DHANALAKSHMI SRINIVASAN ENGINEERING COLLEGE
(AUTONOMOUS)
(Approved by AICTE & Affiliated to Anna University, Chennai)
Re-Accredited by NAAC with 'A' Grade
Accredited by NBA for AERO, BME, CSE, ECE, EEE, IT & MECH.
PERAMBALUR-621212, TAMILNADU, INDIA.
Website: www.dsengg.ac.in

LABORATORY COURSE PLAN

COURSE – INFORMATION:

LAB COURSE TITLE	OPERATING SYSTEMS LABORATORY			
LAB COURSE CODE	U23ITP41			
LAB COURSE STRUCTURE	LECTURE	TUTORIAL	PRACTICAL	CREDIT
	0	0	4	2
REGULATION	BRANCH	YEAR	SEMESTER	ACADEMIC YEAR
2023	IT	II	IV	2025-2026
COURSE INCHARGE				

SYLLABUS

COURSE OBJECTIVE:

The main learning objective of this course is to prepare the students for:

1. To learn Unix commands and shell programming
2. To implement various CPU Scheduling Algorithms
3. To implement Process Creation and Inter Process Communication.
4. To implement Deadlock Avoidance and Deadlock Detection Algorithms
5. To implement Page Replacement Algorithms
6. To implement File Organization and File Allocation Strategies

LIST OF EXPERIMENTS

1. Basics of UNIX commands
2. Write programs using the following system calls of UNIX operating system fork, exec, getpid, exit, wait, close, stat, opendir, readdir
3. Write C programs to simulate UNIX commands like cp, ls, grep, etc.
4. Shell Programming
5. Write C programs to implement the various CPU Scheduling Algorithms
6. Implementation of Semaphores

7. Implementation of Shared memory and IPC
8. Bankers Algorithm for Deadlock Avoidance
9. Implementation of Deadlock Detection Algorithm
10. Write C program to implement Threading & Synchronization Applications
11. Implementation of the following Memory Allocation Methods for fixed partition
    a) First Fit b) Worst Fit c) Best Fit
12. Implementation of Paging Technique of Memory Management
13. Implementation of the following Page Replacement Algorithms
    a) FIFO b) LRU c) LFU
14. Implementation of the various File Organization Techniques
15. Implementation of the following File Allocation Strategies
    a) Sequential b) Indexed c) Linked

TEXT/REFERENCE BOOKS:

1. Abraham Silberschatz, Peter Baer Galvin and Greg Gagne, "Operating System Concepts", 10th Edition, John Wiley and Sons Inc., 2018.
2. Andrew S Tanenbaum, "Modern Operating Systems", Pearson, 5th Edition, 2022 New Delhi.
3. Achyut S.Godbole, Atul Kahate, "Operating Systems", McGraw Hill Education, 2016.

VIRTUAL LAB LINK:

https://naim30.github.io/OS-virtual-lab/
https://ebootathon.com/labs/beta/csit/OS/exp2/
https://erpjietuniverse.in/virtual_lab/Operating_system/labs/index.html
https://erpjietuniverse.in/virtual_lab/Operating_system/labs/exp8/index.html
https://www.classes.cs.uchicago.edu/archive/2020/fall/12100-1/labs/lab0/index.html

EXP. NO.	NAME OF THE EXPERIMENTS	NO. OF PERIODS	CUMULATIVE PERIODS
	CYCLE I		
1	Basics of UNIX commands	4	4
2	Write programs using the following system calls of UNIX operating system fork, exec, getpid, exit, wait, close, stat, opendir, readdir	4	8
3	Write C programs to simulate UNIX commands like cp, ls, grep, etc.	4	12
4	Shell Programming	4	16
5	Write C programs to implement the various CPU Scheduling Algorithms	4	20
6	Implementation of Semaphores	4	24
7	Implementation of Shared memory and IPC	4	28
8	Bankers Algorithm for Deadlock Avoidance	4	32

CYCLE II			
9	Implementation of Deadlock Detection Algorithm	4	36
10	Write C program to implement Threading & Synchronization Applications	4	40
11	Implementation of the following Memory Allocation Methods for fixed partition a) First Fit b) Worst Fit c) Best Fit	4	44
12	Implementation of Paging Technique of Memory Management	4	48
13	Implementation of the following Page Replacement Algorithms a) FIFO b) LRU c) LFU	4	52
14	Implementation of the various File Organization Techniques	4	56
15	Implementation of the following File Allocation Strategies a) Sequential b) Indexed c) Linked	4	60

COURSE OUTCOME

At the end of the course, the student should be able to:

CO1: Illustrate the various CPU scheduling algorithms.

CO2: Apply deadlock avoidance and detection algorithms.

CO3: Implement semaphore concepts.

CO4: Create processes and implement IPC.

CO5: Analyze the performance of the various Page Replacement Algorithms

CO6: Implement File Organization and File Allocation Strategies

CO-PO MAPPING:

CO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2
CO1	3	2	1	1	-	-	-	-	1	-	-	-	2	-
CO2	3	2	1	1	-	-	-	-	1	-	-	-	2	-
CO 3	3	2	1	1	-	-	-	-	1	-	-	-	2	-
CO 3	3	2	1	1	-	-	-	-	1	-	-	-	2	-
CO 4	3	2	1	1	-	-	-	-	1	-	-	-	2	-
CO 5	3	2	1	1	-	-	-	-	1	-	-	-	2	-
CO 6	3	2	1	1	-	-	-	-	1	-	-	-	2	-
AVG:	**3.00**	**2.00**	**1.00**	**1.00**	**-**	**-**	**-**	**-**	**1.00**	**-**	**-**	**-**	**2.00**	**-**

ADDITIONAL EXPERIMENTS		
EXP. NO.	**NAME OF THE EXPERIMENTS**	**IDENTIFIED RESOURCE LINK**
1	Shared Memory and Inter Process Communication.	https://opensource.com/article/19/4/interprocess communication-linux-storage
2	Program Illustrating I/O System Calls.	https://www.educative.io/answers/introduction-to system-calls
3	Install any guest operating system like Linux using VMware.	https://www.geeksforgeeks.org/linux-unix/installing linux-using-virtual-machine/
4	Illustrate the inter process communication strategy.	https://publicvoidlife.com/2014/12/18/c-program implement-inter-process-communication-ipc-system programming/
5	Program for IPC using shared memory	https://dextutor.com/program-for-ipc-using-shared memory/
6	Program to create Threads in Linux	https://dextutor.com/program-to-create-threads-in linux/
7	Program for SSTF Algorithm Program in C	https://dextutor.com/sstf-algorithm-program-in-c/

MODEL LAB DETAILS

BATCH	REGISTER NO.	MODE OF LAB CONDUCT	DATE	TIMING

LIST OF QUESTIONS

1. Write a C program to simulate the following non-preemptive CPU scheduling algorithms to find turnaround time and waiting time. a) FCFS b) SJF c) Round Robin (pre-emptive) d) Priority
2. Write a C program to simulate multi-level queue scheduling algorithm considering the following scenario. All the processes in the system are divided into two categories – system processes and user processes. System processes are to be given higher priority than user processes. Use FCFS scheduling for the processes in each queue.
3. Write a C program to simulate the following file allocation strategies.
   a) Sequential b) Indexed c) Linked
4. Write a C program to simulate the MVT and MFT memory management techniques.
5. Write a C program to simulate the following contiguous memory allocation techniques
   a) Worst-fit b) Best-fit c) First-fit
6. Write a C program to simulate paging technique of memory management.
7. Write a C program to simulate the following file organization techniques
   a) Single level directory b) Two level directory c) Hierarchical
8. Write a C program to simulate Bankers algorithm for the purpose of deadlock avoidance.
9. Write a C program to simulate disk scheduling algorithms
   a) FCFS b) SCAN c) C-SCAN
10. Write a C program to simulate page replacement algorithms
    a) FIFO b) LRU c) LFU
11. Write a C program to simulate page replacement algorithms a) Optimal
12. Write a C program to simulate producer-consumer problem using semaphores.
13. Write a C program to simulate the concept of Dining-Philosophers problem.
14. Write a Program for Search Pattern in File — grep Command
15. Write a C Program using getcwd() Show Present Working Directory — pwd
16. Program to implement simple command execution using `execvp()`.
17. Program to demonstrate `kill()` & signal handling (`signal()`, `sigaction()`).
18. Shell script to validate email/phone number using regex.
19. Script to display top 5 CPU-consuming processes.
20. Script to generate student report based on marks.
21. Write a program to Modify Banker's algorithm to handle dynamic resource requests.
22. Program for Simulation of unsafe state detection before allocation.
23. Graphical visualization of safe sequence generation.
24. Simulation of deadlock detection using cycle detection in graphs.
25. Task manager simulation showing deadlocked processes.
26. Detection + Recovery implementation (rollback, resource preemption).
27. Program to Implement **Next Fit** memory allocation.
28. Implement **Buddy Memory Allocation System**.
29. Visualization of memory blocks after each allocation & deallocation.
30. Implement multi-level index file organization.
31. Implement directory structure simulation (single, two-level).
32. Implement ISAM file organization.

VIVA QUESTIONS

1. What is the difference between an absolute path and a relative path in UNIX?
2. What is the purpose of `ls -l` command?
3. What does `chmod` do? Types of permissions?
4. How is `grep` different from `find`?
5. What is the use of `pipe (|)` in UNIX?
6. What is the difference between `more` and `less`?
7. What does `cat > file` do?
8. Difference between `rm`, `rmdir`, and `rm -r`.
9. What is the purpose of `head` and `tail` commands?
10. Explain hidden files in UNIX.
11. What is a system call?
12. How is it different from a function call?
13. What does the `fork()` system call do?
14. What is the difference between parent and child process?
15. What is returned by `fork()`?
16. What happens after `exec()` is executed?
17. Why is `wait()` used?
18. Difference between `exit()` and `_exit()`?
19. What does `getpid()` return?
20. What does the `stat()` system call fetch?
21. What is the purpose of `opendir()` and `readdir()`?
22. What happens when you call `close(fd)`?
23. What is a file descriptor?
24. What are the inputs required for the Banker's algorithm?
25. What does the **Allocation matrix** represent?
26. What does the **Maximum matrix** represent?
27. What does the **Need matrix** represent? How is it calculated?
28. What is the **Available vector**?
29. Why must Max ≥ Allocation always be true?
30. What does each entry in a page table contain?
31. What is a page table base register (PTBR)?
32. What is a page table length register (PTLR)?
33. What is the size of a page table?
34. How is page table stored in memory?
35. What is multi-level paging? Why is it needed?
36. What is hierarchical paging?
37. How are blocks stored in sequential allocation?
38. What information is stored in the file control block (FCB)?
39. What is the main advantage of sequential allocation?
40. What is the main drawback of sequential allocation?
41. Why does sequential allocation cause external fragmentation?
42. Can a file grow dynamically in sequential allocation? Explain.
43. What is a FAT (File Allocation Table)?
44. Why is FAT required in linked allocation?
45. Does linked allocation suffer from external fragmentation?

46. What is the disadvantage of linked allocation?
47. Why is random access difficult in linked allocation?
48. What is internal fragmentation in linked allocation?
49. What happens if a pointer in the chain is damaged?
50. In Indexed Allocation, how did you store index blocks?
51. What is the input to your program? (file name, block count, start block, etc.)
52. How do you check if a block is already allocated?
53. How did you represent free and allocated blocks?
54. How do you prevent block overlap in the simulation?
55. What output does your program generate? (allocation table, index block, links)
56. How does FCFS scheduling work?
57. Is FCFS preemptive or non-preemptive?
58. Why does FCFS suffer from convoy effect?
59. Does FCFS cause starvation?
60. What is the main disadvantage of FCFS?
61. What is SJF scheduling?
62. What is the difference between SJF and SRTF?
63. Does SJF guarantee minimum average waiting time? Why?
64. Why is SJF considered optimal?
65. What is the drawback of SJF?
66. Does SJF cause starvation? Explain.

Google Classroom Code **:** j7nwtwy
Google Classroom Name **: U23ITP41 - OS LAB**

Prepared By **Verified By**

Approved By

PRINCIPAL