**University of Nicosia, Cyprus**

| **Course Code** | **Course Title** | **ECTS Credits** |
|---|---|---|
| COMP-354 | Operating Systems | 6 |
| **Department** | **Semester** | **Prerequisites** |
| Computer Science | Spring | COMP-211, COMP-335 |
| **Type of Course** | **Field** | **Language of Instruction** |
| Required | Computer Science | English |
| **Level of Course** | **Year of Study** | **Lecturer(s)** |
| 1st Cycle | 3rd | Dr Harald Gjermundrød |
| **Mode of Delivery** | **Work Placement** | **Co-requisites** |
| Face-to-face | N/A | None |

**Objectives of the Course:**

The main objectives of the course are to:

- introduce Operating System structuring methods like monolithic, layered, modular, micro-kernel models
- provide deep knowledge of abstractions, processes, and resources
- make aware the concept of protection through the transition between user and system(kernel) mode
- thoroughly discuss OS structures like ready list, process control blocks, and so forth
- provide deep knowledge of the concept of processes and threads
- thoroughly discuss dispatching, context switching, preemptive, and non-preemptive scheduling
- cover in detail the "mutual exclusion" problem and some solutions
- provide knowledge of deadlock including: causes, conditions, and prevention
- provide knowledge of synchronization models and mechanisms (semaphores, monitors, condition variables, rendezvous)
- explain in detail physical memory, memory management hardware, paging, and virtual memory.

**Learning Outcomes:**

After completion of the course students are expected to be able to:

1. describe how computing resources are used by application software and managed by system software
2. compare and contrast the various ways of structuring an operating system such as object-oriented, modular, micro-kernel, and layered
3. contrast kernel and user mode in an operating system
4. describe the difference between processes and threads
5. compare and contrast the common algorithms used for both preemptive and non-preemptive scheduling of tasks in operating systems, such as priority, shortest job first, round robin, and multi-layer schemes
6. describe reasons for using interrupts, dispatching, and context switching to support concurrency in an operating system.
7. describe the need for concurrency within the framework of an operating

system
8. demonstrate the potential run-time problems arising from the concurrent operation of many separate tasks.
9. summarize the various approaches to solving the problem of mutual exclusion in an operating system.
10. explain memory hierarchy and cost-performance trade-offs.
11. explain the concept of virtual memory and how it is realized in hardware and software.

**Course Contents:**

1. Introduction: History of operating systems, computer-system organization and architecture, operating system structure and operation
2. Operating system structures: Operating system services and interfaces, system programs, operating system design and implementation
3. Processes: Process concepts, process scheduling, operations on processes, and cooperating processes
4. Threads: Motivation, user and kernel threads, multithreading models, thread scheduling
5. Processor (CPU) scheduling: Preemptive vs. non-preemptive, FIFO, SJF, Priority, RR, and multilevel queue
6. Process Synchronization: Producer-consumer problem, mutual exclusion, Peterson solution, lock-based solution, and semaphores
7. Deadlocks: System model, prevention, avoidance, detection, and recovery
8. Memory Management: Logical and physical address space, swapping, and memory allocation
9. Virtual Memory: Paging, segmentation, page replacement strategies
10. File System: Interface and Implementation.

**Learning Activities and Teaching Methods:**

Lectures, Practical Exercises, Projects and Assignments.

**Assessment Methods:**

Projects, Quizzes, Mid-term Exam, Final Exam.

**Required Textbooks/Reading:**

| Authors | Title | Publisher | Year | ISBN |
|---|---|---|---|---|
| A. Silberschatz, P. Gavin, and G. Gagne | Operating Systems Concepts, 8th Edition | Wiley | 2009 | 978-0-470-23399-3 |

**Recommended Textbooks/Reading:**

| Authors | Title | Publisher | Year | ISBN |
|---|---|---|---|---|
| Andrew S. Tanenbaum | Modern Operating Systems | Prentice Hall | 2007 | 0-136-00663-9 |