

COURSE DESCRIPTION

Department and Course Number: CSCI 423

Course Title: Introduction to Operating Systems

Current Catalog Description: Study of the basic concepts of operating systems, including user interfaces, process management, state saving, interprocess communication, input/output, device drivers, timing services, memory management, file management, and system abstractions.

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.

Other required materials: Study guides and miscellaneous class handouts.

References: <http://pix.cs.olemiss.edu/csci423/>

Course Goals: To provide students with the conceptual tools to analyze the design and use of operating systems. Principles of operating systems, design tradeoffs, and a detailed operating system model are presented. Students are introduced to systems programming in C language and IA86 assembly language. The Linux operating system is employed as an example operating system.

Prerequisites by Topic:

1. Basic computer organization (CSCI 223)
2. Programming experience with a general purpose, high-level language such as Java or C and knowledge of basic algorithms, data structures, and file structures.(CSCI 211)

Major Topics Covered in the Course:

1. C programming language and IA86 and PC architecture review (3 hours).
2. Operating system model and overall design tradeoffs (4 hours).
3. Analysis of state management implementation and process context switching (3 hours).
4. Process management (3 hours).
5. Kernel threads and process creation(4 hours).
6. Background system services (sync, automount, power management, quotas, etc) (1 hour).
7. Interprocess communication (3 hours).
8. Timing management (1 hour).
9. Segmentation and paging memory management models (4 hours).
10. Memory management (shared code, free, I/O and file caches) (3 hours).
11. File manager front end (naming & protection) (3 hours).
12. File manager back end (buffer movement, device map, and journalling) (4 hours).
13. Read, write, and close calls (1 hour).
14. Device drivers (1 hour).
15. System abstractions (/dev/null, /proc, /dev/pts, /dev/shm, etc.) (1 hour).
16. Logical device drivers (multi-volume and RAID) (1 hour)
17. Tests (5 hours).

Operating Systems and Languages: Windows, Linux, C, and IA86 assembly

Laboratory projects : Two weeks allotted per assignment.

1. A C program that exercises producer and consumer functions against a buffer of characters to produce a word count.
2. A basic shell-like C program that executes programs from the command line.

3. Extended shell-like C program that passes arguments and either waits for program completion or returns immediately to the user.
4. An extended C program which improves the shell with the ability to provide I/O redirection for one program.
5. An extended C program which improves the shell with the ability to provide pipe I/O communication for up to five programs.

Estimate of ABET/CAC Category Content:

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

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:

The concept of modeling with a model operating system containing hierarchical layers.

Solution Design:

Design and implementation of required programs for the laboratory exercises.