CS 332-Computer Networks-Spring 2010

Professor
Lewis Barnett
Office
Jepson Hall 212-A
(804) 289-8091
lbarnett@richmond.edu

Office hours: MTWR 2:00 pm - 3:00 pm, or by appointment. Please note that I am VERY flexible about office visits. Come on by.

Course Description
I sometimes think of computer networking as the public face of computer science. Increasingly, the Web has become the aspect of computing that the world engages in most frequently, and a huge portion of what we talk about this semester lies just behind that user interface. Network protocols like HTTP and TCP/IP make all of the amazing and appalling stuff that goes on the Internet possible, and network infrastructure like the Domain Name Service and the Internet backbone have become as important or possibly more important to life as we know it than the telephone system.

This course is intended as a general introduction to the principles of computer networking. No previous knowledge of the topic is assumed. Topics will be arranged roughly around the ISO seven layer reference model for networks, which specifies the type of services that should be available at various levels of network interconnection. We will spend proportionately more time at the Network (IP) layer and above, as those are the layers that are of most interest to software developers and architects.

The course will begin by discussing some of the underlying issues in establishing communication between two computers. At the lowest end of the spectrum, we will talk about issues related to the physical connections between computers. From the basics, we will move on to consider international standards for protocols at various levels of the ISO reference model such as Ethernet, TCP/IP, et cetera. These standards span a wide variety of transmission media, network layouts, and access techniques. We will also spend time discussing some of the applications that are being built upon these various types of networks and the implications that such applications have for computer science and for society as a whole.

Prerequisites
CS 301 - Computer Organization, or instructor’s permission
Readings


Additional readings from the technical literature and other sources will be distributed as needed.

Much system-level coding such as the projects you will be doing this semester are written in straight ANSI C, due to considerations of speed, code compactness, and potential portability. C is the base language for C++, but without the object-oriented features. This semester, I will accept your code in either C or C++. I expect that you already have some reference materials on C++. The Donahoo and Calvert book uses ANSI C. You will probably also want a reference on ANSI C. There are a great many similarities between C++ and C, but it is best to have a book specific to C. Here are some that I am familiar with or have heard reasonable things about:

- The definitive reference is Kernighan & Ritchie, *The C Programming Language*, 2nd Edition(ISBN 0131103628). It is a slender volume, somewhat terse (like the language), but it is complete. Though published in 1988, it contains everything you need to know about the core C language (there’s a reason it’s still in print after 20 years in a fast-moving discipline like CS).

- A bit more verbose, but also useful, is Steve Oualline’s book *Practical C* (ISBN 1565923065), but if you have the C++ version, there is a great deal of repetition between the two.

- *C Primer Plus* by Stephen Prata is a more recent book that gets good reviews on Amazon as a book for novice programmers, so you may find it a bit slow. I haven’t seen this book personally.

- I have also heard good things about *Expert C Programming* by Peter Van Der Linden (ISBN 0-13-177429-8), though it is more of an advanced book. This is a great second book on C once you are familiar with the language.

Attendance

Each unexcused absence will result in a 1 point reduction in your final average. I have a fairly broad concept of what constitutes a reasonable excuse for an absence, but I have my limits. Having a lot of work for other courses, hangovers, being in jail, or forgetting how long it actually takes to play “Free Bird” in Guitar Hero won’t cut it.

Help

Programming projects and homework assignments may be discussed with others subject to the "empty hands" policy - you may freely discuss ideas and approaches with other students subject to the restriction that each student must leave the discussion without any
written or otherwise recorded material. In your project write-ups, you must also document any person or source that you consulted for that project. Failure to comply with this policy will be treated as an Honor Code violation. It is never appropriate for code written by another person to appear in your program unless I have given specific permission for you to use the code. If you experience problems in completing a programming assignment, please see me for help during office hours, or make an appointment to see me at another time if you are unable to come during office hours. You may also ask me questions by email. Please begin work early on programming assignments! I may not always be reachable in the final hours before a deadline.

The final project will be done in groups. I understand that there can be problems with group work, but the fact is that almost all computing work, whether it is software development or IT consulting, is group work, and you need to develop the communication and interaction skills to deal with this reality. You will have an opportunity to do peer evaluations, which I will take into account in the grading of the project.

**Grading Policy**

<table>
<thead>
<tr>
<th>Grade Component</th>
<th>Percentage</th>
<th>Due Dates</th>
</tr>
</thead>
<tbody>
<tr>
<td>Labs/Projects</td>
<td>35%</td>
<td></td>
</tr>
<tr>
<td>Quizzes</td>
<td>10%</td>
<td>Weekly</td>
</tr>
<tr>
<td>Tests</td>
<td>35%</td>
<td>Test 1: Thurs., Feb. 18 (tentative) Test 2: Thurs., April 1 (tentative)</td>
</tr>
<tr>
<td>Final Exam</td>
<td>20%</td>
<td>Mon., Apr. 26, 9:00am - 12:00 noon</td>
</tr>
</tbody>
</table>

I expect assignments to be turned in on time. Written assignments not received by the stated due date will receive a grade of 0. With projects and lab work making up 35% of the grade, missing any assignment will have a significant impact on your final grade.

I will be assigning weekly homework problems, but I will not be collecting or grading them. Most of the questions for the weekly quizzes will come from the homework. The quizzes will be short (approximately 10 minutes) so you will need to be very familiar with the concepts and solutions for the homework problems before taking them. I am happy to answer questions about the homework; that’s one of the main purposes for office hours, but in order for this to work for you you have to start the assignments early enough to talk to me before the weekly quiz about problems you don’t understand. And remember: I understand that your schedule may not line up well with my stated office hours. Don’t let that stop you from coming by my office. You may want to check with me before making the trip to my office to be sure I’m available, but don’t get the idea that I consider this some kind of imposition. Yes, I’m the department chair, but that’s just something I do because somebody has to do it. Teaching and working with students is what I get up in the morning for.

Some of the programming project assignments have been used in previous semesters. I don’t mind you talking with students who have taken the course before, but you are not permitted, under any circumstances, to receive or view either hard copies or electronic copies of any source code from any source related to these projects, except source code.
that I may distribute to you. You are also not allowed to disassemble executables and use the resulting code in your projects.

A note on project grading: my grading rubric includes program correctness, programming style and/or design quality, and a written description of your work. Correctness is the major component, and in order to receive credit for program correctness, you program must compile and execute. Please develop a strategy for maintaining a copy of your most complete, stable partial solution to the problems that is separate from your current working copy! The easiest and safest way to do this is to use some source control tool like CVS, RCS, Subversion, Mercurial, or Git. This can prevent a last minute change, an errant paste while “adding comments” (another reason to do that first or as you go along), or some other random event from leaving you in the position of not meeting the “compile/execute” criterion for program correctness.

Labs

We will be scheduling a one hour per week lab session at our first meeting. The labs will be focused on the programming tools and techniques you need to successfully complete the programming projects for the course. Some lab sessions will be used for additional lecture time to introduce new programming topics, some will be exercises in useful tools, some will be launching points for projects, etc.

Course Outline

The following is a tentative (and rather ambitious) list of topics to be covered this semester. I will be focusing on commonly used algorithms and techniques and protocol standards which are in common use rather than on specific implementations of these protocols. So, you are unlikely to hear much about Novell Netware or Microsoft Network in class (though there may be a few case studies to illustrate some of the things we talk about). However, the material presented here should give you a good foundation for understanding the workings and design of any particular networking technology you are confronted with. In particular, you will note that we really won’t be talking about specific networking products or about system administration; if that is where your interests lie, there are plenty of commercially available certification courses you can take. I will suggest that if you successfully complete this course, you should be in a position to teach yourself pretty much anything you would need to know about specific networking hardware or software.

Proposed list of topics:

**Transmission Media**
- Copper, Optical Fiber, Wireless transmission

**Networking Basics**
- Circuit Switching vs Packet Switching
- Point to point transmission techniques
- Shared medium transmission techniques
Network programming techniques

Local Area Networks (LANs)
   IEEE 802.3: Ethernet

Wireless Communication
   IEEE 802.11 networking

Internetworking
   The IP protocol
   Internet routing

End-to-end issues
   The TCP and UDP protocols
   Congestion control

Network Security
   Typical network weaknesses
   Infamous intrusions

Networking Applications
   World Wide Web
   Voice Over IP (VOIP)
   Video conferencing
   Peer-to-Peer networks
   Cloud computing
   (We may not get to all of these)