

Report

운영체제 3분반

2번 과제: 파일시스템 시뮬레이터



이름	이창현
학번	32193605
학과	소프트웨어학과
과목	운영체제 3분반
교수명	김하연
날짜	2024-6-3
이메일	32193605@dankook.ac.kr
전화	01085319558

프로젝트 설계

파이썬을 이용해서 간단한 파일시스템 시뮬레이터를 CLI와 GUI로 설계했다.

프로젝트의 기본 요구사항에 맞춰 설계한 파일시스템 시뮬레이터는 파일 생성, 삭제, 읽기, 쓰기, 디렉토리 생성, 삭제, 이동 및 파일 검색 기능을 수행할 수 있다.

CLI 프로그램의 경우 아래와 같은 방법으로 사용할 수 있다.

create <파일이름> <내용>: 새 파일을 생성을 생성한다. 파일이름은 필수적이고 내용에 아무 입력도 하지 않을 시에는 빈파일로 생성된다.

delete <파일이름>: 파일을 삭제한다. 파일이 존재하지 않을 경우에는 에러 메시지박스를 띄운다.

read <파일이름>: 파일을 읽어와서 출력한다. 파일이 존재하지 않을 경우에는 에러 메시지박스를 띄운다.

write <파일이름> <내용>: 파일에 추가한다. 존재하지 않는 파일일 경우에 새로 파일을 생성할 수 있다.

mkdir <폴더이름>: 새 폴더를 생성한다. 이미 존재하는 파일의 이름일 경우에는 (1), (2), ...(n)의 숫자를 부여하여 파일을 생성한다.

rmdir <폴더이름>: 폴더를 삭제한다. 폴더가 존재하지 않을 경우에는 에러 메시지박스를 띄운다.

cd <경로>: 경로를 이동한다. 경로가 존재하지 않을 경우에는 에러 메시지박스를 띄운다.

search <파일이름>: 파일을 찾아서 경로를 출력한다. 파일이 존재하지 않을 경우에는 에러 메시지박스를 띄운다.

mv <파일이름/폴더이름> <경로>: 파일이나 폴더를 이동합니다. 파일이 존재하지 않을 경우에는 에러 메시지박스를 띄운다.

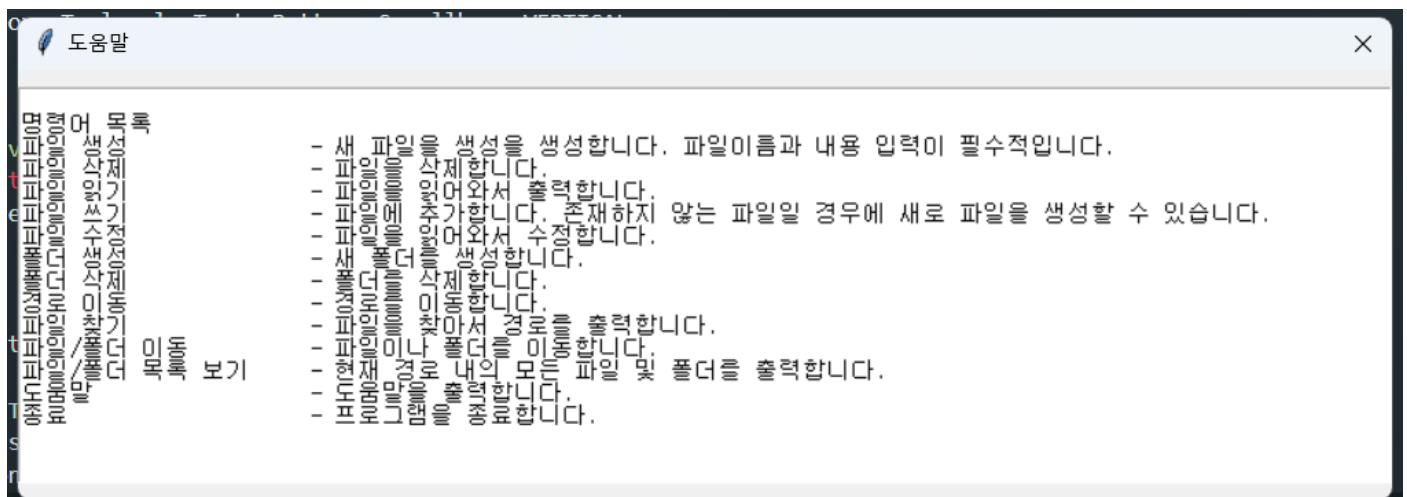
list: 현재 경로 내의 모든 파일 및 폴더를 출력한다.

help: 도움말을 출력한다.

exit: 프로그램을 종료한다.

GUI 프로그램의 구체적인 사용방법은 도움말 버튼을 눌러서 확인할 수 있다.

과제의 최종 목표는 CLI 프로그램이 아닌 GUI 프로그램이므로 아래 내용은 GUI 프로그램을 기준으로 설명한다.



주요 구현 사항은 다음과 같다.

파일 생성 - 새 파일을 생성을 생성합니다. 파일이름과 내용 입력이 필수적입니다.

파일 삭제 - 파일을 삭제합니다.

파일 읽기 - 파일을 읽어와서 출력합니다.

파일 쓰기 - 파일에 추가합니다. 존재하지 않는 파일일 경우에 새로 파일을 생성할 수 있습니다.

파일 수정 - 파일을 읽어와서 수정합니다.

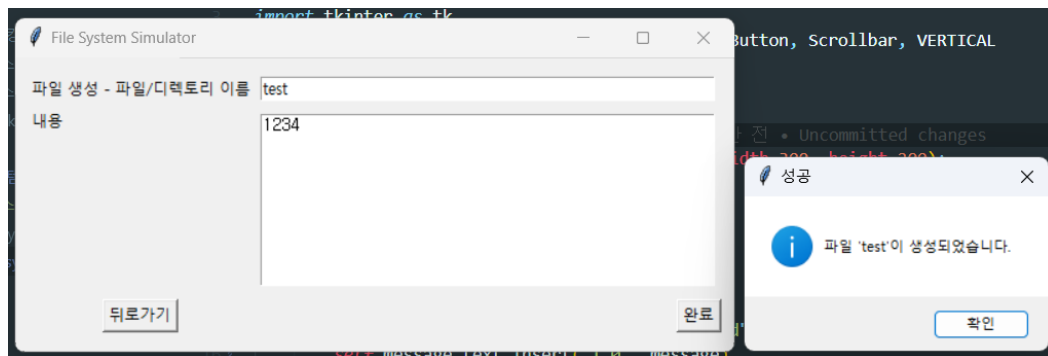
폴더 생성 - 새 폴더를 생성합니다.

폴더 삭제 - 폴더를 삭제합니다.

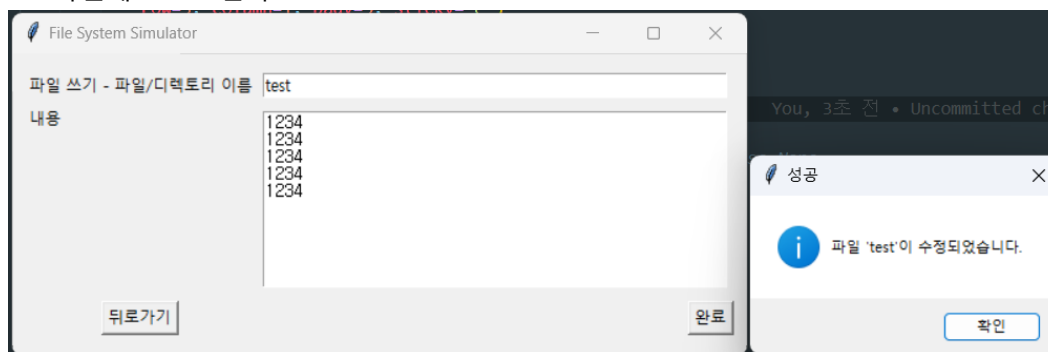
- | | |
|-------------|-------------------------------|
| 경로 이동 | - 경로를 이동합니다. |
| 파일 찾기 | - 파일을 찾아서 경로를 출력합니다. |
| 파일/폴더 이동 | - 파일이나 폴더를 이동합니다. |
| 파일/폴더 목록 보기 | - 현재 경로 내의 모든 파일 및 폴더를 출력합니다. |
| 도움말 | - 도움말을 출력합니다. |
| 종료 | - 프로그램을 종료합니다. |

실행결과

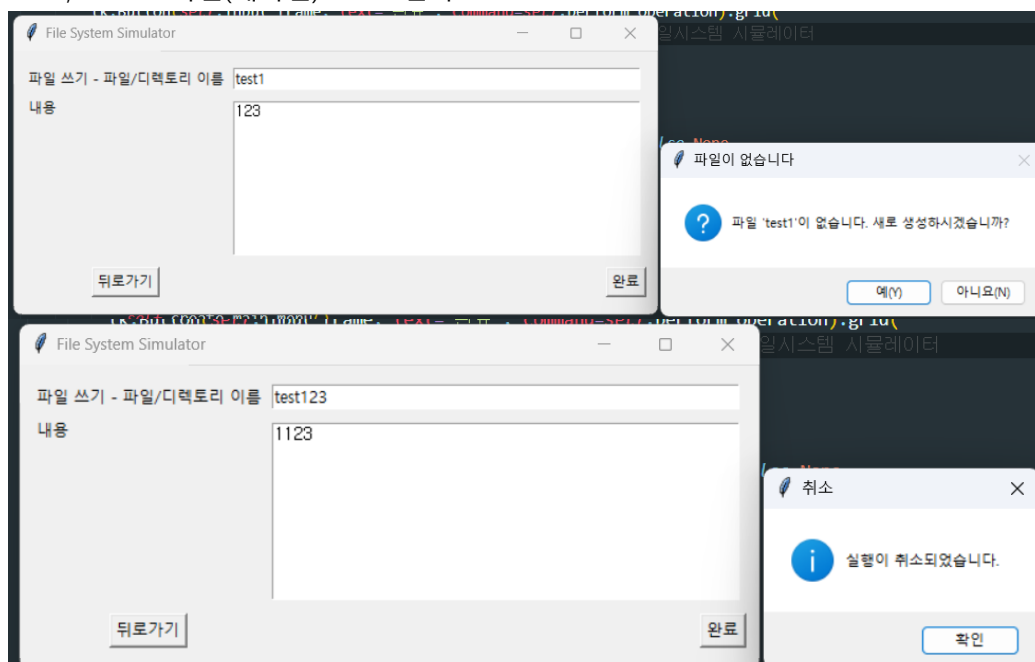
test파일 create 결과



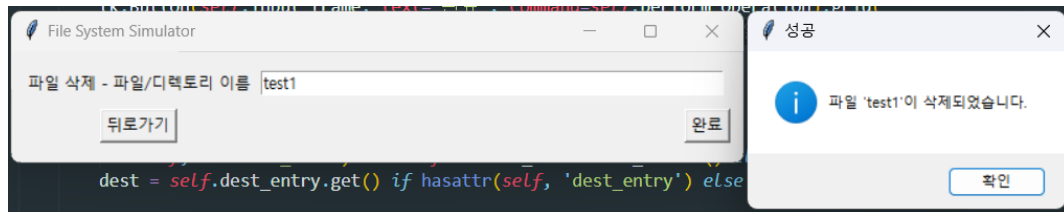
test파일에 write 결과



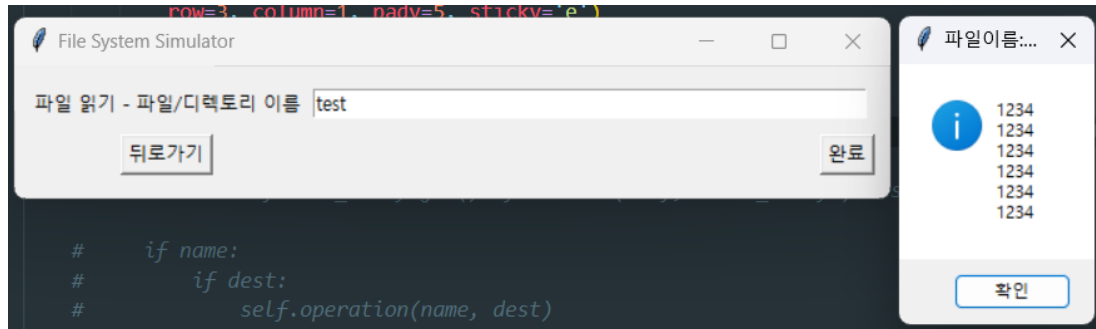
test1, test123파일(새파일) write 결과



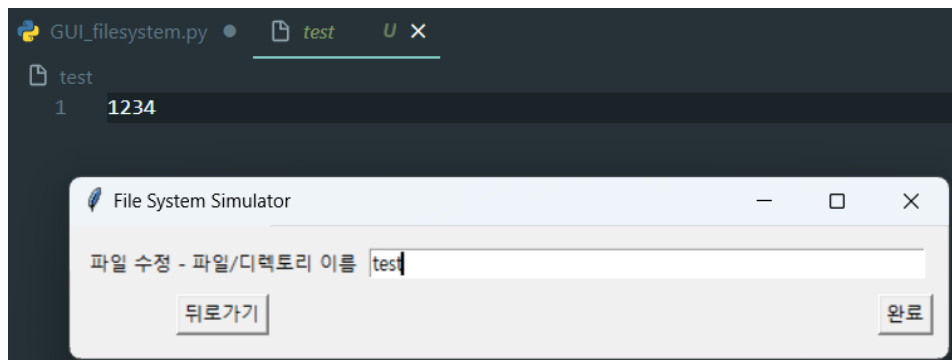
test1파일 delete결과



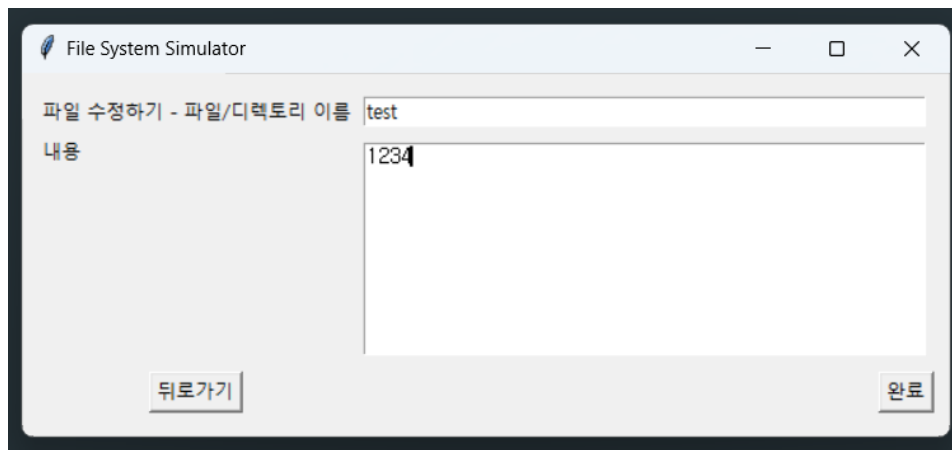
test파일 read결과



test파일 edit 결과

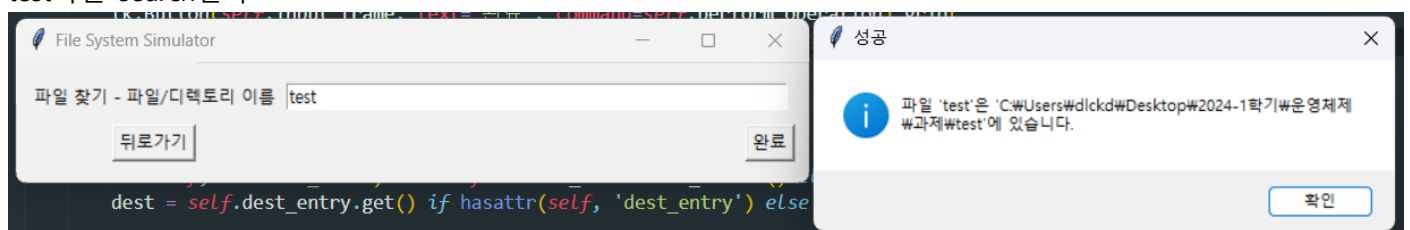


1234가 입력된 test파일을 수정하기 위해 수정하려는 파일의 이름을 입력한다.

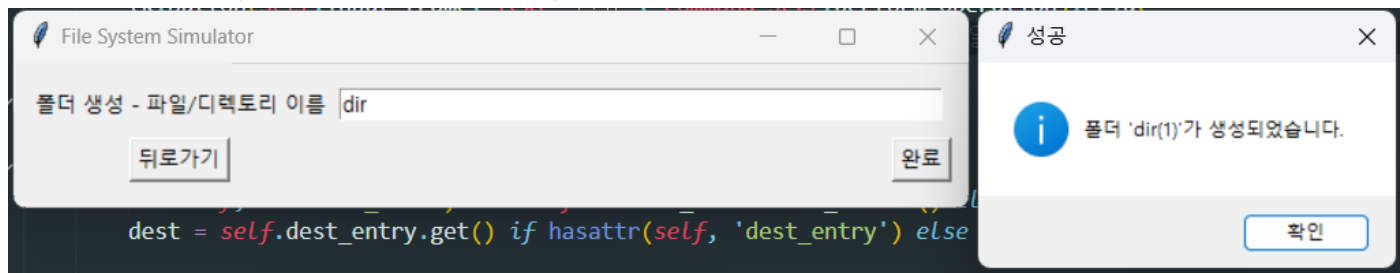


파일 시스템 시뮬레이터가 test파일을 읽어와서 test파일의 내용을 text 입력창에 띄운다. 사용자는 이를 자유롭게 수정해서 단순히 파일쓰기(append)를 넘어서 파일수정(edit)이 가능하다.

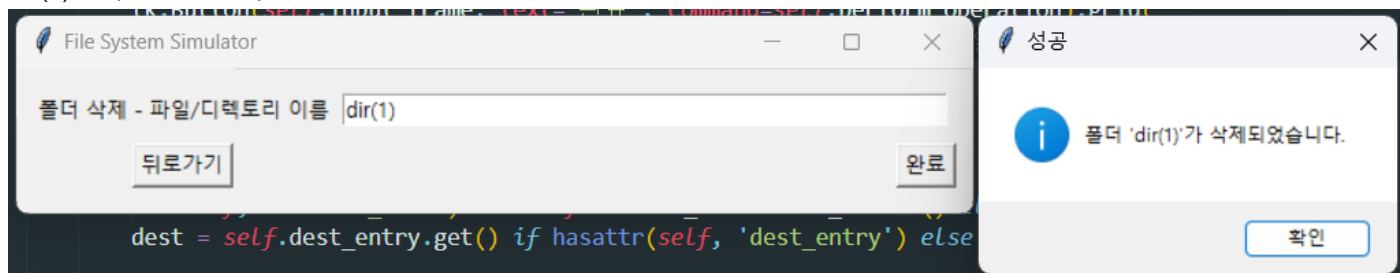
test파일 search결과



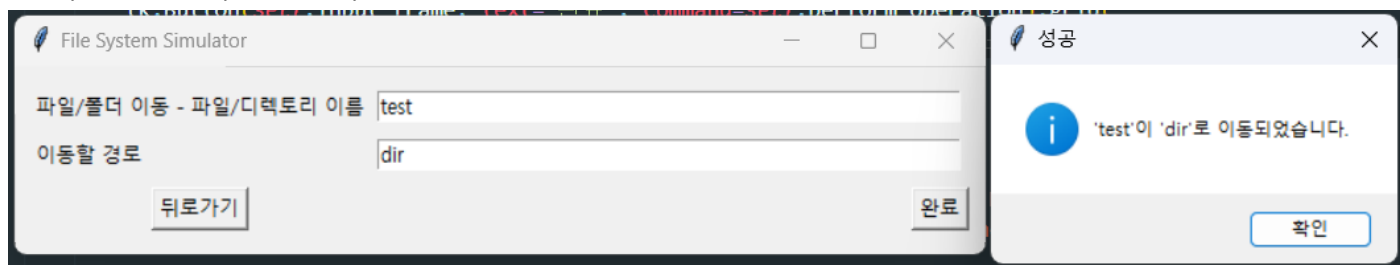
dir폴더 create결과(이미 dir폴더가 존재할 때)



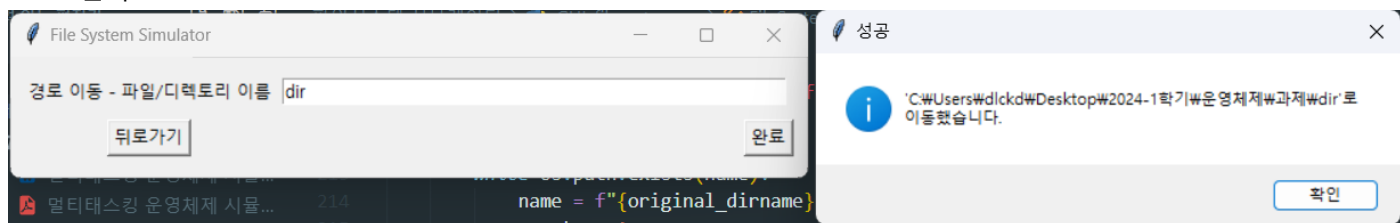
dir(1)폴더 delete결과



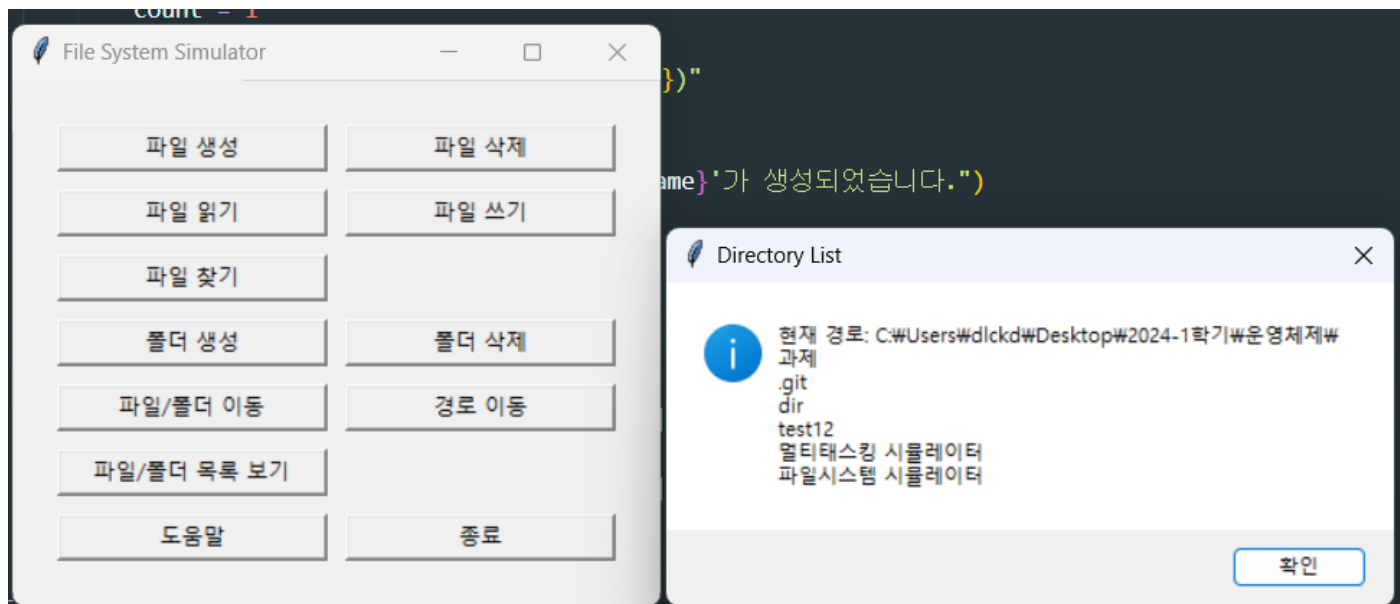
test파일을 dir폴더로 mv결과



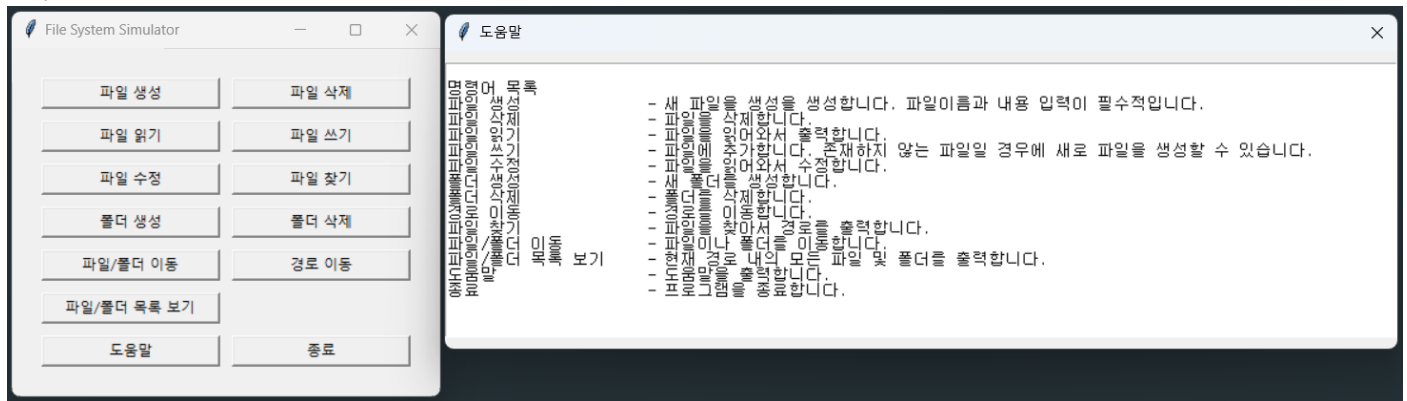
cd dir결과



list결과



help결과



비고

파일 수정 실행 시에 아래와 같은 에러가 발생한다.

```
PS C:\Users\dlckd\Desktop\2024-1학기\운영체제\과제> c::; cd 'c:\Users\dlckd\Desktop\2024-1학기\운영체제\과제'; & 'c:\Users\dlckd\anaconda3\python.exe' 'c:\Users\dlckd\.vscode\extensions\ms-python.debugpy-2024.6.0-win32-x64\bundled\libs\debugpy\adapter\..\..\debugpy\launcher' '52045' '--' 'c:\Users\dlckd\Desktop\2024-1학기\운영체제\과제\파일시스템 시뮬레이터\GUI_filesystem.py'
Exception in Tkinter callback
Traceback (most recent call last):
  File "c:\Users\dlckd\anaconda3\lib\tkinter\_init_.py", line 1921, in __call__
    return self.func(*args)
  File "c:\Users\dlckd\Desktop\2024-1학기\운영체제\과제\파일시스템 시뮬레이터\GUI_filesystem.py", line 177, in perform_operation
    self.operation(name, content)
  File "c:\Users\dlckd\Desktop\2024-1학기\운영체제\과제\파일시스템 시뮬레이터\GUI_filesystem.py", line 224, in edit_file_operation
    self.content_text.insert(1, content)
  File "c:\Users\dlckd\anaconda3\lib\tkinter\_init_.py", line 3782, in insert
    self.tk.call((self.w, 'insert', index, chars) + args)
tkinter.TclError: bad text index "1"
```

이 에러는 Tkinter의 Text 위젯에서 발생하는 것으로 insert 메서드에 전달된 인덱스가 1로 전달되는데 보통 Tkinter의 Text 위젯에서는 '1.0'과 같은 형식으로 지정되기 때문에 발생하는 에러이다. 이는 프로그램 기능 수행에는 문제는 없지만 코드 정확성을 위해서 수정할 필요는 있는 에러이다.

이를 수정하기 위해 해당 인덱스를 '1.0'으로 수정하였다. 그 결과 설계대로라면 2번째 단계인 '파일 수정하기' 단계로 넘어가야 하지만, 실제로는 다시 초기화면으로 돌아가는 치명적인 에러가 발생하게 된다.

실제로 프로그램 작성과정에서 '파일 수정'은 프로그램이 완성된 후에 추가적으로 고안해서 추가한 부분이다. 즉, 프로그램 원래 설계과정에서 포함되어 있던 코드가 아니었기 때문에 다른 기능들은 1단계만에 프로세스를 완료하지만 '파일 수정' 기능은 파일 이름 입력 -> 파일 이름으로부터 context 추출 후 수정 2단계 과정으로 이루어져 있다. 이를 추가하는 과정에서 코드를 수정하다 보니 난잡한 코드가 생성된 것 같다.

따라서 이를 수정하기 위해 test.py라는 파일을 생성해서 open_input_window함수에 name인자를 추가해주고, perform_operation함수에 self.operation(name, content)를 추가하여 인덱스를 edit_file_operation함수 내부에서 text.insert를 할 필요가 없도록 수정했다. 그 결과 기능 수행 후 메시지 박스가 2번씩 호출되고, 아래와 같은 추가적인 에러가 발생하는 추가적인 에러가 발생했다. 첨부된 에러는 사용에 문제가 없고 해결하기 위해서는 perform_operation함수에 content가 None일 때와 아닐 때를 처리해주면 해결될 거라 생각하지만, 메시지 박스 2번 호출되는 에러는 실제 사용에 문제가 되는 에러이다. 이러한 알 수 없는 에러를 잡기 위해서는 디버깅이 필요하다. 그 과정에서 시간이 많이 소요될 것 같아서 최초의 에러인 Tkinter Text위젯에서 발생하는 에러도 실제로 사용에 있어서는 문제가 없기 때문에 해당 버전으로 rollback을 해서 과제를 마무리하였다.

```
'c:\Users\dlckd\.vscode\extensions\ms-python.debugpy-2024.6.0-win32-x64\bundled\libs\debugpy\adapter\..\..\debugpy\launcher' '54660' '--' 'c:\Users\dlckd\Desktop\2024-1학기\운영체제\과제\파일시스템 시뮬레이터\test.py'
Exception in Tkinter callback
Traceback (most recent call last):
  File "c:\Users\dlckd\anaconda3\lib\tkinter\__init__.py", line 1921, in __call__
    return self.func(*args)
  File "c:\Users\dlckd\Desktop\2024-1학기\운영체제\과제\파일시스템 시뮬레이터\test.py", line 179, in perform_operation
    self.operation(name, content)
  File "c:\Users\dlckd\Desktop\2024-1학기\운영체제\과제\파일시스템 시뮬레이터\test.py", line 230, in edit_file_save_operation
    f.write(content.strip())
AttributeError: 'NoneType' object has no attribute 'strip'
```