ToDo List Using LinkedList

Description:

This code represents a To-Do List manager implemented in C++ using a linked list. Let's break down the components:

Node Class (Node):

- Represents a single node in a linked list.
- Contains two data members: data (string) to store the task name and next (pointer to Node) to link to the next node.

To-Do List Class (ToDoList):

- Manages the linked list of tasks.
- Contains a private member head (pointer to Node) representing the head of the linked list.

Main Function (main()):

- Presents a menu-driven interface to interact with the To-Do List:
  - ➢ Adds tasks (Press 1), allowing the user to input task names with spaces using getline(cin, Task).
  - ➢ Deletes tasks (Press 2), asking the user for the task name to be deleted.
  - ➢ Displays all tasks (Press 3) if the list is not empty.
- Uses cin.ignore() after reading integers (choice) to clear the newline character ('\n') from the input buffer before using getline() to read strings. This prevents issues when switching between different input methods.
- Continues the program according to the user's choice ('y' for continuing and any other key for exit).

This program provides a basic command-line interface for managing a To-Do List using a linked list, allowing tasks to be added, removed, and displayed. The Node class represents individual elements, while the ToDoList class organizes and manages these elements. The main() function serves as the user interface to interact with the To-Do List functionalities.

Functionality:

Certainly! Let's break down the functionality of the code:

- **IsEmpty():** Checks if the list is empty by examining whether the head is pointing to nullptr.
- **AddTask(string Task):** Adds a new task to the end of the list.
- **DisplayTask():** Displays all tasks in the list.
- **DeleteTask(string Task):** Removes a specific task from the list.

Code:

- **TDnode.h:**

```
#include<string>

#include<iostream>

using namespace std;

class Node

{

        public:
```

```
    string data;

    Node* next;

    Node(string d=0, Node* n=0)

    {

        data = d;

        next = n;

    }

};
```

- **ToDoList.h:**

```
#include<string>

#include"TDnode.h"

#include<iostream>

using namespace std;


class ToDoList

{

    private:

    Node* head;

    public:

    ToDoList()

    {
        head = 0;
        }
        bool IsEmpty();
        void AddTask(string Task);
        void DisplayTask();
        void DeleteTask(string Task);
};
bool ToDoList::IsEmpty()
{
    if(head == 0)
    {
        return true;
    }
    else
    {
        return false;
    }
}
// Add a new task to the list
void ToDoList::AddTask(string Task)
{
  Node* newTask = new Node(Task);
  if(IsEmpty())
```

```
              {
            head = newTask;
         }
              else
              {
              Node* currnode = head;
            while (currnode->next != 0)
                        {
              currnode = currnode->next;
            }
            currnode->next = newTask;
            currnode = newTask;
         }
         cout<<"Added task: "<<Task<<endl;
      }
      // Display all tasks in the list
      void ToDoList::DisplayTask()
      {
         Node* currnode = head;
         while(currnode != 0)
              {
            cout<<"- "<<currnode->data<<endl;
            currnode = currnode->next;
```

```
    }
}
// Remove a task from the list
void ToDoList::DeleteTask(string Task)
{
    Node* currnode = head;
    Node* prevnode = 0;
    while(currnode!=0 && currnode->data!=Task)
        {
        prevnode = currnode;
        currnode = currnode->next;
    }
    if(currnode == 0)
        {
        cout<<"Task not found."<<endl;
    }
        else
        {
        if (prevnode == 0)
                {
            head = head->next;
            currnode->next = 0;
        }
```

```
                  else

                  {

              prevnode->next = currnode->next;

              currnode->next = 0;

            }

            delete currnode;

            cout<<"Task removed: "<<Task<<endl;

          }

        }
```

- **ToDoList.cpp:**

```
#include<string>
#include"ToDoList.h"
#include<iostream>
using namespace std;
int main()
{
        string Task;
        char con;
        int choice;
        ToDoList td;
        do
        {
                cout<<"\tPress 1 for adding the task in the list"<<endl;
                cout<<"\tPress 2 for deleting the task from the list"<<endl;
                cout<<"\tPress 3 for display the tasks from the list"<<endl;
                cin>>choice;
                cin.ignore();
                switch(choice)
                {
                case 1:
                    cout<<"Enter the task that you want to add in the list: ";
                    getline(cin, Task);
                    td.AddTask(Task);
                    break;
              case 2:
                if(td.IsEmpty())
                    {
                cout<<"List is empty."<<endl;
                }
                else
                    {
                        cout<<"Enter the task that you want to remove from
the list: ";
                        getline(cin, Task);
                        td.DeleteTask(Task);
                    }
                  break;
                case 3:
                    if(td.IsEmpty())
                    {
                cout<<"List is empty."<<endl;
                }
                else
                {
                        cout<<"Tasks in the list are:"<<endl;
                        td.DisplayTask();
                    }
                  break;
                default:
                    cout<<"Sorry! Wrong choice"<<endl;
                    break;
```

```
				}
				cout<<"\nPress (y) for again continue the program and press any key except (y) for exit: ";
				cin>>con;
		}
		while(con == 'y');
		return 0;
}
```