|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Debre Berhan University**  **BSc Degree in Information Technology** | | | | | |
| **Program** | **Information Technology** | | | | |
| **Course Code** | **ITec2022** | | | | |
| **Course Title:** | **Operating Systems** | | | | |
| **Degree Program** | **Information Technology** | | | | |
| **Module Name** | **Computer Systems** | | | | |
| **Module No.** | **02** | | | | |
| **ECTS Credits**  **(CP)** | **05** | | | | |
| **Contact Hours (per week)** | ***Lecture*** | ***Lab/Practical*** | ***Tutorial*** | ***Home Study*** | ***Total*** |
| 2 | 3 | 0 | 6 | 12 |
| **Target Group:** | 2nd Year Information Technology Students | | | | |
| **Year /Semester** | Year: II, Semester: II | | | | |
| **Pre-requisites** | None | | | | |
| **Status of the Course** | Core | | | | |
| **Course Objective** | The course introduces students to basics of operating system design principles and components, and their functions. It will also discuss memory management, processor management, process management and deadlocks, concurrency control, scheduling and dispatching, device management, file systems, file management, security and protection, and system performance evaluation. In addition, students will be introduced with different operating systems, and they will see their similarities and differences. | | | | |
| **Course Outline** | **Chapter 1: History and Overview**   * 1. Indicate some reasons for studying operating systems   2. Indicate some important topic areas such as function and design, concurrency, scheduling, dispatch, memory management, device management, file systems, security, and protection   3. Describe the purpose of an operating system   4. Indicate the meaning of an interrupt   5. Describe the meaning of concurrency and the reasons for its importance   6. Illustrate the manner in which scheduling and dispatch take place in a computer through its operating system   7. Describe the manner and importance of memory management   8. Describe the manner and importance of device management   9. Explore some additional resources associated with operating systems   10. Explain the purpose and role of operating systems in computer engineering   **Chapter 2: Design Principles**   * 1. Functionality of a typical operating system   2. Mechanisms to support client-server models, hand-held devices   3. Design issues (efficiency, robustness, flexibility, portability, security, compatibility)   4. Influences of security, networking, multimedia, windows   5. Structuring methods (monolithic, layered, modular, micro-kernel models)   6. Abstractions, processes, and resources   7. Concepts of application program interfaces (APIs) specific to operating systems   8. Applications needs and the evolution of hardware/software techniques   9. Device organization   10. Interrupts: methods and implementations   11. Concept of user/system state and protection, transition to kernel mode   **Chapter 3: Design Principles**   * 1. States and state diagrams   2. Structures (ready list, process control blocks, and so forth)   3. Dispatching and context switching   4. The role of interrupts   5. Concurrent execution: advantages and disadvantages   6. The “mutual exclusion” problem and some solutions   7. Deadlock: causes, conditions, prevention   8. Models and mechanisms (semaphores, monitors, condition variables, rendezvous)   9. Producer-consumer problems and synchronization   10. Multiprocessor issues (spin-locks, reentrancy)   **Chapter 4: Scheduling and Dispatch**   * 1. Preemptive and non-preemptive scheduling   2. Schedulers and policies   3. Processes and threads   4. Deadlines and real-time issues   **Chapter 5: Memory Management**   * 1. Review of physical memory and memory   2. Management Hardware   3. Overlays, swapping, and partitions   4. Paging and segmentation   5. Placement and replacement policies   6. Working sets and thrashing   7. Caching   **Chapter 6: Device Management**   * 1. Characteristics of serial and parallel devices   2. Abstracting device differences   3. Buffering strategies   4. Direct memory access   5. Recovery from failures   **Chapter 7: Security and Protection**   * 1. Overview of system security   2. Policy/mechanism separation   3. Security methods and devices   4. Protection, access, and authentication   5. Models of protection   6. Memory protection   7. Encryption   8. Recovery management   **Chapter 8: File Systems**   * 1. Files: data, metadata, operations, organization, buffering, sequential, non- sequential   2. Directories: contents and structure   3. File systems: partitioning, mount/unmount, and virtual file systems   4. Standard implementation techniques   5. Memory-mapped files   6. Special-purpose file systems   7. Naming, searching, access, backups | | | | |
| Software Requirements | UNIX/Linux, Windows Operation Systems | | | | |
| **Assessment** | Assignments………………….10%  Quiz ………………………… 5%  Test ………………………….20%  Project Work/ Lab Exam …….…. ….20 %  Final Examination…………45%  **Total ………………...…………………… 100%** | | | | |
| **Reference** | **Text**: Andrew S. Tanenbaum (1992) Modern Operating Systems. Prentice-Hall International Inc.   * Operating Systems, 4th edition by W. Stalling (editor) Prentice-Hall 2001 * Distributed Systems: Concepts and Design (3rd edition), George Coulouris, Jean * Dollimore and Tim Kindberg, Addison Wesley Longman, 2000. * Distributed Programming with Java, Qusay H. Mahmoud, Manning Publications Co., 2000. * Distributed Systems: Principles and Paradigms, Andrew S. Tanenbaum and Maarten van Steen, Prentice-Hall, 2002. | | | | |

|  |  |
| --- | --- |
| Course Title | Operating Systems and System Programming |
| Course Code | SEng2034 |
| CP | 7(3hr Lecture, 3hr Lab, 1 tutor) |
| Module Title | Hardware and System Programming |
| Module Number | 03 |
| Pre-requisites | None |
| Year | II |
| Semester | II |
| Status of Course | Compulsory |

Course Description

This course provides basic concepts of operating systems and system programming, targeting on types and architecture of operating system, major components of operating system and subsystems (Utility programs, multiple-program systems), Unix/Linux OS overview & architecture. Processes, inter-process communication, and synchronization. Memory allocation, segmentation, paging. Loading and linking, libraries. Resource allocation, scheduling, performance evaluation. File systems, storage devices, I/O systems. Protection, security, and privacy. The computer laboratory practice mainly focusing on system level programming based on operating system programming services and other APIs. But, not exclusively on issues related to operating system component design and implementation. Moreover, it includes the concept of process of scheduling, inter-process communication; threads; CPU scheduling, basic concepts, scheduling criteria, scheduling algorithms; process synchronization, the critical section problem, semaphores, monitors, classical synchronization problems; deadlocks, avoidance, prevention, detection; memory management, physical and virtual memory, swapping, allocation, paging, segmentation; file systems, access methods, directory structure, file sharing and protection; security, authentication, intrusion detection, encryption

Course Goals or Learning Outcomes

By the end of this course, students will be able to:

* Explain the objectives and functions of modern operating systems
* Describe the functions of a contemporary operating system with respect to convenience, efficiency, and the ability to evolve.
* Explain the different states that a task may pass through and the data structures needed to support the management of many tasks.
* Explain conditions that lead to deadlock.
* Compare and contrast the common algorithms used for both preemptive and non- preemptive scheduling of tasks in operating systems, such as priority, performance comparison, and fair-share schemes.
* Explain the concept of virtual memory and how it is realized in hardware and software
* Identify the basic components of an operating system, describe their purpose, and explain how they function.
* Know the fundamental system programming unix/linux cornerstones (system call, c library, and gcc compiler) and commands
* Employ practical and technical skills in programming for control process/inter-process communication, thread, signal handling, CPU scheduling, deadlock, memory, input/output, device driver, file, clock, Disks.
* Demonstrate capability in system software programming

Course contents

|  |  |
| --- | --- |
| Week/Date | TOPIC TO BE DISCUSSED |
| Week 1 | Overview  Operating System (OS) and System Programming Overview   * What is OS? * Types of OS and OS Architecture * Component of OS   + File Management   + Process management   + I/O Device Management   + Memory Management   + Network Management   + Secondary-Storage Management   + Security Management * Typical Unix/Linux System Overview   + Unix Architecture, files & directory, input/output   + Programs, Process, Error handle & user identification   + Unix standardization and Implementation |
| Week 2 - 5 | Processes and process management   * The process concepts * The threads concept   Process and Process Management(P&PM)   * Process Concept * Threads Concept   + Thread Identification   + Thread Creation   + Thread Termination   + Thread Synchronization * Inter-process communication * Signals * Process Scheduling * Deadlock * Implementation of P&PM   + Main function, process termination, command line arguments   + Environment list and variables,   + Some system calls (setjmp, longjmp, getrlimit, setrlimit, fork, vfork, exit, wait, and waitpid functions)   + Signal function, unreliable singals, interrupted system calls, kill, raise, alarm, pause, abort, system, sleep, Job-Control SIG |
| Week 6 -8 | Memory Management (MM)   * Review of physical memory and memory management hardware * Overlays, swapping and partitions * Paging and segmentation * Paging placement and replacement policies * Working sets and threshing * Catching * Implementation of MM   + Memory layout of C program   + Memory allocation * Some system calls (malloc, calloc, and realloc functions) |
| Week 9 -11 | Device Management (DM)   * Characteristics of serial and parallel devices * Abstracting device differences * Buffering Strategies * Direct Memory Access * Recovering from failures * Implementation of DM   + Standard I/O Library   + Streams and FILE Objects   + Standard input, output and error   + Opening, reading and writing a stream |
| Week 11-13 | File Management (FM)   * Files, File System, Structure and device drivers * File system techniques (partitioning, mounting and unmounting, virtual file systems) * Memory-mapped files * Special-purpose file systems * Naming, searching, and access * Implementation of FM   + File I/O system calls (like open, create, close, lseek, read, and write functions).   + Files, File Types and Directories (stat, fstat, and lstat) functions   + File Access permission/function, ownership of new files and directories, set user ID and set group ID * Link, unlink, remove, rename, mkdir, rmdir function and reading directories   File systems   * Fundamental concepts (data, metadata, operations, organization, buffering,   + Sequential vs. non-sequential files)   + Content and structure of directories * File system techniques (partitioning, mounting and un mounting, virtual file systems) * Memory-mapped files |
| Week 14 | Secondary-Storage management system (SSMS)   * Overview of Disk-storage * Storage allocation * Free-space management * Disk-scheduling * Implementation of SSMS   + FCFS   + SSTF   + SCAN and C-SCAN |
| Week 15 | Security and Protection (SM)   * Overview of system security * Policy/mechanism separation Security methods and devices Assets and their vulnerability * Protection * Intruders * Malicious and Trusted Software’s * Protection and Security design principles * The Unix/Linux Security Model * Protection, access, and authentication, Models of protection * Memory protection, Encryption * Recovery management |
| Week 16 | FINAL EXAM |

Summary of Teaching Learning Methods

The learning–teaching methodology will be student-centered with appropriate guidance of instructor/s during the students ‘activities. There will be Lecture, Demonstrations, Lab work Tutorials, Reading assignments and Group Discussions.

**Summary of Assessment Methods:** As per the academic regulation

References

* Andrew Tanenbaum, Modern Operating Systems, 2nd Edition, USA, Prentice Hall, 2001
* William Stallings, operating Systems: Internals and Design Principles,5th Edition, Prentice-Hall,2005