CPU Scheduler

Project Overview

This project is a CPU Scheduling Simulator developed using C++ and Qt, designed to compare various CPU scheduling algorithms. The simulator allows users to input process details and visualize the performance of different scheduling algorithms, such as FCFS, SJF, SRTF, Round Robin, and Priority Scheduling (both preemptive and non-preemptive).

Features:

- Multiple Scheduling Algorithms: Supports FCFS, SJF (Preemptive and Non-Preemptive), SRTF, Round Robin, and Priority Scheduling (Preemptive and Non-Preemptive).
- User-Friendly Interface: The software has a Qt-based GUI which makes it easy to add processes, start scheduling, and view results.
- Performance Comparison: Allows for a visual comparison of average waiting and turnaround times across various scheduling algorithms.
- Expandable Design: Simple to add new scheduling algorithms or modify existing ones.

Project Structure

The project is structured into several key files and directories as outlined below:

```
cpu-scheduling/
├── CMakeLists.txt
└── Headers/
├── MainWindow.h
├── Process.h
└── source/
├── main.cpp
├── MainWindow.cpp
├── SchedulingAlgorithms.cpp
├── SchedulingSelector.cpp
├── ui_MainWindow.h
└── resources/
    └── MainWindow.ui
```

File Descriptions

- **CMakeLists.txt**: This is the configuration file for the CMake tool. It's responsible for managing the project build and instructing CMake on how to compile and link libraries to create the final executable.
- **main.cpp**: This is the entry point for the application. It includes the necessary code to initialise the Qt application framework and shows the main window to the user.

- **MainWindow.cpp**: This file implements the main window of the application. It contains the code for UI event handling, including button clicks and input field events.
- **MainWindow.h**: This is the header file for the main window class. It declares all the UI components like buttons, labels, text fields, etc. Additionally, it has slot declarations for handling different UI events.
- **Process.h**: This file defines the `Process` class. Each instance of this class represents an individual process with attributes like PID (Process ID), arrival time, burst time, and priority, which are necessary for process scheduling.

  ```
  struct Process {
      int pid;          // Process ID
      int arrival;      // Arrival time
      int burst;        // Burst time
      int remaining;    // Remaining burst time
      int priority;     // Priority
      int start = -1; // Start time
      int finish = -1;// Finish time
      int waiting = 0;// Waiting time
      int turnaround = 0; // Turnaround time
      int completion = 0; // Completion time
  ```

```
    Process(int p, int a, int b, int pr = 0)
        : pid(p), arrival(a), burst(b), remaining(b), prio
};
```

- **SchedulingAlgorithms.cpp**: This file contains the implementations of various CPU scheduling algorithms. Each algorithm decides the execution order of processes based on different criteria.
- **SchedulingSelector.cpp**: This file implements comparison and selection logic for different scheduling algorithms. It helps to determine which scheduling algorithm should be used based on specific conditions.
- **ui_MainWindow.h**: This is an auto-generated header file from the .ui file created in Qt Designer. It's used by the Qt framework to set up and manage the UI components.
- **resources/MainWindow.ui**: This is a Qt Designer file. It visually defines the layout of the main window UI, including the positioning and properties of all UI components.