

grep 빠른 참조

패턴 매칭, 정규식, 재귀 검색, 컨텍스트, 필터링

기본 사용법

grep 실행

```
grep "pattern" file.txt          # search in file
grep "error" *.log              # search multiple files
grep "hello" file1.txt file2.txt # explicit file list
cat file.txt | grep "pattern"   # pipe input
dmesg | grep -i "usb"          # filter command output
```

주요 플래그

- i 대소문자 무시 매칭
- v 반전 매칭 — 일치하지 않는 줄 출력
- c 일치하는 줄 수 출력
- n 줄 번호 표시
- l 일치하는 파일명만 나열
- L 일치하지 않는 파일명 나열
- w 단어 전체 매칭만
- x 줄 전체 매칭만

정규식 패턴

기본 정규식 (BRE)

- .
- *
- ^
- \$
- [abc]
- [^abc]
- [a-z]
- \<, \>
- \(\), \1

임의의 한 문자
이전 요소 0회 이상
줄 시작
줄 끝
문자 클래스 — a, b, c 중 하나
부정 클래스 — a, b, c 외 모든 것
범위 — 소문자
단어 경계 (GNU)
캡처 그룹과 역참조

BRE 예시

```
grep '^#' file.conf          # lines starting with #
grep 'error$' file.log      # lines ending with error
grep '^$' file.txt          # blank lines
grep 'col[ou]r' file.txt    # match color or colour
```

확장 정규식

확장 정규식 (ERE)

- +
- ?
- {n}
- {n,m}
- (a|b)
- ()

이전 요소 1회 이상
이전 요소 0회 또는 1회
정확히 n회 반복
n회 이상 m회 이하 반복
교체 — a 또는 b 매칭
그룹화 (역슬래시 불필요)

ERE 예시

```
grep -E '[0-9]{3}-[0-9]{4}' f # phone number pattern
grep -E '(error|warn|fatal)' f # multiple patterns
grep -E '^[A-Z][a-z]+' f      # capitalized words
grep -P '\d{1,3}\.\d{1,3}' f  # Perl regex: IP fragments
```

컨텍스트 줄

컨텍스트 예시

```
grep -B 3 "error" app.log      # 3 lines before match
grep -A 5 "FAIL" test.log      # 5 lines after match
grep -C 2 "crash" kern.log     # 2 lines before and after
grep --group-separator="---" -C 1 "err" f # custom separator
```

컨텍스트 플래그

- B N
- A N
- C N
- group-separator=str
- color=auto

각 일치 전 N줄 표시
각 일치 후 N줄 표시
전후 N줄 표시 (컨텍스트)
일치 그룹 간 구분자 (기본 --)
터미널에서 일치 강조

재귀 검색

재귀 예시

```
grep -r "TODO" .              # recursive from current dir
grep -rn "FIXME" src/         # recursive with line numbers
grep -r --include="*.py" "import" . # only .py files
grep -r --exclude="*.log" "error" . # skip .log files
grep -r --exclude-dir=node_modules "require" .
```

재귀 플래그

- r / --recursive
- R
- include=glob
- exclude=glob
- exclude-dir=dir
- include-dir=dir

디렉터리 재귀 검색
-r와 같지만 심볼릭 링크 따라감
글로브와 일치하는 파일만 검색
글로브와 일치하는 파일 건너뛸
이름이 일치하는 디렉터리 건너뛸
이름이 일치하는 디렉터리만 검색

카운팅 & 목록

카운트 & 목록 예시

```
grep -c "error" *.log          # count matches per file
grep -l "TODO" src/*.py        # list files with TODOs
grep -L "test" src/*.py        # files missing "test"
grep -o "http[^ ]*" page.html  # extract matching parts only
grep -c '' file.txt            # count total lines (like wc -l)
```

출력 플래그

- c
- l
- L
- o
- H / -h
- Z

파일별 일치하는 줄 수 출력
일치가 있는 파일명만 출력
일치가 없는 파일명만 출력
일치하는 부분만 출력
파일명 접두사 표시 / 숨기기
null 구분 출력 (xargs -0용)

반전 매칭

반전 & 제외

```
grep -v "^#" config.conf      # remove comment lines
grep -v "^$" file.txt         # remove blank lines
grep -v -e "debug" -e "trace" app.log # exclude two patterns
grep -v "pattern" f | grep "other" # chain: NOT A, then B
```

필터링 전략

- v
- v with -e
- pipe chain
- grep -v '^\$' | grep -v '^#'
- v with -c

반전 매칭 — 일치하지 않는 줄 선택
여러 패턴 제외
복잡한 필터링을 위한 grep 체이닝
빈 줄과 주석 제거
일치하지 않는 줄 수

복수 패턴

복수 패턴 예시

```
grep -e "error" -e "warning" app.log
grep -E "error|warning|fatal" app.log
grep -f patterns.txt file.txt      # patterns from file
grep -w -e "GET" -e "POST" access.log
```

패턴 옵션

- e pattern
- f file
- E 'a|b|c'
- F
- G
- P

패턴 지정 (여러 번 사용 가능)
파일에서 패턴 읽기 (줄당 하나)
복수 패턴용 ERE 교체
고정 문자열 — 정규식 없음, 더 빠름
기본 정규식 (기본 모드)
Perl 호환 정규식 (PCRE)

성능

성능 팁

- F (fgrep)
- LC_ALL=C grep
- include/--exclude
- m N
- q
- ripgrep (rg)

고정 문자열 모드 — 리터럴 문자열에 가장 빠름
ASCII 데이터에서 2-10배 빠른 로케일 우회
열기 전에 검색할 파일 줄이기
파일당 N번 일치 후 중지
조용한 모드 — 첫 번째 일치 시 종료 (스크립트용)
대체 도구, 대형 저장소에서 더 빠름

성능 예시

```
LC_ALL=C grep -F "exact string" huge.log
grep -r -m 1 "needle" /var/log/ # stop after first hit
grep -rq "pattern" . && echo "found" # boolean test
grep -r --include="*.go" "func main" .
```

일반 패턴

한 줄 명령

```
grep -rn "TODO|FIXME|HACK" src/ # find code markers
grep -oP '(?<=)[^]+(=?=)' f      # extract quoted strings
grep -E '^s*$' f | wc -l         # count blank lines
grep -c '*' *.py | sort -t: -k2 -rn # sort files by line count
grep -rn --include="*.yaml" "password" . # audit for secrets
```

레시피

- IP addresses
- Email addresses
- URLs
- Lines between markers
- Unique matches
- Count per pattern

```
grep -oE '[0-9]{1,3}(\.[0-9]{1,3}){3}'
grep -oE '[a-zA-Z0-9._%+-]@[a-z.-]+'
grep -oE 'https?://[^\ ]+'
grep -A999 'START' f | grep -B999 'END'
grep -oE 'pattern' f | sort -u
grep -c 'pat1' f; grep -c 'pat2' f
```