



**Comsats University Islamabad,
Attock Campus**

Department : CS

Name: Laraib Ayaz

Reg No.: SP23-BSE-025

Course: DS

Date: sep 24,2024

Assignment No.: 01

Submitted To: Mr. Kamran

Objective of the Assignment:

The objective of this assignment is to design and implement a simple Task Management System using a singly linked list data structure. The system should allow users to add new tasks, view all tasks, remove the highest priority task, and remove a task by its ID.

Operations Implemented:

The following operations are implemented in the Task Management System:

1. **Add Task:** This operation allows users to add a new task to the system. The task is added based on its priority, with higher priority tasks being added at the beginning of the list.
2. **View Tasks:** This operation displays all tasks in the system, along with their IDs, descriptions, and priorities.
3. **Remove Highest Priority Task:** This operation removes the task with the highest priority from the system.
4. **Remove Task by ID:** This operation removes a task from the system based on its ID.

Code Explanation:

Here's a brief explanation of the logic behind each function implemented in the code:

1. addTask Function:

- This function creates a new task node with the given ID, description, and priority.
- It checks if the list is empty or if the new task has a higher priority than the existing tasks.
- If the list is empty or the new task has a higher priority, it adds the new task at the beginning of the list.
- Otherwise, it traverses the list to find the correct position for the new task based on its priority.

2. viewTasks Function:

- This function checks if the list is empty.
- If the list is not empty, it traverses the list and displays the ID, description, and priority of each task.

3. removeHighestPriorityTask Function:

- This function checks if the list is empty.
- If the list is not empty, it removes the task at the beginning of the list (which has the highest priority).

4. removeTaskByID Function:

- This function checks if the list is empty.
- If the list is not empty, it traverses the list to find the task with the given ID.

- If the task is found, it removes the task from the list.

5. menu Function:

- This function displays a menu with options to add a new task, view all tasks, remove the highest priority task, remove a task by ID, and exit the system.
- It uses a switch statement to handle the user's choice and perform the corresponding operation.

Code 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: 1
Enter Task Description: abc
Enter Task Priority: 1
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: 2
Enter Task Description: def
Enter Task Priority: 2
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: def, Priority: 2
ID: 1, Description: abc, Priority: 1

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
Highest priority task removed 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: 4
Enter Task ID to remove: 2
Task with ID 2 not found.

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...

-----
Process exited after 121.4 seconds with return value 0
Press any key to continue . . .

```

Conclusion:

Through this assignment, I learned about the implementation of a singly linked list data structure and its application in a real-world scenario. I gained hands-on experience with C++ programming and improved my problem-solving skills.

One of the challenges I faced was ensuring that the tasks were added and removed correctly based on their priorities. I overcame this challenge by carefully implementing the logic for adding and removing tasks, and testing the code thoroughly to ensure that it worked as expected.

Overall, this assignment helped me develop a deeper understanding of data structures and algorithms, and I'm confident that I can apply this knowledge to more complex problems in the future.