



COMSATS University Islamabad, Attock

Campus Department of computer Science

Program: BSE

Course: DS

Assignment No.: 01

Registration No.: SP23-BSE-005

Name: Huzaiifa Farooq

Date: 23rd Sep, 2024

Submitted to: Sir Kamran

OBJECTIVE

The objective of this assignment is to implement a task management system using a singly linked list in C++. Each task is stored as a node containing a task ID, description, and priority level. The system allows users to manage tasks by adding them in priority order, viewing all tasks, and removing tasks based on priority or task ID.

OPERATIONS

1. **Add a New Task:** Inserts a task into the linked list at the appropriate position based on its priority level, ensuring higher priority tasks come first.
2. **View All Tasks:** Displays all tasks currently in the list, showing their task ID, description, and priority.
3. **Remove the Task with the Highest Priority:** Removes the task at the start of the list, which has the highest priority.
4. **Remove a Task by ID:** Finds and removes a task from the list using its unique task ID.

CODE EXPLANATION

Insert a new task (insert_task)

- **Purpose:** Adds a new task to the list in the correct position based on its priority. Higher priority tasks are placed before lower priority ones.
- **Logic:**
 - First, the function checks if a task with the given task_id already exists using the task_exists function. If it does, the task won't be added.
 - If the list is empty, the new task becomes the head of the list.
 - If the new task has a higher priority than the head, it is inserted at the start of the list.
 - Otherwise, the function traverses the list until it finds the correct position where the new task's priority is higher than the subsequent nodes. It then inserts the new task in this position.

```

Task Management System Menu:
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: This is task 1
Enter task priority: 8

Task Management System Menu:
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: This is task with same ID
Enter task priority: 5
Task already exists with ID: 1

Task Management System Menu:
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: this is new task

```

```

Enter task priority: 100

Task Management System Menu:
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
-----
2: this is new task - 100
1: This is task 1 - 8
-----

```

Print all tasks (print_tasks)

- **Purpose:** Displays all tasks currently in the list, showing the task_id, description, and priority for each task.
- **Logic:**
 - The function starts at the head of the list and iterates through each node, printing the task information until the end of the list is reached (current is NULL).

```

Enter your choice: 2
-----
2: this is new task - 100
3: this is task with id 3 - 9
1: This is task 1 - 8
-----

```

Delete the task with the highest priority (delete_from_start)

- **Purpose:** Removes the task with the highest priority, which is always the task at the head of the list.

➤ **Logic:**

- If the list is empty, the function outputs a message saying the task list is empty.
- Otherwise, it deletes the head node and updates the head pointer to the next node in the list.

```
-----
2: this is new task - 100
5: new task 2 - 12
3: this is task with id 3 - 9
1: This is task 1 - 8
4: new task - 4
-----

Task Management System Menu:
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

Task Management System Menu:
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
-----
5: new task 2 - 12
3: this is task with id 3 - 9
1: This is task 1 - 8
4: new task - 4
-----
```

Check if a task exists (task_exists)

➤ **Purpose:** Checks whether a task with a specific task_id already exists in the list.

➤ **Logic:**

- The function iterates through the list, comparing each node's task_id with the given tid. If a match is found, it returns true.
- If no match is found, it returns false.

```
Task Management System Menu:
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: This is task 1
Enter task priority: 8

Task Management System Menu:
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: This is task with same ID
Enter task priority: 5
Task already exists with ID: 1
```

Delete a task by id (delete_task_using_id)

- **Purpose:** Deletes a specific task from the list based on its task_id.
- **Logic:**
 - If the list is empty, the function outputs a message indicating that there is nothing to delete.
 - If the task to delete is the head of the list, the head is updated to the next node, and the old head is deleted.
 - If the task is located somewhere else in the list, the function iterates through the list until it finds the node before the target task. It then deletes the required node while retaining the order of tasks list.
 - If the task is not found, it outputs a message saying the task with the given ID does not exist.

```

-----
5: new task 2 - 12
3: this is task with id 3 - 9
1: This is task 1 - 8
4: new task - 4
-----

Task Management System Menu:
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 the task ID to delete: 5

Task Management System Menu:
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
-----
3: this is task with id 3 - 9
1: This is task 1 - 8
4: new task - 4
-----

```

CONCLUSION

Through this assignment, I learned how to effectively use singly linked lists to manage tasks based on priority. One challenge was ensuring tasks were inserted in the correct order based on priority, which required careful manipulation of pointers. The same challenge was presented when we were required to delete a node with task ID somewhere in between the nodes. Overall, this exercise enhanced my understanding of linear data structures and their practical applications in task management systems.