# SAVEETHA SCHOOL OF ENGINEERING

**SAVEETHA INSTITUTE OF MEDICAL AND TECHNICAL SCIENCES COMPUTER SCIENCE AND ENGINEERING PROGRAMME**

# CSA04 – OPERATING SYSTEMS

**LIST OF PROGRAMS (DAY 1)**

1. Create a new process by invoking the appropriate system call. Get the process identifier of the currently running process and its respective parent using system calls and display the same using a C program.
2. Identify the system calls to copy the content of one file to another and illustrate the same using a C program.
3. Design a CPU scheduling program with C using First Come First Served technique with the following considerations.
   1. All processes are activated at time 0.
   2. Assume that no process waits on I/O devices.
4. Construct a scheduling program with C that selects the waiting process with the smallest execution time to execute next.
5. Construct a scheduling program with C that selects the waiting process with the highest priority to execute next.
6. Construct a C program to implement pre-emptive priority scheduling algorithm.
7. Construct a C program to implement non-preemptive SJF algorithm.
8. Construct a C program to simulate Round Robin scheduling algorithm with C.

# SAVEETHA SCHOOL OF ENGINEERING

**SAVEETHA INSTITUTE OF MEDICAL AND TECHNICAL SCIENCES COMPUTER SCIENCE AND ENGINEERING PROGRAMME**

# CSA04 – OPERATING SYSTEMS

**LIST OF PROGRAMS (DAY 2)**

1. Illustrate the concept of inter-process communication using shared memory with a C program.
2. Illustrate the concept of inter-process communication using message queue with a C program.
3. Illustrate the concept of multithreading using a C program.
4. Design a C program to simulate the concept of Dining-Philosophers problem
5. Construct a C program for implementation the various memory allocation strategies.
6. Construct a C program to organize the file using single level directory.
7. Design a C program to organize the file using two level directory structure.
8. Develop a C program for implementing random access file for processing the employee details.

# SAVEETHA SCHOOL OF ENGINEERING

**SAVEETHA INSTITUTE OF MEDICAL AND TECHNICAL SCIENCES COMPUTER SCIENCE AND ENGINEERING PROGRAMME**

# CSA04 – OPERATING SYSTEMS

**LIST OF PROGRAMS (DAY 3)**

1. Illustrate the deadlock avoidance concept by simulating Banker’s algorithm with C. 18 Construct a C program to simulate producer-consumer problem using semaphores.
2. Design a C program to implement process synchronization using mutex locks.
3. Construct a C program to simulate Reader-Writer problem using Semaphores.
4. Develop a C program to implement worst fit algorithm of memory management.
5. Construct a C program to implement best fit algorithm of memory management.
6. Construct a C program to implement first fit algorithm of memory management.
7. Design a C program to demonstrate UNIX system calls for file management.

# SAVEETHA SCHOOL OF ENGINEERING

**SAVEETHA INSTITUTE OF MEDICAL AND TECHNICAL SCIENCES COMPUTER SCIENCE AND ENGINEERING PROGRAMME**

# CSA04 – OPERATING SYSTEMS

**LIST OF PROGRAMS (DAY 4)**

1. Construct a C program to implement the I/O system calls of UNIX (fcntl, seek, stat, opendir, readdir)
2. Construct a C program to implement the file management operations.
3. Develop a C program for simulating the function of ls UNIX Command.
4. Write a C program for simulation of GREP UNIX command
5. Write a C program to simulate the solution of Classical Process Synchronization Problem
6. Write C programs to demonstrate the following thread related concepts.

(i) create (ii) join (iii) equal (iv) exit

1. Construct a C program to simulate the First in First Out paging technique of memory management.
2. Construct a C program to simulate the Least Recently Used paging technique of memory management.

# SAVEETHA SCHOOL OF ENGINEERING

**SAVEETHA INSTITUTE OF MEDICAL AND TECHNICAL SCIENCES COMPUTER SCIENCE AND ENGINEERING PROGRAMME**

# CSA04 – OPERATING SYSTEMS

**LIST OF PROGRAMS (DAY 5)**

1. Construct a C program to simulate the optimal paging technique of memory management
2. Consider a file system where the records of the file are stored one after another both physically and logically. A record of the file can only be accessed by reading all the previous records. Design a C program to simulate the file allocation strategy.
3. Consider a file system that brings all the file pointers together into an index block. The ith entry in the index block points to the ith block of the file. Design a C program to simulate the file allocation strategy.
4. With linked allocation, each file is a linked list of disk blocks; the disk blocks may be scattered anywhere on the disk. The directory contains a pointer to the first and last blocks of the file. Each block contains a pointer to the next block. Design a C program to simulate the file allocation strategy.
5. Construct a C program to simulate the First Come First Served disk scheduling algorithm.
6. Design a C program to simulate SCAN disk scheduling algorithm.
7. Develop a C program to simulate C-SCAN disk scheduling algorithm.
8. Illustrate the various File Access Permission and different types users in Linux.