

REPORT

보고서 작성 서약서

1. 나는 타학생의 보고서를 복사(Copy)하지 않았습니다.
2. 나는 타학생의 보고서를 인터넷에서 다운로드 하여 대체하지 않았습니다.
3. 나는 타인에게 보고서 제출 전에 보고서를 보여주지 않았습니다.
4. 보고서 제출 기한을 준수하였습니다.

나는 보고서 작성시 위법 행위를 하지 않고,
성.균.인으로서 나의 명예를 지킬 것을 약속합니다.

과 목 : 전자전기 프로그래밍실습

과 제 명 : HW3

담당교수 : 민 형 복

학 과 : 전자전기공학부

학 년 : 1

학 번 : 2017311583

이 름 : 정해진

제 출 일 : 2018.4.9

1. Introduction

임의의 file을 binary 형식으로 read하고 해당 파일의 원하는 정보를 얻는 방법을 배운다. File을 read할 때 내가 원하는 파일 형식이 아니거나 내가 입력한 파일 경로의 길이가 문제가 있는 경우, 해당하는 문제를 console창에 출력함으로써 오류 보고도 수행한다.

2. Problem Statement

① Describe what is the problem.

- Main 함수 전에 선언한 "getLink" 함수를 모델링하여 원활한 Main 함수의 실행을 꾀한다.

② Describe how do you solve the problem.

임의의 file이 1개 주어진다. 이 파일에 파일의 정보를 담은 binary data가 들어있다.

- Main 함수

getFileName, getInteger, getLink 함수들을 실행한다. 그 중 완성해야 하는 함수는 getLink 함수이다.

- getLink 함수

1. 첫번째 DWORD data와 LinkCLSID를 읽어 들어서 이 file이 Shortcut file임을 확인한다.

(Shortcut file이 아니면, error message를 주고 종료)

2. Header에서 file이 가리키는 것이 folder인지 file인지 확인한다.

3. Header에서 LinkTargetIDList와 LinkInfo의 존재 여부를 확인한다.

(LinkTargetIDList가 존재하면 건너뛰고, LinkInfo가 없으면 error message와 함께 종료)

4. LinkInfo의 Header에서 LocalBasePath와 CommonPathSuffix의 위치를 찾고, 두 값을 읽어들이어 최종 LinkPath를 출력한다.

file을 binary로 읽는 과정에서 필요한 data들을 잘 구분해서 각각 다른 datatype으로써 저장할 필요가 있다. 파일들은 모두 Microsoft에서 제시한 datatype을 따르기 때문에, DWORD, WORD 타입 등을 활용하여 프로그램을 작성하는게 필요하다.

파일을 읽으면서, 프로그램에서 출력해야 하는 Error 종류는 다음과 같다.

1. 입력한 파일명을 가진 파일이 shortcut file이 아닌 경우

-> 해당 파일의 이름을 가진 파일의 첫번째 DWORD data와 LinkCLSID가 shortcut 파일 형식이 아님을 출력한다.

2. 입력한 파일명을 가진 파일이 LinkFlags에서 LinkTargetIDList나 LinkInfo가 누락된 경우

-> 해당 파일의 이름을 가진 파일이 LinkTargetIDList나 LinkInfo가 없으며, shortcut 파일이 아님을 출력한다.

3. 입력한 파일명을 가진 파일이 가리키는, 파일 경로의 길이보다 작은 수치의 buffer값을 입력한 경우

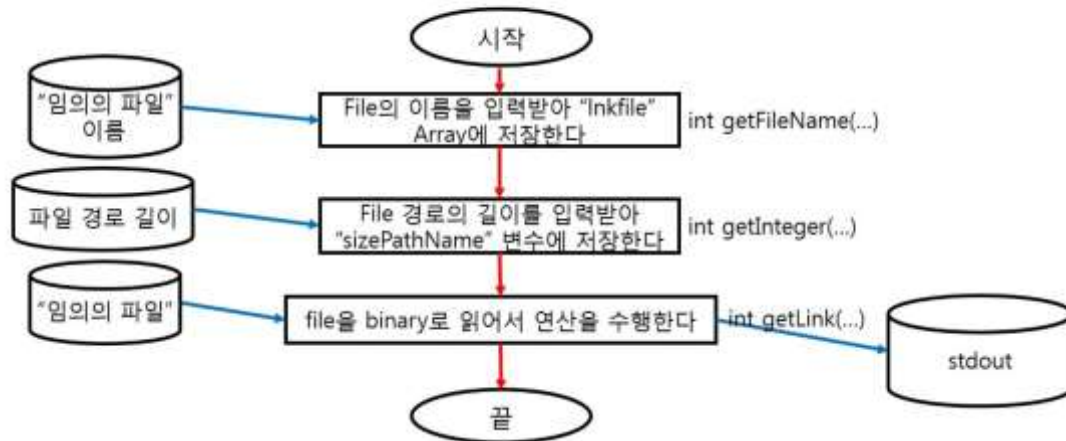
-> 입력한 buffer값과 입력해야 하는 최소 buffer값을 출력한다.

위 1,2,3번 오류들 각각 발생한 경우에 0을 리턴한다.

③ Draw a flowchart of your algorithm

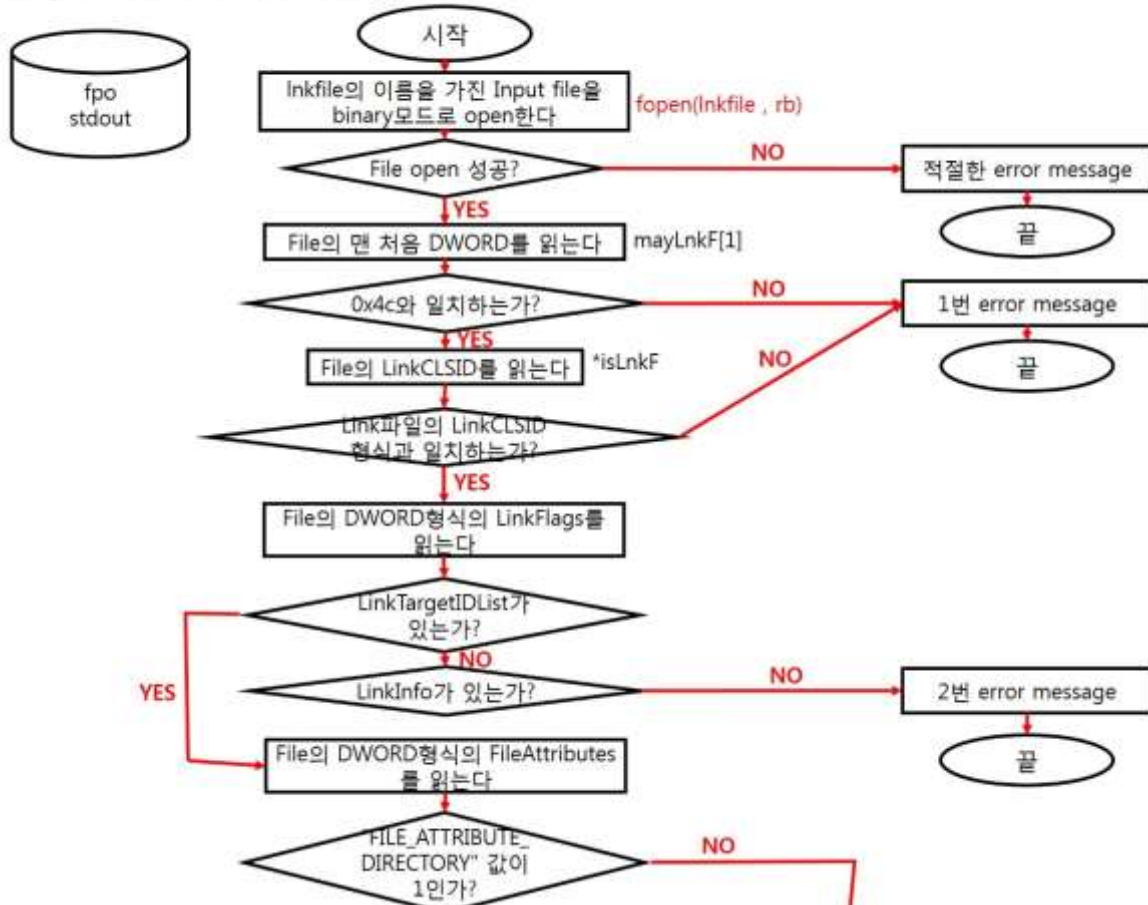
1. Main함수

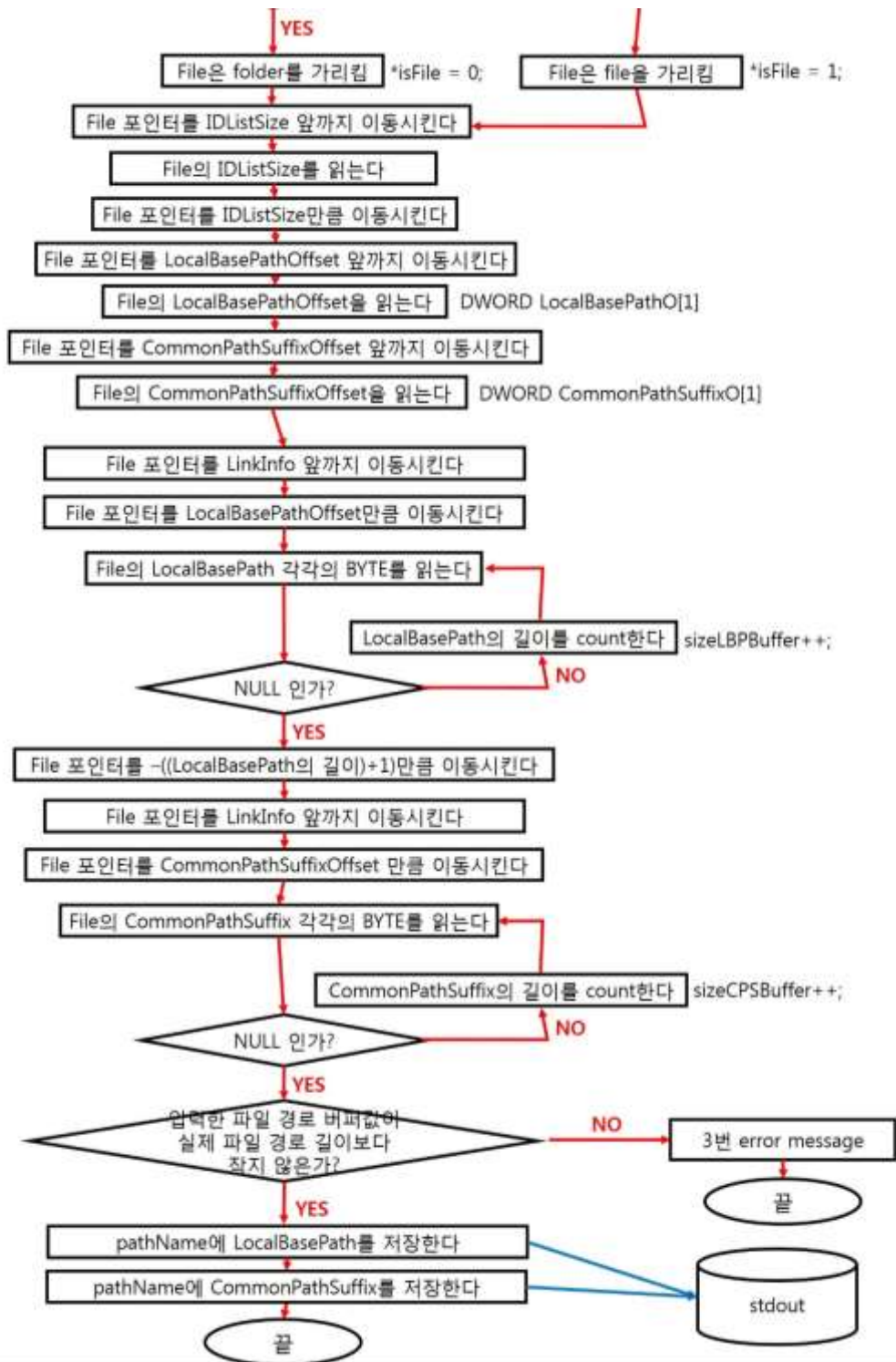
int main(void)



2. getLink함수

**int getLink(char *lnkfile, int *isFile,
char *pathName, int sizePathName)**





3. Implementation

1. Main 함수

임의의 file의 이름을 사용자로부터 입력받아 "lnkfile"이라는 이름의 포인터가 가리키는 Array에 저장한다. 이를 int 자료형의 getFileName 함수를 통해 실행한다. file의 경로의 길이를 사용자로부터 입력받아 "sizePathName"이라는 변수에 저장한다. 이를 int 자료형의 getInteger 함수를 통해 실행한다. 마지막으로 file을 binary로 읽어서 연산을 수행하는 int 자료형의 getLink함수를 실행하고 적절한 출력을 콘솔창에 띄운 다음 프로그램을 종료한다.

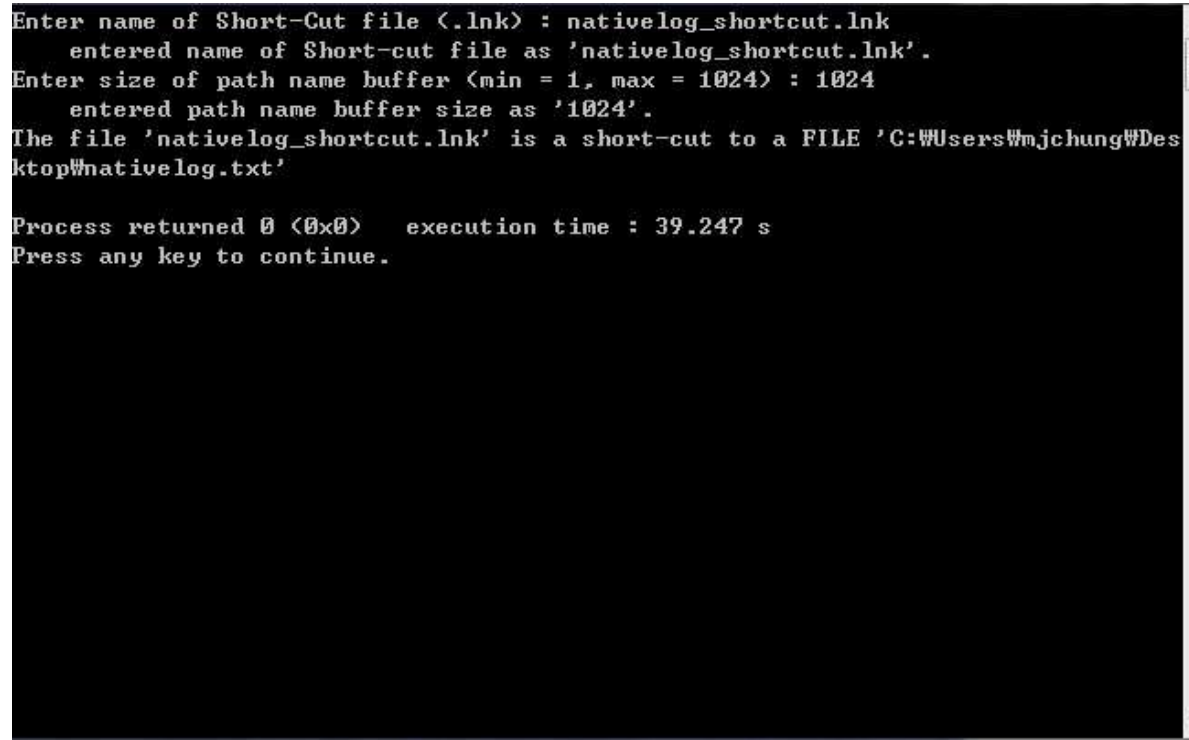
2. getLink 함수

lnkfile이라는 이름을 가진 file을 binary로 연다. File이 존재하지 않으면 적절한 에러메세지를 출력하고 0을 return한다. File의 첫 DWORD 값을 받는다. 이 값이 0x4c와 일치하지 않으면 [WARNING] 메시지를 출력한다. 그 다음으로 바이트 16 길이의 DWORD 값을 받는다. 이 값이 Link 파일이 가지는 LinkCLSID 값과 일치하지 않으면 1번 error 메시지를 출력하고 0을 return한다. File의 DWORD형식의 LinkFlags를 읽고, LinkFlags의 맨 첫번째와 두번째 bit 값을 읽는다. 이 값은 LinkTargetIDList와 LinkInfo의 존재 유무를 판별한다. LinkTargetIDList가 존재하면 if문을 벗어나고 LinkInfo가 존재하지 않으면 2번 error 메시지를 출력하고 0을 return한다. File의 DWORD형식의 FileAttributes를 읽고 FILE_ATTRIBUTE_DIRECTORY 값을 판별한다. 1이면 isFile 값을 0, 0이면 isFile 값을 1로 하여 File이 파일을 가리키는지 폴더를 가리키는지 판별한다.

File이 LinkInfo의 LocalBasePath값과 CommonPathSuffix값을 읽기 위해 불필요한 데이터의 크기만큼 포인터를 이동시키고, LocalBasePath와 CommonPathSuffix의 시작 위치를 읽는다. 그 후, LocalBasePath와 CommonPathSuffix의 길이를 반복문을 통해 측정한다. 만약 사용자로부터 입력받은 "sizePathName"이라는 file의 경로의 길이 buffer가 실제 경로 길이보다 작을 시에 3번 error 메시지를 출력하고 0을 return한다. 마지막으로 LocalBasePath와 CommonPathSuffix를 각각의 길이만큼 읽어 pathName에 이어붙여 저장하고 1을 return한다.

4. Result

"2017311583.JungHaeJin.HW3.c" 를 컴파일 후 실행한 결과이다.



```
Enter name of Short-Cut file (<.lnk>) : nativelog_shortcut.lnk
    entered name of Short-cut file as 'nativelog_shortcut.lnk'.
Enter size of path name buffer <min = 1, max = 1024> : 1024
    entered path name buffer size as '1024'.
The file 'nativelog_shortcut.lnk' is a short-cut to a FILE 'C:\Users\mjchung\Desktop\nativelog.txt'

Process returned 0 (0x0)    execution time : 39.247 s
Press any key to continue.
```

바로가기 파일이 file을 가리킬 경우

file을 가리키며, 그 file의 경로를 원활하게 출력하고 있다.

이 경우에는 "nativelog.txt" 라는 text 파일에 대한 바로가기 파일 분석이다.


```
Enter name of Short-Cut file (<.lnk>) : bin_shortcut.lnk
    entered name of Short-cut file as 'bin_shortcut.lnk'.
Enter size of path name buffer <min = 1, max = 1024> : 1024
    entered path name buffer size as '1024'.
The file 'bin_shortcut.lnk' is a short-cut to a FOLDER 'C:\Users\Wmjchung\Desktop
Wprogram_file(C)\W2017311583.JungHaeJin.HW3\bin'

Process returned 0 (0x0)   execution time : 28.786 s
Press any key to continue.
```

바로가기 파일이 folder를 가리킬 경우

folder를 가리키며, 그 folder의 경로를 원활하게 출력하고 있다.

이 경우에는 "bin" 이라는 폴더에 대한 바로가기 파일 분석이다.

```
Enter name of Short-Cut file (<.lnk>) : main.c
    entered name of Short-cut file as 'main.c'.
Enter size of path name buffer (min = 1, max = 1024) : 1024
    entered path name buffer size as '1024'.
[WARNING] 'main.c' may not be a short-cut file. Header size is '0x636e6923' (expected 0x4c).
[WARNING] Output may be wrong since we do not support this file.
[ERROR] Invalid LinkCLSID of File 'main.c'.
[ERROR] This is not a Windows shortcut file.

Process returned 2 (0x2)    execution time : 11.623 s
Press any key to continue.
```

1번 error에 대한 출력 메시지다.

"main.c"라는 파일은 바로가기 파일이 아니기 때문에 위와 같은 오류 메시지를 출력한다.

```
Enter name of Short-Cut file <.lnk> : bin_shortcut.lnk
    entered name of Short-cut file as 'bin_shortcut.lnk'.
Enter size of path name buffer <min = 1, max = 1024> : 10
    entered path name buffer size as '10'.
[ERROR] Reading pathname from file fails due to small buffer size (10).
[NOTE] We need buffer of size 71 to hold target path name.
Process returned 2 (0x2)    execution time : 25.583 s
Press any key to continue.
```

3번 error에 대한 출력 메시지다.

“bin”이라는 폴더의 경로 길이는 71 이기 때문에 경로 길이 buffer size를 10으로 입력했을 경우 위와 같은 오류 메시지를 출력한다.

2번 error에 대한 콘솔 결과 창은 준비할 수 없었다.

필자의 수준으로는 특정 파일의 포렌식을 변경하여 바로가기 파일이었던 파일을 다른 형식의 파일로 변경하여 시험해볼 수 없었기 때문이다.

하지만 “2017311583.JungHaeJin.HW3.c” 에는 2번 error에 대한 출력 메시지를 원활히 출력하게 하는 명령어가 쓰여 있다.

5. Conclusion & Evaluation

실습을 통해 file을 binary 형식으로 다루는 과정을 좀 더 이해하게 된 것 같다. 파일을 binary로 다룬다는 것이 생소해서 c파일을 작성하는데 어려움이 있었지만, [1]과 관련한 강의 내용을 떠올리며 찬찬히 작성해보고, [2]을 통해 파일의 포렌식을 이해하고 적절히 명령문을 대입하였더니 잘 되었다. 앞으로 이 실습을 진행하는 학생들은 [1]에서 명령문을 작성하는데 많은 힌트, [2]에서 fseek 함수를 통해 이동해야 하는 포인터 위치들을 정확하게 하는 힌트를 얻을 수 있을 것이다.

6. 참고 문헌

[1] Min, H. B. and SKKU, “textfile.pdf”

[2] Microsoft, “[MS-SHLLINK]: Shell Link (.LNK) Binary File Format”, September 15, 2017