust. The topics to be covered include:

<table>
<thead>
<tr>
<th>Topic</th>
<th>Subtopics</th>
</tr>
</thead>
<tbody>
<tr>
<td>Review</td>
<td>C language, make</td>
</tr>
<tr>
<td>UNIX file system interface</td>
<td></td>
</tr>
<tr>
<td>Processes</td>
<td></td>
</tr>
<tr>
<td>Linking and Libraries</td>
<td></td>
</tr>
<tr>
<td>Interprocess communication</td>
<td>signals, pipes, sockets</td>
</tr>
<tr>
<td>Terminal I/O</td>
<td></td>
</tr>
<tr>
<td>Time and Timing</td>
<td></td>
</tr>
<tr>
<td>TCL/TK</td>
<td>(as time allows)</td>
</tr>
</tbody>
</table>

Course Text: *Advanced Programming in the UNIX Environment*, by W. Richard Stevens

The text book is not required. Overheads will be distributed and, together with class notes, are expected to provide adequate information for completion of the programming projects and in-class tests. However, if you can afford it, the textbook will be helpful.

The following reference books may also be helpful:

Grade Determination:

The course grade will be comprised of the following components.

55 % Programming Projects - Programming projects (probably five) will comprise the most significant portion of the course grade. A later project will have an equal or larger weight than an earlier project. For example, if we do five projects, the weight distribution could be 5 %, 5 %, 10 %, 15 %, 20 %. The number of projects is dynamically malloc'ed based on the pace of your learning. A 6-th project may potentially free your worst project grade. Project submissions must be uploaded to canvas. Email submissions will not be accepted.

45 % Tests - There will be two in-class tests and a final exam with weights of 10 %, 15 %, and 20 % respectively.

Programming: The grading criteria for submitted projects includes readability, robustness, and efficiency. Correct operation on normal input will not achieve an A grade. Submitted programs are expected to terminate normally under reasonably foreseeable errors in user input and are expected not to contain gross inefficiencies in resource usage.

Over the course of the semester, you may have five slip days. Each person is given an automatic extension of five calendar days, which you can use on any programming project during the semester, in any increment, as long as the total amount of lateness does not add up to more than five days. For instance, you can hand in one program five days late, or two programs two and three days late respectively. When you hand in a program, you must identify at the top of the assignment, (i) how late this assignment is, and (ii) how much of the total slip time you have left. Each calendar day that passes beyond the due date counts as a slip day. Slip days are used in increments of days, not hours. If an assignment is submitted one hour after the due date, a slip day has been used for that assignment. Once the five slip days have expired, projects can be submitted at a penalty of 20% per day. No slip days can be used beyond the last day of classes, and no projects will be accepted after the last day of classes.

Programming projects are to be performed individually. You may neither show anyone your project code nor look at the code of anyone else. (This policy extends to any external resource, including code found on the web or individuals who are not enrolled in the course.) However, you may engage in empty hands discussions with anyone. No participant in an empty hands discussion should leave with written or printed material he/she did not enter the conversation with. If you are unsure whether or not a particular type of collaboration is allowed, you are expected to check with me before engaging in the collaboration.

Notes:

- Class attendance is required. Attendance will be taken. Your attendance ultimately decides what grade you will get in the class. If your circumstances may force you to miss a large number of classes, you should talk to me first to see what we can do to make sure you can learn the course material.

- A course web page will be maintained at http://www.cs.mtu.edu/~soner/Classes/CS-3411/Teaching.html. This page is maintained only as a convenience. In the event of a conflict, information given during class supersedes any contained on the web page.

- Student work products (exams, essays, projects, etc.) may be used for purposes of university, program, or course assessment. Work used for assessment purposes will not include student identification.