Task Scheduler (Python Mini-Project)

https://github.com/adityakrmishra/DSA/tree/main/Task%20Scheduler%20using%20Priority%20Queue%20(Heap)

Overview

The **Task Scheduler** is a Python-based command-line application that manages tasks using a **priority queue (Min-Heap)**. It allows users to add, execute, and view tasks based on their priority levels.

Features

- ✅ Add tasks with a name and priority level (lower number = higher priority).
- ✅ Execute and remove the highest-priority task.
- ✅ View all pending tasks sorted by priority.
- ✅ Automatically log completed tasks to a file (`tasks_log.txt`).
- ✅ Simple and interactive CLI interface.

Folder Structure

```
task_scheduler/
├── main.py            # Entry point (CLI to interact with the user)
├── task.py            # Defines the Task class
├── scheduler.py       # Manages the priority queue and scheduling
├── utils.py           # Utility functions (e.g., logging tasks)
└── tasks_log.txt      # Stores completed tasks (generated automatically)
```

Installation & Setup

Prerequisites

- Python 3.x installed

Steps

1. **Clone the repository** (or create a new folder and add the provided files).
2. **Navigate to the project directory:**
3. `cd task_scheduler`
4. **Run the script:**
5. `python main.py`

Usage Guide

Once you run `main.py`, you will see a menu with the following options:

```
Task Scheduler Menu:
1. Add Task
2. Execute Task
3. Show Pending Tasks
4. Exit
Enter your choice:
```

1⃣ Add a Task

- Enter the task name and priority (lower value = higher priority).

2⃣ Execute a Task

- Removes and executes the task with the highest priority.
- Completed tasks are logged in `tasks_log.txt`.

3⃣ Show Pending Tasks

- Displays all tasks sorted by priority.

4⃣ Exit

- Exits the application.

Code Explanation

`task.py` (Task Class)

Defines the `Task` class with attributes:

- `name` (Task name)
- `priority` (Lower number = higher priority)
- `timestamp` (For FIFO order when priorities are the same)

`scheduler.py` (TaskScheduler Class)

Handles task management with:

- `add_task(name, priority)`: Adds a task to the queue.
- `execute_task()`: Removes and executes the highest-priority task.
- `show_tasks()`: Displays pending tasks.

`utils.py` (Logging Functionality)

Contains:

- `log_task(task)`: Saves completed tasks to `tasks_log.txt`.

Example Output

```
Task Scheduler Menu:
1. Add Task
2. Execute Task
3. Show Pending Tasks
4. Exit
Enter your choice: 1
Enter task name: Complete project
Enter priority (lower number = higher priority): 1
Added: Task(name=Complete project, priority=1)
```

Future Enhancements

- 🔹 Add a GUI using Tkinter or PyQt.
- 🔹 Implement database storage (SQLite).
- 🔹 Add task scheduling at specific times.

License

This project is open-source and free to use under the MIT License.

🎯 **Enjoy building and enhancing the Task Scheduler!** 🚀