

COP 3402: Systems Software – Fall 2016

Syllabus

Instructor

Dr. Pawel Wocjan

wocjan@cs.ucf.edu

Lectures

MoWe 7:30 – 8:45PM HEC 125

Instructor's Office Hours

TuThu 4:30 – 5:45PM HEC 341

Graduate Teaching Assistants

Mahdi Mahmoudkalayeh

mahdi@knights.ucf.edu

Ramin Izadpanah

r.izadpanah@knights.ucf.edu

Recitations

Mo 6:30 – 7:20PM NSC O108

We 3:30 – 4:20 PM CB1 O307

We 6:30 – 7:20 PM NSC O108

Course Outline

This course is designed to provide a fundamental understanding of real and virtual machines as language processor and the implementation of compilers. We will study the processor as an instruction interpreter. Compilers, assemblers, linkers and loaders, and virtual machines will be presented as systems software for program development. An introduction to Operating system will be given.

Course Topics

Introduction to compilers and interpreters, virtual machines, computer architecture and assembler, loaders and linkers, macro-processors, run time environment and operating systems

Prerequisites

COP 3502 – Computer Science I, COP 3223 Intro to Programming with C, CDA 3101 Computer Logic & Organization

Grading Policy

- (20 points) **Midterm Exam – 10/26 (Wednesday) 7:30 – 8:15PM**
- (20 points) **Final Exam – 12/07 (Wednesday) 7:00 – 9:50PM**
- (15 points) **Programming Project Module 1: P-Machine**
- (15 points) **Programming Project Module 2: Lexical Analyzer**
- (15 points) **Programming Project Module 3: Parser / Code Generator**
- (15 points) **Programming Project Module 4: PL/0 Compiler**
- (5 points) **Recitations:** you can earn an additional 5 points if you come to more than 75% of the recitations

Letter grades

Points	Grade
>= 95	A+
>= 90	A
>= 85	A-
>= 80	B+
>= 75	B
>= 70	B-
>= 65	C+
>= 60	C
>= 55	C-
>= 50	D+
>= 45	D
>= 40	D-
Less	F

Note on Academic Dishonesty

Any academic dishonesty (including, but not limited to: cheating, copying and/or plagiarism) with respect to any exam or assignment in this class will result in a grade of **F**, following by the usual procedures for dealing with such behavior, as describe in the *UCF Golden Rule : a handbook for students*.

Supplementary Texts

Compilers: Principles, Techniques, & Tools, Second Edition by Alfred V. Aho, Monica S. Lam, Ravi Sethi, and Jeffrey D. Ullman. Addison Wesley, 2007

Modern Compiler Implementation in C by Andrew Appel. Cambridge University Press, 1998

Compiler Construction: Principles and Practice by Kenneth C. Loudon, PWS, 1997

System Software: An Introduction to Systems Programming, 3rd Edition by Leland Beck, 1997.

Concepts of Programming Languages, 8th Edition by Robert W. Sebesta. Addison Wesley, 2010.

The Semester Plan (Tentative)

- System Software
- Computer System Structure
- Virtual Machines
- Compilers and Interpreters
- Syntax Analysis
- Grammars and Parsing
- Code Generation
- Assemblers and ELF (Executable Linkable Files)
- Linkers and Loaders
- Operating System Structure
- Interrupt Handling

Programming Project

The programming project consists in implementing a compiler for a tiny programming language.

You will work in teams comprised of 4 members.

The compiler must generate code for a virtual machine. The project will be divided in four modules: 1 - Virtual Machine implementation, 2 - Scanner, 3 - Parser and Code Generation (Part 1) and 4 - Parser and Code Generation (Part 2). Details will be given out well before the due dates for each part of the project.

This project must be written in C (**not C++**) on a UNIX system. The standard for this class will be the Linux system in the main computer lab called Eustis. You are welcome to write and test code on some other system, if you wish, but it will be graded on Eustis and if it does not work there, it does not work. You will be given a Eustis account at the beginning of the semester, and instructions on how to access your account can be found at

http://www.cs.ucf.edu/~wocjan/Teaching/2016_Fall/SS/eustis_tutorial.pdf

To pass this course, you **must** successfully complete at least 3 out of the 4 modules. No exceptions.

Each module will have a due date and you will receive only half of the points for that module if you submit after the due date. If you submit 2 days after the deadline you will get 0 points. **No exceptions.**

Also, you may be required to explain your code in detail (algorithms and data structures, time and space complexity etc.) to the instructor and/or the teaching assistants.

In general, this project will give you a better understanding of the data structures and programming techniques used to build compilers and also provide you with experience in developing and debugging a complex software project.

Important Dates

- Midterm Exam: 10/26 (Wednesday) 3:00 – 4:15PM
- Final Exam: 12/07 (Wednesday) 7:00 – 9:50PM