# Project Title

Implementation of CPU Scheduling Algorithms

## Description

An in-depth paragraph about your project and overview of use.

CPU scheduling is the process of deciding which process will own the CPU to use

while another process is suspended. The main function of CPU scheduling is to

ensure that whenever the CPU remains idle, the OS has at least selected one of

the processes available in the ready-to-use line. The project implements the

below mentioned CPU scheduling algorithms.

1) First Come First Serve(FCFS) scheduling algorithm

2) Shortest Job First(SJF) scheduling algorithm

3) Shortest Job First(SJF) Pre-emptive CPU scheduling algorithm

4) Round Robin scheduling algorithm

The project provides user with options wherein the user can select one of the

CPU scheduling algorithms to execute. Based on user selection, program takes

as input the processes and burst time and then based on the user inputs does

necessary computations to provide details of scheduling of the process based

on the burst time and algorithm under consideration.

## Getting Started

### Dependencies

Requires Free BSD 12.1

### Installing & Executing program

Steps:

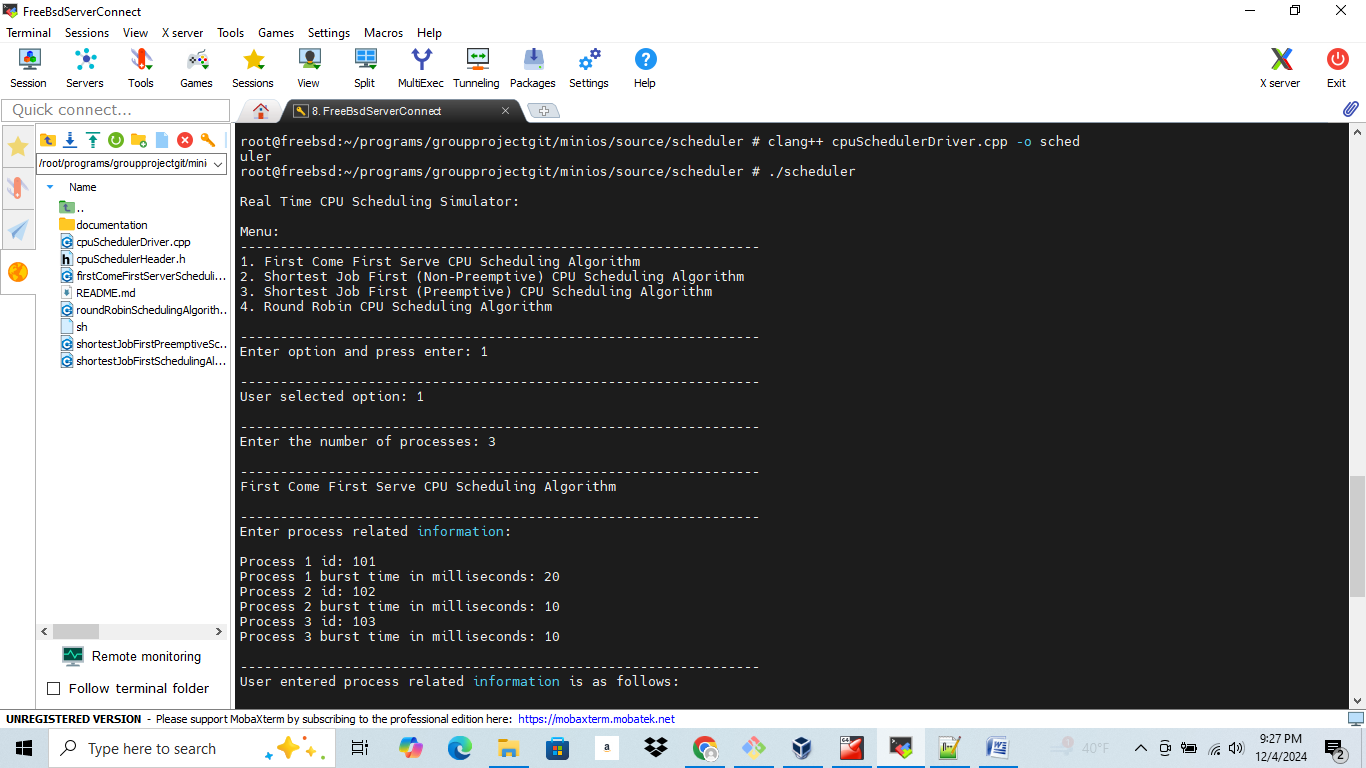
-Navigate to the folder scheduler or

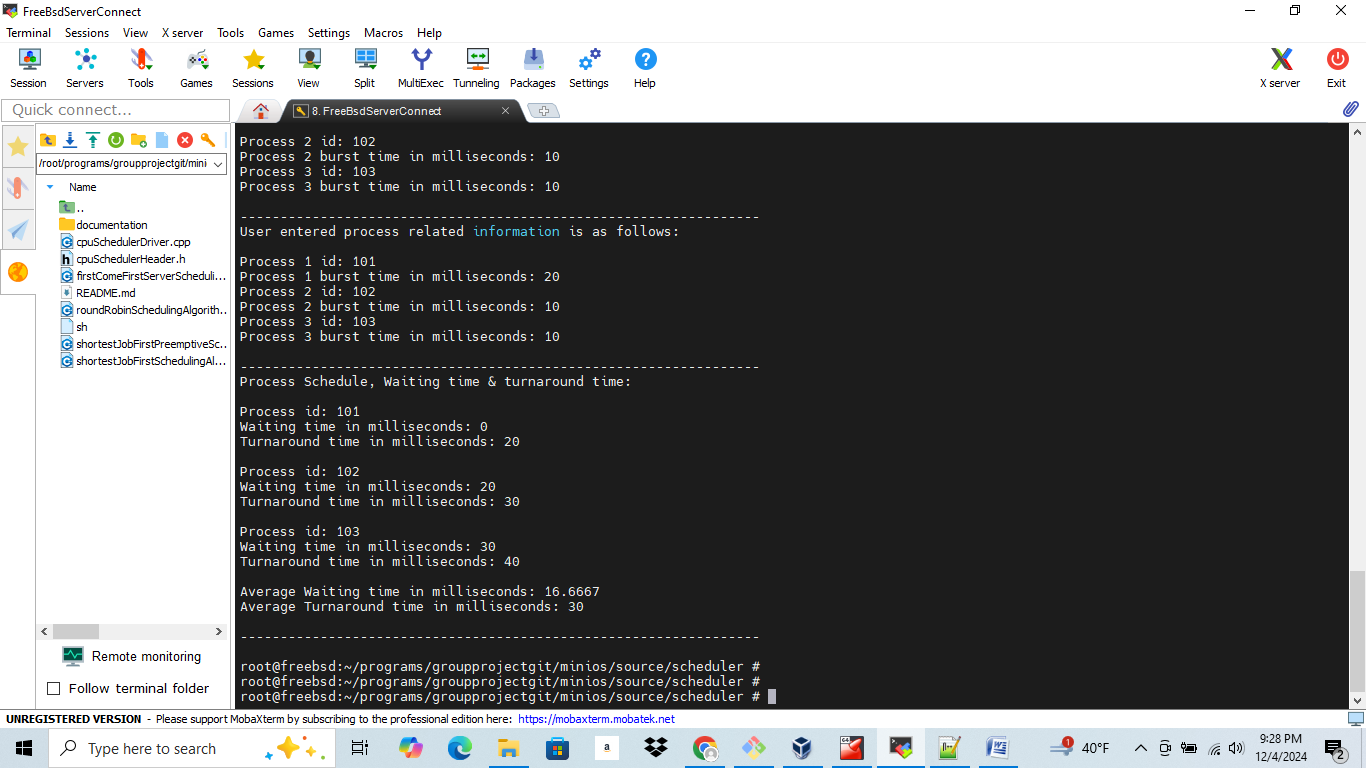
navigate to path minios/source/scheduler/

-Run the file cpuSchedulerDriver.cpp using command mentioned below:

> clang++ cpuSchedulerDriver.cpp -o scheduler

> ./scheduler

## Output Snapshots  
  
## First Come First Serve (FCFS) Scheduling Algorithm Snapshots:   
  




## Sample Output

root@freebsd:~/programs/groupprojectgit/minios/source/scheduler # clang++ cpuSchedulerDriver.cpp -o scheduler

root@freebsd:~/programs/groupprojectgit/minios/source/scheduler # ./scheduler

Real Time CPU Scheduling Simulator:

Menu:

-----------------------------------------------------------------

1. First Come First Serve CPU Scheduling Algorithm

2. Shortest Job First (Non-Preemptive) CPU Scheduling Algorithm

3. Shortest Job First (Preemptive) CPU Scheduling Algorithm

4. Round Robin CPU Scheduling Algorithm

-----------------------------------------------------------------

Enter option and press enter: 1

-----------------------------------------------------------------

User selected option: 1

-----------------------------------------------------------------

Enter the number of processes: 3

-----------------------------------------------------------------

First Come First Serve CPU Scheduling Algorithm

-----------------------------------------------------------------

Enter process related information:

Process 1 id: 101

Process 1 burst time in milliseconds: 20

Process 2 id: 102

Process 2 burst time in milliseconds: 10

Process 3 id: 103

Process 3 burst time in milliseconds: 10

-----------------------------------------------------------------

User entered process related information is as follows:

Process 1 id: 101

Process 1 burst time in milliseconds: 20

Process 2 id: 102

Process 2 burst time in milliseconds: 10

Process 3 id: 103

Process 3 burst time in milliseconds: 10

-----------------------------------------------------------------

Process Schedule, Waiting time & turnaround time:

Process id: 101

Waiting time in milliseconds: 0

Turnaround time in milliseconds: 20

Process id: 102

Waiting time in milliseconds: 20

Turnaround time in milliseconds: 30

Process id: 103

Waiting time in milliseconds: 30

Turnaround time in milliseconds: 40

Average Waiting time in milliseconds: 16.6667

Average Turnaround time in milliseconds: 30

## Version History

\* 1.0

\* Initial Release

\* Implementation of below mentioned CPU scheduling algorithms

\* below mentioned CPU scheduling algorithms.

\* 1) First Come First Serve(FCFS) scheduling algorithm

\* 2) Shortest Job First(SJF) scheduling algorithm

\* 3) Shortest Job First(SJF) Pre-emptive CPU scheduling algorithm

\* 4) Round Robin scheduling algorithm

## License

This project is licensed under the an open source/free software License.

## References

\* Abraham Silberschatz, Peter Galvin, Greg Gagne, “Operating System Concepts”, 2005, Seventh Edition, Chapter 5. CPU Scheduling, pp. 153-172.