

CS260: Computer Networks I -- Interprocess Communication

Prerequisites: CS180 and CS225

Classroom: Michelangelo

Time: Wed/Fri 3:00pm to 4:20pm

Instructor: Siobhan (Steph) Beeman

Contact Info: steph.beeman@digipen.edu

Office Hours: Mon 1pm to 7pm or by appointment

Class Webpage: CS260 at <http://distance.digipen.edu>

Course Description

Computing has fundamentally changed over the past decade. Previously, most computing problems were solved using a single processing core. Today, between multi-core CPUs, general-purpose GPU computing, and of course Internet-connected systems, almost all interesting work is spread across multiple independent processes. Programming in this environment demands special skills to handle the four key challenges of distributed systems: *danger*, *uncertainty*, *scale* and *time*.

In this course, you will get your first introduction to these key issues and technologies involved in modern multi-process computing. We'll look at the evolution of the Internet, UDP and TCP socket programming, the development and use of application-layer protocols, and techniques for synchronizing state among processes. We'll pay particular attention to network security and secure coding practices.

Course Objectives and Learning Outcomes

This course will give you:

- A strong understanding of TCP/IP and its underlying algorithms
- A strong understanding of socket programming on both Windows and Linux
- A good understanding of security threats to networked applications and some of their mitigations
- An introduction to the challenges of networking in game programming and some of their solutions
- A good understanding of threading and how it applies to networked applications
- An introduction to key application layer protocols such as HTTP

Programming Environment

Windows is the most important platform for client software development, while Linux is the most important platform for server development. In this class, we will thus use both.

You should have encountered Linux already in CS180. We will be using the same version of Linux you used there: Mint 17 “Qiana” – Cinnamon (64-bit) (available at <http://www.linuxmint.com/edition.php?id=158>). DigiPen’s computers should all have VirtualBox 4.3.12 already installed on them, allowing you to install Mint in a VM; you can get VirtualBox for your personal computers at <https://www.virtualbox.org/wiki/Downloads>.

If you need a refresher on Linux, *The Linux Command Line* is an excellent reference: <http://linuxcommand.org/tlcl.php>.

On the Windows side, the default environment found on DigiPen computers is sufficient.

All source code from our textbook is available at <https://github.com/MultiplayerBook>. You *may* use the textbook code samples as the starting point for your assignments in this class (and CS 261), but *do not* assume that you can use that code for any other class, particularly GAM project class—other classes have their own IP policies.

Textbooks

Our required textbook is *Multiplayer Game Programming: Architecting Networked Games* by Joshua Glazer and Sanjay Madhav.

The TCP/IP Guide by Charles Kozierok is an excellent companion volume, but it's not required. Its content is available for free online at <http://www.tcpipguide.com/free/index.htm>.

Outline and Tentative Schedule

This schedule is tentative. The following topics are subject to change at any time.

- Week 1: Overview; history of the Internet; Internet protocol
Read pages 15-39
- Week 2: UDP; socket programming
Read pages 65-83; begin assignment 1
- Week 3: TCP
Read pages 42-52, and 83-88
- Week 4: Serialization
Read pages 101-136
- Week 5: Time
Read pages 52-61, 88-98, and 199-205; begin assignment 2
- Week 6: Uncertainty
Read pages 206-231
- Week 7: Midterm review; **midterm exam**
Review the reading assignments from weeks 1-6.
- Week 8: HTTP and other protocols
Read pages 289-310; begin assignment 3
- Week 9: Danger
Reading assignments TBD.
- Week 10: Mitigations and defensive coding
Read pages 265-278, plus additional reading TBD.
- Week 11: Cryptography
Reading assignments TBD; begin assignment 4
- Week 12: **Thanksgiving**
No reading assignment

- Week 13: Scale
Read pages 2-14, 165-169, and 253-264.
- Week 14: Semester review; looking ahead to 261
Review the reading assignments from weeks 1-13.
- Week 15: **Final exam**

Entrance and Exit Tickets

Towards the end of every class, I'll distribute *exit tickets*: slips of paper with two short-response questions. You'll have 5-10 minutes to answer the two questions, and then hand in the ticket on your way out the door.

As you walk out the door, be sure to take an *entrance ticket*. This is likewise a slip of paper with two short-response questions for you to take home and answer as you read the week's reading assignment. You'll hand in the entrance ticket as you walk in the door the next class session.

Participation

You will be expected to actively participate in this class. I'm not just going to stand in front of the room and talk for an hour and a half, as much as I enjoy doing that. There will be questions, discussion topics and in-class activities.

You should be prepared to be called upon in class. Toward that end, always come to class prepared with at least one QQC: a question, quote from, or comment on the week's reading assignment. That way, if I call on you, you'll have something ready to talk about.

Projects

There will be four projects this semester. The first project will be delivered on Windows; the second will be primarily Linux with bonus points for cross-compiling to Windows; and all subsequent projects will be on Linux only.

The assignment for each project will provide a specification for what the program should do and what its output should be. *Please read and follow this specification very carefully.* When implementing a network protocol, getting even a single byte wrong can be fatal to your program--and very hard to track down! In particular, some of the testing on assignments is automated, and any deviation from the spec will break the test and cost you points... potentially a *lot* of points.

For every assignment, once you get your grade you'll have one opportunity to fix errors and resubmit. However, this resubmission comes with a 10-point penalty, so your best bet is to check your work carefully to get the submission right the first try.

Each assignment will have an upload box in Moodle, with a due date marked. Assignments can be turned in up to 12 hours past the due date for no penalty. Any unexcused lateness beyond that point is treated as a resubmission. No assignment will ever be accepted more than one week late without special pleading. If you are unable to

meet a deadline due to extenuating circumstances--illness, personal emergency or disaster--email me immediately and we'll discuss an extension.

Grading Policy

Projects	40%
Midterm	12%
Pre-Midterm Participation	5%
Post-Midterm Participation	5%
Final	24%
Entrance/Exit Tickets	14%

- *Projects*: Each of the four solo projects represents 10% of your grade.
- *Midterm*: This will be an in-class open-note test during week 7, worth 12% of your grade.
- *Pre-Midterm Participation*: At the midterm break, I'll give feedback on my view of your class participation to date. This score includes both in-class participation and your activity on the class's Moodle forums. This will be on a scale of 0-5 points on your final grade. You should *not* expect to get a 5 by default; a 3 reflects perfectly acceptable and expected participation.
- *Post-Midterm Participation*: At the end of the year, you'll receive feedback on your participation for weeks 8 through 14, on the same 0-5 scale.
- *Final*: This will be another open-note test during finals week, worth 24% of your grade.
- *Entrance/Exit Tickets*: For every week in which you turn in the full set of tickets for that week's class sessions, you'll receive 1 point on your final grade.

Scale:

>=93	A
90-92	A-
87-89	B+
83-86	B
80-82	B-
77-79	C+
73-76	C
70-72	C-
60-69	D
<60	F

Standards:

A A-level work is **above professional standards**: All requirements are met, no errors are found, the code is clear and expressive, and the project is very easily deployed. This is code that you'd be proud to contribute to an open-source project.

B B-level work is *professional-grade*: All high- and medium-priority requirements are met, errors are minor and rare, the code is maintainable, and the project deploys correctly according to instructions provided in its README file.

C C-level work is *adequate*: All high-priority requirements are met, no catastrophic errors are present, the code is readable, and the project can be built, deployed and tested with minimal human intervention.

D D-level work is *inadequate*: A high-priority requirement is not met or has fatal errors, the code is hard to understand, or the project requires significant human intervention to build, deploy and test.

F F-level work is *unacceptable*: More than one high-priority requirement is not met or has fatal errors, the code is wholly opaque, or the project cannot be built, deployed or tested at all.

Attendance Policy

The textbook is only part of the material. The lectures build off the textbook, and the notes in the slide decks only partially capture the lecture material. The discussion activity in class is an equally important part of learning, and there's no way to recapture that if you miss it. It's therefore very important that you make it to class every session. At the same time, I understand that reality sometimes gets in the way. If you cannot make it to class, contact me by email and we'll discuss how to make up the lecture material and the entrance/exit tickets.

Missing class is one thing. Missing class without any sort of excuse, and without even contacting me, is another thing. I've found that these sorts of absences are a major warning sign of students who go on to fail the class outright, *and I do not want that to happen*. If by the midterm break you have accumulated two or more unexcused absences, expect me to contact you to discuss whether it makes sense for you to remain in the class or withdraw before the midterm deadline. Every unexcused absence past the second one will cost you 10 points off your final grade.

Academic Integrity Policy

Cheating, or academic dishonesty in any form, will not be tolerated in this course. Penalties for cheating may include receiving a zero on an assignment, a failing grade in the course, or even expulsion from DigiPen. For further details, please consult the DigiPen Academic Integrity Policy.

Disabled Student Services

If students have disabilities and will need formal accommodations in order to fully participate or effectively demonstrate learning in this class, they should contact the Disability Support Services Office at (425)629-5015 or [dss\[at\]digipen\[dot\]edu](mailto:dss[at]digipen[dot]edu). The DSS Office welcomes the opportunity to meet with students to discuss how the accommodations will be implemented. Also, if you may need assistance in the event of an evacuation, please let the instructor know.