Computer Organization & Assembly Language Programming

CSE 2312
Lecture 1  Administration & Introduction

Junzhou Huang, Ph.D.
Department of Computer Science and Engineering
 Administration

• Course CSE2312
  – What: Computer Organization & Assembly Language Programming
  – When: Mon. & Wed. 3:00 ~ 4:20pm
  – Where: SH 332
  – Who: Junzhou Huang (Office ERB 650) jzhuang@uta.edu
  – Office Hour: Mon. & Wed. 1:00 ~ 3:00pm and/or appointments
  – Homepage: http://ranger.uta.edu/~huang/teaching/CSE2312/
    (You’re required to check this page regularly)

• Lecturer
  – PhD in CS from Rutgers, the State University of New Jersey
  – Research areas: medical imaging (hardware and software design), machine
    learning, computer vision, image and signal processing

• GTA
  – Yeqing Li (Office ERB 101), yeqing.li@mavs.uta.edu
  – Office hours: Mon. & Wed. 10:00am ~ 12:00pm and/or appointments
Study Materials

• Prerequisites
  – Intermediate Programming (CSE 1320)
  – Introduction to Computers & Programming (CSE 1310)
  – What this really means:
    ➢ You know at least one programming language.
    ➢ You know something about how to write/run/test programs.
    ➢ Elementary knowledge of math and algorithms

• Text book
  – ISBN-10: 0132916525
Study Materials

• **Text book**
  – We will not cover all the chapters of the book
  – We will not cover all sections of the covered chapters
  – We will not fully follow the order of the book
  – The contents uncovered in slides/lectures are optional

• **Recommended textbooks**

• **Acknowledgments**
  – Class notes partially based on 2312 classes taught at UTA in prior years
  – Material from textbook site
  – Lots of material available on the web (via google search, wikipedia)
Grading

• Distribution
  – 25% Homework Sets (Quizzes)
  – 20% Programming Assignments
  – 25% Midterm Exam
  – 25% Final Exam
  – 5% Class Participation

100%

• Attention
  – Homework is as important as any other aspects of your grade!
  – Attendance though not mandatory, but is HIGHLY encouraged.
  – The university makeup policy will be strictly adhered to. Generally, no make-up exams/quizzes except for university sanctioned reasons.
  – When missing an exam/quiz due to unavoidable circumstances, PLEASE notify the instructor and request a makeup approval ahead of time.
Final Grade

• Final Letter Grade
  – [90 100] --- A
  – [80 90) --- B
  – [70 80) --- C
  – [60 70) --- D
  – [00 60) --- F

• Attention
  – Final letter grades will be assigned based on absolute percentage
  – [ ] denotes inclusion and ( ) denotes exclusion.
  – The instructor reserves the right to move the thresholds down based on the distribution of final percentages, but they will not move up.
Assignments

• **Homework assignments**
  – Assigned in class, typically due one week later at the start of lecture
  – Automatic 20% deduction for each day late
  – Homework is not accepted more than 3 days late

• **Programming assignments**
  – Assigned in class, typically due 1~3 weeks later after assignments
  – They are not created equally.
  – Later programming assignment are generally harder and hence worth more.

• **Collaboration**
  – You may discuss assignments with others, but must write up them individually. Please identify collaborators on your assignment cover sheet
  – Failure to comply with this policy is a violation of academic integrity

• **Start early! Start early! Start early !!!**
Information

• Course Webpage
  – Check the web page regularly (2-3 times per week).
  – Announcements, assignments, and lecture notes will be posted there.

• Grade Appeal
  – You may appeal the grade in writing (email) within 5 class days.
  – Appealed to the appropriate GTA firstly, then to the instructor, if necessary.
  – Please refer to the UTA Catalog for the detailed guide of grade appeals.

• Drop Policy
  – The university withdrawal policy will be strictly adhered to.

• Others
  – Accommodating students with disabilities
  – Student Support Services
  – Etc.
Questions
Course Overview

• **What is it?**
  – Computer Organization
  – Assembly Language Programming

• **Why is a CS course?!?**
  – Hardware & Software

• **Will I really ever use this stuff again?**
  – Necessary knowledge for a CS student
  – You may not become a professional assembly programmer but you need to know how to do it

• **How to succeed in this course?**
Why Are You In This Class?

- **Something interesting about you**  
  - Why you picked your major?  
  - Life Plan

- **To learn background in order to take more advanced classes in computer science and engineering**  

- **Understand the effect of an architecture on the code you write**  
  - Learn how to directly control the computer

- **To have the necessary background to understand innovations in processor and compiler design or related others**  
  - Your desktop, laptop, ipad, iphone, etc.

- **Not necessary to become an assembly programmer**  
  - But to be able to debug  
  - To be able to understand innovative ideas
Why Assembly?

• Two main reasons
  – Accessibility to system hardware
  – Space and time efficiency

• Accessibility to system hardware
  – Assembly Language is useful for implementing system software
  – Also useful for small embedded system

• Space and Time efficiency
  – Understanding sources of program inefficiency
  – Tuning program performance
  – Writing compact code
iPhone Processor

620MHz ARM chip
Inside Blackberry Torch

Research firm iSuppli took apart the new Blackberry Torch 9800 to identify key suppliers and estimate the cost of components. Here’s some of the findings about the smartphone, which the firm said was assembled for RIM in Mexico and likely relies on plastic and stamped metal parts from China.

**Estimated component costs**

- Display & Touchscreen: $34.85
- Memory: $34.25
- Mechanical/Enclosures: $23.35
- Applications Processor: $15
- Radio/Wireless: $24.50
- User Interface: $12.40
- Battery/Power Management: $15.90
- Camera: $10.80
- Total materials: $171.05

Source: iSuppli; With reporting by Don Clark, Phil Dvorak and Jennifer Valentine-Dinh
Ipad A4 Processor

Max. CPU clock rate 1 GHz (iPad)
Instruction set ARM v7 32 bit RISC
Cores 1
L1 cache 64 KB
L2 cache 640 KB
Computer Organization

• **Architecture:**
  – Art or practice of designing

• **What’s Inside?**

• **How is it put together?**

• **How does it work?**
Main Component

• **CPU**
  – Executes Instructions

• **Memory**
  – Stores Programs and data

• **BUS**
  – Transfers data
  – Front Side Bus (FSB)

• **Storage**
  – Permanent

• **I/O devices**
  – Input: Keypad, Mouse, Touch
  – Output: Printer, Screen
  – Both (input and output), such as USB, Wifi, Touch screen, DISK
Von Neumann Architecture

- Model of a computer that used stores programs
  - Both Data and Program stored in memory
  - Allows the computer to be “Re-programmed”
Neumann In practice
Instruction Set Architecture

Application Software

| OS | Compiler |

ISA: Interface between Software & Hardware

software

hardware

Computer Architecture: ISA, memory, I/O, Power
Specialized H/W: GPU, co-processor
Programming Language to Hardware

```
#include <stdio.h>
int main() {
    int x, y, temp;
    x=1; y=2;
    temp =x; x=y; y=temp;
    printf("%d %d %d\n",x,y,temp);
}
```

```
movl $1, -8(%ebp)
movl $2, -12(%ebp)
movl -8(%ebp), %eax
movl %eax, -16(%ebp)
movl -12(%ebp), %eax
movl %eax, -8(%ebp)
movl -16(%ebp), %eax
movl %eax, -12(%ebp)
movl -16(%ebp), %eax
movl %eax, 12(%esp)
movl -12(%ebp), %eax
movl %eax, 8(%esp)
movl -8(%ebp), %eax
movl %eax, 4(%esp)
```

```
7f 45 4c 46 01 01 01
00 00 00 00 00 00 00
00 00 02 00 03 00 01
00 00 00 0f 82 04 08
34 00 00 00 c4 0c 00
00 00 00 00 34 00
```
What will you get out of the course?

• **Basic Elements of Computer Systems**
  – Instructions CPU, Memory, I/O

• **Memory Workings of various sub systems**

• **Design issues**
  – Performance vs cost tradeoffs

• **Interaction of Software (programs) and the underlying hardware (on which programs execute)**
Processor/CPU Performance

- What does a processor do?
Measuring Processor Speed

• How long does it take to execute a program?
  – Quicker the better

• How long does it take to go from point A to Point B

• Need to know:
  – Speed: Constant,
  – Distance: variable
  – Start time,
  – End time
Course Goals

• Teaching Style
  – Architecture course have been traditionally taught in two way: top-down or bottom-up.
  – We’re going in from the middle. Programmer-Centric Approach.

• Teaching Goals
  – Show that by knowing more about the underlying system, one can be more effective as a programmer.
  – Write programs that are more reliable and efficient.
  – Understand how programs interact with the underlying hardware.
  – Learn the insides and outsides of a computer’s architecture.
  – Learn how to do low-level programming.
Course Expectation

• **What to expect from the course:**
  – Will cover key issues and concepts in class.
  – Recitations will provide review and teach you the tools you need.
  – Programming Projects (Don’t freak out… yet)
  – A mid-term exam and a final exam
  – Practice homework and quizzes

• **What do I expect of you:**
  – Come to class
  – Read the textbook (Listening to me is not good enough)
  – Work through the problems in the textbook (not really homework… but it helps)
  – Do the projects
  – Ask questions (**IMPORTANT**)