COEN 180
Spring, 2000

COEN 180- Software Systems

Course Description:
This is a three credit-hour course providing an overview of the major issues in the design and implementation of operating systems and language translators. Operating system topics include: tasking and processing, process coordination and synchronization, scheduling and dispatch, physical and virtual memory organization, device management, file systems, security, communications and networking. Language translation topics include storage management, fine state automata and regular expressions, context-free grammars and push-down automata, code translation and optimization techniques, and programming language semantics.

Prerequisites: COEN 020 and either COEN 120 or COSC 154

Course Materials:
Required:
Leland L. Beck, System Software: An Introduction to Systems Programming, 3/e, Addison Wesley Longman, 1997.

Course Goals:
To prepare students to design and implement assemblers, loaders, linkers, macro processors, and elements of compilers and operating systems.

Course Objectives:
By the end of this course, you should....

- Know the seven elements used to define hardware and its software.
- Be able to evaluate hardware and software using the seven elements, and to identify the advantages and disadvantages of particular design decisions.
- Know the basic algorithm for a two-pass assembler and be able to implement it.
- Know how to modify the assembler to support a variety of machine dependent and machine independent features.
- Be able to assess various design options for an assembler.
- Know the basic algoriths for an absolute loader and be able to implement it.
- Know how to modify the loader to support a variety of machine dependent and machine independent features.
- Be able to assess various design options for an loader/linker.
- Know the basic algoriths for a macro processor and be able to implement it.
- Know how to modify the macro processor to support a variety of machine independent features.
- Be able to assess various design options for a macro processor.
- Know the basic algorithms for the phases of a compiler and be able to implement the algorithm for syntax checking.
- Know how to modify the compiler to support a variety of machine dependent and machine

independent features.
- Be able to assess various design options for a compiler.
- Know the basic algorithms for a single user operating system and be able some simple scheduling or memory management algorithms.
- Know how to modify the operating system to support a variety of machine dependent and machine independent features.
- Be able to assess various design options for an operating system.

Course Topics:	**In the Beck Text**	**Tentative Dates**
Background	Chapter 1	1/19 — 1/21
Assemblers	Chapter 2	1/24 — 2/9
Loaders and linkers	Chapter 3	2/11 — 2/25
Macro processor	Chapter 4	3/13 — 3/17
Compilers	Chapter 5	3/20 — 4/7
Operating systems	Chapter 6	4/10 — 4/28
Other system software	Chapter 7	5/1 — 5/5

Class Schedule: 3 credit course, meeting for 3 50-minute periods each week.

Contribution to Professional Component: Engineering Science 33%
Engineering Design 67%

Contribution to Program Objectives: partial fulfillment of Criterion 3 objectives A, C, E, F, G, H, I, J, and K

Prepared by: Susan A. Riedel December 1, 1999
Course Coordinator: Xin Feng

Last modified: July 13, 2000

COEN 180: Software Systems
ABET Objectives,
Assessment Instruments, and
Assessment Criteria

(A) An ability to apply knowledge of mathematics, science, and engineering:
Successful completion of all assignments, each of which focuses on one or more of the Goals listed in the COEN 180 outline.
Tests, homework, and projects will be completed with a grade of C or better to demonstrate minimum competence.

(C) An ability to design a system (sic) to meet desired needs:
Successful completion of the term projects, each of which asks students to engineer several software systems (assembler, loader, macroprocessor, syntax checker, scheduler) and demonstrate their correctness using benchmark tests.
Projects will be completed with a grade of C or better to demonstrate minimum competence.

(E) An ability to identify, formulate, and solve engineering problems.
The assigned projects and some of the assigned homework problems are open-ended, and require the students to investigate alternative solutions, evaluate those solutions, pick a solution based on quantitative measures of performance, and arrive at a solution to the problems.
Projects will be required to discuss alternative solution strategies and demonstrate the use of performance measures in deciding on a solution strategy.

(F) An understanding of professional and ethical responsibility
Teams are required to act in professional and ethical manner.
Projects will be required to conform to professional standards where they exist, and to make a complete citation of all references used and persons consulted.

(G) An ability to communicate effectively
Communication for projects must consist of both written descriptions and visual aids, such as flow charts, class diagrams, and the like.
Project grades are based both on content and on the effectiveness of the communication. A grade of C or better demonstrates minimum competence in project reports.

(H) The broad education necessary to understand the impact of engineering solutions in a global/societal context.
Successful completion of the assigned projects, which in part ask students to understand their duty to ensure that the software is as fault-free as possible.
Projects will be completed with a grade of C or better to demonstrate minimum competence.

(I) A recognition of the need for and an ability to engage in life-long learning.
Successful completion of the term projects, which asks students to engineer a software system using modern CASE tools, with the knowledge that such tools will evolve extensively throughout their careers. Students are responsible for learning how to use the CASE tools through interactive tutorials and other self-study resources.
Projects will be completed demonstrating beginning mastery with one or more CASE tools.

(J) A knowledge of contemporary issues
Successful completion of the term projects, which asks students to engineer a software system using modern CASE tools, with the knowledge that such tools will evolve extensively throughout their careers. Students are responsible for learning how to use the CASE tools through interactive tutorials and other self-study resources.
Projects will be completed demonstrating beginning mastery with one or more CASE tools.

(K) An ability to use the techniques, skills and modern engineering tools necessary for engineering practice
Successful completion of the term projects, which asks students to engineer a software system using modern CASE tools, with the knowledge that such tools will evolve extensively throughout their careers. Students are responsible for learning how to use the CASE tools through interactive tutorials and other self-study resources.
Projects will be completed demonstrating beginning mastery with one or more CASE tools.

Evaluation of Objective Attainment-Continuing Course Review

At the end of each semester, instructors of this course will submit a short written report to the Course Coordinator for COEN 180 which

r) discusses their perceptions of student knowledge and ability to apply the listed prerequisites, and
s) provides qualitative and quantitative information which discusses the attainment of objectives listed above for COEN 180.

The course coordinator will review these reports with the instructors and in consultation with the instructors recommend modification and/or enhancements to the objectives and criteria as needed. The recommendations will be forwarded to the EECE Undergraduate Committee for approval.

Last modified: July 13, 2000