CS 213: Introduction to Computer Systems

Spring 2001

1. Organization

Instructors:

Guy E. Blelloch	Bruce Maggs
WeH 7125	WeH 4123
x8-6245	x8-7654
blelloch@cs.cmu.edu	bmm@cs.cmu.edu
Mon 2:30-3:30pm	Mon 3:30-4:30pm

TAs:

Umut Acar	Shaheen Gandhi	Urs Hengartner	David Koes
WeH 4130	WeH 3108	WeH 4103	WeH 3108
x8-3053	x2-3001	x8-7571	422-8325
umut@cs.cmu.edu	sgandhi+@andrew	uhengart@cs.cmu.edu	dkoes@andrew
Tue 3-4pm	Wed 1:30-2:30pm	Tue 10:30-11:30am	Tue 7-8pm

Class Secretary:
Keith Ledonne
WeH 7116
x8-7660
ledonne@cs.cmu.edu

Lecture:
Tue Thu 9:00-10:20, Wean Hall 7500

Recitations:

A	Mon	10:30–11:20	OSC (Old Student Center) 203	Urs
B	Mon	11:30–12:20	OSC (Old Student Center) 203	Urs/Umut
C	Mon	12:30–1:20	OSC (Old Student Center) 203	Umut
D	Mon	1:30–2:20	OSC (Old Student Center) 203	Shaheen
E	Mon	2:30–3:20	OSC (Old Student Center) 203	David

Web page: `www.cs.cmu.edu/afs/cs/academic/class/15213-s01/www/`
Newsgroup: `cmu.cs.class.cs213`

2. Objectives

Our aim in CS 213 is to help you become a better programmer by teaching you the basic concepts underlying all computer systems. We want you to learn what really happens when your programs run, so that when things go wrong (as they always do) you will have the intellectual tools to solve the problem.

Why do you need to understand computer systems if you do all of your programming in high level languages? In most of computer science, we're pushed to make abstractions and stay within their frameworks. But, any abstraction ignores effects that can become critical. As an analogy, Newtonian mechanics ignores relativistic effects. The Newtonian abstraction is completely appropriate for bodies moving at less than 0.1c, but higher speeds require working at a greater level of detail.

Our 21X sequence works as follows: 211 is based on a simplified model of program execution. 212 builds further layers of abstraction. 213 introduces greater detail about system behavior and operation. This greater detail is needed for optimizing program performance, for working within the finite memory and word size constraints of computers, and for systems-level programming.

The following "realities" are some of the major areas where the abstractions we teach in 211/212 break down:

1. *Int's are not integers, Float's are not reals.* Our finite representations of numbers have significant limitations, and because of these limitations we sometimes have to think in terms of bit-level representations.
2. *You've got to know assembly language.* Even if you never write programs in assembly, The behavior of a program cannot be understood sometimes purely based on the abstraction of a high-level language. Further, understanding the effects of bugs requires familiarity with the machine-level model.
3. *Memory matters.* Computer memory is not unbounded. It must be allocated and managed. Memory referencing errors are especially pernicious. An erroneous updating of one object can cause a change in some logically unrelated object. Also, the combination of caching and virtual memory provides the functionality of a uniform unbounded address space, but not the performance.
4. *There is more to performance than asymptotic complexity.* Constant factors also matter. There are systematic ways to evaluate and improve program performance
5. *Computers do more than execute instructions.* They also need to get data in and out and they interact with other systems over networks.

By the end of the course you will understand these "realities" in some detail. As a result, you will be prepared to take any of the upper level systems classes at Carnegie Mellon. Even more important, you will have learned skills and knowledge that will help you throughout your career.

Good luck!

3. Textbook

Professor's Randy Bryant and David O'Hallaron are in the process of writing a textbook for this course. We will be using a very early draft of this book, which will be available at the bookstore hopefully by the end of the first week of classes. In addition to this draft book, we will make all the overheads from class available online.

We will also use the following book as a reference text on C.

Samuel P. Harbison and Guy L. Steele Jr., *C: A Reference Manual, Fourth Edition*, Prentice Hall, 1995.

If you have some other book on C, it will probably suffice, but be warned that the Harbison and Steele book has much more precise definitions than many other books.

4. Course Organization

Your participation in the course will involve five forms of activity:

1. Attending the lectures.
2. Participating in the recitations.
3. Homework and laboratory assignments.
4. Reading the text and supplementary handouts.
5. Exams

Attendance will not be taken at the lectures or recitation sections. You will be considered responsible for all material presented at the lectures and recitations. Lectures will cover higher-level concepts. Recitations will be more applied, covering important "how-to's", especially in using tools that will help you do the labs. In addition, the recitations will help clarify lecture topics and describe exam coverage.

There will be two types of assignments in this class. *Lab* assignments will be multi-week efforts (in groups of up to 2) providing in-depth understanding of some aspect of computer systems. Labs will involve some combination of C and assembly programming, and some will involve performance measurements. Labs will be handed out in class on Thursday and due a couple of weeks later on a Wednesday.

Homework assignments will be 1-week efforts (that you work on individually) that involve solving a series of smaller problems. The solution to many of these will involve writing small C programs. Homeworks are designed to be drills to help you get practice for exam problems. Homeworks will be handed out at class on Tuesday and due the following Monday.

5. Getting help

For urgent communication with the teaching staff, it is best to send electronic mail (preferred) or to phone.

If you want to talk to a staff member in person, remember that our posted office hours are merely times when we guarantee that we will be in our offices. You are always welcome to visit us outside of office hours if you need help or want to talk about the course. However, we ask that you follow a few simple guidelines:

- Prof. Blelloch and Prof. Maggs normally work with their office doors open and welcome visits from students whenever their doors are open. However, if their doors are closed, they are busy with a meeting or a phone call and should not be disturbed.
- The TAs share offices with other students. To avoid disturbing these students, please send mail or zephyr before visiting a TA outside of office hours so they can arrange to meet you.

We will use the Web as the central repository for all information about the class. The class home page is at

`www.cs.cmu.edu/afs/cs/academic/class/15213-s01/www/`

Using the Web, you can:

- Obtain copies of any handouts or assignments. This is especially useful if you miss class or you lose your copy.
- Read clarifications and changes made to any assignments, schedules, or policies.
- Find links to any electronic data you need for your assignments

We have also set up a news group for this class, `cmu.cs.class.cs213`. This group will be used by members of the teaching staff to post announcements and clarifications. You may also post to this group to make queries.

6. Policies

Working in Groups

For homework assignments, you will work by yourself. For lab assignments you may work in groups of up to 2 people. It is up to you to form and regulate your own groups. If you are not happy with your partner, you are free to find another partner. You may also work by yourself.

Handing in Assignments

All assignments are due at 11:59pm (one minute before midnight) on the due date specified on the assignment (typically a Wednesday). All handins are electronic, usually consisting of one or more files that are to be copied to a specified directory. The writeup for each assignment will provide details about the handin procedure for that assignment.

Penalties for Late Assignments

Late assignments will be docked 20% each day for the first two days. Assignments more than 2 days late will not be accepted, unless you have arranged for an extension *in advance* with Profs. Blelloch or Maggs. For example, suppose an assignment is due at 11:59pm on Wed. If you hand it in between midnight and 11:59pm Thursday, you will be docked 20%. If you turn it in between midnight and 11:59pm Friday, you will be docked 40%. You won't be able to turn it in at all after 11:59pm Friday.

Making up Exams and Assignments

Missed exams and assignments more than 2 days late can be made up, but only if you make prior arrangements with Profs. Blelloch or Maggs. However you should have a good reason for doing so. It is your responsibility to get your assignments done on time. Be sure to work far enough in advance to avoid unexpected problems, such as illness, unreliable or overloaded computer systems, etc.

Appealing Grades

After each exam, homework, and assignment is graded, we will send each of you a personalized email with your grade. You have seven calendar days from the date we send the email to appeal your grade. If you have questions about the grade you received on an assignment (homework or lab), please talk first to the person in charge of the assignment, who will be clearly identified in the writeup. If you are still not satisfied, please come and visit one of the Professors. If you have questions about an exam grade, please visit Prof. Blelloch directly.

Final Grade Assignment

Each student will receive a numeric score for the course, based on a weighted average of the following:

- **Assignments:** The assignments will count a combined total of 50% of your score. The exact weighting of the different assignments will be determined near the end of the course based on our perception of the relative effort required. In any case, each homework will count 2%, while each lab will count 8–12% of your score. Since small differences in scores can make the difference between two letter grades, you'll want to make a serious effort on each assignment.
- **Exams:** There will be two in-class exams, each counting 12.5%, plus a final counting 25%.

Grades for the course will be determined by a curve. The total score will be plotted as a histogram, and then approximate cutoff points for the different letter grades will be determined. Individual cases, especially those near the cutoff points may be adjusted upward or downward based on factors such as attendance, class participation, improvement throughout the course, final exam performance, and special circumstances.

Cheating

You must do your homework assignments by yourself. Your lab assignments allow collaboration, but only with the other member of your project group. Each lab assignment must be the sole work of the group turning it in. Assignments will be closely monitored by automatic cheat checkers, and students may be asked to explain any suspicious similarities. The following are guidelines on what collaboration outside of your group is authorized and what is not:

What is Cheating?

- *Sharing code or other electronic files*: either by copying, retyping, looking at, or supplying a copy of a file.
- *Sharing written assignments*: Looking at, copying, or supplying a assignment.

What is NOT Cheating?

- Clarifying ambiguities or vague points in class handouts or textbooks.
- Helping others use the computer systems, networks, compilers, debuggers, profilers, or other system facilities.

Be sure to store your work in protected directories.

The usual penalty for cheating is to be removed from the course with a failing grade. We also place a record of the incident in the student's permanent record.

7. Facilities: Intel Computer Systems Cluster

Intel has generously donated a cluster of 25 Linux-based Pentium III servers, specifically for CS 213, that we will use for all labs and assignments. The class Web page has details.

8. Class Schedule

The tentative schedule for the class is given below. The notation "Hi" indicates a homework assignment, while "Li" indicates a lab. Any changes will be announced on the class news group. An updated schedule will be maintained on the class web page. Readings are from Bryant and O'Hallaron (BO) and Harbison and Steele (HS). You will also be responsible for the material covered in lecture. The due dates given in the schedule actually refer to the night before (e.g. Lab 1 is due on Wednesday Feb. 2 at 11:59pm).

The schedule will be updated during the semester, so please use the web pages for the latest version.

Class	Date	Day	Topic	Reading	Assignments (estimate)	Who
1	1/16	Tu	Overview			Both
2	1/18	Th	Bits and bit operation	BO 1.1, HS 6.1–6.2, 7.6	L1 Out	GB
3	1/23	Tu	Integer Representation	BO 1.2, HS 5.1		GB
4	1/25	Th	Integer Arithmetic	BO 1.3		GB
5	1/30	Tu	Machine Model	BO 2.1–2.4	H1a Out	BM
6	2/1	Th	Control Flow	BO 2.5	L1 Due	BM
7	2/6	Tu	Procedures	BO 2.6	H1a Due, H1b Out	BM
8	2/8	Th	Structured Data	BO 2.7, HS 5.3–5.4	L2 Out	BM
9	2/13	Tu	Heterogeneous Data	BO 2.8–2.9, HS 5.6–5.7	H1b Due, H2 Out	BM
10	2/15	Th	Floating Point	BO 1.4, 2.10, HS 5.2		BM
11	2/20	Tu	Linking	BO 5	H2 Due	BM
12	2/22	Th	Exceptional Control Flow	BO 6	L2 Due	BM
13	2/27	Tu	*Exam #1*			
14	3/1	Th	Memory Management I	BO 7.9, HS 16	L3 Out	GB
15	3/6	Tu	Memory Management II	BO 7.10		GB
	3/8	Th	*Mid Semester Break*			
16	3/13	Tu	Memory Referencing Errors	BO 7.11		GB
17	3/15	Th	Memory Technology	BO 4.1-4.3		GB
18	3/20	Tu	Caches	BO 4.4–4.5	H3 Out	GB
19	3/22	Th	Cache Performance	BO 4.6	L3 Due, L4 Out	GB
	3/27	Tu	*Spring Break*			
	3/29	Th				
20	4/3	Tu	Virtual Memory	BO 7.1–7.6	H3 Due, H4 Out	GB
21	4/5	Th	Memory System	BO 7.7–7.8		GB
22	4/10	Tu	Threads	BO 8.1–8.3	H4 Due	GB
23	4/12	Th	Synchronization	BO 8.4–8.6	L4 Due	GB
24	4/17	Tu	*Exam #2*			BM
25	4/19	Th	Network Programming	BO 9.?	L5 Out	
26	4/24	Tu	WWW Computing I	BO 9.?	H5 Out	BM
27	4/26	Th	WWW Computing II	BO 9.?		BM
28	5/1	Tu	Internet Protocol	BO 9.?	H5 Due	BM
29	5/3	Th	Network Programming	BO 9.?	L5 Due	BM