c프로그래밍 및 실습

날씨에 따라 변하는 알람

진척 보고서 #2

제출일자:2023/12/08
제출자명: 서준
제출자학번: 230750

1. 프로젝트 목표 (16 pt)

1) 배경 및 필요성 (14 pt)

날씨에 따라 음악을 추천해주는 프로그램은 현재 인공지능 및 빅데이터 기술의 발전으로 다양한 분야에서 활용되고 있음. 날씨는 우리 일상 생활에 많은 영향을 미치며, 우리의 기분과 활동을 결정하는 요소 중 하나. 하지만 아침에 일어나면 날씨를 봐야 하는데 가끔은 안 보고 나오다가 불편함을 겪는 사례가 빈번함. 따라서, 특정 날씨 조건에 어울리는 음악을 알람을 통해 재생함으로써 따로 날씨를 안 봐도 직감적으로 알 수 있는 프로그램은 사용자의 일상 생활을 더 즐겁게 만들 수 있음.

2) 프로젝트 목표

사용자가 설정한 알람 시간에, 현재 날씨 조건을 고려하여 분위기에 맞는 음악 또는 알람음을 재생함으로써, 사용자가 그날의 날씨에 맞게 일상을 계획하고 시작할 수 있는 프로그램을 만드는 것을 목표로 함.

3) 차별점

기존 프로그램은 알림음을 사용자가 선택하고 알람을 설정하거나 또는 사용자가 루틴을 따로 설정해야 알람이 울리고 일기예보나 뉴스를 들을 수 있음. 그러나 우리는 현재 날씨 조건을 고려하여 프로그램이 자동으로 알람음을 선택하므로 사용자는 날씨에 따라 일상을 더 효과적으로 조정할 수 있다는 것에 기존 프로그램과 차별점이 있음.

2. 기능 계획

1) 알람 설정 기능

- 사용자에게 scanf_s함수를 통해 직접 시간을 입력 받아 알람을 설정하는 기능

2) 날씨 정보 수집 기능

-날씨 조건(맑음, 더움, 흐림, 비 등)의 정보를 수집하는 기능

(1) 랜덤 모듈을 통해 가상 날씨 설정

- 날씨를 수치화(ex. 매우 더움:100~90, 더움:89~76, 맑음:75~60, 선선함:59~50 흐림:49~26, 소나기:25~20, 비:19~0) 하여서 가상 날씨를 랜덤 모듈로 생성한다.

3) 알람음 재생 기능

-날씨를 토대로 알람음을 재생

(1) 조건문을 사용하여 날씨에 맞는 음악 재생

- 수치화 한 날씨를 조건문으로 하여 흐린 날엔 잔잔한 음악, 맑은 날에는 쾌할한 노래를 선정해 문구로 출력한다.

(예비) 파일 입출력에서 기능을 배운다면 사용자의 플레이 리스트를 입력 받아 제목에 들어간 단어를 기준으로 음악을 추천하여 사용자화 할 것.

4) 알람 해제 기능

-사용자가 알람이 울릴 때 해제 할 수 있게 함

(1) 알람을 해제할 때 현재 날씨를 확인할 수 있게 함

-사용자가 알람을 종료할 때 날씨를 문구로 출력하고 재확인을 한 후 알람이 종료 되면서 프로그램이 종료된다.

3. 진척사항

1) 기능 구현

(1) 알람 설정 기능

- 입출력

현재 시간이 입력되고 사용자가 입력한 시간을 상수에 곱해 클럭 단위로 변환하고 알람음이 울리기까지 계속 루프문을 돌며 대기한다. setAlarm함수를 통해 알람을 입력받으면

초로 변환하여 구조체에 hour과 minte의 멤버로 접근해 지연 시간을 입력받는다. newAlarm은 동적으로 할당된 알람 구조체의 포인터. 이 구조체는 함수 호출자에게 반환되어, 생성된 알람의 정보를 포함하고 있다.

- 설명

clock() 함수를 사용하여 현재 시간을 측정하고 사용자가 입력한 시간(초)을 CLOCKS_PER_SEC 상수와 곱하여 클럭 단위로 변환한 후, wait_time 변수에 저장

while 루프를 사용하여 현재 시간이 시작 시간에 대기 시간을 더한 값보다 작을 때까지 루프를 실행 한다. 사용자가 입력한 시간 정보는 알람 구조체의 hour와 minute 멤버에 저장되고, 또한 release 멤버는 널 문자로 초기화된. 이렇게 생성된 알람 구조체의 포인터가 반환되어, 이후에 프로그램에서 알람을 관리하거나 사용할 수 있도록 한다.

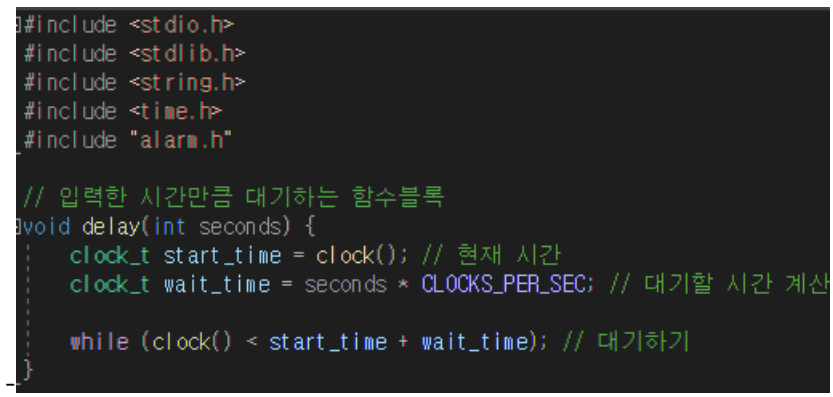
- 적용된 배운 내용 (예: 반복문, 조건문, 클래스, 함수, 포인터 등)

함수선언 및 정의, 루프, 구조체, 동적메모리 할당, 포인터, 구조체 포인터, 헤더파일

-코드 스크린샷

```c
typedef struct {
    int hour;
    int minute;
    char release[10];
} Alarm;

void delay(int seconds);
int generateRandomWeather();
void printWeatherInfo(int weatherValue);
void chooseMusic(int weatherValue);
Alarm* createAlarm(int hour, int minute);
void destroyAlarm(Alarm* alarm);
void setAlarm(Alarm* alarm);
void releaseAlarm(Alarm* alarm);
```



```c
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <time.h>
#include "alarm.h"

// 입력한 시간만큼 대기하는 함수블록
void delay(int seconds) {
    clock_t start_time = clock(); // 현재 시간
    clock_t wait_time = seconds * CLOCKS_PER_SEC; // 대기할 시간 계산

    while (clock() < start_time + wait_time); // 대기하기
}
```

```c
// 알람을 설정하고 딜레이 후 알람을 울리는 함수블록
void setAlarm(Alarm* alarm) {
    printf("알람이 설정되었습니다. %d:%d 후에 울립니다.₩n", alarm->hour, alarm->minute);
    delay(alarm->hour * 3600 + alarm->minute * 60);
    printf("일어나세요!₩n");
}
```

```c
// 알람 구조체를 동적으로 생성하는 함수블록
Alarm* createAlarm(int hour, int minute) {
    Alarm* newAlarm = (Alarm*)malloc(sizeof(Alarm));

    newAlarm->hour = hour;
    newAlarm->minute = minute;
    newAlarm->release[0] = '₩0'; // release 문자열 초기화

    return newAlarm;
}
```

```c
int main() {
    int hour, minute;

    printf("몇 시간 후에 알람을 울릴까요? (24시간 형식)₩n");
    printf("시간: ");
    scanf_s("%d", &hour);

    printf("분: ");
    scanf_s("%d", &minute);

    // 구조체와 동적 메모리 할당을 이용한 알람 생성
    Alarm* myAlarm = createAlarm(hour, minute);

    setAlarm(myAlarm);
```

(2) 날씨 정보 수집 기능

- 입출력 (데이터 입장에서 어떤게 입력되고 어떤게 출력되는지)

rand() 함수를 사용하여 0부터 100까지의 난수를 생성하고, 이를 반환한다. printWeatherInfo 함수는 입력으로 받은 weatherValue에 따라 날씨 정보를 결정하고 출력한다. const char* weatherString은 날씨 정보를 담는 문자열 포인터이다.

- 설명

이 함수는 generateRandomWeather 함수에서 받은 날씨 값을 기반으로 날씨 정보를 결정하고 출력한다. weatherValue 값에 따라 다양한 날씨 상태에 해당하는 문자열을 weatherString 포인터에 할당한다. 각 조건문은 특정 범위의 weatherValue 값을 기준으로 날씨를 판별하고, 적절한 문자열을 weatherString에 할당한다. 마지막으로, printf 함수를 사용하여 현재 날씨 정보를 출력한다.

- 적용된 배운 내용 (예: 반복문, 조건문, 클래스, 함수, 포인터 등)

포인터 배열, 함수 선언 및 정의, 랜덤 모듈, 조건문, 함수 호출, 난수 생성, 해더파일

- 코드 스크린샷

```
// 랜덤 모듈을 사용하여 가상 날씨를 생성하는 함수블록
int generateRandomWeather() {
    srand((unsigned int)time(NULL)); //시드 초기화

    int randomValue = rand() % 101;
    randomValue = (randomValue + rand()) % 101;

    return randomValue;
}
```

```
// 날씨 정보 출력
void printWeatherInfo(int weatherValue) {
    const char* weatherString = NULL;

    if (weatherValue >= 90)
        weatherString = "매우 더움";
    else if (weatherValue >= 76)
        weatherString = "더움";
    else if (weatherValue >= 60)
        weatherString = "맑음";
    else if (weatherValue >= 50)
        weatherString = "선선함";
    else if (weatherValue >= 26)
        weatherString = "흐림";
    else if (weatherValue >= 20)
        weatherString = "소나기";
    else
        weatherString = "비";

    printf("현재 날씨: %s₩n", weatherString);
}
```

```
// 가상 날씨 생성
int weatherValue = generateRandomWeather();

// 날씨 정보 출력
printWeatherInfo(weatherValue);

// 날씨에 따른 음악 선택 및 출력
chooseMusic(weatherValue);
```

(3) 알람음 재생 기능

- 입출력 (데이터 입장에서 어떤게 입력되고 어떤게 출력되는지)

weatherValue에 날씨 값을 나타내는 정수가 들어가서 날씨에 따라 적절한 음악을 선택한다. 이 값이 26이상이면 "선택된 음악: 쾌할한 노래 "이라는 메시지가 출력된다. 아니라면 "선택된 음악: 잔잔한 음악 "라는 메시지가 출력된다.

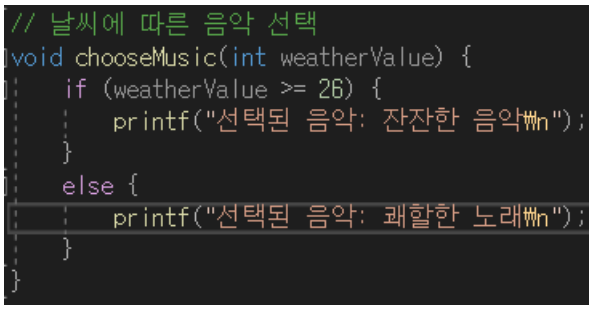
- 설명

weatherValue는 날씨 값을 나타내는 정수다. 이 값은 이 함수에 전달되어 날씨에 따라 적절한 음악을 선택하는 기준으로 사용된다. chooseMusic 함수는 조건문을 통해 날씨 값을 분석하고, 그 결과에 따라 적절한 음악을 선택하여 출력한다. weatherValue값에 따라 "선택된 음악: 잔잔한 음악" 또는 "선택된 음악: 괘할한 노래"라는 메시지가 출력된다.

- 적용된 배운 내용 (예: 반복문, 조건문, 클래스, 함수, 포인터 등)

함수 선언 및 정의, 함수 호출, 조건문, 헤더파일

- 코드 스크린샷



```
// 날씨에 따른 음악 선택
void chooseMusic(int weatherValue) {
    if (weatherValue >= 26) {
        printf("선택된 음악: 잔잔한 음악₩n");
    }
    else {
        printf("선택된 음악: 괘할한 노래₩n");
    }
}
```

```
// 날씨에 따른 음악 선택 및 출력
chooseMusic(weatherValue);
```

(4) 알람 해제 기능

- 입출력 (데이터 입장에서 어떤게 입력되고 어떤게 출력되는지)

releaseAlarm 함수에서 사용자로부터 'haeje'를 입력받아 alarm->release에 저장, releaseAlarm 함수에서 사용자에게 'haeje'를 입력할 것을 출력

- 설명

사용자가 'haeje'를 입력하여 알람을 해제한다. 올바른 문자열이 입력된 경우, 가상 날씨를 생성하고 출력하며, 날씨에 따른 음악을 선택하여 출력. 그렇지 않은 경우에는 "알람이 계속 울립니다." 메시지를 출력. destroyAlarm 함수를 호출하여 free함수로 동적으로 할당된 메모리를 해제 시킨다..

- 적용된 배운 내용 (예: 반복문, 조건문, 클래스, 함수, 포인터 등)

함수 선언 및 정의, 함수 호출, 조건문, 문자열 처리, 동적 메모리 할당 및 해제, 구조체

배열, 헤더파일, 문자열 비교

- 코드 스크린샷

```c
typedef struct {
    int hour;
    int minute;
    char release[10];
} Alarm;

void delay(int seconds);
int generateRandomWeather();
void printWeatherInfo(int weatherValue);
void chooseMusic(int weatherValue);
Alarm* createAlarm(int hour, int minute);
void destroyAlarm(Alarm* alarm);
void setAlarm(Alarm* alarm);
void releaseAlarm(Alarm* alarm);
```

```c
// 동적으로 할당된 알람 구조체를 해제하는 함수블록
void destroyAlarm(Alarm* alarm) {
    free(alarm);
}
```

```c
// 알람을 해제하는 함수블록
void releaseAlarm(Alarm* alarm) {
    printf("알람을 해제하려면 'haeje'를 입력하세요: ");
    scanf_s("%s", alarm->release, sizeof(alarm->release));
}
```

```c
// 알람이 해제되었을 경우
if (myAlarm->release[0] == 'h' && myAlarm->release[1] == 'a' && myAlarm->release[2] == 'e' && myAlarm->release[3] == 'j' && myAlarm->release[4] == 'e' && myAlarm->release[5] == '₩0') {
    // 가상 날씨 생성
    int weatherValue = generateRandomWeather();

    // 날씨 정보 출력
    printWeatherInfo(weatherValue);

    // 날씨에 따른 음악 선택 및 출력
    chooseMusic(weatherValue);
}
else {
    printf("알람이 계속 울립니다.₩n");
}

// 동적 메모리 해제
destroyAlarm(myAlarm);
```

2) 테스트 결과

(1) 알람 설정 기능

- 설명

현재 시간을 잘 받아 왔는지, 입력한 시간을 잘 변환해서 알람이 울리는지를 테스트 하기 위해서

- 테스트 결과 스크린샷

```
몇 시간 후에 알람을 울릴까요? (24시간 형식)
시간: 0
분: 0
알람이 설정되었습니다. 0:0 후에 울립니다.
일어나세요!
```

(2) 날씨 정보 수집 기능

- 설명

계속 똑같은 날씨가 출력되길래 난수를 출력해보았는데 똑같은 난수가 나와서 문제 해결법을 찾다가 시드 초기화를 해야한다는걸 알고 수정하였다.

- 테스트 결과 스크린샷

```
Random Value: 13
```

```
Random Value: 96
```

(3) 알람음 재생 기능

- 설명

26이상이면 쾌활한 노래가 나오고 26미만이면 잔잔한 음악이 나온다.

- 테스트 결과 스크린샷

```
Random Value: 96
선택된 음악: 쾌활한 노래
```

(3) 알람 해제 기능

- 설명

알람음이 해제 되는 기능을 '해제'를 입력하면 release배열에 해제가 들어가면서 인덱싱

하여 0번째가 해 1번째가 제 2번째가 /0가 들어오면서 if문이 실행되어 현재 날씨와 음악을 출력해주는 기능을 구현하려고 했는데 if문이 작동을 하지 않아서 전체 배열을 출력해 보았고 다음으로는 사용자의 입력을 출력해 보았고 한글은 똑같이 해제로 입력받고 해제가 저장되길래 NULL을 입력 받는지 확인해 보았고 각 문자 하나하나 별로 일치하는지 테스트 해보았는데 다 불일치 하길래 인코딩의 문제같아서 영어로 입력했더니 해결되어서 해제 대신 haeje로 입력하기로 수정하였다.

- 테스트 결과 스크린샷

```
void releaseAlarm(Alarm* alarm) {
    printf("알람을 해제하려면 '해제'를 입력하세요: ");
    scanf_s("%s", alarm->release, sizeof(alarm->release));
    printf("현재 입력된 release 배열: %s₩n", alarm->release);
```

테스트 1

```
몇 시간 후에 알람을 울릴까요? (24시간 형식)
시간: 00
분: 00
알람이 설정되었습니다. 0:0 후에 울립니다.
일어나세요!
알람을 해제하려면 '해제'를 입력하세요: 해제
현재 입력된 release 배열: 해제
알람이 계속 울립니다.
```

테스트 1결과

```
printf("사용자 입력: %s₩n", myAlarm->release);
```

테스트 2

```
몇 시간 후에 알람을 울릴까요? (24시간 형식)
시간: 00
분: 00
알람이 설정되었습니다. 0:0 후에 울립니다.
일어나세요!
알람을 해제하려면 '해제'를 입력하세요: 해제
현재 입력된 release 배열: 해제
사용자 입력: 해제
알람이 계속 울립니다.
```

테스트 2 결과

```
printf("조건식 결과: %d₩n", (myAlarm->release[0] == '해' && myAlarm->release[1] == '제' && myAlarm->release[2] == '₩0'));
```

테스트 3

```
몇 시간 후에 알람을 울릴까요? (24시간 형식)
시간: 00
분: 00
알람이 설정되었습니다. 0:0 후에 울립니다.
일어나세요!
알람을 해제하려면 '해제'를 입력하세요: 해제
현재 입력된 release 배열: 해제
사용자 입력: 해제
조건식 결과: 0
알람이 계속 울립니다.
```

테스트 3 결과

```
//각 문자 비교
printf("1번째 문자 비교: %d₩n", myAlarm->release[0] == '해');
printf("2번째 문자 비교: %d₩n", myAlarm->release[1] == '제');
printf("3번째 문자 비교: %d₩n", myAlarm->release[2] == '₩0');
```

테스트 4

```
몇 시간 후에 알람을 울릴까요? (24시간 형식)
시간: 0
분: 0
알람이 설정되었습니다. 0:0 후에 울립니다.
일어나세요!
알람을 해제하려면 '해제'를 입력하세요: 해제
사용자 입력: 해제
조건식 결과: 0
1번째 문자 비교: 0
2번째 문자 비교: 0
3번째 문자 비교: 0
알람이 계속 울립니다.
```

테스트 4 결과

```
printf("조건식 결과: %d₩n", (myAlarm->release[0] == 'h' && myAlarm->release[1] == 'a' && myAlarm->release[2] == 'e' && myAlarm->release[3] == 'j' && myAlarm->release[4] == 'e' && myAlarm->release[5] == '₩0'));

// 문자열 비교를 통한 알람 해제 확인
if (myAlarm->release[0] == 'h' && myAlarm->release[1] == 'a' && myAlarm->release[2] == 'e' && myAlarm->release[3] == 'j' && myAlarm->release[4] == 'e' && myAlarm->release[5] == '₩0') {
```

마지막 테스트

```
몇 시간 후에 알람을 울릴까요? (24시간 형식)
시간: 00
분: 00
알람이 설정되었습니다. 0:0 후에 울립니다.
일어나세요!
알람을 해제하려면 'haeje'를 입력하세요: haeje
조건식 결과: 1
현재 날씨: 비
선택된 음악: 잔잔한 음악
```

마지막 테스트 결과(성공)

4. 계획 대비 변경 사항 (기능에 대한 것만 일정변경X)

1) 변경 내역 제목

- 이전

  실시간 날씨 데이터를 가져오기

- 이후

  날씨 데이터를 미리 저장해놓기

- 사유

  사용자가 산행을 계획하고 있다. 그러나 여행 장소에는 불안정한 인터넷 연결이 예상되고 그 지역에서의 데이터 서비스가 부족한 상황이라 실시간으로 날씨를 불러오는 건 어렵다고 판단하였다.

5. 프로젝트 일정

(진행한 작업과 진행 중인 작업 등을 표기)

업무		11/3	11/10	11/17	11/23	12/10	12/15
제안서 작성		완료					
알람 설정 기능	X		완료				
날씨 정보 수집 기능	랜덤 모듈을 통해 가상 날씨 설정		완료				
알람음 재생 기능	조건문을 사용하여 날씨에 맞는 음악 재생		완료				
알람 해제 기능	알람을 해제할 때 현재 날씨를			완료			

	확인할 수 있게 함					
테스트						**---->**