

# **OS ASSIGNMENT-3**

## **GROUP-32**

**ARYAN VASHISHTHA & SAMAYRA MEENA**

### **OVERVIEW:**

This project implements a round-robin process scheduler with the assistance of a simple shell using C. The shell allows users to submit executable jobs into a ready queue, manage their execution, and then start the round-robin scheduler. The scheduler runs multiple jobs for a fixed time slice based on the available CPUs, ensuring each job gets executed in a round-robin fashion.

### **CODE WALKTHROUGH:**

The shell takes the input for nCPU & tSlice & submits the jobs for execution into the readyQueue. Once command start is entered, the scheduler is called & it starts running all the processes in the readyQueue in Round-Robin mode. nCPU processes run simultaneously for tSlice milliseconds & then the next nCPU processes are run in the same fashion. The scheduler checks if the process present exists or not. It stores the PID values of each process and also keep a track of the waiting and completion time of each process. It keeps a check if any process gets completed in between & neglects it for running again. In this way scheduler run all the processes, & store their waiting time & completion time. Once the shell is terminated using the Ctrl+c interrupt, the scheduler also terminates & it prints all the processes in the readyQueue, their PIDs, their waiting time & their completion time & the program is completely executed.

### **HOW TO USE?**

**COMPILATION:** gcc shell.c scheduler.c -o s

**RUNNING:** ./s

**SUBMIT:** pvtshell:~\$ submit ./fib

**START:** pvtshell:~\$ start

**DISPLAY OF HISTORY:** pvtshell:~\$ ^C (Interrupt)

## **CREDITS:**

**ARYAN**- implementation of the scheduler and round robin.

**SAMAYRA**- modifications in the shell to introduce the scheduler, error handling  
Debugging, testing and documentation were mutual.

**[GITHUB LINK](#)**