COMP2035

G52OSC
Operating Systems & Concurrency Readme

Date : 7/12/2023
Group number : PGA17

Group Members :

Name	Student ID	OWA	Course Name
Chu Wen Xin Mabel	20512066	hcywc4	Computer science with AI
Tan Chui En	20510133	hcyct4	Computer Science
Hooi Wern Ning	20501309	hcywh4	Computer science with AI
Li Zhengyoule	20495589	hcyzl4	Computer Science

Process Scheduling Algorithms

The code includes implementations of three classic process scheduling algorithms. Users can choose between these algorithms to simulate the scheduling of processes based on their arrival times, burst times and time quantum (only for round robin and Multi-level Feedback Queue).

Getting Started

Follow these steps to run the program on your local machine.

Prerequisites

You need a C compiler installed on your system.
If you don't have one, you can install GCC (GNU Compiler Collection).
You can use Code::Blocks which supports many compilers, such as GNU GCC (MinGW and Cygwin).

Instructions

1) Open Code::Blocks:

2)Open the Code::Blocks IDE on your system.

3)Open the project file (Group(17)-OSC-CW-System-algorithms.c) using Code::Blocks.

4)Click on the "Build" menu.Choose "Build" or press F9 to compile the project.

5)After a successful build, click on the "Run" menu.Choose "Run" or press Ctrl + F10 to execute the program.

6)Choose a scheduling algorithm from the menu:
Enter 1 for FCFS.
Enter 2 for Round Robin.
Enter 3 for MLFQ.
Enter 0 to exit the program.

7)Follow the prompts to provide details about the processes:
-Number of processes.
-Arrival time and burst time for each process.
-Time quantum (for RR and MLFQ).

8)View the detailed output, including Gantt charts and performance metrics.

9)Repeat the process or choose another scheduling algorithm or exit by inputting 0.

Input Format

Number of Processes: Enter the total number of processes.

Process Details: For each process, enter the arrival time and burst time.

Time Quantum (for RR and MLFQ): Enter the time quantum for the Round Robin or MLFQ algorithms.

Output

The program generates detailed information for each process, including arrival time, burst time, completion time, turnaround time, and waiting time. It also displays Gantt charts showing the sequence of process execution. The average waiting time, average turnaround time, and throughput (processes completed per unit time) are provided.

Features

The ability to choose:
Users can choose the algorithm they want to simulate.

Detailed Gantt Charts:
Gantt charts for both Round Robin and MLFQ algorithms are generated

User-Friendly Interface:
prompts users to input essential details

Sorting and Indexing:
Order processes are sorted based on arrival time and process ID.

Dynamic Memory Allocation:
create an array of processes to ensure efficient memory usage and flexibility in handling varying workloads.

Average Metrics Calculation:
The project calculates and displays average waiting time, average turnaround time, and throughput for each scheduling algorithm.