

lf (list files) cheatsheet

개요 및 기본 개념

lf는 Go 언어로 개발된 터미널 기반 파일 관리자입니다. ranger에서 영감을 얻었으며, 서버-클라이언트 아키텍처를 채택하여 여러 인스턴스 간에 상태를 공유할 수 있는 것이 특징입니다. Vim과 유사한 키 바인딩을 제공하여 파일 시스템을 매우 효율적으로 탐색하고 관리할 수 있습니다.

파일 탐색 및 이동

- k** 또는 **↑**: 위로 이동
- j** 또는 **↓**: 아래로 이동
- h** 또는 **←**: 상위 디렉토리로 이동
- l** 또는 **→**: 선택한 대상이 파일이면 열고, 디렉토리면 내부로 진입
- gg**: 목록의 맨 처음으로 이동
- G**: 목록의 맨 마지막으로 이동
- H**: 현재 화면의 맨 윗줄로 이동
- M**: 현재 화면의 중간 줄로 이동
- L**: 현재 화면의 맨 아랫줄로 이동
- ~** 또는 **gh**: 홈(Home) 디렉토리로 즉시 이동

파일 및 디렉토리 관리

- Space**: 현재 항목 선택 또는 해제 (다중 선택 가능)
- d**: 선택한 항목을 휴지통(~/.local/share/lf/trash)으로 이동
- D**: 선택한 항목을 영구 삭제
- y**: 선택한 항목 복사 (Yank)
- c**: 선택한 항목 잘라내기 (Cut)
- p**: 복사하거나 잘라낸 항목을 현재 위치에 붙여넣기 (Paste)
- r**: 파일 또는 디렉토리 이름 변경 (Rename)
- :mkdir <이름>**: 새 디렉토리 생성
- :touch <이름>**: 새 파일 생성
- :link <대상>**: 심볼릭 링크 생성

파일 검색 및 실시간 필터링

- /**: 검색 모드 진입. 입력 시 목록이 실시간으로 필터링됨
- f**: 필터 모드 진입. find 명령어와 유사하게 작동
- ?**: 이전 검색 기록 확인

외부 명령어 및 셸 연동

- !**: 외부 셸 명령어 실행 (예: `!echo "hello"`)
- \$**: 현재 디렉토리에서 셸 프롬프트 열기. 작업을 마친 후 `exit`를 입력하면 lf로 복귀

- c**: 셸 명령어를 통해 lf와 상호작용하는 클라이언트 모드 실행

보기 옵션 및 정렬 방식

- zh**: 숨김 파일 표시 여부 전환 (Toggle)
- zr**: 정렬 순서 반전 (Reverse)
- zs**: 정렬 기준 변경 (이름: n, 크기: s, 시간: t)
- zp**: 미리보기(Preview) 창 표시 여부 전환

탭 및 다중 인스턴스 활용

lf는 자체적인 탭 기능을 내장하고 있지는 않지만, tmux나 zellij 같은 터미널 멀티플렉서와 병용하면 강력한 효과를 냅니다. 서버-클라이언트 구조 덕분에 서로 다른 창에서 실행 중인 lf 인스턴스끼리도 복사/붙여넣기 등의 상태가 실시간으로 공유됩니다.

사용자 설정 및 커스터마이징

~/.config/lf/lfrc 파일을 편집하여 사용자 취향에 맞게 고도로 설정할 수 있습니다.

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


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

프로그램 종료 및 디렉토리 동기화

- q**: lf 종료. 종료 시 에디터에서 마지막으로 머물렀던 디렉토리로 셸의 작업 경로를 자동으로 변경하도록 설정할 수 있습니다.

```
# .bashrc 또는 .zshrc 설정 예시
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
}
# 이제 'lf' 명령어 대신 'lfcd'를 사용하면 됩니다.
```