## List of Commands

The following commands are working in my project.

| Command | Remarks |
|---|---|
| (Default) | This is default command. System will be asked to initialize the system. User will provide necessary information for the system like resources, memory, frame size etc. |
| create-process | This is used to create process for the system. User will create process and provide necessary information about the process. |
| ready-process | To transfer process from Job Queue to Ready Queue and Device Queue. |
| ready-IO-process | To transfer process form Device Queue to Ready Queue. |
| show-ready-queue | Print the process currently inside the Ready Queue. |
| show-job-queue | Print the process currently inside the Job Queue. |
| show-device-queue | Print the process currently inside the Device Queue. |
| Show-terminated-process | Print the recent Terminated(executed) process. |
| execute-process | Execute the Process currently inside the Ready Queue. |
| reexecute-process | Re-execute the process currently inside the Terminated Queue. |
| exit | Exit form running program. |

## Data Structure used

I have used the data structure, vector.

## CPU Scheduling Algorithms

The following CPU scheduling algorithms are implemented.

- FCFS
- SJF
- SRTF
- Priority Based Scheduling
- Preemptive Priority Scheduling
- Round Robin

## Banker's Algorithm for Deadlock Avoidance

This algorithm is implemented in my project.

**Memory Management**

I have considered memory management techniques in my project. There is enough free memory needed to transfer process from Job Queue to Ready Queue and Device Queue to Ready Queue, otherwise the process will remain their current position. Memory will be available after executing the process.