

TRƯỜNG ĐẠI HỌC GIAO THÔNG VẬN TẢI
PHÂN HIỆU TẠI THÀNH PHỐ HỒ CHÍ MINH
BỘ MÔN CÔNG NGHỆ THÔNG TIN



BÁO CÁO BÀI TẬP LỚN
ĐỀ TÀI: TẠO TODOLIST QUẢN LÝ THỜI GIAN

Giảng viên hướng dẫn : ThS. TRẦN THỊ DUNG
Sinh viên thực hiện : PHẠM VĂN GIANG
Lớp : CÔNG NGHỆ THÔNG TIN
Khóa : K62

Tp. Hồ Chí Minh, năm 2022

TRƯỜNG ĐẠI HỌC GIAO THÔNG VẬN TẢI
PHÂN HIỆU TẠI THÀNH PHỐ HỒ CHÍ MINH
BỘ MÔN CÔNG NGHỆ THÔNG TIN



BÁO CÁO BÀI TẬP LỚN
ĐỀ TÀI: TẠO TODOLIST QUẢN LÝ THỜI GIAN

Giảng viên hướng dẫn : ThS. TRẦN THỊ DUNG
Sinh viên thực hiện : PHẠM VĂN GIANG
Lớp : CÔNG NGHỆ THÔNG TIN
Khóa : K62

Tp. Hồ Chí Minh, năm 2022

NHIỆM VỤ THIẾT KẾ BÀI TẬP LỚN
BỘ MÔN: CÔNG NGHỆ THÔNG TIN

-----***-----

Mã sinh viên: 6251071031

Khóa: 62

Mã sinh viên: 6251071055

Khóa: 62

Mã sinh viên: 6251071025

Khóa: 62

Mã sinh viên: 6251071027

Khóa: 62

Họ tên SV: Đinh Nguyễn Minh Hoàng

Lớp: CNTT-K62

Họ tên SV: Nguyễn Văn Long

Lớp: CNTT-K62

Họ tên SV: Phạm Văn Giang

Lớp: CNTT-K62

Họ tên SV: Nguyễn Nhật Hào

Lớp: CNTT-K62

1. Tên đề tài:

Xây dựng phần mềm TODOLIST để quản lý thời gian dễ dàng.

2. Mục đích, yêu cầu.

- Mục đích:
 - Mang đến cho bạn một danh sách các việc ưu tiên để đảm bảo rằng bạn không quên bất cứ việc gì.
 - Lập kế hoạch hiệu quả cho các nhiệm vụ của mình để có thể hoàn thành trong khung thời gian chính xác.
- Yêu cầu:
 - Hiểu rõ todolist dùng để làm gì dùng cho mục đích gì .
 - Lên thời khóa biểu cho bản thân để hoàn thành việc một cách tốt nhất.

3. Nội dung và phạm vi đề tài.

- Nội dung:
 - Xây dựng chương trình quản lý thời gian.
 - Giúp người dùng không bỏ quên việc cần thiết phải làm.
- Phạm vi đề tài : xây dựng todolist .

4. Công nghệ, công cụ và ngôn ngữ lập trình.

- Ngôn ngữ C/C++.
- Notepad++.
- MonoDevelop.
- GNAT Programming Studio.

- C++ Builder.
- Microsoft Visual Studio.
- Vim.
- CodeLite.
- Eclipse.
- Visual Studio Code.

5. Các kết quả chính dự kiến sẽ đạt được và ứng dụng.

-Tạo một phần mềm ứng dụng giúp cho người sử dụng có thể quản lý thời gian một cách hiệu quả nhất.

6. Giáo viên và cán bộ hướng dẫn.

Họ tên: Trần Thị Dung.

Đơn vị công tác: Bộ môn Công Nghệ Thông Tin – Trường Đại học Giao Thông Vận tải phân hiệu tại phân hiệu tại TP HCM.

Điện thoại: 0388389579

Email: ttdung@st.utc2.edu.vn

Ngày 17 tháng 5 năm 2022

Trưởng BM Công Nghệ Thông Tin

Đã giao nhiệm vụ BTL

Giáo viên hướng dẫn

ThS. Trần Phong Nhã

ThS. Trần Thị Dung

Đã nhận nhiệm vụ BTL

Sinh viên: Đinh Nguyễn Minh Hoàng

Điện thoại: 0337972340

Sinh viên: Nguyễn Văn Long

Điện thoại: 0398481719

Sinh viên: Phạm Văn Giang

Điện thoại: 0976320547

Sinh viên: Nguyễn Nhật Hào

Điện thoại: 0386982857

Ký tên:Hoàng

Email:6251071031@st.utc2.edu.vn

Ký tên:Long

Email:6251071055@st.utc2.edu.vn

Ký tên:Giang

Email: 6251071025@st.utc2.edu.vn

Ký tên:Hào

Email: 6251071027@st.utc2.edu.vn

LỜI CẢM ƠN

Lời đầu tiên, em xin gửi đến quý Thầy Cô của **Bộ môn Công Nghệ Thông Tin trường Đại học Giao Thông Vận tải - Phân hiệu tại Tp Hồ Chí Minh** lời chúc sức khỏe và lòng biết ơn sâu sắc nhất. Thầy cô đã tận tình hướng dẫn, truyền đạt kiến thức, kinh nghiệm cho em trong suốt quá trình học tập và rèn luyện ở trường. Đặc biệt, em xin gửi lời cảm ơn đến cô Trần Thị Dung, cô đã tận tình hướng dẫn chỉ bảo để em có thể tiếp cận với nhiều bài toán thực tế để hoàn thành bài tập lớn. Nhờ cô mà em đã khắc phục được những khuyết điểm của bản thân trong quá trình làm bài tập lớn.

Mặc dù em đã cố gắng hết sức trong quá trình làm bài tập lớn, nhưng vì thời gian có hạn và chưa có kinh nghiệm với bài toán thực tế, nên em chắc chắn sẽ có nhiều thiếu sót nên mong thầy cô bỏ qua.

Cảm ơn tập thể lớp công nghệ thông tin K62 và các anh chị khóa trên đã luôn giúp đỡ em trong quá trình học tập.

Lời sau cùng, em xin chúc quý thầy cô trong **Bộ môn Công Nghệ Thông Tin trường Đại học giao thông Vận tải – Phân hiệu tại Tp Hồ Chí Minh** luôn mạnh khỏe và đạt được nhiều thành công trong công việc.

Em xin chân thành cảm ơn !

Tp. Hồ Chí Minh, ngày 17, tháng 5, năm 2022

Sinh viên thực hiện

Phạm Văn Giang

NHẬN XÉT CỦA GIÁO VIÊN

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

Tp. Hồ Chí Minh, ngày tháng năm 2022

Giáo viên hướng dẫn

Trần Thị Dung

DANH MỤC CHỮ VIẾT TẮT

STT	Mô tả	Ý nghĩa	Ghi chú
1	TODOLIST	Danh sách quản lí công việc	
2	BTL	Bài Tập Lớn	
3	C/C++.	Ngôn ngữ lập trình.	
4	Notepad++	Phần mềm soạn thảo đa tính năng.	
5	MonoDevelop	một open-source GNOME IDE (intergrated development environment) hỗ trợ lập trình đa ngôn ngữ (với mục tiêu ban đầu là .Net) trên các hệ điều hành Linux, Mac OS X và Windows	
6	C++ Builder	một môi trường phát triển ứng dụng nhanh (RAD)	
7	Vim	một trình text editor, được tích hợp sẵn trên một số distro của Linux, giúp bạn có thể chỉnh sửa text.	
8	CodeLite	Công cụ dùng để lập trình.	
9	Eclipse	là một công cụ hỗ trợ người dùng lập trình mã nguồn mở	
10	Visual Studio Code	là một trong những trình soạn thảo mã nguồn phổ biến nhất được sử dụng bởi các lập trình viên	

MỤC LỤC

LỜI CẢM ƠN	iv
NHẬN XÉT CỦA GIÁO VIÊN	v
DANH MỤC CHỮ VIẾT TẮT	vi
MỞ ĐẦU	3
CHƯƠNG 1: GIỚI THIỆU ĐỀ TÀI	4
1.1 Tổng quan đề tài.....	4
1.2 Mục tiêu nghiên cứu.....	4
CHƯƠNG 2 : CƠ SỞ LÝ THUYẾT	5
2.1 TỔNG QUAN VỀ NGÔN NGỮ C.	5
CHƯƠNG 3 : XÂY DỰNG CHƯƠNG TRÌNH.....	6
3.1 CẤU TRÚC.....	6
3.1.1 Thư Viện.....	6
3.1.2 Các hàm được khởi tạo và có sẵn được sử dụng.....	6
3.2 Phân tích code.....	6
3.2.1 Tạo struct todo, date.	6
3.2.2 Tạo câu lệnh để con trỏ t có giá trị thực của thời gian.....	7
3.2.3 Tạo hàm adjustcount để cập nhật số lượng node của các todo.	7
3.2.4 Tạo hàm createtodo để nhập các todo.....	7
3.2.5 Tạo hàm seetodo để xem các todo sau khi đã tạo.	10
3.2.6 Tạo hàm unfinishTodo để xem các todo chưa hoàn thành.	11
3.2.7 Tạo hàm overdueTodo để xem các todo đã quá hạn.	12
3.2.8 Tạo hàm rankTodo để sắp xếp các todo theo mức độ công việc.	12
3.2.9 Tạo hàm completeTodo để xem các todo đã hoàn thành.	14
3.2.10 Tạo hàm repairTodo để sửa todo đã tạo từ trước.	15
3.2.11 Tạo hàm deletetodo để xóa các todo đã tạo trước đó.	19
3.2.12 Tạo hàm command để thực thi các lựa chọn của người dùng.....	20
3.2.13 khởi tạo hàm interface để tạo giao diện khi vừa khởi động app. ...	22
3.2.14 hàm main hàm chính của chương trình.....	23
CHƯƠNG 4. KẾT QUẢ	24
4.1 Kết quả đạt được.....	24

Tài liệu tham khảo.....	25
--------------------------------	-----------

MỞ ĐẦU

1. Phần mềm ToDoList.

Phần mềm quản lý công việc todolist ở các doanh nghiệp hay các cá nhân thì việc quản lý công việc và sắp xếp thời gian sao cho hợp lý được đặt lên hàng đầu.

Với một số lượng công việc quá lớn và phức tạp thì việc quản lý bằng sổ sách đã không còn phù hợp thay vào đó cần một công cụ hỗ trợ giúp chủ doanh nghiệp, con người quản lý và sắp xếp được công việc của mình một cách dễ dàng hơn Từ nhu cầu thiết yếu trên mà phần mềm quản lý Todolist đã ra đời.

Phần mềm todolist là một phần mềm hỗ trợ cho người sử dụng có thể quản lý thông tin về công việc và thời gian của mình hoặc sắp xếp thời gian và công việc cho các nhân viên của mình một cách đơn giản. Cung cấp hỗ trợ người dùng một giao thức đơn giản. Tạo điều kiện thuận lợi cho việc quản lý.

2. Mục đích.

Việc quản lý công việc và thời gian là rất cần thiết ở mỗi cá nhân hay các doanh nghiệp.

Phòng quản lý công việc và thời gian cho nhân viên của một doanh nghiệp làm việc khá vất vả và có nhiều sai sót.

3. Đối tượng và phạm vi áp dụng.

- Học sinh, sinh viên, giáo viên, giảng viên.
- Nhân viên làm việc trong các công ty, xí nghiệp lớn...vv.

CHƯƠNG 1: GIỚI THIỆU ĐỀ TÀI

1.1 Tổng quan đề tài.

Todolist là gì?

- Todolist là một danh sách các nhiệm vụ cần phải hoàn thành trong ngày/ tuần/ tháng... được sắp xếp theo một trình tự ưu tiên từ rất quan trọng đến không quan trọng. Nói cách khác, Todolist như là một kế hoạch được phác thảo lên để hướng đến mục đích bạn mong muốn.

- Một khi lập nên Todolist rõ ràng, bạn sẽ chú tâm vào thực hiện các nhiệm vụ và tránh lãng phí thời gian vào những vấn đề xao lãng xung quanh, đáp ứng đúng tiến độ đã đặt ra.

1.2 Mục tiêu nghiên cứu.

- Xây dựng 1 chương trình quản lý thời gian cho đối tượng sử dụng bao gồm:
 - + Xác định nhiệm vụ phải làm.
 - + Sự ưu tiên công việc.
 - + Tính cam kết làm việc mình đã đề ra.
- Lợi ích:
 - + Giảm bớt sự căng thẳng.
 - + Cảm giác hoàn thành công việc tốt hơn.
- Ngôn ngữ sử dụng nghiên cứu : C/C++.
- Chương trình todolist được viết dựa vào ngôn ngữ lập trình C.

CHƯƠNG 2 : CƠ SỞ LÝ THUYẾT

2.1 TỔNG QUAN VỀ NGÔN NGỮ C.

- C là ngôn ngữ lập trình cấp cao, được sử dụng rất phổ biến để lập trình hệ thống cùng với Assembler và phát triển các ứng dụng.

- Vào những năm cuối thập kỷ 60 đầu thập kỷ 70 của thế kỷ XX, Dennis Ritchie (làm việc tại phòng thí nghiệm Bell) đã phát triển ngôn ngữ lập trình C dựa trên ngôn ngữ BCPL (do Martin Richards đưa ra vào năm 1967) .

- Năm 1978, Dennis Ritchie và B.W Kernighan đã cho xuất bản quyển “Ngôn ngữ lập trình C” và được phổ biến rộng rãi đến nay.

Ưu điểm:

1. Ngôn ngữ lập trình C là một ngôn ngữ mạnh, mềm dẻo và có thể truy nhập vào hệ thống, nên thường được sử dụng để viết hệ điều hành, các trình điều khiển thiết bị, đồ họa, có thể xây dựng các phân mềm ngôn ngữ khác , ...
2. Ngôn ngữ lập trình C có cấu trúc module, từ đó ta có thể phân hoạch hay chia nhỏ chương trình để tăng tính hiệu quả, rõ ràng, dễ kiểm tra trong chương trình.

Nhược điểm:

1. Một số kí hiệu của ngôn ngữ lập trình C có nhiều ý nghĩa khác nhau. Ví dụ toán tử * là toán tử nhân, cũng là toán tử thay thế, hoặc dùng khai báo con trỏ. Việc sử dụng đúng ý nghĩa của các toán tử phụ thuộc vào ngữ cảnh sử dụng.
2. Vì C là một ngôn ngữ mềm dẻo, đó là do việc truy nhập tự do vào dữ liệu, trộn lẫn các dữ liệu, ... Từ đó, dẫn đến sự lạm dụng và sự bất ổn của chương trình.

CHƯƠNG 3 : XÂY DỰNG CHƯƠNG TRÌNH

3.1 CẤU TRÚC.

3.1.1 Thư Viện.

- Stdio.h là thư viện chuẩn định nghĩa 3 kiểu biến, một số macro và các hàm đa dạng để thực hiện input và output.
- Stdlib.h là thư viện chuẩn định nghĩa 4 kiểu biến, một số macro và các hàm đa dạng để thực hiện các tính năng chung.
- Unistd.h là thư viện cung cấp quyền truy cập vào API hệ điều hành POSIX. Nó được định nghĩa bởi tiêu chuẩn POSIX.1 , cơ sở của đặc điểm kỹ thuật unix đơn , và do đó phải có sẵn trong bất kì trình biên dịch và hệ điều hành nào tương thích với POSIX.

3.1.2 Các hàm được khởi tạo và có sẵn được sử dụng.

- adjustcount : dùng để xác định số node (số todo) có trong bộ nhớ.
- createtodo : dùng để tạo todo mới.
- seetodo : xem todo đã tạo.
- deletetodo: xóa todo đã tạo.
- repairTodo: để sửa todo đã tạo.
- unfinishTodo: để xem todo chưa hoàn thành.
- overdueTodo: để xem các todo quá hạn.
- rankTodo: để sắp xếp các todo theo mức độ công việc.
- completeTodo: để xem các todo chưa hoàn thành.
- option :đưa ra các lựa chọn cho người dùng.
- interface: dùng để render ra giao diện ban đầu cho application.
- system("cls"): hàm xóa màn hình.
- sleep(): dùng để tạo 1 khoảng thời gian “ngủ” cho chương trình (n giây).
- system(“color mã màu”): dùng để sét màu nền.
- system("pause"): dùng để dừng chương trình để chuẩn bị cho 1 lần nhập phím.
- Và một số lệnh/hàm cơ bản khác(printrf() ,scanf () ,fflush(stdin) ,...);.

3.2 Phân tích code.

3.2.1 Tạo struct todo, date.

```
struct ToDo // tạo struct todo
{
    char buffer[101]; // chứa dữ liệu của các node
    struct ToDo *next; // dùng để liên kết các node với nhau
    int count; // biến count dùng để lưu số lượng node
```

```

};
typedef struct ToDo todo;
todo *start = NULL; // khởi tạo con trỏ/node start với giá trị bằng NULL
    struct Date{      // tạo một struct Date
        char startTime; // tạo 2 biến có kiểu dữ liệu char có tên :startTime, endTime
        char endTime;
    };
typedef struct Date date;

```

3.2.2 Tạo câu lệnh để con trỏ t có giá trị thực của thời gian.

```

time_t now = time(NULL);
tm *t= localtime(&now);
char *ch1 = asctime(t);

```

3.2.3 Tạo hàm adjustcount để cập nhật số lượng node của các todo.

```

void adjustcount()
{
    todo *temp;
    int i = 1;
    temp = start;
while (temp != NULL)
    {
        temp->count = i;
        i++;
        temp = temp->next;
    }
}

```

3.2.4 Tạo hàm createtodo để nhập các todo.

```

todo *addGlo, *tempGlo;
void createtodo()
{
    char check;
    system("cls");

while (1)

```

```

{
    printf("\nNHAP 'Y' DE TAO TODO-NHAP 'N' DE THOAT ");
    fflush(stdin);
    scanf("%s", &check);
    if (check == 'N' || check=='n')
        break;
    else if (check == 'Y' || check == 'y')
    {
        if (start == NULL)
        {
            addGlo = (todo *)calloc(1, sizeof(todo));
            start = addGlo;
            printf("\nMOI BAN NHAP TODO MOI:");
            fflush(stdin);
            gets(addGlo->buffer);
            printf("\nMOI BAN NHAP MUC DO QUAN TRONG CUA TODO: ");
            fflush(stdin);
            fflush(stdin);
        }
        do{
            printf("\nMUC DO QUAN TRONG(1:CUC KI QUAN TRONG, 2 QUAN TRONG, 3 IT QUAN TRONG): ");
            scanf("%d",&addGlo->severity);
        }
        while(addGlo->severity !=1 && addGlo->severity!=2 && addGlo->severity!=3);
        printf("\nMOI BAN NHAP THOI GIAN BAT DAU CONG VIEC: ");
        fflush(stdin);
        scanf("%d",&addGlo->data.startTime);
        do{
            printf("\nMOI BAN NHAP THOI GIAN KET THUC CONG VIEC: ");
            fflush(stdin);
            scanf("%d",&addGlo->data.endTime);
        }
        if(addGlo->data.endTime < addGlo->data.startTime || addGlo->data.endTime >24){
            printf("\nNHAP SAI! THOI GIAN KET THUC PHAI LON HON THOI GIAN BAN DAU.MOI BAN NHAP LAi! ");
        }
    }
}

```



```

    }
}
while(addGlo->data.endTime < addGlo->data.startTime || addGlo->data.endTime>24);
    addGlo->complete=0;
    addGlo->count = 1;
    start->next = NULL;
else
{
    tempGlo = (todo *)calloc(1, sizeof(todo));
    printf("\n\nMOI BAN NHAP TODO MOI:");
    fflush(stdin);
    gets(tempGlo->buffer);
    fflush(stdin);
    do{
        printf("\nMUC DO QUAN TRONG(1:CUC KI QUAN TRONG, 2 QUAN
        TRONG, 3 IT QUAN TRONG): ");
        scanf("%d",&tempGlo->severity);
    }
    while(tempGlo->severity !=1 && tempGlo->severity!=2 && tempGlo->severity!=3);
        printf("\nMOI BAN NHAP THOI GIAN BAT DAU CONG VIEC: ");
        fflush(stdin);
        scanf("%d",&tempGlo->data.startTime);
        do{
            printf("\nMOI BAN NHAP THOI GIAN KET THUC CONG VIEC: ");
            fflush(stdin);
            scanf("%d",&tempGlo->data.endTime);
            if(tempGlo->data.endTime < tempGlo->data.startTime || tempGlo->data.endTime >24 ){
                printf("\nNHAP SAI! THOI GIAN KET THUC PHAI LON HON THOI GIAN
                BAN DAU.MOI BAN NHAP LAi! ");
            }
        }
        while(tempGlo->data.endTime < tempGlo->data.startTime || tempGlo-
        >data.endTime>24);
            tempGlo->complete=0;

```

```

        tempGlo->next = NULL;
        addGlo->next = tempGlo;
        addGlo = addGlo->next;
    }
    adjustcount();
}
else
{
    printf("\n\n\t\t\t\t\tSAI CU PHAP, MOI BAN NHAP LAI!");
    sleep(1);
    createtodo();
}
}
}

```

3.2.5 Tạo hàm seetodo để xem các todo sau khi đã tạo.

```

void seetodo()
{
    system("cls");
    todo *temp;
    temp = start;
    if (start == NULL)
        printf("\n\n\t\t\t\t\tTUYET VOI BAN DA HOAN THANH TODO HOM NAY\n");
    while (temp != NULL)
    {
        printf("\nVIEC %d.", temp->count);
        puts(temp->buffer);
        fflush(stdin);
        printf("\nMUC DO QUAN TRONG CUA CONG VIEC: ");
        if(temp->severity==1){
            printf("CUC KI QUAN TRONG");
        } else if(temp->severity==2){
            printf("QUAN TRONG");
        }
    }
}

```

```

} else{
    printf("IT QUAN TRONG");
}

fflush(stdin);

printf("\nTHOI GIAN THUC HIEN CONG VIEC:%dH-%dH",temp-
>data.startTime,temp->data.endTime);
if(temp->data.endTime < t->tm_hour){
    printf("\nMUC DO HOAN THANH:DA QUA HAN");
}
else{
    if(temp->complete==0){
        printf("\nMUC DO HOAN THANH:CHUA HOAN THANH");
    } else {
        printf("\nMUC DO HOAN THANH:DA HOAN THANH");
    }
}

temp = temp->next;
}

printf("\n\n\n");
}

```

3.2.6 Tạo hàm unfinishTodo để xem các todo chưa hoàn thành.

```

void unfinishTodo()
{
    system("cls");
    todo *temp=start;
    int i=0;
    if(temp ==NULL){
        printf("\nBAN CHUA NHAP TODO NGAY %d/%d/%d",t->tm_mday,t-
>tm_mon+1,t->tm_year+1900);
    }else{
        while(temp!=NULL){
            if(temp->complete==0 && temp->data.endTime > t->tm_hour){
                i++;
                printf("\n%d) %s ",temp->count,temp->buffer);
            }
        }
    }
}

```

```

        printf("THOI GIAN CON LAI DE HOAN THANH:%dH:%dP",(temp-
>data.endTime-t->tm_hour),t->tm_min);
    }
    temp=temp->next;
}
}

printf("\n\n\n");
system("pause");
}

```

3.2.7 Tạo hàm overdueTodo để xem các todo đã quá hạn.

```

void overdueTodo()
{
    system("cls");
    todo *temp=start;
    if(temp==NULL){
        printf("\nTUYET VOI KHONG CO TODO NAO HET HAN");
    } else{
        while(temp!=NULL){
            if(temp->data.endTime < t->tm_hour && temp->complete !=1){
                printf("\nTO DO: %s DA HET HAN TU %d:%dPHUT TRUOC ",temp-
>buffer,(t->tm_hour-temp->data.endTime),t->tm_min);
            }
            temp=temp->next;
        }
    }

    printf("\n\n\n");
    system("pause");
}

```

3.2.8 Tạo hàm rankTodo để sắp xếp các todo theo mức độ công việc.

```

void rankTodo()
{
    system("cls");
    todo *temp=start,*temp2=start,*temp3=start;
    if(temp==NULL){

```

```

        printf("\nTUYET VOI BAN DA HOAN THANH TODO HOM NAY");
    }
    else{
        int i;
        i=1;
        printf("\nDANH SACH TODO CAN LAM SAP THEO MUC DO CONG VIEC:");
        while(temp != NULL){
            if(temp->severity==1 && temp->data.endTime > t->tm_hour){
                printf("\n%d.%s",i,temp->buffer);
                printf("\nMUC DO: CUC KI QUAN TRONG");
                printf("\nTHOI GIAN CON LAI DE HOAN THANH:%dH:%dP",(temp->data.endTime-t->tm_hour),t->tm_min);
                i++;
            }
            temp=temp->next;
        }
        i=1;
        while(temp2 != NULL){
            if(temp2->severity==2 && temp2->data.endTime > t->tm_hour){
                printf("\n%d.%s",i,temp2->buffer);
                printf("\nMUC DO: QUAN TRONG");
                printf("\nTHOI GIAN CON LAI DE HOAN THANH:%dH:%dP",(temp2->data.endTime-t->tm_hour),t->tm_min);
                i++;
            }
            temp2=temp2->next;
        }
        i=1;
        while(temp3 != NULL){
            if(temp3->severity==3 && temp3->data.endTime > t->tm_hour){
                printf("\n%d.%s",i,temp3->buffer);
                printf("\nMUC DO: IT QUAN TRONG");
            }
        }
    }
}

```

```

        printf("\nTHOI GIAN CON LAI DE HOAN THANH:%dH:%dP",(temp3-
>data.endTime-t->tm_hour),t->tm_min);
        i++;
    }
    temp3=temp3->next;
}
}

printf("\n\n\n");
system("pause");
}

```

3.2.9 Tạo hàm completeTodo để xem các todo đã hoàn thành.

```

void completeTodo()
{
    system("cls");
    todo *temp1=start,*temp2=start;
    int i=0;
    while(temp1 !=NULL){
        if(temp1->complete==1){
            i++;
        }
        temp1=temp1->next;
    }
    if(i>0){
        if(i !=1){
            printf("\nCAC TODO DA HOAN THANH: ");
        } else{
            printf("\nTODO DA HOAN THANH: ");
        }
    }
    while(temp2->complete==1){
        printf("\n%s",temp2->buffer);
        temp2=temp2->next;
    }
    } else {

```

```

        printf("\nBAN CHUA HOAN THANH TODO NAO TRONG NGAY
%d/%d/%d",t->tm_mday,t->tm_mon+1,t->tm_year+1900);
    }

    printf("\n\n");
    system("pause");
}

```

3.2.10 Tạo hàm repairTodo để sửa todo đã tạo từ trước.

```

void repairTodo()
{
    system("cls");
    int x,index;
    todo *repair;
    repair = start;
    if(repair==NULL){
        printf("\nCHUA CO TODO NAO DUOC TAO");
    } else{
        seetodo();
        printf("NHAP THU TU TODO MUON BAN MUON SUA: ");
        scanf("%d",&x);
        system("cls");
        printf("\nNHAP '1' DE SUA TOAN BO TODO");
        printf("\nNHAP '2' DE SUA TEN TODO");
        printf("\nNHAP '3' DE SUA MUC DO HOAN THANH CUA TODO");
        printf("\nNHAP '4' DE SUA MUC DO QUAN TRONG CUA TODO");
        printf("\nNHAP '5' DE SUA THOI GIAN BAT DAU VA KET THUC CUA
TODO");
        printf("\nNHAP '6' DE QUAN LAI");
        printf("\nNHAP LUA CHON CUA BAN:");
        scanf("%d",&index);
        system("cls");
        switch (index)
        {
            case 1:
                if(repair!= NULL){

```

```

while(17){
    if (repair->count == x)
    {
        printf("\n\nMOI BAN NHAP LAI TODO: ..... \n");
        fflush(stdin);
        gets(repair->buffer);
        fflush(stdin);
        do{
            printf("\nMUC DO QUAN TRONG(1:CUC KI QUAN TRONG, 2 QUAN
TRONG, 3 IT QUAN TRONG): ");
            scanf("%d",&repair->severity);
            if(repair->severity !=1 && repair->severity!=2 && repair->severity!=3){
                printf("\nSAI CU PHAP VUI LONG NHAP LAI: ");
            }
        }

        while(repair->severity !=1 && repair->severity!=2 && repair-
>severity!=3);
        printf("\nMOI BAN NHAP THOI GIAN BAT DAU CONG VIEC: ");
        fflush(stdin);
        scanf("%d",&repair->data.startTime);
        do{
            printf("\nNHAP THOI GIAN KET THUC CONG VIEC: ");
            fflush(stdin);
            scanf("%d",&repair->data.endTime);
            if(repair->data.endTime < repair->data.startTime || repair->data.endTime
>24){
                printf("\nNHAP SAI! THOI GIAN KET THUC PHAI LON HON THOI
GIAN BAN DAU.MOI BAN NHAP LAI! ");
            }
        }

        while(repair->data.endTime < repair->data.startTime || repair-
>data.endTime>24);
        do{
            printf("\nMUC DO HOANH THANH CONG VIEC(NHAP 1 KHI 'HOAN
THANH', 0 NEU CHUA HOAN THANH): ");

```



```

scanf("%d",&repair->complete);
if(repair->complete !=1 && repair->complete !=0){
printf("\nSAI CU PHAP VUI LONG NHAP LAI! ");
}
}

while(repair->complete !=1 && repair->complete !=0);
break;

}

else{
repair=repair->next;
}
}
}

break;
case 2:
while(17){
if (repair->count == x)
{
printf("\nMOI BAN NHAP LAI TEN CONG VIEC:");
fflush(stdin);
gets(repair->buffer);
break;
}else{
repair=repair->next;
}
}

break;
case 3:
while(repair !=NULL){
if (repair->count == x)
{
printf("\nMUC DO HOANH THANH CONG VIEC(NHAP 1 KHI 'HOAN
THANH', 0 NEU CHUA HOAN THANH): ");

```

```

        fflush(stdin);
        scanf("%d",&repair->complete);
        break;
    }else{
        repair=repair->next;
    }
}

break;
case 4:
    while(repair !=NULL){
        if (repair->count == x)
        {
            printf("\nMUC DO QUAN TRONG(1:CUC KI QUAN TRONG, 2 QUAN
TRONG, 3 IT QUAN TRONG): ");
            fflush(stdin);
            scanf("%d",&repair->severity);
            break;
        }

        else{
            repair=repair->next;
        }
    }

    break;
case 5:
    while(17){
        if (repair->count == x)
        {
            printf("\nMOI BAN NHAP THOI GIAN BAT DAU CONG VIEC: ");
            fflush(stdin);
            scanf("%d",&repair->data.startTime);
            printf("\nMOI BAN NHAP THOI GIAN KET THUC CONG VIEC: ");
            fflush(stdin);
            scanf("%d",&repair->data.endTime);

```

```

        break;
    }else{
        repair=repair->next;
    }
}

    break;
    default:
        repairTodo();
        break;
    }
}

    printf("\n\n\n");
}

```

3.2.11 Tạo hàm deletetodo để xóa các todo đã tạo trước đó.

```

void deleteTodo()
{
    system("cls");
    int x;
    todo *del, *temp;
    if (start == NULL)
        printf("\n\n\t\t\t\tTUYET VOI BAN DA HOAN THANH TODO HOM NAY\n\n");
    else
    {
        seetodo();
        printf("\n\n\t\t\t\tNHAP DIA CHI TODO BAN MUON XOA:");
        scanf("%d", &x);
        del = start;
        temp = start->next;
        while (1)
        {
            if (del->count == x)
            {

```

```

        start = start->next;
        free(del);
        adjustcount();
        break;
    }
    if (temp->count == x)
    {
        del->next = temp->next;
        free(temp);
        adjustcount();
        break;
    }
    else
    {
        del = temp;
        temp = temp->next;
    }
}

system("pause");
}

```

3.2.12 Tạo hàm command để thực thi các lựa chọn của người dùng.

```

void command(char choice)
{
    int choiceTempo;
    choiceTempo = choice;
    switch (choiceTempo)
    {
    case '1':
        seetodo();
        system("pause");
        break;
    case '2':

```

```

        createtodo();
        break;
case '3':
    deleteTodo();
    break;
case '4':
    repairTodo();
    system("pause");
    break;
case '5':
    unfinishTodo();
    break;
case '6':
    overdueTodo();
    break;
case '7':
    rankTodo();
    break;
case '8':
    completeTodo();
    break;
case '9':
    exit(1);
    break;
default:
    printf("\n\n\t\tLUA CHON KHONG HOP LE :-(\n");
    printf("\n\n\t\tMOI BAN NHAP LAI SU LUA CHON: ");
    scanf("%s", &choiceTempo);
    command(choiceTempo);
    break;
    }
}

```

3.2.13 khởi tạo hàm interface để tạo giao diện khi vừa khởi động app.

void interface()

```
{
    system("color 3F");
    printf("\n\n\n\n");
    printf("\t~~~~~"
           "~~~~~"
           "~~~~~"
           "~~~~~\n");
    printf("\t~~~~~"
           "~~~~~"
           "~~~~~"
           "~~~~~\n\n");
    printf("\t} : } : } : } : } : } "
           ": } : } : } : "
           " "
           "CHAO BAN DEN VOI TODO LIST : { : { : { : { : { "
           ": { : { : { : { \n\n");
    printf("\t~~~~~"
           "~~~~~"
           "~~~~~"
           "~~~~~"
           "~~~~~\n");
    printf("\t~~~~~"
           "~~~~~"
           "~~~~~"
           "~~~~~"
           "~~~~~\n");
    system("pause");
}
```

3.2.14 hàm main hàm chính của chương trình.

```
int main()
{
    char choice,choice2;
    interface();
while (17)
    {
system("cls");
        printf("\n\n\n\n\n\n\n\n");
        printf("\t\t\t\t\tTODO      NGAY:      %d/%d/%d\n",t->tm_mday,t->tm_mon+1,t-
>tm_year+1900);
        printf("\t\t\t\t\tCHON '1' DE XEM TODOS TRONG NGÀY %d/%d/%d\n",t-
>tm_mday,t->tm_mon+1,t->tm_year+1900);
        printf("\t\t\t\t\tCHON '2' DE TAO TODO MOI CHO RIENG BAN\n");
        printf("\t\t\t\t\tCHON '3' DE XOA TODO\n");
        printf("\t\t\t\t\tCHON '4' DE SUA TODO\n");
        printf("\t\t\t\t\tCHON '5' DE XEM TODO CHUA HOAN THANH\n");
        printf("\t\t\t\t\tCHON '6' DE XEM CAC TODO QUA HAN\n");
        printf("\t\t\t\t\tCHON '7' DE SAP XEP CAC TODO THEO MUC DO CONG
VIEC\n");
        printf("\t\t\t\t\tCHON '8' DE XEM CAC TODO DA HOAN THANH\n");
        printf("\t\t\t\t\tCHON '9' DE THOAT KHOI UNG DUNG\n");
        printf("\t\t\t\t\tNHAP LUA CHON CUA BAN:");
        scanf("%s", &choice);
        command(choice);
    }
}
```

CHƯƠNG 4. KẾT QUẢ

4.1 Kết quả đạt được.

Sau quá trình tìm hiểu và thực hiện, đề tài đã đáp ứng được những yêu cầu ban đầu được đưa ra. Chương trình todolist được sử dụng với các chức năng cơ bản:

- Tạo 1 todolist của bản thân.
- Nhập todolist của bạn.
- Xem todolist của bạn đã tạo.
- Xóa todolist của bạn đã tạo.
- Sửa todolist của bạn đã tạo.
- Xem các todolist của bạn đã quá hạn.
- Xem các todolist của bạn đã hoàn thành.
- Xem các todolist của bạn chưa hoàn thành.
- Thoát ra khỏi ứng dụng.

4.2 Kiến nghị.

Do thời gian tìm hiểu, phân tích, thiết kế và hiện thực còn hạn chế nên phần mềm vẫn còn một số thiếu sót như:

Giao diện chưa đáp ứng về mặt thẩm mỹ.

Chưa hoàn hảo bằng các ứng dụng ghi chú thời gian và công việc hiện nay

Tài liệu tham khảo

1. [C Programming Absolute Beginner's Guide](#)
2. [The C Programming Language \(2nd Edition\)](#)
3. [Learn C the Hard Way](#)
4. [Head First C](#)
5. [C Programming: A Modern Approach](#)