University of Engineering and Technology
School of Computer Science
Syllabus of Course
Academic Period 2018-II

1. **Code and Name:** CS2S01. Operating systems
2. **Credits:** 4
3. **Hours of theory and Lab:** 2 HT; 4 HL;
4. **Professor(s)**

   Meetings after coordination with the professor
5. **Bibliography**

[Avi12] Greg Gagne Avi Silberschatz Peter Baer Galvin. *Operating System Concepts, 9/E*. John Wiley & Sons, Inc., 2012. ISBN: 978-1-118-06333-0.

[Mat99] Luis Mateu. *Apuntes de Sistemas Operativos*. Universidad de Chile, 1999.

[Sta05] William Stallings. *Operating Systems: Internals and Design Principles, 5/E*. Prentice Hall, 2005. ISBN: 0-13-147954-7.

[Tan01] Andrew S. Tanenbaum. *Modern Operating Systems, 2/E*. Prentice Hall, 2001. ISBN: 0-13-031358-0.

[Tan06] Andrew S. Tanenbaum. *Operating Systems Design and Implementation, 3/E*. Prentice Hall, 2006. ISBN: 0-13-142938-8.

6. **Information about the course**

   (a) **Brief description about the course** An Operating System is a program that acts as an intermediary between the user and the machine.

   The purpose of an operating system is to provide an environment in which the user can run their applications.

   In this course the design of the core of the operating systems will be studied. In addition, the course includes practical activities in which problems of concurrency will be solved and the operation of a pseudo Operating System will be modified.

   (b) **Prerrequisites:** CS2201. Computer Architecture. (3^{rd} Sem)

   (c) **Type of Course:** Mandatory

   (d) **Modality:** Face to face

7. **Specific goals of the Course**

   - Know the basic elements of the design of the operating systems.

8. **Contribution to Outcomes**

   b) An ability to design and conduct experiments, as well as to analyze and interpret data. (**Assessment**)

   b) An ability to design and conduct experiments, as well as to analyze and interpret data. (**Assessment**)

9. **Competences (IEEE)**

   C1. An intellectual understanding and the ability to apply mathematical foundations and computer science theory.⇒ **Outcome b**

   C6. Ability to design and implement larger structural units that utilize algorithms and data structures and the interfaces through which these units communicate.⇒ **Outcome b**

CS8. Apply the principles of human-computer interaction to the evaluation and construction of a wide range of materials including user interfaces, web pages, multimedia systems and mobile systems..⇒ **Outcome b**

C1. An intellectual understanding and the ability to apply mathematical foundations and computer science theory.⇒ **Outcome b**

C6. Ability to design and implement larger structural units that utilize algorithms and data structures and the interfaces through which these units communicate.⇒ **Outcome b**

CS8. Apply the principles of human-computer interaction to the evaluation and construction of a wide range of materials including user interfaces, web pages, multimedia systems and mobile systems..⇒ **Outcome b**

10. List of topics

1. Overview of Operating Systems
2. Operating System Principles
3. Concurrency
4. Scheduling and Dispatch
5. Memory Management
6. Security and Protection
7. Virtual Machines
8. Device Management
9. File Systems
10. Real Time and Embedded Systems
11. Fault Tolerance
12. System Performance Evaluation

11. Methodology and Evaluation
Methodology:

Theory Sessions:
The theory sessions are held in master classes with activities including active learning and roleplay to allow students to internalize the concepts.

Lab Sessions:
In order to verify their competences, several activities including active learning and roleplay will be developed during lab sessions.

Oral Presentations:
Individual and team participation is encouraged to present their ideas, motivating them with additional points in the different stages of the course evaluation.

Reading:
Throughout the course different readings are provided, which are evaluated. The average of the notes in the readings is considered as the mark of a qualified practice. The use of the UTEC Online virtual campus allows each student to access the course information, and interact outside the classroom with the teacher and with the other students.
Evaluation System:

12. Content

Unit 1: Overview of Operating Systems (3)	
Competences Expected: C1	
Learning Outcomes	**Topics**
• Explain the objectives and functions of modern operating systems [Familiarity] • Analyze the tradeoffs inherent in operating system design [Assessment] • Describe the functions of a contemporary operating system with respect to convenience, efficiency, and the ability to evolve [Familiarity] • Discuss networked, client-server, distributed operating systems and how they differ from single user operating systems [Familiarity] • Identify potential threats to operating systems and the security features design to guard against them [Familiarity]	• Role and purpose of the operating system • Functionality of a typical operating system • Mechanisms to support client-server models, handheld devices • Design issues (efficiency, robustness, flexibility, portability, security, compatibility) • Influences of security, networking, multimedia, windowing systems
Readings : [Avi12], [Sta05], [Tan06], [Tan01], [Mat99]	

Unit 2: Operating System Principles (6)	
Competences Expected: C1	
Learning Outcomes	**Topics**
• Explain the concept of a logical layer [Familiarity] • Explain the benefits of building abstract layers in hierarchical fashion [Familiarity] • Describe the value of APIs and middleware [Familiarity] • Describe how computing resources are used by application software and managed by system software [Familiarity] • Contrast kernel and user mode in an operating system [Assessment] • Discuss the advantages and disadvantages of using interrupt processing [Familiarity] • Explain the use of a device list and driver I/O queue [Familiarity]	• Structuring methods (monolithic, layered, modular, micro-kernel models) • Abstractions, processes, and resources • Concepts of application program interfaces (APIs) • The evolution of hardware/software techniques and application needs • Device organization • Interrupts: methods and implementations • Concept of user/system state and protection, transition to kernel mode
Readings : [Avi12], [Sta05], [Tan06], [Tan01], [Mat99]	

Unit 3: Concurrency (9)	
Competences Expected: C6	
Learning Outcomes	**Topics**
• Describe the need for concurrency within the framework of an operating system [Familiarity] • Demonstrate the potential run-time problems arising from the concurrent operation of many separate tasks [Usage] • Summarize the range of mechanisms that can be employed at the operating system level to realize concurrent systems and describe the benefits of each [Familiarity] • Explain the different states that a task may pass through and the data structures needed to support the management of many tasks [Familiarity] • Summarize techniques for achieving synchronization in an operating system (eg, describe how to implement a semaphore using OS primitives) [Familiarity] • Describe reasons for using interrupts, dispatching, and context switching to support concurrency in an operating system [Familiarity] • Create state and transition diagrams for simple problem domains [Usage]	• States and state diagrams • Structures (ready list, process control blocks, and so forth) • Dispatching and context switching • The role of interrupts • Managing atomic access to OS objects • Implementing synchronization primitives • Multiprocessor issues (spin-locks, reentrancy)
Readings : [Avi12], [Sta05], [Tan06], [Tan01], [Mat99]	

Unit 4: Scheduling and Dispatch (6)	
Competences Expected: CS8	
Learning Outcomes	**Topics**
• 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 [Assessment] • Describe relationships between scheduling algorithms and application domains [Familiarity] • Discuss the types of processor scheduling such as short-term, medium-term, long-term, and I/O [Familiarity] • Describe the difference between processes and threads [Familiarity] • Compare and contrast static and dynamic approaches to real-time scheduling [Assessment] • Discuss the need for preemption and deadline scheduling [Familiarity] • Identify ways that the logic embodied in scheduling algorithms are applicable to other domains, such as disk I/O, network scheduling, project scheduling, and problems beyond computing [Familiarity]	• Preemptive and non-preemptive scheduling • Schedulers and policies • Processes and threads • Deadlines and real-time issues
Readings : [Avi12], [Sta05], [Tan06], [Tan01], [Mat99]	

Unit 5: Memory Management (6)	
Competences Expected: C1	
Learning Outcomes	**Topics**
• Explain memory hierarchy and cost-performance trade-offs [Familiarity] • Summarize the principles of virtual memory as applied to caching and paging [Familiarity] • Evaluate the trade-offs in terms of memory size (main memory, cache memory, auxiliary memory) and processor speed [Assessment] • Defend the different ways of allocating memory to tasks, citing the relative merits of each [Familiarity] • Describe the reason for and use of cache memory (performance and proximity, different dimension of how caches complicate isolation and VM abstraction) [Familiarity] • Discuss the concept of thrashing, both in terms of the reasons it occurs and the techniques used to recognize and manage the problem [Familiarity]	• Review of physical memory and memory management hardware • Working sets and thrashing • Caching
Readings : [Avi12], [Sta05], [Tan06], [Tan01], [Mat99]	

Unit 6: Security and Protection (6)	
Competences Expected: C1	
Learning Outcomes	**Topics**
• Articulate the need for protection and security in an OS [Familiarity] • Summarize the features and limitations of an operating system used to provide protection and security [Familiarity] • Explain the mechanisms available in an OS to control access to resources (cross reference IAS/Security Architecture and Systems Administration/Access Control/Configuring systems to operate securely as an IT system) [Familiarity] • Carry out simple system administration tasks according to a security policy, for example creating accounts, setting permissions, applying patches, and arranging for regular backups (cross reference IAS/Security Architecture and Systems Administration) [Familiarity]	• Overview of system security • Policy/mechanism separation • Security methods and devices • Protection, access control, and authentication • Backups
Readings : [Avi12], [Sta05], [Tan06], [Tan01], [Mat99]	

Unit 7: Virtual Machines (6)	
Competences Expected: CS8	
Learning Outcomes	**Topics**
• Explain the concept of virtual memory and how it is realized in hardware and software [Familiarity] • Differentiate emulation and isolation [Familiarity] • Evaluate virtualization trade-offs [Assessment] • Discuss hypervisors and the need for them in conjunction with different types of hypervisors [Familiarity]	• Types of virtualization (including Hardware/Software, OS, Server, Service, Network) • Paging and virtual memory • Virtual file systems • Hypervisors • Portable virtualization; emulation vs. isolation • Cost of virtualization
Readings : [Avi12], [Sta05], [Tan06], [Tan01], [Mat99]	

Unit 8: Device Management (6)	
Competences Expected: C6	
Learning Outcomes	**Topics**
• Explain the key difference between serial and parallel devices and identify the conditions in which each is appropriate [Familiarity] • Identify the relationship between the physical hardware and the virtual devices maintained by the operating system [Familiarity] • Explain buffering and describe strategies for implementing it [Familiarity] • Differentiate the mechanisms used in interfacing a range of devices (including hand-held devices, networks, multimedia) to a computer and explain the implications of these for the design of an operating system [Familiarity] • Describe the advantages and disadvantages of direct memory access and discuss the circumstances in which its use is warranted [Familiarity] • Identify the requirements for failure recovery [Familiarity] • Implement a simple device driver for a range of possible devices [Usage]	• Characteristics of serial and parallel devices • Abstracting device differences • Buffering strategies • Direct memory access • Recovery from failures
Readings : [Avi12], [Sta05], [Tan06], [Tan01], [Mat99]	

Unit 9: File Systems (6)	
Competences Expected: CS8	
Learning Outcomes	**Topics**
• Describe the choices to be made in designing file systems [Familiarity] • Compare and contrast different approaches to file organization, recognizing the strengths and weaknesses of each [Assessment] • Summarize how hardware developments have led to changes in the priorities for the design and the management of file systems [Familiarity] • Summarize the use of journaling and how log-structured file systems enhance fault tolerance [Familiarity]	• Files: data, metadata, operations, organization, buffering, sequential, nonsequential. • Directories: contents and structure. • File systems: partitioning, mount/unmount, virtual file systems. • Standard implementation techniques • Memory-mapped files • Special-purpose file systems. • Naming, searching, access, backups • Journaling and log-structured file systems
Readings : [Avi12], [Sta05], [Tan06], [Tan01], [Mat99]	

Unit 10: Real Time and Embedded Systems (6)	
Competences Expected: C1	
Learning Outcomes	**Topics**
• Describe what makes a system a real-time system [Familiarity] • Explain the presence of and describe the characteristics of latency in real-time systems [Familiarity] • Summarize special concerns that real-time systems present, including risk, and how these concerns are addressed [Familiarity]	• Process and task scheduling • Memory/disk management requirements in a real-time environment • Failures, risks, and recovery. • Special concerns in real-time systems
Readings : [Avi12], [Sta05], [Tan06], [Tan01], [Mat99]	

Unit 11: Fault Tolerance (3)	
Competences Expected: C1	
Learning Outcomes	**Topics**
• Explain the relevance of the terms fault tolerance, reliability, and availability [Familiarity] • Outline the range of methods for implementing fault tolerance in an operating system [Familiarity] • Explain how an operating system can continue functioning after a fault occurs [Familiarity]	• Fundamental concepts: reliable and available systems • Spatial and temporal redundancy • Methods used to implement fault tolerance • Examples of OS mechanisms for detection, recovery, restart to implement fault tolerance, use of these techniques for the OS's own services.
Readings : [Avi12], [Sta05], [Tan06], [Tan01], [Mat99]	

Unit 12: System Performance Evaluation (3)	
Competences Expected: C1	
Learning Outcomes	**Topics**
• Describe the performance measurements used to determine how a system performs [Familiarity] • Explain the main evaluation models used to evaluate a system [Familiarity]	• Why system performance needs to be evaluated? • What is to be evaluated? • Systems performance policies, e.g., caching, paging, scheduling, memory management, and security • Evaluation models: deterministic, analytic, simulation, or implementation-specific • How to collect evaluation data (profiling and tracing mechanisms)
Readings : [Avi12], [Sta05], [Tan06], [Tan01], [Mat99]	