

## 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://www.cs.uwindsor.ca/~angom>

### Contact information

**Telephone** 519-253-3000 extension 3789. Never call me unless it is *very* important

**Fax** 519-973-7093. Never fax me

**E-Mail** [angom@cs.uwindsor.ca](mailto:angom@cs.uwindsor.ca)

**Office Hours** Thursday: 8:30 AM to 12:00 PM.

## Course Description, Philosophy, Objectives, and Prerequisite

**Course Description** Principles of assembler-level programming of common microprocessors. Machine language. Data representation. Assembly language fundamentals. Memory segmentation and access. Addressing modes. Register set and operations. Instruction set. 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 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.
- Assembler is very useful for work on operating systems, file systems, network systems and most embedded applications.

- 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, Wednesday 08:30-09:50 AM, MH 109.

### Laboratories

Section 51: Monday, 10:00-11:20 AM, ER 3107

Section 52: Wednesday, 10:00-11:20 AM, ER G125

**Course Textbook** Kip R. Irvine, *Assembly Language for Intel-Based Computers*, Fifth Edition, Prentice Hall, 2006, ISBN 0-13-238310-1

**Course Outline** (tentative and subject to change)

1. Introduction to Microprocessor Programming (assembling, linking, loading and 265 review)
2. Assembly Language Fundamentals (registers, variables, integer 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. Modular Programming
12. Parameter Passing
13. High-Level Language Interface
14. Input and Output (programmed I/O, interrupt-driven I/O and DMA)
15. Interrupts and Exception Handling
16. ...

# Course Work and Grading

**Course Work** Grades are based the following:

**Q** Average over a maximum of 6 Quizzes, worth 8%

**A** Average over a maximum of 4 Assignments, worth 12%

**M** One Midterm Exam, worth, 30%

**F** Final Exam, worth 50%

## Exam Dates

Midterm Exam: Wednesday, February 16-th, 2011.

Final Exam: Date determined by the Faculty of Science.

**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  $(0.30 \times M) + (0.50 \times F) < 40$  Then

$$G = (0.30 \times M) + (0.50 \times F)$$

Else

$$G = (0.08 \times Q) + (0.12 \times A) + (0.30 \times M) + (0.50 \times F)$$

**Letter Grading** The final letter grade,  $L$ , will be given from the numeric grade based on the following conversion rule:

Letter Grade	Numeric Grade Range
A+	$93 \leq G < 100$
A	$86 \leq G < 93$
A-	$80 \leq G < 86$
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	$35 \leq G < 50$
F-	$0 \leq G < 35$

## 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. Notetaking 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** *Makeup work or incomplete grade are only given in unusual circumstances, and only when work has been completed satisfactorily up to the point when the incomplete was requested. If you suspect that you will be unable to attend an examination because of a **valid and verifiable reason**, you **must** give me a prior notice, **at least** one full day before the examination. Even if you are sick or face unavoidable circumstances, you **must** notify me or the department through phone, email, fax, etc. along with a valid documentary evidence. I **must** receive a **proper documentary evidence within a week** of the examination. **In the absence of such notice and a proper documented proof, makeup examination(s) will not be allowed.** Unless mentioned otherwise, all examinations will be closed book, closed notes and closed neighbors. Date and place for makeup examination will be announced at an appropriate time. It ! will be your responsibility to get the necessary information about the makeup examination. **Please read the University of Windsor regulations.***

**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**