Department of Electrical Engineering

College of Engineering

Salahaddin University-Erbil

Subject: Operating Systems

Course Book – (Third Year C/C)

Lecturer's name: Asst. Prof. Dr. Diary R. Sulaiman

Goran Wnis Hama Ali (M.Sc.)

Academic Year: 2020/2021 (Semester 6)

Course Book

1. Course name	**Operating Systems**
2. Lecturers in charge	**Asst. Prof. Dr. Diary R. Sulaiman** **Goran Wnis Hama Ali (M.Sc)**
3. Department/ College	**Electrical Engineering / College of Engineering**
4. Contact	**e-mail: diary.sulaiman@su.edu.krd** **goran.hamaali@su.edu.krd**
5. Time (in hours) per week	**3 hrs**
6. Office hours	**15**
7. Course code	
8. Teacher's academic profile	**Asst. Prof. Dr. Diary R. Sulaiman** B.Sc. Salahaddin University-Erbil in 1990. M.Sc. Salahaddin University-Erbil in 2005. Ph.D. Salahaddin University-Erbil in 2018. **Goran Wnis Hama Ali (M.Sc.)** B.Sc. Salahaddin University-Erbil in 2011. M.Sc. Salahaddin University-Erbil in 2018.
9. Keywords	**OS, Processes, Deadlock, Synchronization, CPU Scheduling, Mutex Lock, Swapping and Mounting.**

10. Course overview:

This Course explains what operating systems are, what they do, and how they are designed and constructed. Then describes the process concept and concurrency as the heart of modern operating systems. After that the methods for process synchronization and deadlock handling will be explored. Finally, it deals with methods of; management of main memory during the execution of a process, management of mass storage, and how file systems and Virtual Machines are handled in a modern computer system.

11. Course objective:

1. To provide a thorough understanding of fundamental principles of operating system design
2. To illustrate the application of these principles by discussing their implementation in Windows UNIX, and/or other modern operating systems.
3. To develop an ability to analyse systems and to understand how to evaluate various options.
4. To give practical experience in the implementation of operating system theory so that the trade-offs between theory and practice can be better understood.
5. Make use of semaphores, monitors, and locks to synchronize concurrent processes and threads.
6. Learn algorithms and data structures for scheduling, memory management and file system organization.

12. Student's obligation

For the student to achieve a level of excellence in this subject, the following points should be

given utmost consideration: -

- Class attendance on regular basis for the purpose of learning. On administrative level if the student absence rate exceeded (15 %) of the total lecture hours, the student will be expelled.
- Active participation in class discussions.
- Reviewing the lecture notes and topics on weekly basis, noting the ambiguous points, if any, and requesting clarification during tutor office hours.
- Visiting the library on regular basis and checking the Internet for other approaches or simplifications of topics and ideas.
- Giving adequate and sufficient priority to preparing for weekly, monthly and final tests.

13. Forms of teaching

- By using one of the modern learning techniques which is using data show with power point slides which is clear, simple and helps students concentrate more. As well another advantage of using data show over the chalkboard or marker-board is that the notes can be pre-prepared which saves time, furthermore enables teachers and lecturers to build up banks of notes, diagrams, tables, use of animation, figures etc. that can be used over and over again. After showing and explaining power point slides, students start participating by asking questions, giving answers and discussing the subject.

14. Assessment scheme

In this system the maximum mark is (100%). The grading system is based on the summation of two categories of evaluations:

First, (40%) of the mark is based on the academic year effort of the student which includes

- One examination during the academic year = 30%, for which the study material is set for the topics reviewed in that particular exam.
- Assignments, Quizzes exams, Seminar presentation and active participation of the student in the classroom attendance, activities and discussions (10%).

Second, (60%) of the mark is based on final examination that is comprehensive for the whole of the study material reviewed during the semester and it usually occurs during the month of June.

At the end of the evaluation process, if the students could not secure a minimum of (50%), they are given a chance to repeat the final exam in June or July and they should be able by then to equal or exceed the (50%) limit otherwise they will have to repeat this subject during the next academic year if it did not contradict with the administrative regulations.

15. Student learning outcome:

The learning outcomes for this course (i.e., what you should be able to do at the end of the course) are as follows:

- Describe the main responsibilities of a contemporary operating system (OS) and to explain the history leading to their current form.
- List the most fundamental subsystems of an OS and the functions that each subsystem is responsible of.

- Describe how multitasking is possible to implement in a uniprocessor system, and how application programming differs between a multicore system and a uniprocessor one
- Provide an annotated state diagram that describes the states and state transitions during the whole lifetime of a process; likewise, interpret such a state transition diagram
- Identify and list application scenarios in which it is useful to use multiple threads of execution (including the fundamental need for multitasking in an OS).
- Explain the concept of a process and the process control block (PCB) in a typical OS; recognize a PCB upon seeing the C code of such, and assess whether such a data structure contains everything that is necessary to handle the main tasks of a modern OS
- Explain the typical (physical) computer memory hierarchy and the compromises involved in using such a hierarchy.
- Know what the principle of locality stands for, how it is used in a typical memory system, and how the principle can be used in applications other than computer technology and OSs.
- Explain the principles of paging virtual memory (VM) and describe the data structures and components (both hardware and software) that are necessary to implement paging VM
- Describe the layered structure of input/output (I/O) software and give a broad overview of I/O interrupt handling.
- Describe the phases required of hardware and software components for persistent storage and retrieval of persistent data.
- Understand the idea of direct memory access (DMA) with its advantages and implications.

16. Course Reading List and References:

1. Operating System Concepts, 10th edition, Avi Silberschatz et al, John Wiley & Sons, 2018.

2. Operating Systems: Internals and Design Principles, 9th edition, William Stallings, Prentice Hall, 2018.

17. The Topics:	Lecturer's name
Week 1: Introduction to Operating Systems (Computing Environment) Week 2: Operating System Structures Week 3: Processes (Concept, Scheduling and Operations) Week 4: Threads and Concurrency (Multicore Programming) Week 5:	Asst. Prof. Dr. Diary R. Sulaiman Goran Wnis Hama Ali (M.Sc.) 3 hours/week

Threads and Concurrency (Multithreading Model) Week 6: CPU Scheduling (Concepts, Criteria and Algorithms) Week 7: CPU Scheduling (Multiprocessor Scheduling and Real Time Scheduling) Week 8: Process Synchronization (Critical Section's Problem, Peterson's Solution, Mutex Locks and Semaphores). Week 9: Deadlock (Handling, Prevention, Avoidance and Detection), Recovery from Deadlock. Week 10: Main Memory (Background, Contiguous Memory Allocation, Paging and Swapping). Week 11: Virtual Memory (Background, Demand Paging, Copy-on-Write, Thrashing). Week 12: Mass Storage Management (HDD Scheduling, NVM(SSD) Scheduling, Error Detection and Correction) Week 13: I/O System Management and Kernel I/O Subsystem Week 14: File System Management (File Concept, Access Methods, Disk and Directory Structure, Mounting and Sharing). Week 15: Virtual Machines (Virtualization and Operating System Components)	

18. Practical Topics (if there is any).

1. Linux Main Commands and Basics of C language.
2. Process Creation and Termination.
3. Interprocess Communications (Pipes and Message Passing).
4. Suspending processes.
5. Scheduling (Short, Medium and Long Term).
6. Process Control Block (Linux and Windows API).
7. Working on Thread Libraries.
8. Pthread Scheduling API.
9. Mutex Locks.
10. Virtual OS (Linux on Windows and vice-versa).

19. Examinations:

A) What is the main advantage of the microkernel approach to system design? How do user programs and system services interact in a microkernel architecture? What are the disadvantages of using the microkernel approach?

B) How are iOS and Android similar? How are they different?

C) Design a program using ordinary pipes in which one process sends a string message to a second process, and the second process reverses the case of each character in the message and sends it back to the first process. For example, if the first process sends the message Hi There, the second process will return hI tHERE. This will require using two pipes, one for sending the original message from the first to the second process and the other for sending the modified message from the second to the first process. You can write this program using either UNIX or Windows pipes.

D) The two variables a and b have initial values of 1 and 2, respectively. The following code is for a Linux system:

Thread 1	Thread 2
`a = 3;`	—
`mb();`	—
`b = 4;`	`c = b;`
—	`rmb();`
—	`d = a;`

What possible errors are avoided by the use of the memory barriers?

20. Extra notes:

Null

21. Peer review

The course is approved by **Asst. Prof. Dr. Jalil Aziz Hamadamin**