

Assignment No.4

Name : Prarthana Kumbhar

PRN : B25CE2014

Class: SY1

Batch : C

TITLE:-

Write a program that implements a simple task scheduler using a singly linked list. Each node in the linked list represents a task with its priority and execution time. Tasks are scheduled based on their priority, with higher priority tasks being executed first.

CODE:-

```
#include <iostream>
#include <string> using
namespace std; struct
Task {    string name;
int priority;    int
execTime;
    Task* next;
    Task(const string& n, int p, int e) : name(n), priority(p), execTime(e), next(nullptr) {}
};

class TaskScheduler { private:
    Task* head;

public:
    TaskScheduler() : head(nullptr) {}
    ~TaskScheduler() {
while (head) {        Task*
temp = head;        head =
head->next;        delete
temp;
    }
}
```

```

void addTask(const string& name, int priority, int execTime) {
    Task* newTask = new Task(name, priority, execTime);

    if (!head || head->priority < priority) {
newTask->next = head;        head = newTask;
    } else {
        Task* current = head;        while (current->next && current-
>next->priority >= priority) {        current = current->next;
    }
        newTask->next = current->next;        current-
>next = newTask;
    }
}

void executeTasks() {        Task* current =
head;        while (current) {        cout <<
"task: " << current->name
        << " priority: " << current->priority
        << " time: " << current->execTime << " units\n";
current = current->next;
    }
}

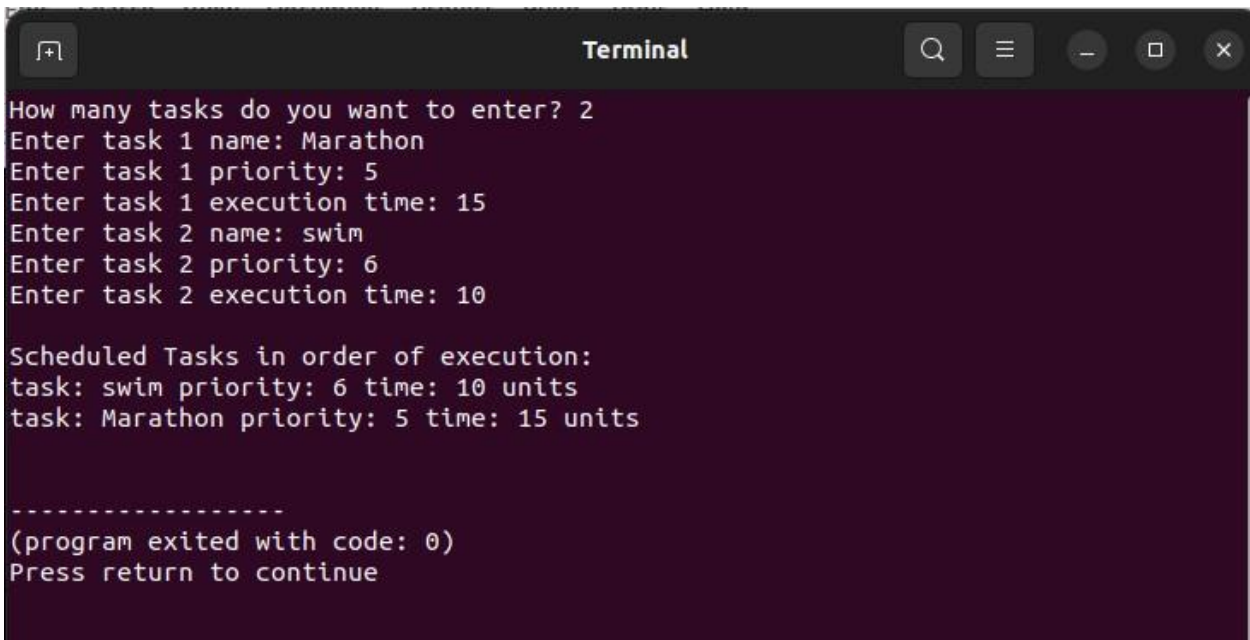
};

int main() {
    TaskScheduler scheduler;
    int n;
    cout << "How many tasks do you want to enter? ";    cin
>> n;
    cin.ignore();    for (int i = 0; i < n; ++i) {
string name;        int priority, execTime;        cout
<< "Enter task " << i + 1 << " name: ";
    getline(cin, name);
        cout << "Enter task " << i + 1 << " priority: ";
    cin >> priority;
        cout << "Enter task " << i + 1 << " execution time: ";
    cin >> execTime;        cin.ignore();
    scheduler.addTask(name, priority, execTime);
    }
}

```

```
    cout << "\nScheduled Tasks in order of execution:\n";  
    scheduler.executeTasks();    return 0;  
  
}
```

OUTPUT:-

A terminal window titled "Terminal" with standard macOS window controls (zoom, search, menu, zoom out, close). The terminal shows the following text:

```
How many tasks do you want to enter? 2  
Enter task 1 name: Marathon  
Enter task 1 priority: 5  
Enter task 1 execution time: 15  
Enter task 2 name: swim  
Enter task 2 priority: 6  
Enter task 2 execution time: 10  
  
Scheduled Tasks in order of execution:  
task: swim priority: 6 time: 10 units  
task: Marathon priority: 5 time: 15 units  
  
-----  
(program exited with code: 0)  
Press return to continue
```