Course Outcomes	**CSC 4103**

CSC 4103: Operating Systems

Credit Hours: 3 hours

Prerequisites:

CSC 3102

Prerequisites By Topic:

Formal ADT representations such as linked lists, analysis of algorithms, algorithm design techniques, programming knowledge in C, C++ or JAVA.

Catalog Course Description:

Design techniques, process management, processor scheduling; deadlocks, memory management, secondary memory management, file management; I/O systems, Unix systems.

Course Outcomes

1. Master functions, structures and history of operating systems
2. Master understanding of design issues associated with operating systems
3. Master various process management concepts including scheduling, synchronization, deadlocks
4. Be familiar with multithreading
5. Master concepts of memory management including virtual memory
6. Master system resources sharing among the users
7. Master issues related to file system interface and implementation, disk management
8. Be familiar with protection and security mechanisms
9. Be familiar with various types of operating systems including Unix

Texts and Other Course Materials

A. Silberschatz, P.B. Galvin and G. Gagne, “Operating System Concepts, (6th or Newer Edition) Wiley.

Reference: M. Kiefer and S.A. Smolka, “OSP: An Environment for Operating System Projects,” Addison-Wesley.

Major Topics

- Introduction including history and types of operating systems
- Processes and threads, scheduling policies, algorithms, multilevel queuing

- Process synchronization concepts, classical examples, communication, interrupts, semaphores, monitors
- Resource allocation, deadlock prevention, avoidance and detection
- Main memory, paging, multilevel paging, segmentation, virtual memory, thrashing, page replacement algorithms
- File system interface, implementation, file organization, storage allocation, access methods, access control
- I/O system and services, disk management, disk scheduling, RAID structures
- System performance, protection and security, policies and methods, Access matrix

Assignments/Projects/Laboratory Projects/Homeworks

- Individual written homework assignments (4). Each consists of ten short, long and mixed questions including those from the exercises of the text book.
- Individual programming assignments (4). Each takes about one to two weeks to complete. Sample projects:
    - Implement various modules for CPU scheduling, memory management in OSP simulator
    - Perform various case studies using outputs produced by OSP simulator, e.g., difference between I/O bound versus CPU bound processes.
    - Write programs for different algorithms related to operating systems such as a bounded buffer problem.

Curriculum Category Content (estimated in semester hours)

Area	**Core**	**Advanced**	**Area**	**Core**	**Advanced**
Algorithms	20	15	Data Structures	2	
Software Design	5		Prog. Languages	3	
Computer Architecture			Mathematical fundamentals		

Relationship to Criterion 3 Outcomes

A	**B**	**C**	**D**	**E**	**F**	**G**	**H**	**I**	**J**	**K**
*	*	*		*		*	*	*		*

- Data Structures -- 2 hr core

Use of various basic data structures such as linked lists, queues, stacks, search trees, hashing and graphs in the context of operating system.

- Algorithms and Software -- 25 hr core / 15 hr advanced
  Fundamentals of process management, memory management, I/O system, storage management, and protection and security

- Computer Organization and Architecture:

- Concepts of Programming Languages -- 3 hr core
  C, C++ and JAVA languages, which are used for programming assignments.

- Social and Ethical Issues:

- Oral Communication (presentations)

- Written Communication:
  Students are required to submit 4 written home works involving discussions of algorithm design issues. The programming assignments (4) also require the students to write comments/specifications of the functions used in their programs. Also some assignments involve preparation of a report on analysis of the results from OSP simulator

Course Coordinator: Dr. Bijaya B Karki
Last Modified: January 8, 2007