

CSL 204 Operating Systems Lab - Program List

Jyothi Engineering College, Cheruthuruthy, Thrissur, Kerala

Department of Computer Science & Engineering

2021 - 2025 (A) Batch

February 2023 - July 2023

Source Code Download Link

Viva Voce Questions

Programs & Viva by Alwin Mathew, Alan Jose, Athul Murali, Adithya E, & Abhijith Suresh

GCC & GEANY Installation on Windows

Cycle 1

1. To study the basics of UNIX/**LINUX OS and its commands**.

2. Shell Scripts

1. Write a shell program to check whether the given number is odd or even.
2. Write a shell program to swap two integers.
3. Write a shell program to find the factorial of a number using for loop.
4. Write a shell program to find the sum on the first n numbers using a while loop.
5. Write a shell program to implement a basic calculator.
6. Write a shell script to convert temperature from Celsius to Fahrenheit.
7. Write a shell script to find the largest among the three numbers inputted.
8. Write a shell script to find the gcd of two numbers.
9. Write a shell script to generate the Fibonacci series up to n.
10. Write a shell script to check whether a given number is prime or not.
11. Write a shell script to find the average of n numbers.
12. Write a shell script to print given three alphabets in alphabetical order. (Input should be taken from the command line.)
13. Write a shell script to print the given three strings in dictionary order.
14. Write a shell script to perform the following task according to the user choice menu.
 1. Area of Circle
 2. Circumference of Circle
 3. Area of Rectangle
 4. Area of Square
15. Write a shell script to do the following in order:
 1. Check whether a directory by the name "bash" exist

2. If not create one.
 3. List all the directories.
-

Cycle 2**3. Programs on Processes, Threads, System Calls, and IPC**

1. Create a new process and print its process id and its parent's process id in C (use fork() and wait() system call).
2. Using C, create a new process and replace the child process with the "ps" process. (use execl() system call).
3. Write a program using fork() system call to create two children of the same process i.e., Parent P having child process P1 and P2.
4. Write a program to create a thread T1. The main process passes two numbers to T1. T1 calculates the sum of these numbers and returns the sum to the parent process for printing.
5. Write a program to create two Threads T1 and T2. Thread T1 creates a file named Thread.txt while T2 writes "Hello its T2" into the Thread.txt
6. Write a program to create a thread T1. The main thread reads an int number and checks whether it is prime or not. T1 calculates the factorial of the same number at the same time.
7. Write a program to create two Threads "Even" and "Odd". Thread "Even" calculates the sum of even numbers and Thread "Odd" calculates the sum of odd numbers. The main thread should display the even and odd numbers.
8. Write a C Program to extract File information using stat() system call.
9. Write a C program for implementing Directory management using the following system calls: opendir, readdir.
10. Write a C program that does the following(use open, read, write system call):
 1. Takes two filenames as argument.
 2. Creates a file with a name as the second argument(use command-line argument)
 3. Copies content from file_1 to file_2.
 4. First line of file_2 should be START and the last line should be STOP.
11. Write a C program to send a message from the parent process to the child process using pipe() system call.
12. Write a program where the first process sends a number to the second process and the second process calculates the factorial of that number(Use shared memory concept).

4. CPU Scheduling

1. Simulate FCFS cpu scheduling
2. Simulate SJF(non-preemptive) cpu scheduling
3. Simulate Round Robin CPU scheduling
4. Simulate Priority(non-preemptive) CPU scheduling

5. Process Synchronization using Semaphore

1. Write a program to achieve synchronization between multiple threads. Both threads make use of semaphore variables to update a shared variable so that only one of the threads is executing in its critical section(use semaphore functions).
2. Simulate Bounded-buffer (or Producer-Consumer) problem solution using Semaphore.
3. Simulate Reader's and Writers' problem solutions using Semaphores.
4. Simulate Dining-Philosophers problem solution using Semaphore.

6. Deadlock Avoidance

1. Simulate Banker's Algorithm

Cycle 3

7. Memory Allocation Methods

1. Write a C Program to implement the First fit Algorithm.
2. Write a C Program to implement the Best fit Algorithm.
3. Write a C Program to implement the Worst fit Algorithm.

8. Page Replacement Algorithms

1. Write a C Program to simulate the FIFO page replacement algorithm.
 2. Write a C Program to implement the LRU page replacement algorithm.
 3. Write a C Program to implement the Optimal page replacement algorithm.
-

Cycle 4

9. Disk Scheduling Algorithms

1. Write a C Program to simulate the FCFS disk scheduling algorithm.
2. Write a C Program to implement the SSTF disk scheduling algorithm.
3. Write a C Program to implement the SCAN disk scheduling algorithm.

10. File Allocation Techniques

1. Write a C Program to simulate the sequential file allocation technique.
2. Write a C Program to implement the linked file allocation technique.
3. Write a C Program to implement the indexed file allocation technique.

11. Build and Install the latest version of the Linux kernel.
