ENG EC440 Introduction to Operating Systems

2008-2009 Catalog Data:
Prereq: ENG EC 327 and ENG EC 312. Operating system concepts and design objectives. Concurrent processes, process synchronization, and deadlocks. Resource management including virtual memory, CPU scheduling, and secondary storage. File structures, input/output, and distributed systems. Case studies of popular operating systems. 4 cr.

Class/Lab Schedule:
LEC: 4hrs/week

Status in the Curriculum: Required

Textbooks and other required materials:
Tanenbaum, Modern Operating Systems, 3rd edition, Prentice Hall, 2008.

References:
Silberschatz and Galvin, Operating Systems Concepts, 8th edition, Wiley, 2009
Bovet and Cesati, Understanding the Linux Kernel, 3rd edition, O'Reily, 2006

Coordinator:
Thomas P. Skinner, Associate Professor, ECE Department

Prerequisites by topic:
Prereq: CAS CS 113 and ENG SC 312.

Goals:
To provide students with a knowledge of the theory of operating system design and their underlying algorithms and hands-on experience in software engineering and development in the operating system context
Show by examples of modern operating systems how design theory has been applied to current systems.
Integrate the student's knowledge of basic computer hardware and software.

Course Outcomes:

As an outcome of completing this course, students should be able to:

1. Understand the need for an operating system
2. Relate hardware concepts to the requirements of the programmer and operating system
3. Understand the structures of operating systems
4. Understand the techniques and algorithms used to manage processes
5. Understand the requirements for interprocess communication and synchronization
6. Understand how memory management is performed within the operating system

7. Understand the incorporation of input/output devices in the operating system
8. Understand how file systems are implemented
9. Demonstrate the application of the underlying theory to current production operating systems
10. Employ operating system services in user application programs
11. Design, implement and test aspects of operating system functionality in C
12. Benchmark the space and time performance of operating system services and user applications
13. Analyze impact of operating system interfaces and implementations on user applications
14. Appreciate the complexity of designing and developing large software systems
15. Write design specifications and implementation reports for software projects and present them

Course Outcomes mapped to Program Outcomes:

Program Outcomes:	**a**	**b**	**c**	**d**	**e**	**f**	**g**	**h**	**i**	**j**	**k**
Course Outcomes:	1-13	11-13	1-14	1,2,7,13	9-15	1,2,11,13	15	1, 9	1-9,14	9-10	10-15
Emphasis:	5	4	5	3	5	3	3	3	4	5	5

1=not at all; 5=a great deal;

Contribution of Course to Meeting the Professional Component:
Engineering topics: 100%
Math & Basic Science: 0%
General Education: 0%

Prepared by: Edward A Bach, Lecturer **Date:** June 10, 2009