CS 485-001 Systems Programming
University of Kentucky
Department of Computer Science
Fall 2015

Course Information

Meeting Times
MWF 1:00 pm - 1:50 pm, Room CB 334

Instructor
Jim Griffioen, Hardymon Bldg 229, 257-6746, griff@netlab.uky.edu

Office Hours
MW 2:00pm - 3:00pm (Hardymon 229), or by appointment. You can also send questions via email.

Course Description

This course provides an introduction to computer systems and explores computer architecture, operating systems, and networks from a programmer’s perspective. The course also introduces advanced programming and debugging tools. Topics include hardware instruction sets, machine language and C language program representations, linking/loading, operating systems (process management, scheduling, memory management, interprocess communication, and file systems), network programming (socket programming and web protocols), and common security attacks and solutions.

Course Objectives

The goal of the course is to provide an overall view of the components that make up a computer system and the ways in which programs interact with those components. The course touches on a broad range of topics including hardware, assembly language, compilers, operating systems, networking, and graphical users interfaces. Topics are presented from a programmer’s perspective with the goal of developing a complete understanding of what happens when a program is executed. The course also introduces useful programming and debugging tools that can assist in creating robust and efficient code. The course provides foundational programming skills required by many upper level classes.

Hands-on projects and in-class assignments will make the concepts real and expose students to the challenges of developing robust and efficient software.
**Learning Outcomes**

Students will acquire knowledge of computer systems and how to program them. More specifically, students will be able to

1. Demonstrate an understanding of overall computer system structure and operating system design.
2. Demonstrate an understanding of how specific high-level language program constructs are implemented in assembly language.
3. Be able to effectively use a debugger.
4. Demonstrate an understanding of the concepts of processes and threads and use them to achieve concurrent execution in applications.
5. Demonstrate an understanding of the interfaces provided by the memory management component of the operating system.
6. Use the interfaces provided by the file system in developing applications.
7. Use a networking API to create networked applications

**Measures of Learning Outcomes**

The above outcomes will be evaluated based on the programming projects, in-class/computer lab assignments, and exams. They will also be evaluated based on the students’ self-assessment of their mastery of these outcomes performed at the end of the semester.

**Course Prerequisites**

The course assumes a basic understanding of computer hardware. It also assumes a basic level of proficiency in the C or C++ programming language. Students must have completed EE 280. Students should also have taken CS 216.

**Textbook (Required)**

The following textbook is required for the course:

  (The webpage for the book can be found at http://csapp.cs.cmu.edu/2e/home.html).

**Exams**

- **Exams**: There will be two exams:
  - Midterm Exam: Friday, October 9, 1:00pm (In-class)
  - Final Exam: Wednesday, December 16, 1:00pm, (Scheduled Exam Slot)
• Programming Projects:
  Programming projects will be given regularly throughout the semester (roughly every three weeks, although the exact dates and timeframes will vary depending on the particular project). The goal of the projects is to reinforce the concepts and material discussed in class. All projects must be implemented using the computing system and development environment specified in the project description. In some cases, the development environment may be constrained for the purposes of teaching a particular programming skill.

  To provide hands-on experience working with various software tools, the Friday class period will often be used for in-class assignments. Students are required to bring their laptops on Fridays and must complete the assignment during the class period.

  In general, most projects and in-class assignments will be based on the C programming language and the Linux development environment. Students are expected to already know how to program in C/C++. Please consult http://www.cs.uky.edu/%7eraphael/programming.html for a list of programming skills that students should have familiarity with prior to taking the course. Unless specified otherwise, projects will be done individually by each student. Students must also read and adhere to good programming standards (see http://www.cs.uky.edu/%7eraphael/checklist.html for a checklist of good programming characteristics). Programs will be penalized for poor programming style.

  All programming projects and in-class assignment will be submitted electronically. An explanation of how to submit your work will be given with each project.

  The class textbook has an associated “student site” that can be found at http://csapp.cs.cmu.edu/public/students.html. The student site has pointers to helpful resources that better describe the C language and Linux operating system/programming environment. Additional resources for specific chapters in the textbook are also available on the student site.

Policies

Grading

Grades will be determined based on programming assignments (50%), exams (45% = 20% Midterm Exam + 25% Final Exam), and in-class/computer lab assignments (5%). Final grades will be assigned according to the following scale:

A=90-100%, B=80-89%, C=70-79%, D=60-69%, E=0-59%.

No incomplete grades will be assigned unless there exist exceptional, extenuating circumstances.
Late Penalty

Late programs will be penalized 10% (of the total possible points) per day for each day late not including weekend or holiday days. No project may be submitted more than five days late not including weekend or holiday days.

Academic Integrity

Per university policy, students shall not plagiarize, cheat, or falsify or misuse academic records. Students are expected to adhere to University policy on cheating and plagiarism in all courses. The minimum penalty for a first offense is a zero on the assignment on which the offense occurred. If the offense is considered severe or the student has other academic offenses on their record, more serious penalties, up to suspension from the university may be imposed.

Plagiarism and cheating are serious breaches of academic conduct. Each student is advised to become familiar with the various forms of academic dishonesty as explained in the Code of Student Rights and Responsibilities. Complete information can be found at the following website: http://www.uky.edu/Ombud. A plea of ignorance is not acceptable as a defense against the charge of academic dishonesty. It is important that you review this information as all ideas borrowed from others need to be properly credited.

Part II of Student Rights and Responsibilities (available online http://www.uky.edu/StudentAffairs/Code/part2.html) states that all academic work, written or otherwise, submitted by students to their instructors or other academic supervisors, is expected to be the result of their own thought, research, or self-expression. In cases where students feel unsure about the question of plagiarism involving their own work, they are obliged to consult their instructors on the matter before submission.

When students submit work purporting to be their own, but which in any way borrows ideas, organization, wording or anything else from another source without appropriate acknowledgement of the fact, the students are guilty of plagiarism. Plagiarism includes reproducing someone elses work, whether it be a published article, chapter of a book, a paper from a friend or some file, or something similar to this. Plagiarism also includes the practice of employing or allowing another person to alter or revise the work which a student submits as his/her own, whoever that other person may be.

Students may discuss assignments among themselves or with an instructor or tutor, but when the actual work is done, it must be done by the student, and
the student alone. When a student’s assignment involves research in outside sources of information, the student must carefully acknowledge exactly what, where and how he/she employed them. If the words of someone else are used, the student must put quotation marks around the passage in question and add an appropriate indication of its origin. Making simple changes while leaving the organization, content and phraseology intact is plagiaristic. However, nothing in these Rules shall apply to those ideas which are so generally and freely circulated as to be a part of the public domain (Section 6.3.1).

Please note: Any assignment you turn in may be submitted to an electronic database to check for plagiarism.

Attendance Policy

Daily attendance is not taken; however, you are nevertheless expected to attend daily lectures. The number one factor that correlates with success in engineering is going to class. Attendance in lecture is required on the days of in-class exams and in-class labs. In case of a legitimate unforeseen conflict (for example, illness or death in the immediate family), if the student contacts the instructor within a week of the test date and provides proof of the legitimate excuse, the student may take the exam/in-class lab (or a substitute exam/in-class lab) at a later time.

In case of other conflicts, the student must inform the instructor in advance as far as possible. Students must take exams before the scheduled time, and must arrange a time with the instructor to make up in-class labs. Advance notification of such a conflict does not automatically guarantee a special exam or in-class lab will be scheduled; the instructor will make the final decision.

Excused Absences

Students need to notify the professor of absences prior to class when possible. S.R. 5.2.4.2 defines the following as acceptable reasons for excused absences: (a) serious illness, (b) illness or death of family member, (c) University-related trips, (d) major religious holidays, and (e) other circumstances found to fit “reasonable cause for nonattendance” by the professor.

Students anticipating an absence for a major religious holiday are responsible for notifying the instructor in writing of anticipated absences due to their observance of such holidays no later than the last day in the semester to add a class.

Verification of Absences

Students may be asked to verify their absences in order for them to be considered excused. Senate Rule 5.2.4.2 states that faculty have the right to request
“appropriate verification” when students claim an excused absence because of illness or death in the family. Appropriate notification of absences due to university-related trips is required prior to the absence.

Making up Missed Work

Students missing any graded work due to an excused absence bear the responsibility of informing the Instructor about their excused absence within one week following the period of the excused absence (except where prior notification is required), and of making up the missed work. Work missed due to an excused absence should be made up, if feasible, before the end of the semester. Students should meet with the instructor to devise a plan and establish a timeline for completing makeup work.

Accommodations due to disability

If you have a documented disability that requires academic accommodations, please see me as soon as possible during scheduled office hours. In order to receive accommodations in this course, you must provide me with a Letter of Accommodation from the Disability Resource Center (Room 2, Alumni Gym, 257-2754, email address jkarnes@email.uky.edu) for coordination of campus disability services available to students with disabilities.

Additional References

The following books are not required, but you may find them helpful if you feel you could use additional sources of information:

- Structured Computer Organization, 5/e, Andrew S. Tannenbaum, Prentice Hall, 2006
- The Linux Kernel Primer, Rodriguez, Fischer, and Smolski, Prentice Hall, 2006
Schedule

The following is a tentative outline of the material that will be covered each week. Specific reading assignments to accompany the lectures will be assigned throughout the semester.

(1) Course Overview, Introduction to Machine Data Representation
(2) Machine Level Representation of Programs: Basic Programs and Control
(3) Using Programming Tools: Compilers, Debuggers, Disassemblers
(4) Machine Level Representation of Programs: Procedures and Data Structures
(5) Linking and Loading
(6) Memory Management: Physical Memory, Allocation/Reclamation, Swapping
(7) Processes: System Calls, Process Control and Concurrent Execution
(8) Process Management: Process/Thread creation, Context Switching
(9) Process Management: Scheduling, Synchronization
(10) Virtual Memory: Address Translation, Page Replacement
(11) File Systems: APIs, Naming, Access Control, and File Structures
(12) Internetworking: Internet Protocol
(13) Network Programming: Network API Calls
(14) Web Services: Client/Server and Web Protocols
(15) Graphic User Interfaces: Windowing Systems and Devices
(16) Miscellaneous: Virtual Machines, the Cloud, Security