



9TEAM

VISUAL SHELL EXPLORER

NO CLI, YES GUI.

PRESENTATION OUTLINE

Why should it be CLI?

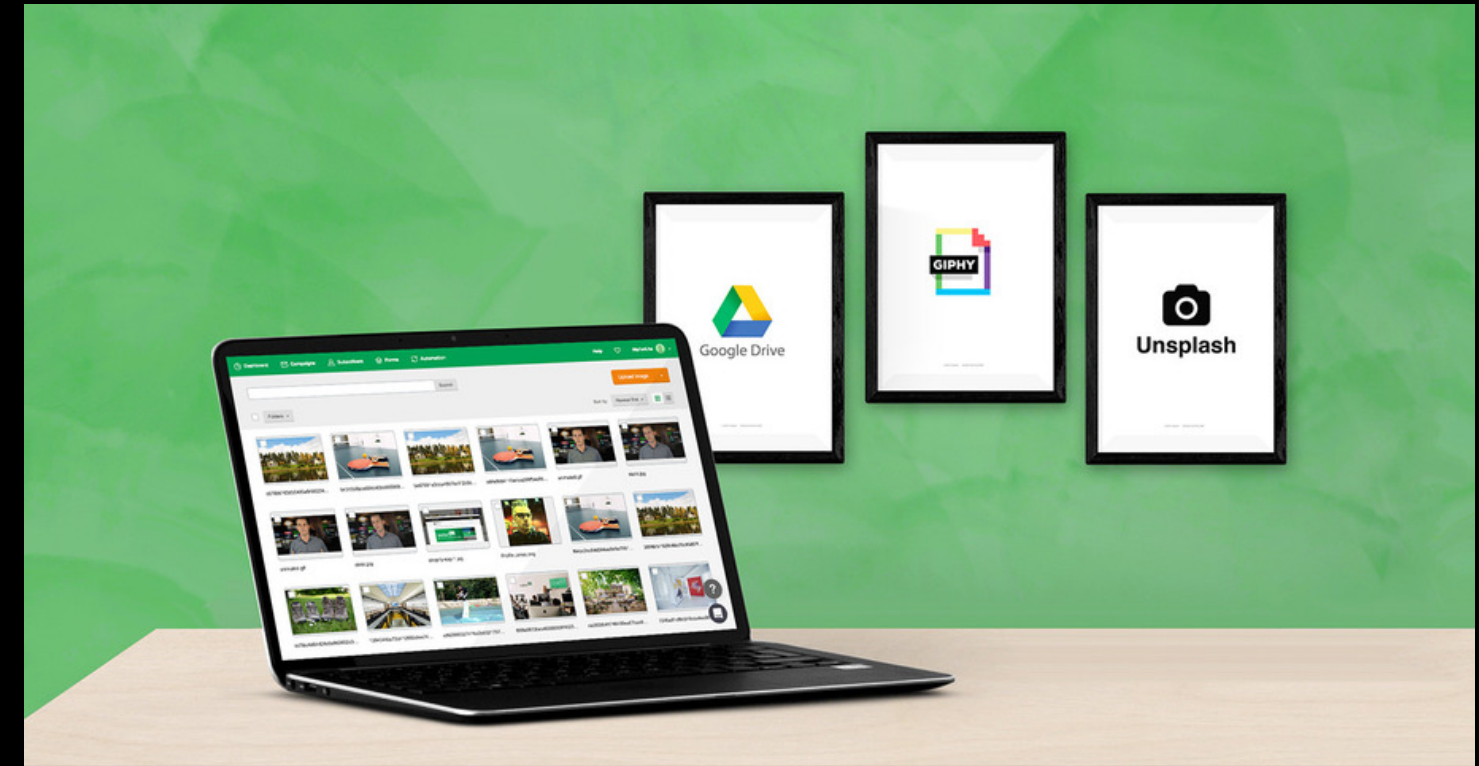


CLI(Command Line Interface)
라 불리는 이 환경은 이것을 처음 접하는
초보사용자 스스로 익숙해지는데 많은
시간이 걸리게 하며 최악의 경우 사용자가
사용의지를 완전히 잃게 되는 일도 생긴다.
또한, 능숙한 사용자라 할지라도 복잡한
명령어들이 많아 나중에는 잊어버릴 수
있으며 명령을 직접 타이핑 하므로 오타에
의한 명령 실행 오류가 발생할 수 있다.
즉CLI환경은 직관적이지 못할 뿐만 아니라
진입장벽이 높다는 큰 단점을 가지고 있다.



THEREFORE...

We must use the GUI.

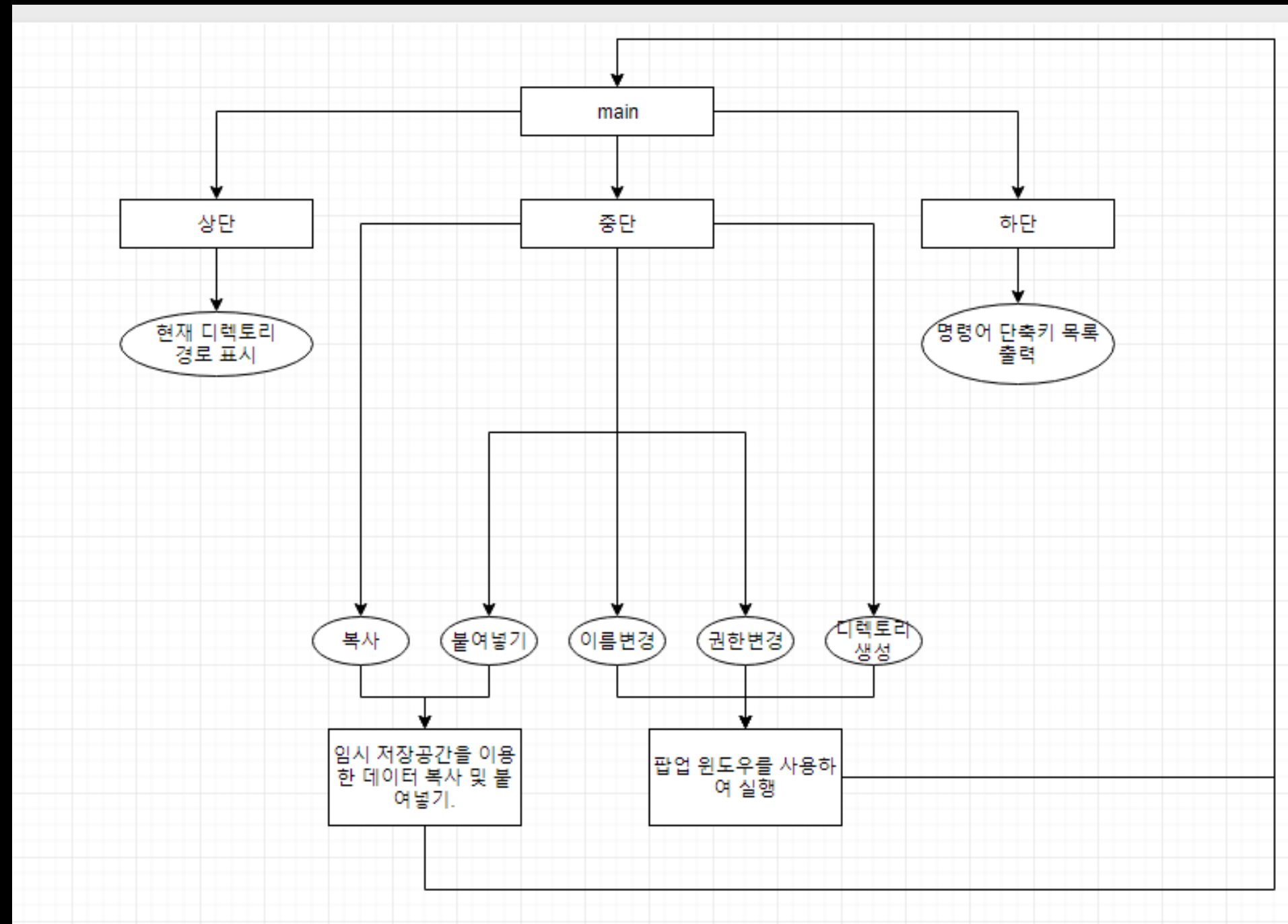


사용자의 편의성을 증대시키기 위해 GUI환경을 제공하는 파일탐색기를 이용해야한다.
대표적인 파일탐색기로 Windows의 탐색기, Ubuntu의 File Manager가 있다.

이런 파일탐색기 기능을 구현하였고 덧붙여 기존의 탐색기와는 다른 우리가 직접
써보면서 불편함을 느낀 부분을 더 추가하였다.
예를들어 리눅스의 TEXT EDITOR중에 하나인 NANO처럼 창 아래 쪽에 단축키 목록을
보여주어 처음사용자도 바로바로 큰 어려움없이 사용할 수 있도록 하였다.

VISUAL SHELL EXPLORER을 실행하면 사용자는 현재 디렉토리내의 목록을 볼 수 있고
방향키와 엔터키를 통해 디렉토리를 이동할 수 있다.
또한 현재 선택된 파일및 디렉토리를 하단에 있는 단축키 목록을 보고 여러가지
기능을 수행할 수 있다.

PROGRAM CONCEPT figure



INITIAL LANUCH SCREEN

키보드 방향키를 이용해 상하좌우로 이동이 가능하고 ENTER을 눌렀을 때 선택된 객체가 디렉토리라면 선택한 디렉토리로 이동.

화면의 좌상단에 보이는 "." 을 선택하면 현재 디렉토리에서 상위 디렉토리로 이동.

하단 블록 부분에 단축키가 명시되어 있지만 현재 구동 가능한 단축키는 디렉토리생성 명령어인 Mkdir과 디렉토리및 파일 이름 변경 명령어 Rename만 가능.

프로그램을 종료 시키려면 F12키를 입력하면 종료된다.

개체	이름 / 크기	종료 : [F12]
.	[디렉토리]	vs 80.0.B
..	[디렉토리]	vs1.c 32.0.B
.vs2.c.swp	32.0.B	vs1_exe 32.0.B
a.out	32.0.B	vs2 48.0.B
gggg.c	8.0.B	vs2.c 24.0.B
test.c	160.0.B	vs2_rename,mkdir.c 40.0.B
visualShell.c	56.0.B	vs3 48.0.B
visualShell_exe	32.0.B	백은규 [디렉토리]

Ctrl+ Copy(c) Move(o) Paste(v) Rename(n) Chmod(h) Mkdir(k) Delete(d)

INITIAL LANUCH SCREEN

개체의 이름을 입력받으면 크기를 계산하고
BYTE 사이즈를 KB, MB, GB 단위로 축약해주는
entity_size_calc()함수를 이용하여 디렉토리 하
위 목록에서 파일이 있을시 그 파일의 크기를 명
시한다.

디렉토리 하위 목록에 디렉토리가 있을시에는 "디
렉토리"라고 표기한다.

```
for(i=0; i < *n_choices;i++){
    pce_rtn_val = print_current_entity((((*d_ptrArray)[i])->d_name), 1, NULL);

    switch(pce_rtn_val){
        case 0:
            entity_size_calc( ((*d_ptrArray)[i])->d_name, (*rtn_entity_size_chr)[i]);
            (*my_items)[i] = new_item((((*d_ptrArray)[i])->d_name, (*rtn_entity_size_chr)[i]);
            break;
        case 1:
            (*my_items)[i] = new_item((((*d_ptrArray)[i])->d_name, "[디렉토리]");
            break;
        case 2:
            entity_size_calc( ((*d_ptrArray)[i])->d_name, (*rtn_entity_size_chr)[i]);
            (*my_items)[i] = new_item((((*d_ptrArray)[i])->d_name, (*rtn_entity_size_chr)[i]);
            set_item_userptr( (*my_items)[i], execFunc );
            break;
    }
}
```

INITIAL LAUNCH SCREEN

menu 라이브러리의 `set_menu_mark` 함수를 이용하여 주어진 메뉴에 대한 표시 문자열을 #으로 설정한다.

menu 라이브러리의 `set_menu_format` 함수를 이용하여 주어진 메뉴의 최대 디스플레이를 설정한다. 디렉토리의 하위 목록이 많으면 13행 2열이거나 13행 1열로 한다.

(기본형식은 16행 1열)

`set_menu_spacing` 함수를 이용하여 메뉴 아이템과 설명의 간격을 3, 아이템간 열간격을 5로 공백을 넣어 설정한다.

`menu_opts_off` 함수를 이용하여 아이템간 다중선택이 가능하도록 `O_ONEVALUE` 옵션과 메뉴가 열로서 정렬되도록 `O_ROWMAJOR` 옵션을 준다. 또한 루트디렉토리가 아닐때 `".."`이 선택되지 않도록 `O_SELECTABLE` 옵션을 준다.

```
set_menu_mark(*my_menu, "# ");

if(max_y-5 < *n_choices ){
    set_menu_format(*my_menu, max_y-4, 2);
} else {
    set_menu_format(*my_menu, max_y-4, 1);
}

set_menu_spacing(*my_menu, 3, 0, 5 );

menu_opts_off(*my_menu, O_ONEVALUE);

menu_opts_off(*my_menu, O_ROWMAJOR);

if(strcmp(absDIRstr, "/") != 0)
{
    item_opts_off((*my_items)[0], O_SELECTABLE);
}

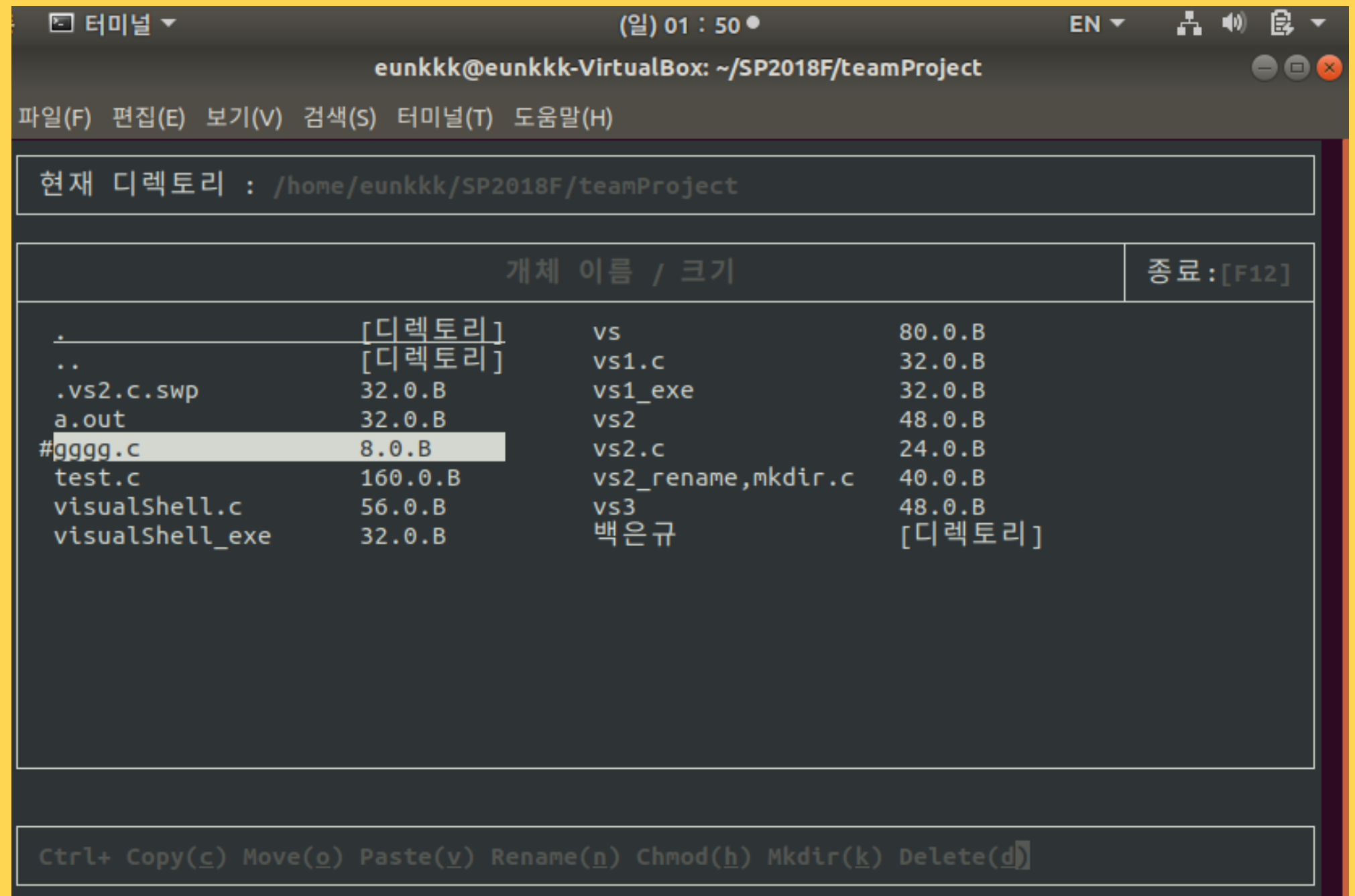
post_menu(*my_menu);
```


RENAME

파일 목록중 test.txt파일에 커서를 두고
Ctrl + n 버튼을 눌러 Rename명령시

새로 지정할 이름에 finaltest.txt입력후 확인

파일이름이 새로 지정된것을 볼 수 있다.



The screenshot shows a terminal window titled "터미널" (Terminal) with the user "eunkkk" at "eunkkk-VirtualBox" in the directory "~/SP2018F/teamProject". The current directory is "/home/eunkkk/SP2018F/teamProject". A file listing is displayed with columns for "개체 이름 / 크기" (Object Name / Size) and "종료 : [F12]" (End : [F12]). The file "test.c" is highlighted. Below the listing, the command "Ctrl+ Copy(c) Move(o) Paste(v) Rename(n) Chmod(h) Mkdir(k) Delete(d)" is shown.

개체 이름 / 크기	종료 : [F12]
.	[디렉토리]
..	[디렉토리]
.vs2.c.swp	32.0.B
a.out	32.0.B
#gggg.c	8.0.B
test.c	160.0.B
visualShell.c	56.0.B
visualShell_exe	32.0.B
vs	80.0.B
vs1.c	32.0.B
vs1_exe	32.0.B
vs2	48.0.B
vs2.c	24.0.B
vs2_rename, mkdir.c	40.0.B
vs3	48.0.B
백은규	[디렉토리]

Ctrl+ Copy(c) Move(o) Paste(v) Rename(n) Chmod(h) Mkdir(k) Delete(d)

RENAME

글꼴 및 글자색 스타일을 적용시켜
버튼 만들기

```
void Dlg_Rename(WINDOW *dlg_win_rename, const char *prevName, char *newName)
{
    box(dlg_win_rename, 0, 0);

    wattron(dlg_win_rename, A_BOLD);
    mvwprintw(dlg_win_rename, 0, 5, " 이름 변경 (Rename)");
    wattroff(dlg_win_rename, A_BOLD);

    attr_style = A_BOLD | COLOR_PAIR(8);
    wattron(dlg_win_rename, attr_style);
    mvwprintw(dlg_win_rename, 2, 3, " 이전 이름  ");

    mvwprintw(dlg_win_rename, 4, 3, " 새로운 이름 ");

    mvwprintw(dlg_win_rename, 6, 24, "  확 인  ");

    mvwprintw(dlg_win_rename, 6, 40, "  취 소  ");
    wattroff(dlg_win_rename, attr_style);

    attr_style = A_BOLD | COLOR_PAIR(7);
    wmove(dlg_win_rename, 2, 16);

    wattron(dlg_win_rename, attr_style);
}
```



RENAME

사용자로부터 변경할 디렉토리 및 파일명을 입력받고 echo() 함수를 이용해 화면에 출력

```
void Dlg_Rename(WINDOW *dlg_win_rename, const char *prevName, char *newName)
{
    wattroff(dlg_win_rename, attr_style );

    for( i=16; i<max_x-2; i++)
    {
        mvwaddch(dlg_win_rename, 2, i, ' ');
        mvwaddch(dlg_win_rename, 4, i, ' ');
    }
    mvwprintw(dlg_win_rename, 2, 17, prevName);

    echo( );
}
```

RENAME

사용자로부터 방향키 입력이 들어오면
그에따라 실행이 되어야하는 함수를 실행
UP키 같은경우 확인버튼에서 다시 입력을
받아야하므로 위에 for문을 다시실행

```
void Dlg_Rename(WINDOW *dlg_win_rename, const char *prevName, char *newName)
{
    while( ( ipt=wgetch(dlg_win_rename) )!='\n')
    {
        switch (ipt)
        {
            case KEY_LEFT :
                wmove(dlg_win_rename, 6, 24);
                wchgat(dlg_win_rename, 9, A_BOLD|A_UNDERLINE, 8, NULL );
                mvwchgat(dlg_win_rename, 6, 40, 9, A_NORMAL, 8 ,NULL);
                wmove(dlg_win_rename, 6, 24);
                break;

            case KEY_RIGHT:
                wmove(dlg_win_rename, 6, 40);
                wchgat(dlg_win_rename, 9, A_BOLD|A_UNDERLINE, 8, NULL );
                mvwchgat(dlg_win_rename, 6, 24, 9, A_NORMAL, 8 ,NULL);
                wmove(dlg_win_rename, 6, 40);

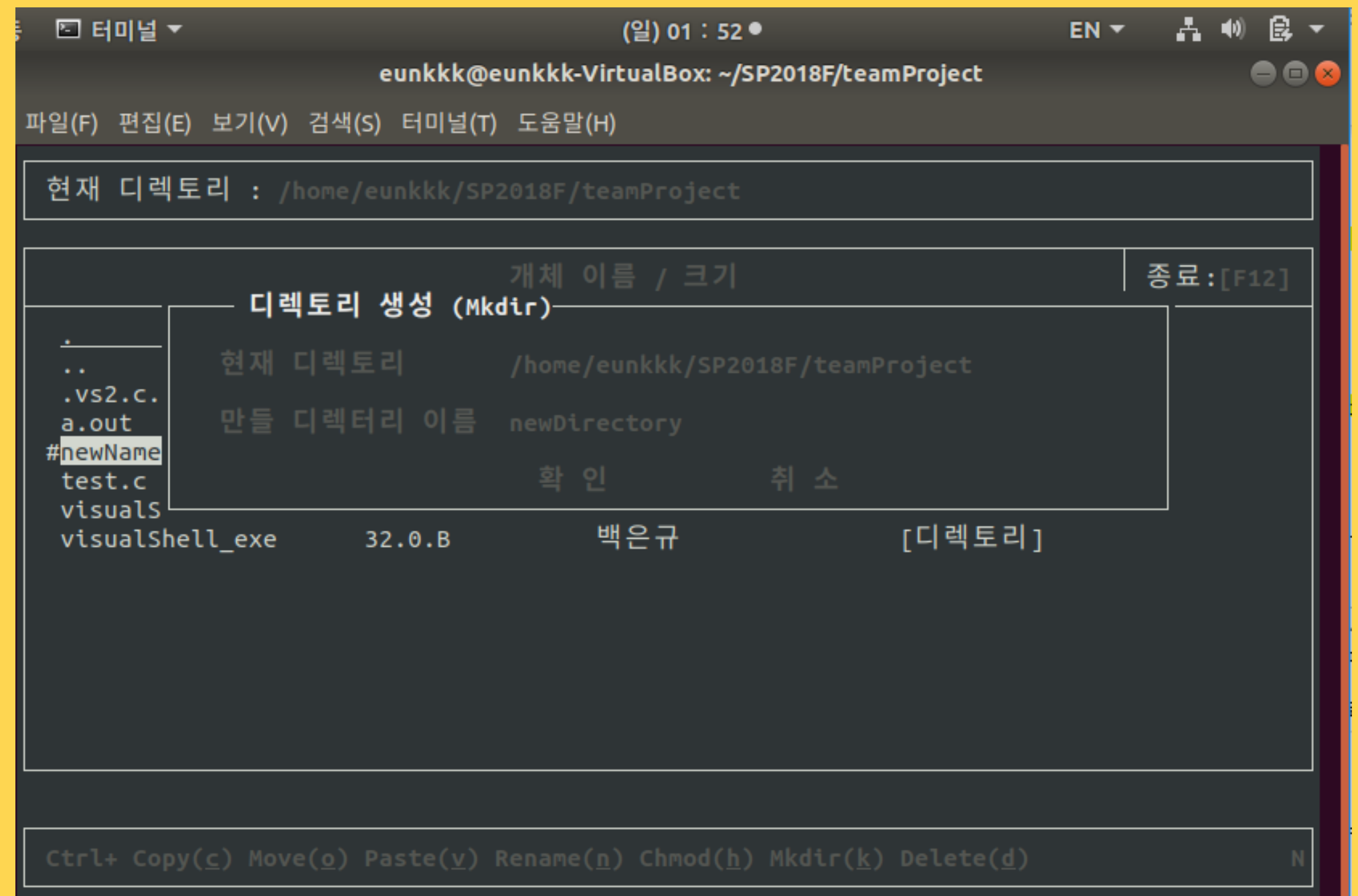
                break;
            case KEY_UP:
                mvwchgat(dlg_win_rename, 6, 24, 9, A_NORMAL, 8 ,NULL);
                mvwchgat(dlg_win_rename, 6, 40, 9, A_NORMAL, 8 ,NULL);

                watttrn(dlg_win_rename, attr_style );
        }
    }
}
```

MKDIR

새로 디렉토리 및 파일 생성 명령어인
Ctrl + k 입력후 디렉토리 이름을 testDir로
입력후 확인

testDir디렉토리가 생성된 것을 확인할 수 있다.



PROBLEMS AND SOLUTIONS

Problem 1.

코어 덤프로 인해 제대로 출력이 되지 않는 문제가 발생

> 메모리 할당 초기화와 배열 길이를 초과해서 계산 하려는 부분을 개선하여 해결

Problem 2.

한글이 제대로 출력되지 않은 문제가 발생

> locale.h 헤더파일을 가져와 setlocale() 함수를 이용하여 한글을 표기할 수 있게 해결





GITHUB.COM/BAEKEUNKYU/SP_
PROJECT_TEAM9.GIT

