

Course Overview

Policies, Grading & Outline

CIT 593
Fall 2006

1

Resources

- References (on reserve in Library):
 - Structured Computer Organization – 5th Edition by Andrew S. Tanenbaum
 - Principle of Computer Architecture by Miles J. Murdocca and Vincent P. Heuring
 - Computer Organization and Architecture by Linda Null and Julia Lobur
- C references
 - The C Programming language (2nd Edition, Paperback) by Brian W. Kernighan, Dennis Ritchie
 - C Language Tutorial (http://www.physics.drexel.edu/courses/Comp_Phys/General/C_basics/c_tutorial.html)
 - Programming in C:UNIX System Calls and Subroutines using C (<http://www.cs.cf.ac.uk/Dave/C/>)

4

Instructor & Teaching Assistant

- Diana Palsetia
(palsetia@cis.upenn.edu)
 - location: Moore 174, phone: 215-898-0382
 - office hours: MW 2:30 - 4pm
 - course web: <http://www.cis.upenn.edu/~palsetia>
 - Black Board:
<https://courseweb.library.upenn.edu/>
- Arthit Hongchintakul
(arthith@seas.upenn.edu)
 - office location and hours: TBA

2

Grading

- Course grades will be curved
 - 90% or above is not necessarily an A
- Assignments (55% of grade)
 - Reading and Answering (preferably typed)
 - Programming(C and assembly)
- Quizzes (10 % of grade)
 - this is to ensure you keep pace with the class
 - includes anything covered so far
 - cover assigned readings not taught in class
- Two exams (35% of grade)
 - Midterm (15%) & Final (20%)
 - exams will be closed book

5

Textbooks

Required
Introduction to Computing Systems from bits & gates to C & beyond
Author: Yale N. Patt and Sanjay J. Patel
ISBN: 0072467509 (2nd Ed)

C Language (online)
A Concise Introduction to C

LC3 Assembly Language(online)
LC3 Simulator Manual
Using LC3 Simulator

3

Dates

- Midterm exam: 19 or 24 Oct (class vote)
- Final exam: Dec 19th (tentative)
- Assignments (regularly check website)
 - Programming/Homework: As they are posted
 - Quizzes will be announced a week in advanced

6

Assignments

- if late, 10% per day for five days; then no credit
 - weekend days count as late days
 - system delays are no excuse for lateness
 - disk failures not an excuse – back up!
- program grading is generally:
 - good documentation 25%
 - sound structure 25%
 - correct output or results 50%

7

How much time to put into the course ?

- On average, at least **10 hours** per week, assuming:
 - you take 3-4 courses
 - you are a full-time student
 - your studies occupy a 40-hour work week
- 3.5 hours in class + office hours
- 3.5 hours on projects and practice
- 3 hours on readings
 - ~20-40 pages per week
 - possibly difficult readings for beginners

10

Programming

- C programs - edit, compile and run on eniac.seas.upenn.edu
- For Assembly we will use LC3 Simulator
- Always due by midnight
- send an archive (.tar file) to BlackBoard's "Drop Box"
 - <https://courseweb.library.upenn.edu/>

8

Honesty

- you may:
 - discuss the assignments with one another
 - help others debug their work
- you may *not*:
 - work together on programs for this course
 - copy another's code, or allow your code to be copied
 - lend your code to someone else, or allow them to copy it
 - use any code from a book or the web without my permission
- penalty for first offense:
 - you will be reported to the Office of Student Conduct
 - you will receive an F in the course
- if you may have accidentally broken a rule, talk to me immediately

11

Computing Facilities

- Penn Eng.: <http://www.seas.upenn.edu/cets>
 - Information on accounts, labs, software
- If working from home
 - use secureCRT from PennNet CD-ROM
 - <http://www.seas.upenn.edu/cets/answers/remote.html>
 - PennNet CD-ROM is available in CETS Office (Moore 169)
 - you also need FileZilla for secure FTP
 - <http://www.seas.upenn.edu/cets/answers/filezilla.html>

9

Why take CIT 593 ?

- A software engineer or scientific programmer interested in high performance studies learns how to arrange/design programs to gain maximum performance from a given architecture.
- Your company assigns benchmarking tasks where you compare computer architectures for a purchase decision as a member of the IT computer purchasing team.
- Writing machine dependent software such as compilers, operating systems, and device drivers

What does CIT 593 offer ?

- Introduction to computer architecture
 - How is data represented inside a computer?
 - What are the pieces of a computer?
 - How do computers work?
- Programming
 - How do I "talk" directly to the machine?
 - How do I program in high-level language like C?
- Computer Systems and Computation
 - How do simple HW/SW elements come together to realize complex computations?

13

Why learn Assembly ?

- Computers can only understand binary representation (0 or 1) a.k.a machine language
- Assembly is the lowest form of human-readable notation for the machine language
- a.k.a low-level language

16

Big Picture

- Hardware
 - Representing data, transistors, gates, digital logic structures, von Neumann machine model
- Instruction Set Architecture (ISA)&Assembly language
 - Instructions, (structured) programming, input/output, *relationship to hardware*
- C programming
 - Syntax, operators, control structures, functions, *pointers*, recursion, data structures, *relationship to assembly language*

14

Why Learn C Language ?

- What is C?
 - High-level language
 - "Portable assembly language"
 - In between assembly and Java/VB/C#
- Very common
 - Operating systems and even general applications
 - Foundation for C++/C#/Java
 - Assembly-to-C migration for embedded applications

17

Why learn Hardware ?

- To build an effective system
 - It is the foundation
 - How do algorithms transform to electrons
- To know its limitations
 - What makes a computer system tick before you can attempt to optimize programs ?

15

Course Topics

- Bits and Bytes
 - How do we represent information using electrical signals?
- Assembly Language Programming
 - How do we use processor instructions to implement algorithms?
 - How do we write modular, reusable code? (subroutines)
- I/O, Traps, and Interrupts
 - How does processor communicate with outside world?
- C Programming
 - How do we write programs in C?
 - How do we implement high-level programming constructs?

Transition to cit 595

- Processor and Instruction Set
 - How do we build a processor out of logic elements?
 - What operations (instructions) will we implement?

Questions?



20