

## 기본 개념

lf는 Go로 작성된 터미널 파일 관리자입니다. ranger에서 큰 영감을 받았으며, 서버-클라이언트 아키텍처를 사용하여 여러 인스턴스 간의 상태를 공유합니다. Vim과 유사한 키 바인딩을 사용하여 파일 시스템을 효율적으로 탐색하고 관리할 수 있습니다.

## 기본 이동

- k, up**: 위로 이동
- j, down**: 아래로 이동
- h, left**: 상위 디렉토리로 이동
- l, right**: 파일인 경우 열고, 디렉토리인 경우 안으로 들어감
- gg**: 목록의 맨 위로 이동
- G**: 목록의 맨 아래로 이동
- H**: 화면의 맨 위로 이동
- M**: 화면의 중간으로 이동
- L**: 화면의 맨 아래로 이동
- ~**: 홈 디렉토리로 이동
- gh**: 홈 디렉토리로 이동 (go home)

## 파일 조작

- space**: 현재 파일을 선택/선택 해제. 여러 파일 선택 가능.
- d**: 선택된 파일(들)을 삭제 (~/.local/share/lf/trash로 이동).
- D**: 선택된 파일(들)을 영구적으로 삭제.
- y**: 선택된 파일(들)을 복사 (yank).
- c**: 선택된 파일(들)을 잘라내기 (cut).
- p**: 현재 디렉토리에 복사/잘라내기 한 파일(들)을 붙여넣기 (paste).
- r**: 파일 이름 변경 (rename).
- :mkdir <name>**: 새 디렉토리 생성.
- :touch <name>**: 새 파일 생성.
- :link <target>**: 심볼릭 링크 생성.

## 검색 및 필터링

- /**: 검색 모드로 진입. 입력하면 실시간으로 파일 목록이 필터링됨.
- f**: 필터 모드로 진입. find와 유사하게 작동.
- ?**: 검색 기록 보기.

## 셸 명령어 실행

- !**: 외부 셸 명령어를 실행 (예: !echo "hello").

- \$**: 셸 프롬프트를 열어 현재 디렉토리에서 작업을 수행. lf로 돌아오려면 exit.
- c**: 셸 명령어를 사용하여 lf와 상호작용하는 클라이언트 모드.

## 보기 모드 및 정렬

- zh**: 숨김 파일 표시/숨기기 토글.
- zr**: 정렬 순서 반전 (reverse).
- zs**: 정렬 기준을 이름(n), 크기(s), 시간(t)으로 변경.
- zp**: 미리보기 창 표시/숨기기 토글.

## 탭 및 창

lf는 자체적인 탭 기능을 내장하고 있지는 않지만, 터미널 멀티플렉서(tmux, zellij 등)와 함께 사용하면 여러 lf 인스턴스를 동시에 실행하여 탭처럼 활용할 수 있습니다. 서버-클라이언트 아키텍처 덕분에 복사/붙여넣기 등의 상태가 모든 인스턴스 간에 공유됩니다.

## 사용자 정의

lf는 ~/.config/lf/lfrc 파일을 통해 고도로 사용자 정의할 수 있습니다.

- 키 바인딩 변경**: map gh cd ~
- 사용자 정의 명령어**: cmd my-command echo "hello"
- 셸 스크립트로 미리보기 설정**:
 

```
# lfrc 예시
set previewer "/path/to/your/preview/script.sh"
```
- 색상 테마 설정**: set colors ...

## 종료

- q**: lf 종료. 종료 시 마지막 디렉토리로 cd하도록 셸 설정을 할 수 있습니다.

```
# .bashrc 또는 .zshrc에 추가
lfcd () {
    tmp="$(mktemp)"
    lf -last-dir-path="$tmp" "$@"
    if [ -f "$tmp" ]; then
        dir="$(cat "$tmp")"
        rm -f "$tmp"
        if [ -d "$dir" ]; then
            if [ "$dir" != "$(pwd)" ];
            then
                cd "$dir"
            fi
        fi
    fi
}
```

```
fi
fi
}
# 이제 'lf' 대신 'lfcd'를 사용
```