Syllabus

CS2110 Computer Organization and Programming
Summer 2018

Introduction

This is not a class where we are going to "learn you" about how computers work and how to program in C. This is a class where we are going to teach and you are going to do the learning. What does this mean? It means that you must become highly engaged in active learning. Read the textbooks (more than once is allowed), search the web, and above all CODE. After every lecture and lab go home and try stuff out. Verify that what we told you really works. Test your knowledge by extending what we tell you. (i.e., *if I understand this then I should be able to make the following happen*.) Imagine trying to learn to play tennis by listening to someone describe the theory and then watching another person play. The absolute key to success in this course is to code constantly. Good luck.

Instructors

Dan Forsyth – Section A
Office: CCB 242 (use back stairs or elevator only); text 678 992 9248 in urgent situations.
Hours: T Th 10-11:30AM, and by appointment
Email: dan.forsyth@cc.gatech.edu

Caleb Southern – Section B
Office: CCB 105 (inside CCB 104B)
Office Hours: W Th 10-11:30AM, and by appointment
Email: caleb.southern@gatech.edu (include 2110 in the subject line)
:

Teaching Assistants

(See the course website for contact details and office hour schedule)

Austin Adams – Head TA
Cem Gokmen – Senior TA
Madeleine Brickell
Daniel Becker
James "Jim" Harris
Preston Olds
Joshua Viszlai
Madison Grams
Vivian De Sa Thiebaut
Sam Gilson
Hanwen "Michael" Xu
Jered Tupik
Minsung "Daniel" Jung
Clifford "Cliff" Panos

Lauren Chen
Shannon Ke
Bharat Srirangam
Jarrett Schultz

Purpose

1. To understand the structure and operation of a modern computer from the ground up.
2. Understand basic hardware concepts: digital circuits, gates, bits, bytes, number representation
3. Understand the Von Neumann model and the structure and operation of a basic data path
4. Understand the structure and function of machine language instructions
5. Understand the structure and function of a symbolic assembly language
6. Understand basic concepts of computer systems such as the runtime stack, simple I/O devices
7. Introduce the C language with particular emphasis on the underlying assembly and machine language as well as interaction with hardware.

Outcomes

- (Competency Knowledge) Be able to identify and/or construct basic digital structures such as MOS FET logic gates, decoders, multiplexors, adders, memory.
- (Competency Application) Be able to construct a state machine diagram and then implement it as a finite state machine circuit.
- (Competency Comprehension) Understand data representation. Be able to convert numbers between various representations: Binary, octal, decimal, hexadecimal, and IEEE Floating Point.
- (Competency Knowledge) Be able to identify the component parts of the Von Neumann Model of computer and be able to explain the purpose of each component.
- (Competency Synthesis) Be able to write, debug and run assembly language programs including recursive subroutines, traps, basic input/output.
- (Accomplishment Synthesis) Be able to write, debug and run multi-file C programs several hundred lines long using "make" to compile and execute said programs.
- (Competency Synthesis) Be able to utilize (in C programs) proper typing and casting constructs, structs, pointers and arrays, functions, function pointers, dynamic memory allocation and variables of different storage classes (auto, static, volatile, etc.)

Course Topics

1. Course introduction and overview
2. Bits, Data Types, and Operations
3. Digital Logic Structures
4. The Von Neumann Model
5. Introduction to a simple microprocessor, the LC-3
6. Programming
7. Assembly Language
8. I/O

9. TRAP Routines and Subroutines
10. Stacks, programming examples.
11. Introduction to Programming in C
12. Variables and Operators
13. Control Structures
14. Functions
15. Debugging
16. Recursion
17. Pointers and Arrays
18. I/O in C
19. Data Structures in C

Textbooks

Required

Introduction to Computing Systems, Second Edition

Yale N. Patt, Sanjay J. Patel
McGraw-Hill ISBN: 0073890480
Book Website: http://www.mhhe.com/patt2

The C Programming Language

Brian Kernighan, Dennis Ritchie
Prentice Hall: ISBN 0131103628

Recommended

This is a recommended book if you are new to Linux. Buy it if you are the type of person that likes to have a reference book when learning something new, but know that this content is available from various places on the internet. There will be no required readings or assignments from this book.

Mastering Linux, Second Edition

Paul S. Wang
Chapman & Hall/CRC
Taylor & Francis Group
6000 Broken Sound Parkway NW, Suite 300
Baton Raton, FL 33487-2742
ISBN-13: 978-1-4398-0686-9

References

These internet references will be discussed during the semester; if you want to know more, please explore them.

Lions Commentary:

http://www.lemis.com/grog/Documentation/Lions/book.pdf
https://minnie.tuhs.org//Archive/Distributions/Research/Dennis_v6/

UNIX Seventh Edition User Manual

(Recommendations: For C, ch 14, 16, 17, 33, 19, 20;
for Unix/Linux command line, ch. 3, 6, 4, 5, 24, 25;
for UNIX API, ch 2, 17, 31, 32, 37)

https://9p.io/7thEdMan/v7vol2a.pdf
https://9p.io/7thEdMan/v7vol2b.pdf

International Obfuscated C Code Contest

https://www.ioccc.org

Course Structure

- The course will consist of lecture and a mix of labs and recitations we call "lecitations."
- There will be lots of homework assignments, each usually being seven (7) days in length.
- Homework assignments will be mostly based on material already covered in lecture.
- One of the purposes of the lab is to give students help with homework assignments.
- Labs will be a mix of instructional labs (e.g. might have a lab on "makefiles") and assessments (i.e., tests: *did students learn what they were supposed to from the homework?*)
- Homework can be, at the student's discretion, collaborative. Nevertheless, each student is required to turn in an assignment that he or she produced, that is not substantially identical to the collaborators', and that he or she understands fully.
- In the schedule, there are reading assignments and suggested exercises that are to be completed **before** the lecture on which they are listed. There is no explicit credit for completing these exercises.
- There will be some quizzes in lecitation. Some will be timed coding exercises; some will be paper and pencil quizzes. Or they might be online.
- There will be a final examination

Patt Book Table of Contents

1. Welcome Aboard
2. Bits, Data Types, and Operations
3. Digital Logic Structures
4. The Von Neumann Model
5. The LC-3
6. Programming
7. Assembly Language

8. I/O
9. TRAP Routines and Subroutines
10. And, Finally...The Stack
11. Introduction to Programming in C
12. Variables and Operators
13. Control Structures
14. Functions
15. Testing and Debugging
16. Pointers and Arrays
17. Recursion
18. I/O in C
19. Data Structures

Appendix A. The LC-3 ISA
Appendix B. From LC-3 to x86
Appendix C. The Microarchitecture of the LC-3
Appendix D. The C Programming Language
Appendix E. Useful Tables

Course Evaluation

The number of and value of each assignment type is show below (subject to change):

Item	Number (approx.)	Totals
Timed Labs	6	24%
Homework	11	22%
Labs	12	5%
Quizzes	6	24%
Final	1	25%
TOTAL		100%

Schedule

The initial course schedule is published to give students a view of the plan for the semester. As the semester progresses this schedule may change. The schedule document will not be updated. Rather students should consult the course website to see the official due date for assignments.

Final Exam Procedures

The last two weeks of class are divided among Final Instruction Days, Reading Periods and Final Exam Days as shown below.

12/3	Mon	16	Final Instructional Days (No quizzes, tests, exams)	29	Review / Discussion of course mechanics	HW 11 due
12/4	Tue					
12/5	Wed		Reading Period			
12/6	Thu		Half Day Reading Period (until 2:20pm)		Section B Final 2:40-5:30	
12/7	Fri		Final Exams			
12/10	Mon	17	Final Exams		Section A Final 11:20-2:10	
12/11	Tue		Half Day Reading Period (until 2:20pm)			
12/12	Wed		Final Exams			
12/13	Thu		Final Exams			

Academic Misconduct

- Academic misconduct is taken very seriously in this class and it greatly upsets your instructors. While you may think it is "just a homework", academic misconduct in the professional world can end a career instantly. Let's just not encourage such a habit.
- If you get behind in your assignments, turn in what you've got and talk with your instructor, not necessarily in that order. You'll ultimately come out better than if you turn in someone else's work.
- Quizzes, timed labs and the final examination are individual work. Period.
- Homework assignments are collaborative, In addition many if not all homework assignments will be evaluated via demo or code review. During this evaluation, you will be expected to be able to explain every aspect of your submission. Homework assignments will also be examined using electronic computer programs to find evidence of unauthorized collaboration.
- Studying is collaborative. You are encouraged to study old tests, do suggested problems, etc. together. Explaining an answer is one of the best ways to learn it.
- What is unauthorized collaboration? You must code each individual programming assignment by yourself. You may work with others, but each student should be turning in their own version of the assignment. Submissions that are essentially identical will receive a zero and will be sent to the Dean of Students' Office of Academic Integrity. Submissions that are copies that have been superficially modified to conceal that they are copies are also considered unauthorized collaboration.
- You are expressly forbidden to supply a copy of your homework to another student via electronic means. If you supply an electronic copy of your homework to another student and they are charged with copying you will also be charged. This includes storing your code on any site which would allow other parties to obtain your code including (but not limited to) public repositories, etc.

Rules and Regulations

1. **You are responsible for turning in assignments on time. This includes allowing for unforeseen circumstances. You are also responsible for ensuring that what you turned in is what you meant to turn in. Each assignment will have an official due date and time and a 6-hour grace period. If an assignment is submitted during the grace period it will receive a 25% penalty. After the grace period no credit will be given. It is your responsibility to plan and insure that you have backups, early safety submissions, etc.**

2. In general, programming assignments should be turned in with a Makefile and all files needed to compile and run the program. The TA grading your submission should be able to make and run your program without adding files, repairing things etc.

3 Quizzes and examinations must be taken at the scheduled date and time. Please do not ask for special treatment because you (or your parents) have purchased non-refundable airline tickets. The safe time to travel is at the end of or after finals week**.**

4. If you need a certain grade in order to stay in school, maintain a scholarship, etc., the time to worry about this is right from the beginning of the course not during the week before finals. Grades are based on demonstrated performance not individual need-based on factors external to the course. Please do not request special consideration based on this type of situation. There is no "extra credit" given in this course.

5. Final grades will be available from OSCAR normally sometime the week after finals. Once you have taken the final exam you should direct any and all questions to your instructor, not your TA's. You may review your final and discuss your grades during the **following semester** in which you are attending Ga Tech. Grades will not be discussed during the break except in emergency cases.

6. If you have any major personal problems (family/illness/etc.), please go to the Dean of Student's office located in the Student Services Building (Flag Building) next to the Student Center. The Dean and staff are equipped and authorized to verify the problems, and they will issue a note to all your instructors making them aware of the problem and requesting whatever extensions, etc. that may be necessary. If you have issues of any kind that affect your performance in the course: Personal problems, illness, accidents, etc., please let your course instructor know .

7. The official announcements and any email from the class should be checked and read every day. Our official course site is at http://canvas.gatech.edu. Make sure you log in at least once to check that your Notification preferences for Announcements do not delay them. Announcements about course matters will be posted to Canvas. Piazza is for posting technical questions about assignments, tests, etc. Complaints, questions about your personal problems, etc. should be discussed with your instructor in person or via email.

8. Out of consideration to your fellow students, please turn off cell phones, beepers, wristwatch alarms, etc. Also, make every effort to be on time for class. If you are unavoidably late, please sit near the door and try to avoid as much disruption to the class as possible.

9. If you are graduating and need this course to do so, please inform your instructor as soon as possible.

10. Complaints about TA's should be directed to the course instructors during office hours or via email as soon as possible.

11. The deadline for re-grades is 2 weeks after an assignment grade is posted or returned to you. This deadline also applies to picking up items that are returned in class. After this deadline no grade changes will be made and tests not picked up will be destroyed.

12. Students participating in approved university activities MAY be given extensions, allowed to take tests at different times, etc. It is important to request such items well in advance and to supply supporting documentation.

13. Please remember that your TA's are also students. They have very full schedules and they are TAing out of a genuine desire to help you learn the material.

14. In order to help you in your programming assignments, we may provide you with "autograder" test cases that will exercise your code. It is certain that these test cases will not test every feature in your code, so thorough testing is still **your** responsibility. In addition, there may be unintentional errors in these test cases that cause the autograder to give null or wrong answers. The programming assignment is always the authoritative document that describes the correct results. Please plan accordingly.

15. You are responsible for backing up your computer. We highly recommend some sort of automatic off-site backup. If you have a catastrophic computer failure we will work with you while you get your hardware fixed but we cannot accept having no backup as a reason for an extension.

16. Note that in unusual circumstances, grades may need to be changed to correct grading errors after they are released. You will be notified if changes occur and will have the opportunity to request reconsideration.

Demos

1. You can sign up with any TA for any time slot, but must do so at least 24 hours in advance. Sign up for whatever's convenient for you!
2. If you miss your demo without a valid excuse, you will get a zero and you may not sign up for another demo slot. If something comes up after your demo is locked in that might make you miss your demo, tell us immediately. We can work with you or possibly allow you to change your demo time if you let us know BEFORE your demo.
3. Your demo time is locked in 24 hours before your demo. You cannot change your demo time after that time and we assume that you will be there. See the previous item. If you try to change your demo time without talking to us, we will know it and you will get a zero.
4. **Each student will be allowed to re-demo one assignment per semester with a 50% penalty**

Lecitations

The term "Lecitation" is derived from the words "lab" and "recitation." While lectures are held on Tuesday and Thursday, lecitations are held during the blocks on Monday and Wednesday. Participation and attendance to lecitation are required. All students have signed up for those blocks, so there should not be any time conflicts with other courses. Attendance will be checked periodically.

Lecitations are designed as follows:

- Attendance is taken at Lecitations; more than 3 officially unexcused absences will result in a deduction of 10% of the lab grade per additional absence.
- The goal of lecitation is not for TAs to stand in front of the room and lecture on the same material that was already covered in class.
- Lecitation is designed to have students actively working on assignments that deal with the material that was previously taught.
- The TAs cover any overarching announcements for the course and then let the students work on their assignment for the day.
- Students can still come to the TAs with any questions about the course or the assignments.
- Depending on the day, students can expect to either work on a regular lab, take a quiz, or take a Timed Lab.
- On regular days, a “lab” is given to the students to be completed.
    - This is an assignment that reinforces the concepts that were taught in lecture.
    - Generally, these assignments also deal with similar concepts that the current homework covers.
    - Students are encouraged to talk and to help each other as they work through the lab.
    - The TAs will address any questions on individual basis, or to the class as a whole if those questions persist.
    - The students are required to submit their lab assignments by midnight of the day on which they are assigned.
- On quiz days, a paper quiz will be given to the students for a specified time.
    - Collaboration on the quizzes is not allowed.
    - Use of any outside resources is prohibited.
    - The quizzes usually take a small fraction of the lab time.
    - After all the quizzes are collected, there will generally be a regular lab assignment to be completed.
- On Timed Lab days, a “Timed Lab” assignment will be given to the students.
    - A Timed Lab resembles a regular lab or a homework problem.
    - Collaboration on the Timed Labs is not allowed.
    - Students may use any previous labs or homework assignment that they have worked on as reference for the Timed Lab. No other material may be used
    - Students have the whole lecitation period to work on the Timed Lab, and they must submit it before leaving.

Requirements for C Programs

Your programs will be tested and graded on Ubuntu 16.04 LTS 64-bit. If you are using something else and it works on your computer but not on ours, you risk getting a zero. So our advice is to run one of

these two (even if you have a Mac). You may use Virtual Box, the VMWare Player or VMWare Fusion. You will be supplied information about setup and installation.

All programming assignments must:

- build cleanly (i.e. no warnings or errors) on a linux box with:

```
gcc -std=c99 -Wall -pedantic -Wextra \
      -Werror -O2 -Wstrict-prototypes -Wold-style-definition
```

- exit gracefully (no segfaults, bus errors, etc) if appropriate
- produce useful output where applicable (i.e. error messages)
- not leak memory if applicable (we use valgrind to check this, you should too!)
- Not produce spurious output (no debugging output, extra messages to the user, etc. - this breaks the grader and gives you a zero)

Capital crimes (== automatic 0):

- non-building (how can we grade it if it doesn't build?)
- core dumps, including segmentation faults (or any other ungraceful exit)
- infinite loops (makes grading quite difficult, don't do it!)
- spurious output (see note above)