

OS ASSIGNMENT BY GROUP-71

GROUP MEMBERS CONTRIBUTION:

- 1) Ayush Singhal: Implemented shared memory part.
- 2) Sanjeet Kumar Patel: Implemented scheduler using round robin method

This assignment was a combined effort from both of us where we both have contributed to our best.

SUMMARY OF SIMPLE SCHEDULER IMPLEMENTATION

Header Inclusions: The code includes several standard C libraries for functionalities such as I/O, process management, synchronization, and shared memory management.

Constants:

- **MAX_QUEUE_SIZE:** Defines the maximum size of a process queue.
- **Queue:** A data structure representing a process queue, which includes an array of process IDs, front and rear indices, a count of processes in the queue, and a semaphore for mutual exclusion.

Helper Functions:

- **min(a, b):** A simple utility function that returns the minimum of two values.
- **dequeue(Queue* q):** Removes and returns the front process ID from the queue.
- **enqueue(Queue* q, pid_t to_enqueue):** Adds a process ID to the rear of the queue.

round robin Function:

- This is the core scheduling function that implements the round-robin scheduling algorithm.
- It takes as input a pointer to the `Queue`, the number of available CPUs (`ncpu`), and the time slice (`tslice`) for each process.
- Inside a continuous loop, it:
 - Selects up to `ncpu` processes from the queue, placing them in the `to_run` array.
 - Sends a `SIGCONT` signal to each selected process to start or resume its execution.
 - Sleeps for a fixed amount of time (`m`) to simulate time passing.
 - Sends a `SIGSTOP` signal to each running process, effectively suspending them.
 - Enqueues the suspended processes back into the queue.

- Repeats the loop until the program is terminated.

`main` Function:

- The `main` function is the entry point of the program.
- It expects two command-line arguments: the number of CPUs (`NCPU`) and the time slice (`TSlice`) for the scheduling algorithm.
- It opens a shared memory segment, retrieves the `Queue` structure from it using `mmap`, and initializes the `roundrobin` function with the specified parameters.

Overall, this code is a basic implementation of a round-robin scheduler that simulates the scheduling of processes on a multi-CPU system with the specified time slice. It uses shared memory and signals to manage processes and their scheduling.

GITHUB LINK:

[LINK](#)