Task Scheduling

System

Joel M Thomas

43

CSEB

18/07/2024

INTRODUCTION

This project involves developing a task scheduling system in C. The system allows users to add tasks, view tasks sorted by priority, and view tasks sorted by deadlines.

Managing tasks efficiently is crucial for any project. The challenge is to design a system that can prioritize tasks based on their importance and deadlines.

The objective of this project is to implement a task scheduling system that can handle multiple tasks, prioritize them based on user-defined criteria, and schedule them effectively.

**System Requirements**

**Hardware Requirements**

* Processor: Intel i3 or higher
* RAM: 4GB or more
* Hard Disk: 500GB or more

**Software Requirements**

* Operating System: Windows/Linux/MacOS
* Compiler: GCC or any C compiler
* Text Editor: Code::Blocks, Visual Studio Code, or any other

The program consists of several functions that allow users to add tasks, view tasks sorted by priority, and view tasks sorted by deadlines. The tasks are stored in a struct array, and sorting is done using bubble sort in two separate functions for priority and deadlines.

1. Start

2. Display menu

3. Get user choice

4. If choice is "Add Task":

* Prompt for task details
* Add task to the array

5. If choice is "Display Tasks by Priority":

* Sort tasks by priority
* Display sorted tasks

6. If choice is "Display Tasks by Deadline":

* Sort tasks by deadline
* Display sorted tasks

7. If choice is "Exit":

* Exit the program

8. Repeat steps 2-7 until exit

OUTPUT

A screenshot of a computer program

Description automatically generated

A screenshot of a computer program

Description automatically generated

A screenshot of a computer program

Description automatically generated

The results demonstrate that tasks are added, viewed, and sorted correctly based on the given criteria. The sorting mechanisms prioritize tasks correctly according to their priority and deadline.

**Conclusion**

**Summary of the Project**

This project successfully implements a task scheduling system in C that can add, view, and schedule tasks based on priority and deadlines. The program is menu-driven and user-friendly.

**Future Enhancements**

* Implementing more sophisticated scheduling algorithms
* Adding features to edit and delete tasks

REFERENCE

* Online(chrome)

APPENDICES

#include <stdio.h>

#include <stdlib.h>

#include <string.h>

struct Task {

    char name[50];

    int priority;

    int deadline;

};

struct Task tasks[50];

int num\_tasks = 0;

void addTask();

void displayTasksByPriority();

void displayTasksByDeadline();

void displayTask(int task\_id);

int main() {

    int choice;

    do {

        printf("\n=== Task Scheduling System ===\n");

        printf("1. Add Task\n");

        printf("2. Display Tasks by Priority\n");

        printf("3. Display Tasks by Deadline\n");

        printf("4. Exit\n");

        printf("Enter your choice: ");

        scanf("%d", &choice);

        switch (choice) {

            case 1:

                addTask();

                break;

            case 2:

                displayTasksByPriority();

                break;

            case 3:

                displayTasksByDeadline();

                break;

            case 4:

                printf("Exiting the program...\n");

                break;

            default:

                printf("Invalid choice. Please enter a number between 1 and 5.\n");

        }

    } while (choice != 4);

    return 0;

}

void addTask() {

    if (num\_tasks >= 50) {

        printf("Cannot add more tasks. Maximum limit reached.\n");

        return;

    }

    struct Task new\_task;

    printf("Enter task name: ");

    scanf(" %[^\n]s", new\_task.name);

    printf("Enter priority (1 - low, 2 - medium, 3 - high): ");

    scanf("%d", &new\_task.priority);

    printf("Enter deadline (in days from start): ");

    scanf("%d", &new\_task.deadline);

    tasks[num\_tasks++] = new\_task;

    printf("Task added successfully.\n");

}

void displayTasksByPriority() {

    if (num\_tasks == 0) {

        printf("No tasks to display.\n");

        return;

    }

    for (int i = 0; i < num\_tasks - 1; i++) {

        for (int j = 0; j < num\_tasks - i - 1; j++) {

            if (tasks[j].priority > tasks[j + 1].priority) {

                struct Task temp = tasks[j];

                tasks[j] = tasks[j + 1];

                tasks[j + 1] = temp;

            }

        }

    }

    printf("\nTasks sorted by priority:\n");

    for (int i = 0; i < num\_tasks; i++) {

        displayTask(i);

    }

}

void displayTasksByDeadline() {

    if (num\_tasks == 0) {

        printf("No tasks to display.\n");

        return;

    }

    for (int i = 0; i < num\_tasks - 1; i++) {

        for (int j = 0; j < num\_tasks - i - 1; j++) {

            if (tasks[j].deadline > tasks[j + 1].deadline) {

                struct Task temp = tasks[j];

                tasks[j] = tasks[j + 1];

                tasks[j + 1] = temp;

            }

        }

    }

    printf("\nTasks sorted by deadline:\n");

    for (int i = 0; i < num\_tasks; i++) {

        displayTask(i);

    }

}

void displayTask(int task\_id) {

    printf("Task ID: %d\n", task\_id + 1);

    printf("Name: %s\n", tasks[task\_id].name);

    printf("Priority: %d\n", tasks[task\_id].priority);

    printf("Deadline: %d days\n", tasks[task\_id].deadline);

    printf("\n");

}