**Operating Systems Project List**

**(Spring-2022)**

The following list is project ideas that can be selected.

| [**S.NO**](http://s.no/) | **Project name** |
| --- | --- |
| **1** | **System Call for Semaphore (Example: Reader- Writer Problem)** |
| **2** | **System Call for Semaphore- Chain Smoker Problem** |
| **3** | **Ice Cream Factory Problem (System call)** |
| **4** | **Dining Philosophers Problem (System Call)** |
| **5** | **Spin Lock (System Call)** |
| **6** | **Sleeping Barber problem (system call)** |
| **7** | **Real-Time Scheduling on RaspberryPi/MCU32** |
| **8** | **Cache Cleaner (No scripting)** |
| **9** | **Voice-controlled Shell** |
| **10** | **Modification of System Call - Open() and Close()**  **/ must display with additional information** |
| **11** | **Preemptive Scheduler on kernel Threads** |
| **12** | **Implementation of A\* Search Algorithm with Threads** |
| **13** | **Comparison between Process And Threads**  **(Ex. merge, quicksort with process and**  **threads, any five algorithms)** |
| **14** | **Process Communication between sockets with GUI** |
| **15** | **Parallel Programming Comparison of sorting**  **Algorithms using Pthreads vs.**  **OpenMP vs. serial, (3 algorithms)** |
| **16** | **Unix Shell (Combining commands)** |
| **17** | **Lightweight Virtual Machine** |
| **18** | **Task Manager implementing at least 5 customized operations** |
| **19** | **Implementing Shared Files between two processes** |
| **20** | **Page replacement algorithm by kernel threads** |
| **21** | **Comparison between Process and Threads in Android**  **(Ex. merge, quicksort with**  **process and threads, any five algorithms)** |
| **22** | **Comparisons of IPC mechanisms on Android** |
| **23** | **Shell script for any sensor using RaspberryPi/MCU32** |
| **24** | **Parallel Programming in Android/iOS - Comparison of sorting**  **Algorithms using threads (3 Sorting algorithms)** |
| **25.** | **Hardware-based synchronization implementation using Semaphore** |

***Note: For system call projects, kernel configuration should be done with student ID.***

***Requirements:***

1. **Group Instructions:**
   1. Maximum 3 Group members
   2. Cross-section of the same teacher of the class is allowed ONLY
   3. Same project is not allowed within a section. and approval will be granted on an FCFS basis.
   4. Please note that there is only one project in the course. project demo and viva marks will be put in theory.
   5. Only C or C ++ language preferred. No python-based projects in OS.
2. **Deliverables and Due Dates:**
   1. Project Proposal (project title, introduction, methodology) submission on **14th March 2022**
   2. Final Demonstration with following requirements**:**
      1. Final Working Project Demo
      2. Project Viva
      3. Final Report soft copy (objectives, project details, results (comparison via graphs), conclusion.
      4. Code repository on GitHub

***Note: Final project demonstration is tentatively scheduled from* 16th May 2022**