**Course Syllabus**

**Course Objective and Outcome Form**

Department of Electrical and Computer Engineering

North South University, Bashundhara, Dhaka-1229, Bangladesh

**Course Number and Title:** CSE 323: Operating Systems Design

**Credits:** 3 SCH

**Course type:** Required, Engineering, Lecture

**Course Prerequisites:** CSE 225 (Data Structure and Algorithm)

and CSE 232 (Computer Organization and Design)

**Course Schedule/Timing:** Lecture – 3 Hours/Week

**Course Assessment:** Homework/Programming Assignment: 4

                        Exam: Quizzes – 3, Midterm - 1, Final – 1

                        Assessment: Presentation-1, Final Project -1

**Grading policy:**

Homework-10%,

Quiz – 15%,

Midterm-30%,

Presentation and Final Project – 15%,

Final – 30%,

**Course Objective**

Upon the completion of the course, the student should be able to

* ***Analyze and Problem Solving:*** Familiarity of analysis and problem solving in the systems perspective
* ***Foundation Knowledge:*** Understanding the basic components of computing and their architecture, connection and relationship with users
* ***Programming:*** Introduction to system programming where students are required to work on a project which involves communicating directly to hardware.

**Catalog Description (**Syllabus**):  Operating Systems Design:** Operating Systems Design: An introduction to the structure of modern operating systems. Topics include operating systems structure, a synchronism, mutual exclusion, deadlocks, monitors, process state transition, interrupts, context switching, storage management for both real and virtual storage, processor scheduling, multi-processing, auxiliary storage management, computer systems performance, network and security.

**Textbook and related course materials:**

* Abraham Silberschatz ,“Operating Systems Concept”, Wiley; 8 edition (July 29, 2008)
* Kay A. Robins, Steve Robins, “UNIX Systems Programming: Communication, Concurrency and Threads”, Prentice Hall; 2 edition (June 27, 2003)

**Topics covered and level of coverage (**Topic/Hours**):**

|  |  |
| --- | --- |
| ***Course Topics*** | **Coverage** |
| Introduction to Operating System | 1.5 Hours |
| Process management [Programming Assignment] | 3. 0 Hours |
| Threading – introduction to multithreaded environment [Programming Assignment] | 1.5 Hours |
| Process Scheduling: Introduction to various scheduling mechanism and rigorous problem solving [Programming Assignment] | 4.5 Hours |
| Process Synchronization [Programming Assignment] | 3 Hours |
| Deadlock Handling [Start working on Final Project] | 3 Hours |
| Memory Management and virtual memory management | 6 Hours |
| File System Management and Storage Management | 6 Hours |
| Protection and Security [End working on Final Project] | 3 Hours |
| Final Project Presentation and demonstration | 3 Hours |
| Presentation of advanced topic of modern OS | 1.5 Hours |

**Material available to students and department at the end of the course:**

*Course Objectives and Outcomes Form:* Student, Department, Instructor

*Lecture notes, homework assignments and solutions:* Student, Department, Instructor

*Student work sample solutions (homework, quiz, exam, report etc.):* Department

*Course performance form including student surveys:* Department, Instructor

**Involve computer assignments?** Yes

**Will this course have TA(s) when it is offered?** No

**Level of contribution of course to Learning Outcomes (**a-k: Strong, average, low**)**

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| ***Learning Outcome*** | **A** | **B** | **C** | **D** | **E** | **F** | **G** | **H** | **I** | **J** | **K** | **L** |
| ***Contribution of Course*** | **\*\*\*** |  | **\*\*** |  |  |  |  |  |  | **\*\*** |  |  |

**\*\*\* Strong      \*\* Moderate     \*Low**

**How Learning outcomes are covered by the specific course objective/outcome**

|  |
| --- |
| Course Objective/outcome |
| Introduction to Operating System |
| Process management [Programming Assignment] |
| Threading – introduction to multithreaded environment [Programming Assignment] |
| Process Scheduling: Introduction to various scheduling mechanism and rigorous problem solving [Programming Assignment] |
| Process Synchronization [Programming Assignment] |
| Deadlock Handling [Start working on Final Project] |
| Memory Management and virtual memory management |
| File System Management and Storage Management |
| Protection and Security [End working on Final Project] |
| Final Project Presentation and demonstration |
| Presentation on Contemporary OS topics |

**Student Outcome:  BS -CSE Program**

The BS-CSE has twelve (a-l) student learning outcomes adopted from the preferred outcomes of both Computing Accreditation Commission (CAC) and Engineering Accreditation Commission (ECA) of ABET. The students who complete the BS-CSE program will have:

(A) an ability to apply knowledge of computing, mathematics, science and engineering appropriate to the discipline

(B) an ability to analyze a problem, and identify and define the computing requirements appropriate to its solution

(C) an ability to design a computer-based system, component, or process to meet desired needs within realistic constraints such as economic, environmental, social, political, ethical, health and safety, manufacturability, and sustainability

(D) an ability to function effectively on multidisciplinary teams to accomplish a common goal

(E) an understanding of professional ethical,leagal, security and social issues and  responsibilities

(F) an ability to communicate effectively with a range of audiences

(G) the broad education necessary to analyze and understand the impact of local and global engineering and computing solutions on individuals, organizations, global, economic, environmental, and societal context

(H) a recognition of the need for, and an ability to engage in life-long learning and continuing professional development

(I) an ability to use current techniques, skills, and tools necessary for engineering and computing practice.

(J) An ability to apply mathematical foundations, algorithmic principles, and computer science theory in the modeling and design of computer-based systems in a way that demonstrates comprehension of the tradeoffs involved in design choices.

(K) an ability to apply design and development principles in the construction of software systems of varying complexity.

(L)  a knowledge of contemporary issues