

COMSATS UNIVERSITY ISLAMABAD

ATTOCK CAMPUS

Department Of Computer Science
Program: BS-SE

Assignment

Data Structure Lab

Submitted by

Shahan Waheed

Registration no

SP23-BSE-030

Submitted to

Mr Muhammad Kamran

Date

24-9-2024

Report on Simple Task Management System

1. Introduction

The Simple Task Management System is a basic program that helps users keep track of their tasks. It uses a linked list to store each task, making it easy to add, view, and remove tasks based on their importance. This system aims to help users manage their time better by organizing tasks in order of priority.

2. Objectives

The main goals of the Simple Task Management System are:

- To create an easy way for users to manage their tasks.
- To store tasks in a linked list, which allows for efficient adding and removing of tasks.
- To make sure that more important tasks are handled first.
- To learn about linked lists and object-oriented programming in C++.

3. Functionalities

The system offers the following features:

Add a New Task: Users can enter a unique task ID, a description, and a priority level for the task. The task will be placed in the correct spot in the list based on its priority.

View All Tasks: Users can see all the tasks in the list along with their details (ID, description, and priority).

Remove the Highest Priority Task: Users can remove the task with the highest priority from the list.

Remove a Task by ID: Users can remove a specific task by entering its unique ID.

Exit the System: Users can close the application.

4. System Design

The system consists of two main parts:

❖ Task Class

This class represents each task in the linked list.

It has the following parts:

TaskID: A unique number for each task (integer).

description: A brief explanation of the task (string).

priority: A number indicating how important the task is (integer).

next: A pointer that points to the next task in the list.

Constructor:

```
Task(int id, string desc, int prio);
```

❖ TaskList Class

This class manages the linked list of tasks.

It includes the following functions:

Constructor: Starts with an empty task list.

addTask: Adds a new task in the right place based on priority.

removeHighestPriorityTask: Removes the task with the highest priority.

removeTaskByID: Removes a specific task using its ID.

viewTasks: Shows all tasks in the list.

5. Code Explanation

➤ Task Class

Constructor: Sets up a new task with a unique ID, description, priority, and a pointer to the next task set to nullptr.

Member Variables: Stores the task details, making it easy to manage each task as a node in the linked list.

➤ TaskList Class

Constructor: Starts the task list with head set to nullptr, meaning the list is empty.

addTask Function:

Creates a new task node.

If the list is empty or the new task has a higher priority than the first task, it becomes the first task.

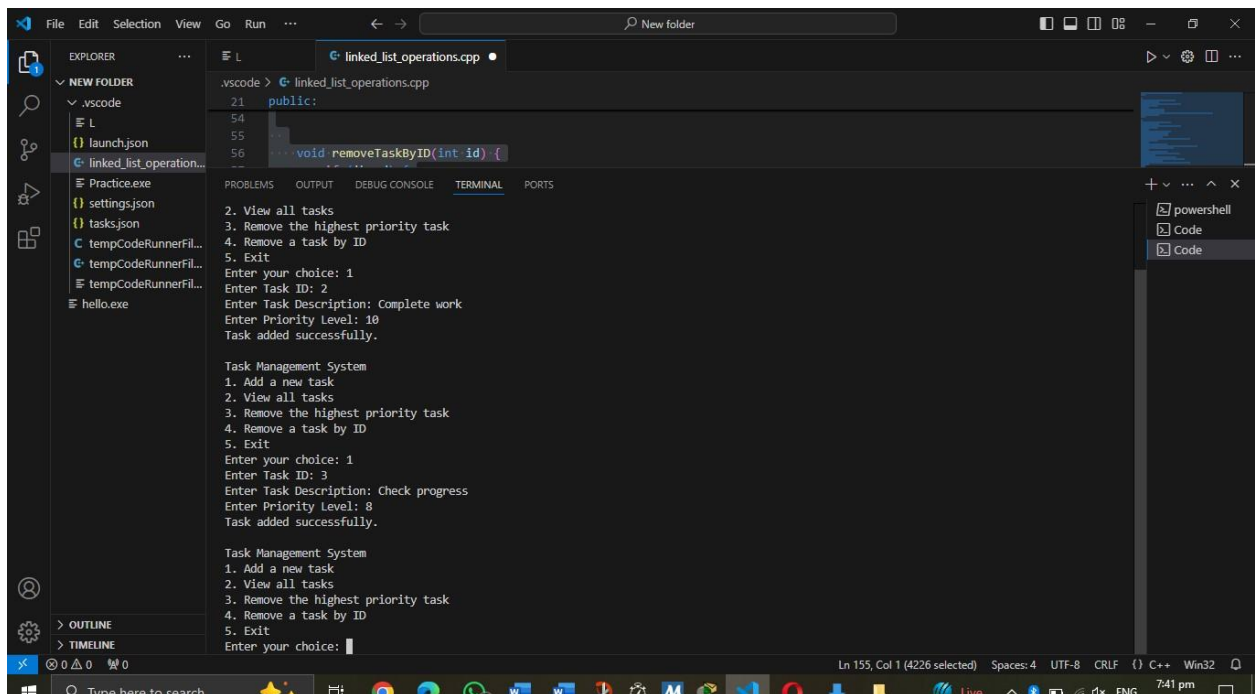
Otherwise, it finds the right place in the list for the new task based on priority.

removeHighestPriorityTask Function:

Checks if the list is empty. If not, it removes the first task (the one with the highest priority) and adjusts the head pointer.

6.Screenshots:

When I enter 1:



```
File Edit Selection View Go Run ...
New folder

EXPLORER
NEW FOLDER
.vscode
  L
  launch.json
  linked_list_operation...
Practice.exe
settings.json
tasks.json
tempCodeRunnerFil...
tempCodeRunnerFil...
tempCodeRunnerFil...
hello.exe

linked_list_operations.cpp
21 public:
54
55
56 void removeTaskByID(int id) {

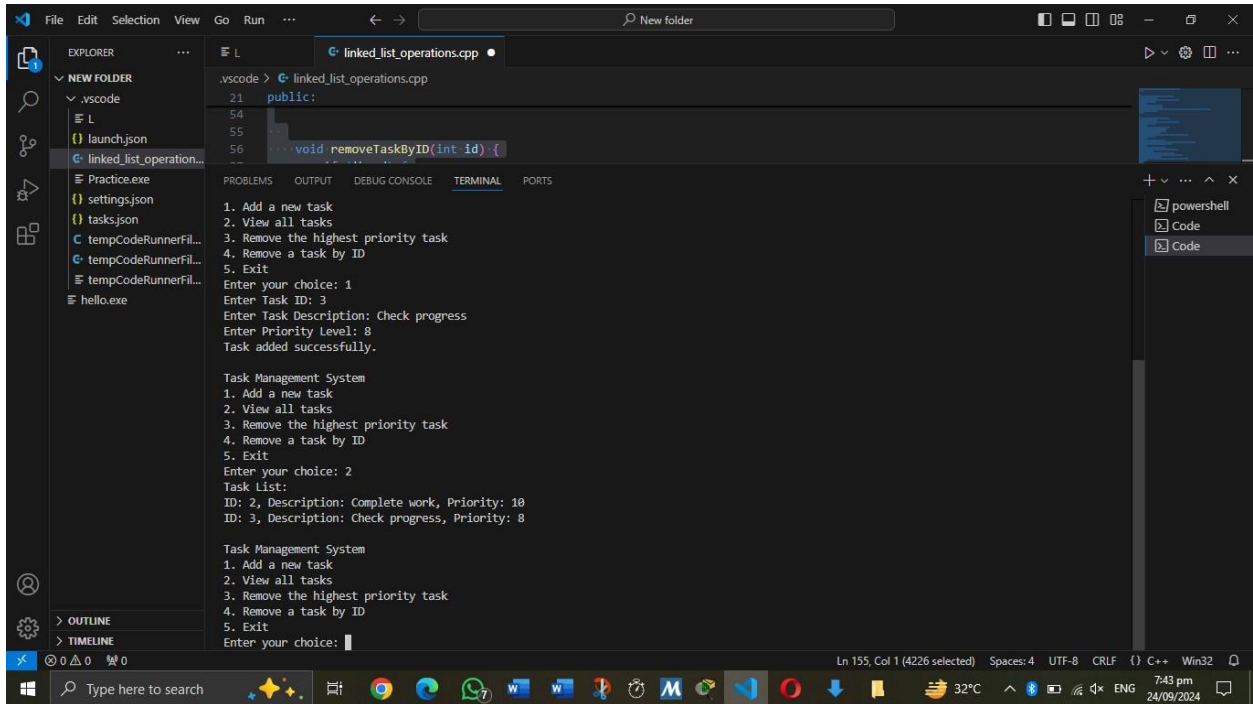
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
2. View all tasks
3. Remove the highest priority task
4. Remove a task by ID
5. Exit
Enter your choice: 1
Enter Task ID: 2
Enter Task Description: Complete work
Enter Priority Level: 10
Task added successfully.

Task Management System
1. Add a new task
2. View all tasks
3. Remove the highest priority task
4. Remove a task by ID
5. Exit
Enter your choice: 1
Enter Task ID: 3
Enter Task Description: Check progress
Enter Priority Level: 8
Task added successfully.

Task Management System
1. Add a new task
2. View all tasks
3. Remove the highest priority task
4. Remove a task by ID
5. Exit
Enter your choice:

Ln 155, Col 1 (4226 selected) Spaces: 4 UTF-8 CRLF {} C++ Win32
7:41 pm
```

When I enter 2:



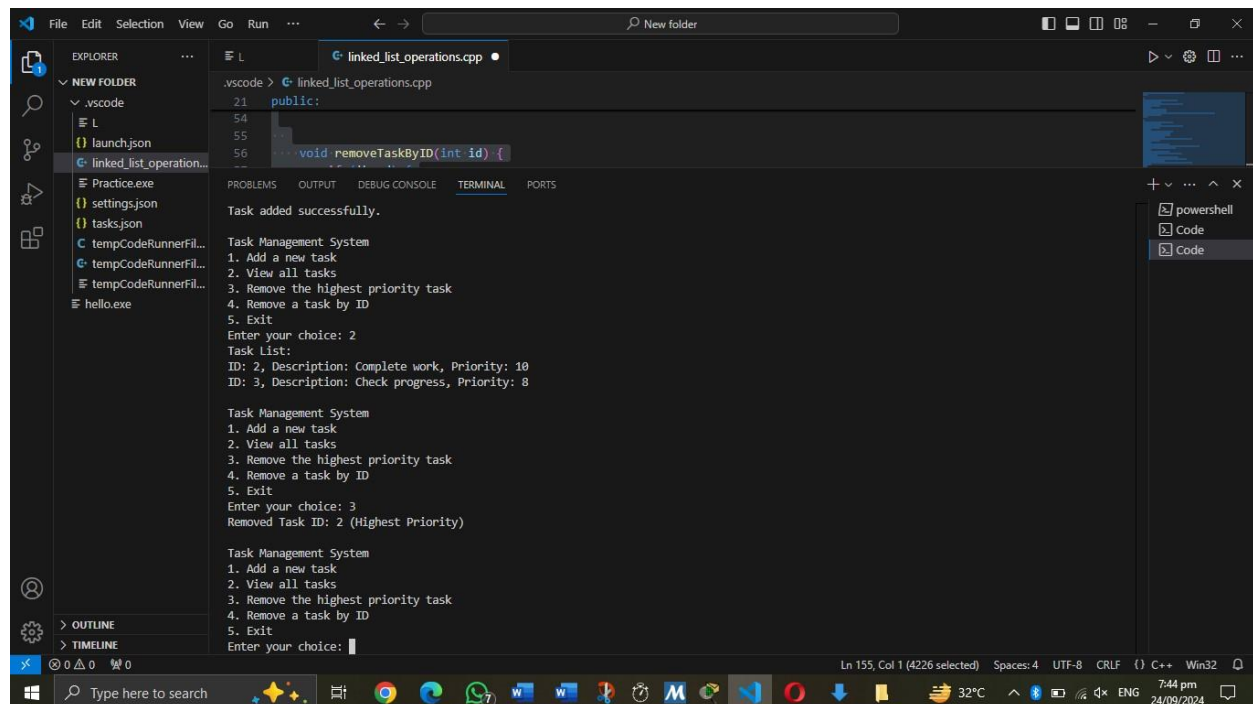
The screenshot shows the Visual Studio Code interface with a C++ project named 'linked_list_operations'. The terminal window displays the following output:

```
Task Management System
1. Add a new task
2. View all tasks
3. Remove the highest priority task
4. Remove a task by ID
5. Exit
Enter your choice: 1
Enter Task ID: 3
Enter Task Description: Check progress
Enter Priority Level: 8
Task added successfully.

Task Management System
1. Add a new task
2. View all tasks
3. Remove the highest priority task
4. Remove a task by ID
5. Exit
Enter your choice: 2
Task List:
ID: 2, Description: Complete work, Priority: 10
ID: 3, Description: Check progress, Priority: 8

Task Management System
1. Add a new task
2. View all tasks
3. Remove the highest priority task
4. Remove a task by ID
5. Exit
Enter your choice: 3
```

When I enter 3:



The screenshot shows the Visual Studio Code interface with the same C++ project. The terminal window displays the following output:

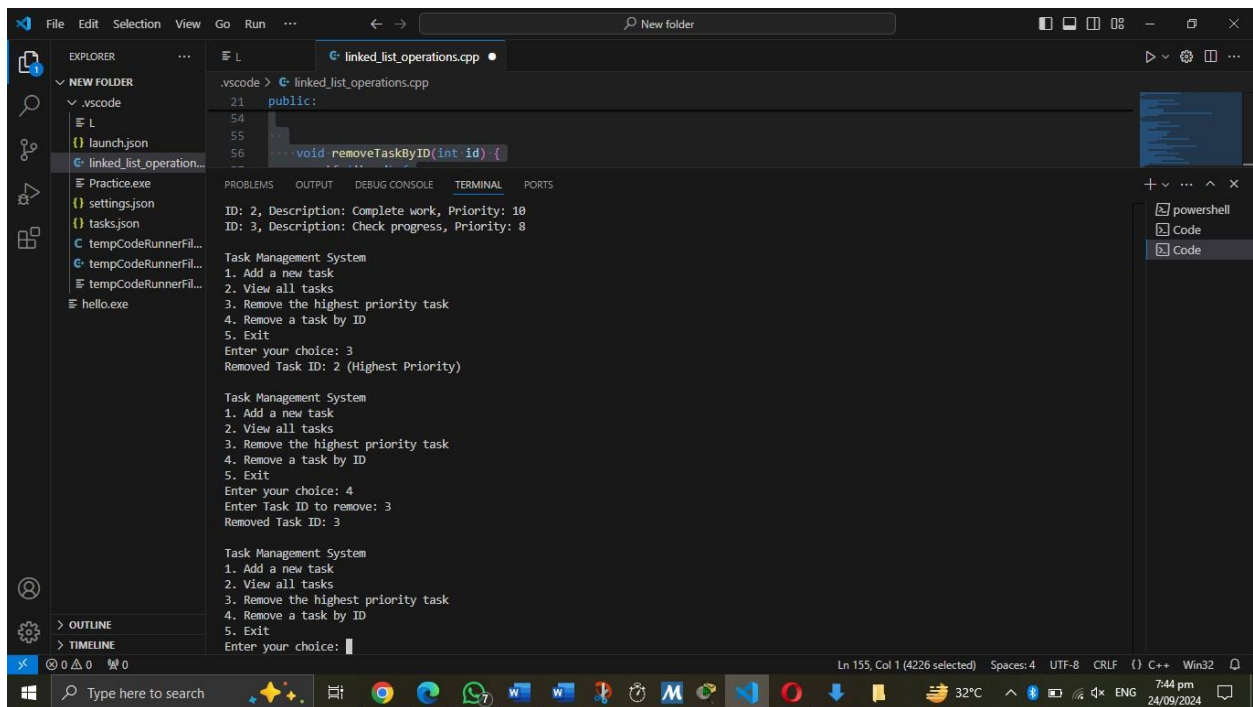
```
Task added successfully.

Task Management System
1. Add a new task
2. View all tasks
3. Remove the highest priority task
4. Remove a task by ID
5. Exit
Enter your choice: 2
Task List:
ID: 2, Description: Complete work, Priority: 10
ID: 3, Description: Check progress, Priority: 8

Task Management System
1. Add a new task
2. View all tasks
3. Remove the highest priority task
4. Remove a task by ID
5. Exit
Enter your choice: 3
Removed Task ID: 2 (Highest Priority)

Task Management System
1. Add a new task
2. View all tasks
3. Remove the highest priority task
4. Remove a task by ID
5. Exit
Enter your choice: 3
```

When I enter 4:



```
.vscode > linked_list_operations.cpp
21 public:
54
55
56 void removeTaskByID(int id) {

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

ID: 2, Description: Complete work, Priority: 10
ID: 3, Description: Check progress, Priority: 8

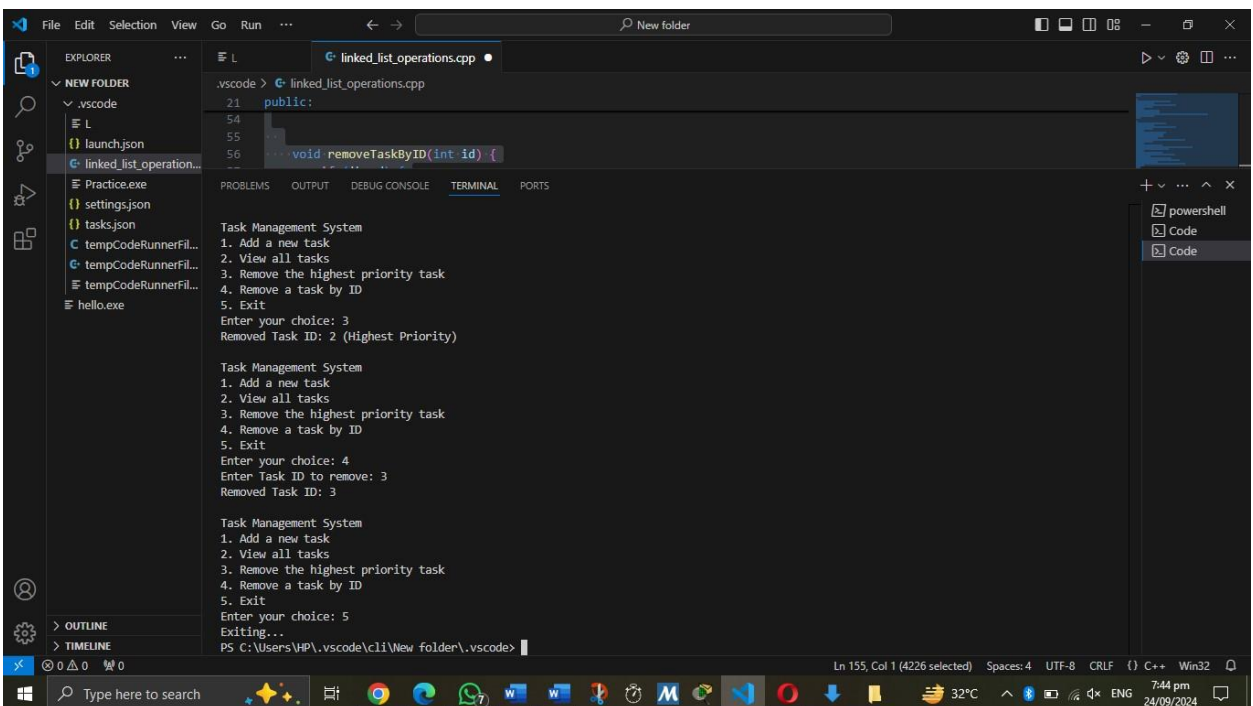
Task Management System
1. Add a new task
2. View all tasks
3. Remove the highest priority task
4. Remove a task by ID
5. Exit
Enter your choice: 3
Removed Task ID: 2 (Highest Priority)

Task Management System
1. Add a new task
2. View all tasks
3. Remove the highest priority task
4. Remove a task by ID
5. Exit
Enter your choice: 4
Enter Task ID to remove: 3
Removed Task ID: 3

Task Management System
1. Add a new task
2. View all tasks
3. Remove the highest priority task
4. Remove a task by ID
5. Exit
Enter your choice:

Ln 155, Col 1 (4226 selected) Spaces: 4 UTF-8 CRLF C++ Win32
Type here to search 32°C 7:44 pm 24/09/2024
```

When I enter 5:



```
.vscode > linked_list_operations.cpp
21 public:
54
55
56 void removeTaskByID(int id) {

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

Task Management System
1. Add a new task
2. View all tasks
3. Remove the highest priority task
4. Remove a task by ID
5. Exit
Enter your choice: 3
Removed Task ID: 2 (Highest Priority)

Task Management System
1. Add a new task
2. View all tasks
3. Remove the highest priority task
4. Remove a task by ID
5. Exit
Enter your choice: 4
Enter Task ID to remove: 3
Removed Task ID: 3

Task Management System
1. Add a new task
2. View all tasks
3. Remove the highest priority task
4. Remove a task by ID
5. Exit
Enter your choice: 5
Exiting...
PS C:\Users\HP\.vscode\cli\New folder\.vscode>

Ln 155, Col 1 (4226 selected) Spaces: 4 UTF-8 CRLF C++ Win32
Type here to search 32°C 7:44 pm 24/09/2024
```

7. Conclusion

Through this assignment, I learned how to use linked lists to manage a collection of data. I faced some challenges in making sure tasks were added and removed correctly based on their priority. Working on this project helped me improve my understanding of C++ programming and data structures.

8. Future Work

In the future, I could:

- Add the ability to view all tasks in the list.
- Include error checks for user inputs.
- Save and load tasks from a file.
- Create a graphical interface for a better user experience.

THE END