

C프로그래밍 및 실습

파일 정리 시스템

진척 보고서 #1

제출일자: 2023.11.03

제출자명: 이현승

제출자학번: 236179

1. 프로젝트 목표

1) 배경 및 필요성

회사나 공동으로 업무를 하는 공간에서 시간이 지날수록 개인 PC의 자료들은 많이 쌓일 수 밖에 없음. 이를 보통은 개인이 개인의 분류 기준에 의해서 파일들을 분류 해놓거나 정리를 전혀 하지 않기도 함. 그래서 타인의 PC에서 자료를 찾으려 할때 필요없는 시간이 걸리거나, PC의 주인의 도움이 필요함. 그리고 중요한 파일들을 정리를 해놓지 않아 파일 위치를 잊거나 파일을 실수로 지워버려서 시간이 지나 그 파일을 찾을때 시간이 걸리거나 찾지 못하는 경우가 생김. 그렇기 때문에 같은 기준으로 파일을 정리해야 할 필요성이 있지만, 파일을 정리할 인력, 시간이 필요하거나 그럼에도 불구하고 정리를 전혀 하지 않는 인원이 생길 수 있음. 이러한 이유들로 인해 파일을 자동으로 분류할 수 있는 프로그램이 필요함.

2) 프로젝트 목표

사용자가 정해진 규칙에 의해 파일을 자동으로 정리하고, 그 규칙들을 관리할 수 있도록 함. 또한 파일들이 자동으로 정리된 이후 그 히스토리를 시각화 하여 파일이 어떻게 정리 되었는지 사용자가 확인 할 수이 있게 하는것을 목표로 함.

3) 차별점

기존 프로그램들은 분류 정해진 기준에 의해 분류를 했음. 이 기준들이 변경될 수 있는 점을 고려해서 기준을 추가, 수정, 삭제 할 수있도록 기준관리 기능 추가 함. 그리고 추가로 파일이 이동, 삭제, 수정 등 정리가 된 이후의 히스토리를 파일에 저장하여 파일이 어떻게 정리가 되었는지 히스토리 파일에서 사용자가 한번에 알 수있도록 하기 때문에 기존의 프로그램과 차별점이 있음.

2. 기능 계획

1) 기능 1

- 설명 : 파일 목록을 수집, 정보 분석

(1) 세부 기능 1

- 설명 : 대상 디렉토리를 입력받아서 저장함. 그 디렉토리 내의 파일들을 순회하여 확장자, 생성일, 크기, 내용, 제목 등(변경 될 수 있음)을 추출함

2) 기능 2

- 설명 : 확장자, 생성일, 크기 등의 분류 기준을 설정하며, 조건문을 통해 처리

(1) 세부 기능 1

- 설명 : 분류 기준을 설정한 파일을 만들고, 그 파일에 접근해서 분류 기준을 가져옴.

(2) 세부 기능 2

- 설명 : 기준을 관리할 수 있는 메뉴를 만들어 메뉴에 접근하게 되면, 기준 추가, 삭제, 변경 등을 할 수 있게 함

3) 기능 3

- 설명 : 수집된 정보, 정해진 기준에 따라 파일을 분류하고, 이동, 반복

(1) 세부 기능 1

- 설명 : 설정한 기준에 따라 파일들을 분류하고, 각각의 파일을 적절한 디렉토리로 이동함.

(2) 세부 기능 2

- 설명 : 프로그램이 어떤 주기마다 실행 될 것인지 설정함. 시스템의 스케줄러, 프로그램 내에서 sleep 함수 사용.

4) 기능 4

- 설명 : 파일 정리 히스토리를 파일에 저장

(1) 세부 기능 1

- 설명 : 히스토리파일에 접근하여 정리가 완료된기록을 파일에 저장 ("file1.csv" 가 folder1 → folder2 로 이동 되었습니다. / "trash.txt" 가 "기준"에 의하여 folder1 → trash_folder1 로 이동 되었습니다.)

3. 진척사항

1) 기능 구현

(1) 파일 목록 수집 및 정보 분석

- 입출력

입력 : 대상 디렉토리를 사용자로부터 입력받음

출력 : 없음

- 설명

- collect_file_info함수는 사용자로부터 대상 디렉토리를 입력받아 해당 디렉토리 내의 파일들의 정보를 수집하고 분석하는 기능을 구현함
- 함수 내부에서 대상 디렉토리를 열고, 파일들에 접근하여 파일 경로를 가져옴
- 가져온 파일 경로를 통해 파일의 정보 (확장자, 생성일 등)를 분석하여FileInfo구조체에

저장함

- 파일 내용과 제목을 읽어와FileInfo구조체에 저장함

- 적용된 배운 내용

- 반복문, 조건문, 문자열

- 코드 스크린샷

```
// 파일 목록 수집 및 정보 분석
void collect_file_info(FileManager* file_manager) {
    char target_dir[MAX_PATH_LENGTH];
    printf("대상 디렉토리를 입력하세요: ");
    fgets(target_dir, sizeof(target_dir), stdin); // 대상 디렉토리 입력 받음
    target_dir[strcspn(target_dir, "\n")] = '\0'; // 개행 문자 제거

    WIN32_FIND_DATA file_data;
    HANDLE dir_handle;

    char search_pattern[MAX_PATH_LENGTH];
    sprintf(search_pattern, sizeof(search_pattern), "%s\\*", target_dir);

    dir_handle = FindFirstFileA(search_pattern, &file_data);
    if (dir_handle == INVALID_HANDLE_VALUE) {
        fprintf(stderr, "디렉토리를 열 수 없습니다."); // 예외처리
        exit(1);
    }
}
```

```
do {
    if (!!(file_data.dwFileAttributes & FILE_ATTRIBUTE_DIRECTORY)) {
        char file_path[MAX_PATH_LENGTH];
        sprintf(file_path, sizeof(file_path), "%s\\*", target_dir, file_data.cFileName);

        struct stat file_stat;
        if (stat(file_path, &file_stat) == 0) {
            FILE* file;
            if (fopen_s(&file, file_path, "r") == 0) {
                FileInfo file_info;
                strcpy_s(file_info.path, sizeof(file_info.path), file_path);
                strcpy_s(file_info.extension, sizeof(file_info.extension), strrchr(file_data.cFileName, '.') + 1);
                file_info.created = file_stat.st_ctime;
                file_info.size = file_stat.st_size;
                // 파일 정보 분석 및 처리
                fclose(file);
            }
        }
    }
} while (FindNextFileA(dir_handle, &file_data));

FindClose(dir_handle);
```

2) 테스트 결과

(1) 파일 목록 수집 및 정보 분석

- 설명

1. 사용자로부터 대상 디렉토리를 입력받는다

- 사용자에게 "대상 디렉토리를 입력하세요: "라는 메시지를 출력한 후, fgets 함수를 사용하여 사용자로부터 대상 디렉토리 경로를 입력받는다

- 입력된 대상 디렉토리 경로는 target_dir 변수에 저장

2. 대상 디렉토리를 열고, 파일들에 접근하여 파일 경로를 가져옴

- FindFirstFileA 함수를 사용하여 대상 디렉토리의 첫 번째 파일 정보를 가져옴

- 가져온 파일 정보를 통해 현재 파일이 디렉토리인지 아닌지를 확인하고, 파일 경로를 생성

- 생성한 파일 경로를 통해 stat 함수를 사용하여 파일의 상태 정보를 가져옴

3. 파일 정보를 수집하고 FileInfo 구조체에 저장

- FileInfo 구조체를 선언하고, 파일의 경로, 확장자, 생성일, 크기 정보를 해당 구조체에 저장

- 이 과정을 반복하여 대상 디렉토리 내의 모든 파일에 대해 파일 정보를 수집하고 저장

- 테스트 결과 스크린샷

A screenshot of a Windows command prompt window with a black background and white text. The text shows the execution of a program named 'Project_ws.exe' located at 'C:\Users\user\Desktop\현승\23_1학기_학교\C 및 PY\C_프로젝트\Project_ws\x64\Debug\'. The process ID is noted as 20956. The message states that the program has finished execution with code 0. It also provides instructions on how to automatically close the console when debugging is stopped, via the 'Tools' -> 'Options' -> 'Debugging' menu. Finally, it prompts the user to press any key to close the window.

- 아직 완벽하게 구현되지 않음

4. 계획 대비 변경 사항

- 없음

5. 프로젝트 일정

업무		11/3	11/6~12	11/20~26	11/27~12/3
제안서 작성		완료			
기능1	세부기능1		진행중		
기능2	세부기능1			진행중	
	세부기능2				----->

업무		12/4~10	12/11~14	12/15~22
기능3	세부기능1	----->			
	세부기능2		----->		
기능4	세부기능1			----->	