CSE3201: Operating System: Course Plan

Dr. Mosaddek Hossain Kamal Tushar, Professor, CSE, DU

Third Year, Second Semester, 2019; Credits: 3

E-mail: tushar@cse.du.ac.bd
Office Hours: **S 10-11:30am**
Office: Science Complex 318

Web: Google Scholar

Class Hours: **Tue 8:30-10:00am Thu 10:00-11:30am**
Class Room: **Science Complex 413**

Lab Course: **CSE3211**
Lab Room: **Science Complex 427**

Lab Hours: **Tue 10:00-13:00**
Wed 10:00-13:00

1 Course Introduction

This course introduces the basic design in depth and services of the modern operating systems. The course aimed to understand the management of multiple tasks that execute at the same time and share resources. In this connection, the course includes concepts, critical investigation and design of processes and threads, context switching, synchronization, scheduling, and deadlock. Next, the operating system course study the problems of memory management. It will cover topics of well-established solutions of the issues such as linking, dynamic memory allocation, dynamic address translation, virtual memory, and demand paging. Furthermore, the course includes detail design and concepts of file systems, including topics such as storage devices, disk management and scheduling, directories, protection, and crash recovery.

2 Course Objectives

This course comprehends the students to the fundamental design concepts and issues of the operating system. The course aims to teach students in the fundamental concepts and components of operating systems, the relevant characteristics of hardware, and the trade-offs between conflicting goals faced by the operating systems in efficiently handling a wide range of applications. This course contributes a perception of the underlying operating systems which students have intrinsically relied upon when developing applications in the courses within Computer Science and Engineering. The knowledge obtained will continue to apply to future careers when designing complex systems and large applications.

2.1 Specific Objectives/Outcome

Understand the fundamental concepts, mechanism, and explain the reason why the operating systems are built the way they are, design implications and lessons for other complex software systems.

- Educational

  E1 : visualization and understanding of design trade-offs and design decisions and their dependence on the target environment;

  E2 : appreciation of the distinction between mechanisms and policies, and why this is important.

  E3 : exposure to system low-level code, and

  E4 : exposure to current trends in operating systems research and development.

- Technical

  T1 : operating system system calls and interaction between applications and os kernel

  T2 : processes and process management, including threads and concurrency control mechanism,

  T3 : techniques for physical and virtual memory management,

  T4 : on-line storage system and methodologies,

  T5 : user and software security and protection mechanism,

3 Prerequisites/Corequisites

- Data Structure
  - Stack, Queue, Lists, hash tables, tree, heaps, etc ...
- Microprocessor and Interfacing
  - Assembly programming
  - Mapping of high-level procedural language to assembly language
- Competent Programmer
  - We will use C programming language
  - The dominant language for OS implementation.
  - Need to understand pointers, pointer arithmetics, explicit memory allocation, etc.

4 Course Content

Introduction: Operating system overview, computer system structure, structure and components of an operating system. **System calls**: class of system calls and description. **Process and threads**: process and thread model, process and thread creation and termination, user and kernel level thread, scheduling, scheduling algorithms, dispatcher, context switch, real time scheduling. **Concurrency and synchronization**: IPC and inter-thread communication, critical region, critical section problems and solutions. **Resource management**: introduction to deadlock, ostrich algorithm, deadlock detection and recovery, deadlock avoidance, deadlock prevention, starvation. **File management**: File Naming and structure, file access and attributes, system calls, **File organization**: OS and user perspective view of file, memory mapped file, file directories organization.

File System Implementation: implementing file, allocation strategy, method of allocation, directory implementation, UNIX i-node, block management, quota, and example file system. **Memory management**: basic memory management, fixed and dynamic partition, virtual memory, segmentation, paging and swapping, MMU. **Virtual memory management**: paging, page table structure, page replacement, TLB, exception vector, demand paging and segmentation, thrashing and performance. **I/O management**: I/O Devices, I/O Bus architecture and controller, interrupts, DMA, programmed I/O. **Disk I/O management**: structure, performance, low-level disk formatting, Disk arm scheduling algorithm, error handling, and stable storage.

5 Lectures

Note: Maximum Three Hours (two class period) is Required for each Lecture.

- Lecture 1: Introduction to Operating System
- Lecture 2: Computer Hardware Review
- Lecture 3: Operating System Components
- Lecture 4: The MIPS R2000/R3000
- Lecture 5: The MIPS R2000/R3000 (Contd.)
- Lecture 6: System Calls
- Lecture 7: Processes and Threads
- Lecture 8: Concurrency and Synchronization
- Lecture 9: Deadlock
- Lecture 10: File Management
- Lecture 11: Unix File Management
- Lecture 12: Memory Management
- Lecture 13: Virtual Memory
- Lecture 14: Virtual Memory II
- Lecture 15: Process and Thread Scheduling
- Lecture 16: Real-time Scheduling
- Lecture 17: I/O Management
- Lecture 18: Disk I/O Management
- Lecture 19: Security
- Lecture 20: OS Case Study (Optional)

6 Unit-wise Learning Outcomes

- Lecture 1, 2, 3: Outcome/Objective: T1, E4
- Lecture 4, 5, 6: Outcome/Objectives: T1, E3
- Lecture 7, 8, 9: Outcome/Objectives: T2, E1, E2
- Lecture 10, 11: Outcome/Objectives: T4, E1, E2
- Lecture 12, 13, 14: Outcome/Objectives: T3, E1, E2, E3
- Lecture 15, 16: Outcome/Objectives: T2, E4, E2
- Lecture 17, 18: Outcome/Objectives: T4, E1, E4
- Lecture 19, 20: Outcome/Objectives: T5, E4, E2

7 Lab Courses (CSE3211)

The student will use OS/161 which is an educational operating system running on system/161 and developed by systems group at Harvard. The operating system OS/161 contains roughly 20,000 lines of code and comments. Student will be given four assignments to achieve the specific objectives described in Section 2.1. The student will modify the source code and add functionalists to solve specific problems of process/thread concurrency management, virtual memory, file systems and system calls.

8 Instructional Strategies

Multimedia presentation, white board, Home work, Assignment, Group discussion.

9 Assessments

9.1 In-course Exam

- Midterm Exam: 20%
- Home work /Class performance: 15%
- Theoretical Course Attendance : 5%

Final Exam

- Marks: 60%
- Duration Three hours
- Six/Five problem solving/analytic questions

9.2 Lab

- Number of Assignments: 04
- Marks Distribution : 25.00%/assignment
- Attendance : 5.00%

9.2.1 Assignment Submission

- Assignment Submission : Electronic Submission
- No hard copy accepted
- Codes and Design documents, etc (TBA)
- reduce 5% of total assignment value per day late
- Assignment only accepted up to one week late and 8+ days late marks = 0

10 Reading Materials

- Text Book
  - Modern Operating System – Andrew S. Tanenbaum
- References
  - A. Silberschatz and P.B. Galvin, Operating System Concepts, Addison Wesley
  - William Stallings, Operating Systems: Internals and Design Principles, Pearson; 8 edition (February 2, 2014)
  - A. Tannenbaum, A. Woodhull, Operating Systems– Design and Implementation, Pearson; 3 edition, Jan 2006
  - John O'Gorman, Operating Systems, MacMillan
  - Uresh Vahalla, UNIX Internals: The New Frontiers, Prentice Hall, 1996
  - McKusick et al., The Design and Implementation of the 4.4 BSD Operating System, Addison Wesley, 1996