Department of Software & Informatics Engineering

College of Engineering

Salahaddin University

Subject: Operating Systems

Course Book – Second Year

Lecturer's name: Mohammed Nasseh

Academic Year: First Semester- 2023/2024

Course Book

1. Course name	**Operating Systems**
2. Lecturer in charge	**Mohammed Nasseh**
3. Department/ College	**Software Engineering/College of Engineering**
4. Contact	**e-mail: mohammed.mohammed@su.edu.krd**
5. Time (Hours / Week)	**Theory: 2** **Practical: 2**
9. Keywords	**Process, CPU Scheduling, Deadlock, and Synchronization**

- Course Overview: general overview of the course
- Introduction to Operating System: Operating system definition and its main functions
- Operating-System Structure and Operations
- System Calls: what are system calls and how they are invoked
- Operating system Design: main designs and models of operating systems
- The Process Concept: what is process and how its represented in memory and in operating system
- Operation on processes: general overview on main operations of operating system on process including creating, terminating, suspending and resuming, scheduling, …
- Scheduling: what are schedulers, their types (long term and short term)
- CPU Scheduling: including CPU Scheduling criteria's, context switching, …
- FCFS Algorithm with examples and its advantages and disadvantages
- SJF Algorithm: both non-preemptive and preemptive(SRTF) schemes with examples and advantages and disadvantages
- Priority and RR Algorithms with examples, their advantages and disadvantages including starvation problem and aging solution
- Multiple-Processor Scheduling
- Deadlocks: what is a deadlock prevention method
- Deadlock Avoidance Algorithm including the definition of safe state and safe sequence
- Banker's Algorithm with examples
- Deadlock detection algorithms
- Deadlock recovery methods including process termination and pre-empting resources
- Process Synchronization: what is process synchronization

- Race condition: what is race condition with examples
- Critical Section Problem Peterson's Solution
- Synchronization hardware: including test and set instruction and swap instruction
- Semaphores: what is semaphore and how it could be used in synchronization
- Classic problems of synchronization: Bounded-Buffer Problem, Readers and Writers Problem and Dining-Philosophers Problem
- Monitors: synchronization mechanism

11. Course objective:

Operating systems are an essential part of any computer system. Similarly, a course on operating systems is an essential part of any computer-science education. An operating system is a program that manages the computer hardware. It also provides a basis for application programs. It acts as an intermediary between a user of a computer and the computer hardware for this reason it's very important to study this course. At the end of this course, students will be able to:

- Describe the purpose and function of operating systems.
- Identify the importance of operating systems in computer systems.
- Understand the fundamental of structure and architecture of operating systems.
- Install and gain basic of how operating system processes are scheduled.
- Understand the fundamental concepts of operating system processes.
- Write fundamental programs of operating system algorithms.

12. Student's obligation

Homework is normally given and unexpected quizzes provide an active way to keep the students active and more in touch with the subject. In addition quizzes, students attendance and their activity on the lectures will all collected together to form the 3% assessment of each semester.

In the laboratory there will be weekly programmes to be written by the students and the achievement of these programmes will be graded. We may have assignments and practical exams also.

13. Forms of teaching

Lectures:

power point presentations are used in addition to the pen and board which are mostly used to make a frequent step by step communication with the students

Practices:

In the lab the students deal with their computers and any explanation or clarification will be done by the projectors which is a dynamic tool for such needs.

14. Assessment scheme

The Grade is generated from the examination result(s) with the following weights (w):

- 50% Quizzes, Homework, Assignments and Activities
- 50% Final exam

Note: There will be random quizzes.

15. Student learning outcome:

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

1- Describe the purpose and function of operating systems.

2- Identify the importance of operating systems in computer systems.

3- Understand the fundamental of structure and architecture of operating systems.

4- Install and gain basic of how operating system processes are scheduled.

5- Understand the fundamental concepts of operating system processes.

6- Write fundamental programs of operating system algorithms.

16. Course Reading List and References:

▪ Key references: “Operating System Concepts”, 8th Edition. ABRAHAM SILBERSCHATZ, PETER BAER GALVIN and GREG GAGNE. JOHN WILEY & SONS. INC.

▪ Useful references: “Operating Systems Design and Implementation”, Third Edition. By Andrew S. Tanenbaum and Albert S. Woodhull. Prentice Hall.

17. The Topics:	Lecturer's name
1. Introduction to Operating System: Operating-System Structure and Operations 2. System Calls: what are system calls and how they are invoked 3. The Process Concept: what is process and how its represented in memory and in operating system 4. Operation on processes: general overview on main operations of operating system on process including creating, terminating, suspending and resuming, scheduling, … 5. Scheduling: what are schedulers, their types (long term and short term). CPU Scheduling: including CPU Scheduling criteria’s, context switching, …	Mohammed Nasseh

6. FCFS Algorithm with examples and its advantages and disadvantages. SJF Algorithm: both non-preemptive and preemptive(SRTF) schemes with examples and advantages and disadvantages 7. Priority and RR Algorithms with examples, their advantages and disadvantages including starvation problem and aging solution. Multiple-Processor Scheduling 8. Deadlocks: what is a deadlock prevention methods 9. Deadlock Avoidance Algorithm including the definition of safe state and safe sequence 10. Banker's Algorithm with examples 11. Deadlock detection algorithms 12. Deadlock recovery methods including process termination and preempting resources 13. Process Synchronization: what is process synchronization 14. Race condition: what is race condition with examples 15. Critical Section Problem Peterson's Solution 16. Semaphores: what is semaphore and how it could be used in synchronization 17. Classic problems of synchronization: Bounded-Buffer Problem, Readers and Writers Problem and Dining-Philosophers Problem	

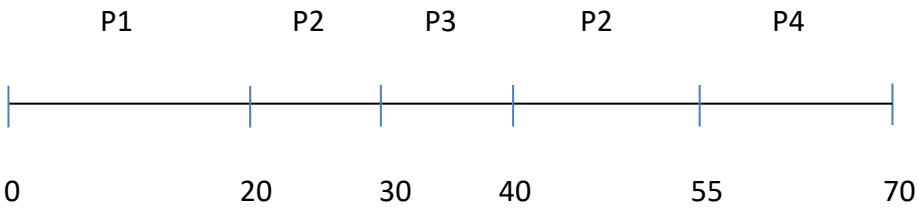
18. Examinations:

1. Compositional:

Example 1- Table given below shows the arrival time and execution time for different processes. Shortest Remaining Time First (SRTF) process scheduling algorithm is used by the operating system. What is the total waiting time and turnaround time for each process?

Process	Execution Time	Arrival Time
P1	20	0
P2	25	15
P3	10	30
P4	15	45

Solution 1-



Process	Waiting time	Turnaround time
P1	0	20
P2	15	40
P3	0	10
P4	10	25

Example 2- Consider a system that has three resources types A, B and C and are shared by three processes. There are 5 instances of each resources type. In the table shown below column Alloc denotes the number of instances of each resource type allocated to each process and the column Request denotes the number of instances of each resource type requested by a process in order to complete execution. From the processes P0, P1, P2 which one will finish last? Why?

	Alloc			Request		
	A	B	C	A	B	C
P0	1	2	1	1	0	3
P1	2	0	1	0	1	2
P2	2	2	1	1	2	0

Solution 2-

As there is 5 instances of each resource type so the availability vector initially is Available = [0 1 2]

By searching for a safe sequence:

Initially:

Work = available = [0 1 2]

Then we should search for a process that its request is smaller than or equal work and it is not finished

P1 is selected because it can continue as request1 = [0 1 2] = work

After finishing P1 work = work + Alloc1 = [0 1 2] + [2 0 1]

Work = [2 1 3]

Now P2 can't continue because request2 [1 2 0] is not smaller than or equal work [2 1 3] while P0 can as request0 [1 0 3] is smaller than work [2 1 3]

After finishing P0 work =work + Alloc0 = [2 1 3] + [1 2 1] = [3 3 4]

Now P2 can work So P2 will finish last

Example 3- The following pair of processes share a common variable X :

Process A

A0: int Y;

A1: Y = X*2;

A2: X = Y;

Process B

B0: int Z;

B1: Z = X+1;

B2: X = Z;

X is set to 5 before either process begins execution. How many different values of X are possible after both processes finish executing? Why?

Solution 3-

there are 4 possible values of X

Explanation: Here are the possible ways in which statements from A and B can be interleaved.

A1 A2 B1 B2: X = 11

A1 B1 A2 B2: X = 6

A1 B1 B2 A2: X = 10

B1 A1 B2 A2: X = 10

B1 A1 A2 B2: X = 6

B1 B2 A1 A2: X = 12

Example 4 -Describe the differences between fixed size and variable size message passing systems?

Solution 4-

If only fixed-sized messages can be sent, the system-level implementation is straightforward. This restriction, however, makes the task of programming more difficult. Variable-sized messages require a more complex system-level implementation, but the programming task becomes simpler.

2. True or false type of exams:

Example - Answer with yes or no? Explain why in both cases.

a. Round robin scheduling algorithm could result in starvation.
b. It is possible to have a deadlock involving only one process.
c. Internal fragmentation is decreased with small blocks.

d. First fit is the storage placement strategy in which a program is placed in the smallest available hole in the main memory.

Answer -

a. No, never because in round robin algorithm, the CPU time is equally divided between all the processes. For example if there are n processes in the ready queue, then each process gets 1/n of the CPU time.
b. No, It is impossible to have deadlock involving only one process because some necessary conditions will never hold like circular waiting
c. Yes, because internal fragmentation is happens when allocated memory (blocks) may be slightly larger than requested memory. So when the block size is small the remaining allocated memory (partition of a block) that is not in use will be small also.
d. No, because first fit is the storage placement strategy in which a program is placed in the first available hole in the main memory that is big enough. While best fit is the storage placement strategy in which a program is placed in the smallest available hole in the main memory.

3. Multiple choices:

Example – Select the correct answer :

1. A small page size causes large page tables
   a. No
   b. Sometimes
   c. Yes
   d. It depends on the system
2. In which of the storage placement strategies a program is placed in the largest available hole in the main memory?
   a. Best fit
   b. First fit
   c. Worst fit

d. None of the above

3. Banker's algorithm deals with:
   a. Deadlock prevention
   b. Deadlock detection
   c. Deadlock recovery
   d. Deadlock avoidance
   e. All the above
4. In Segmentation, logical address consists of ……. tuple.
   a. One
   b. Two
   c. Three
   d. None of the above
5. Job queue is :
   a. A set of all processes in the system
   b. A set of all processes residing in main memory, ready and waiting to execute
   c. A set of processes waiting for an I/O device
   d. None of the above

Answers -

1. c) yes
2. c) worst fit
3. d) deadlock avoidance
4. b) two
5. a) A set of all processes in the system

19. Extra notes:

20. Peer review **پێداچوونەوەی هاوەڵ**