**CENTRAL UNIVERSITY**

**SCHOOL OF APPLIED SCIENCES**

**DEPARTMENT OF COMPUTER SCIENCE & INFORMATION TECHNOLOGY**

**Course Code**: ITEC 307 **Credit Hour(s)**:**3 Course Title**: Operating Systems

**Course Lecturer:** *K. Sarkodie Obeng Kwakye, Ph.D* Computer Science Dept.



*obengsark@yahoo.com*  **0554636015**

**Office Hours:**  Time:

**Course Objective**

The objective of this course is to introduce students to the general understanding of modern computers’ structure, purpose, structure and functions of operating systems and illustration of key OS aspects by example.

Students will demonstrate a knowledge of process control, threads, concurrency, memory management scheduling, I/O and files, distributed systems, security, and networking.

**Course Description**

This course introduces the general principles of systems analysis and design, and places these activities within the context of the overall software development process.

This course will introduce you to modern operating systems. We will focus on Windows-based operating systems, though we will also learn about alternative operating systems, including Linux. The course will begin with an overview of the structure of modern operating systems. Over the course of the subsequent units, we will discuss the history of modern computers, analyze in detail each of the major components of an operating system (from processes to threads), and explore more advanced topics in the field, including memory management and file input/output. The class will conclude with a discussion of various system-related security issues.

**Learning Outcome**

On completion of this course the student should be able to accomplish the following:

* Understand and evaluate operating system implementations.
* Understand the implementation of fundamental OS structures, including
  + Processes, system calls, scheduling, virtual memory, and file systems

Students would also be able to understand the following:

* OS - Types
* OS - Services
* OS - Properties
* OS - Scheduling algorithms
* OS - Multi-threading
* OS - Memory Management
* OS - I/O Hardware
* OS - I/O Software
* OS - Security
* OS - Linux

**Course Delivery Methods**

To encourage active learning and the achievement of learning objectives, the course will be delivered through a group discussion, interactive teaching, class discussions, and guided practice throughout the semester. A course whatsapp group for sharing content, collaborative learning, and student-teacher engagement will supplement these methods. All students must subscribe to the class whatsapp group.

**Recommended Textbooks:**

Tomsho, G. (2016). *Guide to Operating Systems* (5th ed.). San Francisco, CA, Cengage Learning.

Tanenbaum, A. S. Bos, H. (2014). *Modern operating systems* (4th ed.). Boston, MA: Pearson Education.

Mohan , C. I. (2013). *Operating Systems.* Delhi, New Delhi, PHI Learning

Nemeth, E. Snyder G. [Hein](https://www.amazon.com/Trent-R-Hein/e/B001IGFJ5Q/ref=dp_byline_cont_book_3), T. R. (2017) *UNIX and Linux System Administration Handbook* (5th ed.). Boston, Addison-Wesley Professional Publication.

**Assessment / Evaluation Techniques**

There will be a class discussion and everyone is expected to participate.

Mid-semester test will be conducted at the end of October 2019.

The final project will be presented before the revision week.

Then three (3) hours internal closed book examination.

Evaluation is comprised of:

|  |  |
| --- | --- |
| Class Attendance | 10% |
| Group Participation | 10% |
| Mid-Semester Test | 20% |
| End of Semester Exam | 60% |
| **TOTAL** | **100%** |

**Grading Policy and Scheme**

Continuous Assessment (40%)

End-of-semester examination (60%)

Kindly refer to Central University Undergraduate Student Handbook available on school website for grading system, bases for incomplete grade, and bases of grade appeals.

**Examination / Academic Integrity Other Policies**

Please refer to the Central University Undergraduate Student Handbook available on the school website.

**Course contents and schedule**

|  |  |  |  |
| --- | --- | --- | --- |
| **Session** | **Topic** | **Concepts** | **Learner-centered Activities** |
|  | Introduction to Operating Systems | * Definition of OS * Examples of OS * History of OS * Functions OS |  |
|  | Functions of Operating Systems | * Memory Management * Processor Management * Device Management * File Management and Security * Other important activities / functions | Class discussion |
|  | Types of OS | * Batch operating system * Time-sharing operating systems * Distributed operating System * Network Operating System * Real Time operating System * Advantages of OS | Lectures |
|  | Operating Systems Services | * Program execution * I/O operations * File System manipulation * Communication * Error Detection * Resource Allocation * Protection | Class discussion |
|  | Processes and Threads Synchronization CPU Scheduling | Central Process Unit (CPU) scheduling deals with having more processes/threads than processors to handles those tasks, meaning how the CPU determines which jobs it is going to handle in what order. A good understanding of how a CPU scheduling algorithm works is essential to understanding how an Operating System works; a good algorithm will optimally allocate resources, allowing an efficient execution of all running programs. Finally, we will conclude the module with a discussion on some of the more common algorithms found in UNIX-based Operating Systems. |  |
|  | Deadlock | Deadlock is a paralyzing process state resulting from improper CPU scheduling, process management, and synchronization management. Although it cannot be guaranteed that deadlock may be avoided 100% of the time, it is important to know how to avoid the deadlocked state and how to recover from it once it has been achieved. we will talk about how to prevent and avoid deadlock. Finally, we will learn about deadlock detection, as well as methods for recovering from a deadlocked state. |  |
|  | MID-SEMESTER EXAMINATION | **Mid-Semester Assessment Task**  Multiple choice exams to assess students on what they have learned so far. | |
|  |  | Memory is the oil that keeps the computer running smoothly. It is present in various forms throughout the entire computer system. As software developers, it is absolutely essential to have a solid understanding of the role memory plays so that you are able to efficiently use memory in your programs, as well as understand what is going on "under the hood” should a problem arise. We will discuss the role of memory in an Operating System, first with an overview of the memory hierarchy and how memory and the OS interact with each other. Next, we will move on to discussing how memory is allocated for different purposes. Finally, we will discuss the two main topics regarding memory access: segmentation and paging. |  |
|  | Memory Management | * Process Address Space * Static vs Dynamic Loading * Virtual Memory |  |
|  | File systems | * I/O Hardware * I/O software * File systems |  |
|  | Security | * Authentication * One Time passwords * Program Threats * System Threats * Computer Security Classifications | * Class discussion |
|  | Linux | * Components of Linux System * Kernel Mode vs User Mode * Basic Features * Architecture | |
|  | Revision | | |
|  | End-of-semester examinations | | |
|  | End-of-semester examinations  Examination Break | | |