CS-118 Intro to Assembly Language Programming

January 17, 2016

Contact info:

- Instructor: David Topham : dtopham@ohlone.edu
  - NC2320
- Computer Science: `http://www.ohlone.edu/instr/computerscience/`
- Piazza: `http://piazza.com/ohlone/spring2016/cs118/home/`
- Blackboard: `https://ohlone.blackboard.com/webapps/blackboard/execute/courseMain?course_id=_19440_1`

COURSE DESCRIPTION:

- This course is an introduction to the Intel IA-32 Assembly language. Topics include numbering systems, IA-32 processor architecture, native machine instructions, memory addressing, subroutines, and file I/O.

Pre-requisites:

- Intro to Programming
  - CS 102 or equivalent

Schedule 1/25/2016-5/20/2016

- *Classes:*
  - CS-118-01 (065110) TTh 1:00 PM - 4:15 PM, Room NC2318
    - **Final Exam - Tuesday 5/17 12-2 PM**
- *Holidays* (Classes do not meet on these days):
  - Fri Feb 12 - Mon Feb 15 (*President's Day*)
  - Mon Mar 21 - Sun Mar 27 (*Spring Break*)
  - Mon May 30 (*Memorial Day*)

Textbook

The course is based on labs and discussions; however we may use one or both of these books and other materials as references as needed.

These are free, online books!

- Below C Level: An Introduction to Computer Systems by Norm Matloff
- `http://heather.cs.ucdavis.edu/~matloff/50/PLN/CompSystsBook.pdf`
- Programming from the Ground Up by Jonathan Bartlett
- `http://www.bartlettpublishing.com/site/bartpub/section/9`

3 Units: 36 hrs. Lecture and 72 hrs. Lab

- At 16 weeks per semeseter, that is 6.75 hours per week of "instructor-student contact time".
- For college-level courses there is an expectation that an equivalent amount of time will be spent on reading and studying the materials to prepare for quizzes, labs, and exams. I suggest that you form informal study groups with other students in the course, and take advantage of office hours to help clarify difficult concepts.

Academic Honesty

Please read this link to familiarize yourself with Ohlone's policy on academic honesty. This is very important to me and no form of cheating is acceptable. You must do your own work for all graded material. Electronic sharing of lab/hw files is forbidden and both the person sharing and the people receiving files in this way will get zero credit. Repeated offenses will result in failing the course.

- Ohlone Academic Honesty Policy:
  - `http://dev2.ohlone.edu/people/dtopham/files/cs102/dishonesty.html`

Mental Health and Stress Management

- Student Health Center

The staff and faculty of Ohlone College are here to see you succeed academically and care about your emotional and physical health. You can learn more about the broad range of confidential student services, including counseling and mental health services available on campus by visiting the Student Health Center in building 7 or visit this webpage:
`http://stepupohlone.org/oncampus-resources/`.

How grades are calculated

- Lab assignments - 30%
- Quizzes(Unannounced) - 30%
- Final Exam - 40%

There is no make-up on any of this graded work. If you are late or miss it, no matter what the reason, then the score is zero. If you miss something due to sickness, emergencies, etc., talk to me *at the very end* of the semester and we will determine the effect (if any) at that time on your grade. I will not make any exceptions to this policy no matter how different you think your particular circumstance is to others.

Student Learning Outcomes

- Examine the decimal, binary and hexadecimal numbering systems. Convert values between each of the systems.
- Demonstrate knowledge of functions and characteristics of an assembly language
- Demonstrate knowledge of the use of registers, method of addressing, common instruction formats, stack processing, array and indirect processing.
- Demonstrate the ability to program elementary programs in assembly language and provide appropriate documentation.

Student Learning Outcomes

- Evaluate basic boolean logic and analyze how it applies to programming and computer hardware.
- Show how high-level languages translate statements into native machine code.
- Compare ways that application programs communicate with the operating system using interrupt handlers, system calls, and common memory areas.
- Survey CS concepts such as virtual machines, memory management, instruction execution, pipelining, cache, paging, and error-correction codes.

Basic Concepts

- Translate C/C++ expressions to ASM
- Convert between different data representations
- Boolean Functions

x86 Architecture and ASM Fundamentals

- Clock cycles
- Logical and physical addresses
- ASM Data representations

Data Transfers, Addressing, and Arithmetic

- Data Addressing
- Data Arithmetic
- Data Directives
- Indirect Addressing
- Loops

Procedures

- Using library procedures
- Stack Operations
- Defining procedures

Conditional Processing

- Boolean Operations
- Conditional Loops
- Control Structures

Arithmetic

- Shift and Rotate Instructions
- Multiplication and Division

Advanced procedures

- Stack Frames
- Recursive Procedures

Structures, Strings and Arrays

- Using Data Structures
- String and Array Processing

Real Numbers

- Floating-Point Processing
- Binary Representation
- Instruction Encoding

High-Level Language Interface

- C/C++ inline ASM
- Mixing C/C++ and ASM