03–60–266
Computer Architecture II: Microprocessor Programming
Winter 2017, 3hrs/wk, 3 credits

Instructor's Information

Name and Title Dr. Alioune Ngom, Associate Professor of Computer Science.

Office Location 5107 Lambton Tower, School of Computer Science, 401 Sunset Avenue, University of Windsor.

Web Page http://angom.myweb.cs.uwindsor.ca/

Contact information

Telephone 519-253-3000 extension 3789.

Fax 519-973-7093.

E-Mail angom@uwindsor.ca

Office Hours Fridays 8:30AM to 11:30AM.

Course Description, Philosophy, Objectives, and Prerequisite

Course Description Basic Concepts, principles of assembler-level programming of common microprocessors. Machine language. Data representation and types. Assembly language fundamentals. Memory segmentation and access. Data transfer, addressing modes, and arithmetic. Register set and operations. Instruction set. Arithmetics. Procedures, calling conventions, parameter passing, recursion and macros. Input/Output operations. Stack operations and high-level language interfacing. BIOS and DOS interrupts.

Course Philosophy In the previous course, 03–60–265, you have learned the subject of computer architecture by following a *bottom-up* approach. By starting from basic hardware components (transistors and logic gates) to construct more sophisticated circuits (adders, decoders, flip-flops, registers, ...), you have seen how the processor, the memory and a whole computer system is structured. Conversely, this course examines computer architectures by following a *top-down* strategy. It builds from the knowledge you have about high-level languages like C, C++ or Java, and teaches computer architecture from the programmer's perspective. Hence, you will learn the actions that the processor must do to perform the tasks that are formulated in a high-level language. This implies, that you need to learn the set of *basic actions* that a processor can do: its *instruction set*, and how a high-level language compiler decomposes the high-level language commands into machine-level instructions. However, this can be achieved only by learning the instruction set of a processor at the symbolic (and humanly understandable) level: the *assembly language level*. Consequently, you will learn how to program a processor in assembly language in order to perform the tasks that are normally formulated in high-level language. We will use the *Pentium* processor from Intel and the *Turbo Assembler*, *TASM32*, from Borland.

Course Objectives This course examines the link between high-level languages (processed by a compiler) and the microprocessor, which is the level of *assembly language*. Assembler is tailored specifically toward a particular processor architecture, and yet is intended to be sufficiently friendly that a computer science student can write reliable, bug-free, programs in any assembler. The computer science student should learn at least the fundamentals of assembler for the following reasons:

- Most hardware drivers and systems software are written in a mixture of C, C++, and assembler. Some drivers are written entirely in assembler. In any case, an in-depth understanding of machine and assembler concepts is essential to designing and writing a reliable hardware driver.
- A detailed knowledge of assembly is *required* for compiler development and code optimization.

- As a vital link between compilers and the microprocessor architecture, one's education is incomplete without an appreciation of assembler. It is the conceptual bridge between the world of abstract, high-level language tools and the microprocessor.

The goals of the course can be summarized as follows:

- Understand the difference between machine language, assembly language and high-level languages.
- Become familiar with the instruction set of a processor, namely the Intel's Pentium.
- Understand how high-level language commands are broken into processor-level instructions.
- Acquire a good understanding of interrupts and exceptions.

Course Prerequisite Minimum grade of C- in 03–60–265.

Basic Course Information

Lectures and Laboratories

Lectures Monday and Wednesday 08:30–09:50 AM, BB 121.

Laboratories

Section 51: Monday, 10:00–11:20 AM, LT 3107

Section 52: Wednesday, 10:00–11:20 AM, LT 3107

Section 53: Tuesday, 0:00–11:20 AM, LT 3107

Section 54: Thursday, 0:00–11:20 AM, LT 3107

Course Textbook Kip R. Irvine, *Assembly Language for x86 Processors*, Sixth Edition, Prentice Hall, 2011, ISBN 0-13-602212-X-1

Course Website At https://blackboard.uwindsor.ca/

Course Outline (tentative and subject to change)

1. Introduction to Microprocessor Programming (assembling, linking, loading and 265 review)
2. Assembly Language Fundamentals (registers, variables, basic arithmetics, . . .)
3. Control Flow Instructions
4. Addressing Modes
5. Logic, Shift and Rotate Instructions
6. The Stack (activation records, . . .)
7. Procedures
8. Multiplication, Division and Numerical Conversions
9. Floating Point Representations (arithmetic, . . .)
10. Floating Point Directives and Instructions
11. Advanced Procedures and Parameter Passing
12. Modular Programming and Macros

Course Work and Grading

Course Work Grades are based the following:

Q Average over a maximum of 4 Quizzes, worth 10%

A Average over a maximum of 4 Assignments, worth 20%

M One Midterm Exam, worth, 30%

F Final Exam, worth 40%

Exam Dates

Midterm Exam: Monday, February 27th, 2017.

Final Exam: week of April 8th-21th, 2017, [to be announced by U Windsor].

Grading ***To pass this course, one must have at least 50% of the weighted sum of midterm and final examinations***. That is, the final numeric grade G (total: 100 points) will be calculated as follows:

If $(0.30 \times M) + (0.40 \times F) < 35$ Then
  $G = (0.30 \times M) + (0.40 \times F)$
Else
  $G = (0.10 \times Q) + (0.20 \times A) + (0.30 \times M) + (0.40 \times F)$

Letter Grading The University of Windsor uses a percentage marking and grading scale as of Fall 2013. ***Only raw % scores (e.g., 75%) are assigned in course work and only raw scores are used in the computation of cumulative, major and sessional grade point averages. The meaning of scores (or logical conversion of raw scores to grades) in transcripts are with a conversion scale different from the one used previously. This new conversion scale can be found through*** www.uwindsor.ca/calendar, then grading and marks/grades descriptors link. ***Only raw % scores are assigned in course work and meaning of scores in transcripts are***:

Letter Grade	Raw %Score Range
A+	$90 \leq G < 100$
A	$85 \leq G < 90$
A-	$80 \leq G < 85$
B+	$77 \leq G < 80$
B	$73 \leq G < 77$
B-	$70 \leq G < 73$
C+	$67 \leq G < 70$
C	$63 \leq G < 67$
C-	$60 \leq G < 63$
D+	$57 \leq G < 60$
D	$53 \leq G < 57$
D-	$50 \leq G < 53$
F	$0 \leq G < 50$

In computing a student's average, grades from 0% to 22% are calculated as 22%. Grades from 23% to 40% calculated as 40%. Grades from 40% to 49% are calculated as is into the student's average. All grades are recorded in the transcript as is. All grades below 50% are considered failures. (see mark/grades descriptor page of calendar www.uwindsor.ca/calendar for details).

Teaching Evaluation

Student Evaluation of Teaching (SET) forms will be administered during the last two weeks of the class schedule.

Course Policies

Attendance and preparation Lecture attendance is mandatory and students are expected to come well-prepared for every class. Note taking is encouraged to help understand ideas more deeply.

Assignment submission All assignments must be handed in to me in classroom at the beginning of the lecture on the due dates and in envelopes with the School of Computer Science and University of Windsor logo on them. ***Late submission will not be accepted (tolerated)***. Students are responsible for making sure that I receive their assignments by or on the due dates. All assignments as well as envelopes must be clearly marked with the student name, student number, course name and number, section number and the instructor's name.

Academic honesty *You are expected to do all of your work on assignments and examinations individually. That is, collaboration on the assignments and/or plagiarism is not accepted; what you turn in should be your own work.* ***Anyone found cheating on any graded assignment or examination will get no points at all for that homework assignment or question in exam****. The instructor reserves the right to assign anyone involved in cheating a failing grade (F-) and will initiate the proceedings for disciplinary actions by the department and the university. This will be irrespective of who cheated from whom. In other words, you are responsible to protect your work from others.* ***Please read the University of Windsor regulations on cheating****.*

Makeup/Incomplete THERE IS NO MAKE-UP EXAM FOR MIDTERM AND QUIZZES. Missing a quiz or the midterm will greatly affect your grade since they carry some weight. In the case of illness or serious and unavoidable reason (as per the Senate Bylaws), please consult with the instructor in advance if possible to make alternate arrangements. You must formally inform the instructor in writing and present proper supporting documents (stating that the student was unable to attend the exam at the specified time and date) within a week from a quiz or the midterm, or earlier to be verified. If the reason is deemed valid, the weight of the midterm will be added to that of the final exam. (i.e. the final exams weight will increase to include the missed midterm weight). Likewise, the weight of a missed quiz will be added to the following quiz or midterm, depending on which quiz has been missed. All students are required to write and PASS the final exam in order to pass the course. Final examination must be taken to obtain a final score in the course. If a final exam is missed for valid medical or emergency reason (proof needed), student is allowed to write a makeup final exam in the first week of the next semester (following semester), for all students who missed the final exam.

Appeal Students who wish to appeal an assignment or exam mark should do it within two weeks of the reception of the mark. I will be glad to remark your work and explain my marking scheme to you. Numerical errors in adding marks will be corrected when identified. In case of a total disagreement on a mark, you must then submit a formal appeal. **Please read the University of Windsor regulations on appealing**