COURSE DESCRIPTION

Department and Course Number: CSCI 523

Course Title: Operating Systems

Current Catalog Description: Design and construction of operating systems for shared program computers; various contemporary operating systems.

Total Credits: 3 hours

Coordinator: P. Tobin Maginnis, Associate Professor of Computer and Information Science.

Textbook: Bovet and Cesati, *Understanding the Linux Kernel*, O'Reilly, 2001.
Gary Nutt, *Kernel Projects for Linux*, Addison-Wesley 2000.

Other required materials: Study guides and miscellaneous class handouts.

References: http://www.kernelnewbies.org/
http://www.kernelnewbies.org/documents/
http://jungla.dit.upm.es/~jmseyas/linux/kernel/hackers-docs.html

Course Goals: Juniors, seniors, and graduate students are provided with the conceptual tools to analyze operating systems. The Linux operating system is examined at the source code level.

Prerequisites by Topic:

1. Programming expertise in a high-level language such as Java, C, or Pascal (CSCI 211).
2. Familiarity with operating system concepts such as process management, memory management, file structures, device drivers, and the user interface (CSCI 423).

Major Topics Covered in the Course:

1. Introduction and overview (1 hour).
2. C programming language and Intel 8088 architecture review (3 hours).
3. Analysis of the Linux state management implementation and the "block move" problem (3 hours).
4. System bootstrapping and PCB initialization (3 hours).
5. Kernel-level IPC and process management (2 hours).
6. Logical memory drivers (1 hour).
7. Floppy disk driver (3 hours).
8. Timing service (1 hour).
9. Console terminal driver (4 hours).
10. Process duplication and creation (4 hours).
11. Linux memory management (1 hour).
12. Data caching (2 hours).
13. Open call (4 hours).
14. Name service (3 hours).
15. Read, write, and close calls (3 hours).
16. Tests (4 hours)

Operating Systems and Languages: Linux and C

Laboratory projects (one or two weeks per assignment):

1. Recompile the Linux kernel generating new messages as the system boots.
2. Display Linux kernel message traffic.
3. Add priority scheduling to Linux.
4. Enable the 3COM Ethernet driver.
5. Install a SSF I/O queue in a disk driver.
6. Create a logical window device driver.
7. Add record locking to the file manager.
8. Add a fourth group to owner, group, and others called net in the file manager.
9. Add an ACL to the file manager.
10. Install a garbage collector in the memory manager.
11. Create a kernel server thread and have it honor a signal to display the system state information.

Estimate of ABET/CAC Category Content:

	CORE	ADVANCED		CORE	ADVANCED
Data Structures		1	Computer Organization and Architecture		
Algorithms		1	Concepts of Programming Languages		1
Software Design					

Oral and Written Communications:

Not a significant focus of this course.

Social and Ethical Issues:

Not a significant focus of this course.

Theoretical Content (Foundations:

Not a significant focus of this course.

Problem Analysis:

Not a significant focus of this course.

Solution Design:

Not a significant focus of this course.