Linux Commands

A Useful Document

Nathan Warner

Northern Illinois
University

Computer Science
Northern Illinois University
United States
December 10, 2023

Contents

Find and Replace For a Directory

Suppose we have some codebase, and we decide that we want to change every occurrence of the word ‘foo‘ with the word ‘bar‘.

```
find . -type f -exec grep -l 'foo' {} \; -exec sed -i
↪  's/foo/bar/g' {} +
```

Where

- find . -type f: Find all files (-type f) in the current directory (.) and its subdirectories.
- -exec grep -l 'foo' : For each file found, use grep to search for 'foo'. The -l option makes grep only list the names of files with matching lines.
- -exec sed -i 's/foo/bar/g' +: For each file where 'foo' is found, use sed to replace 'foo' with 'bar'. The -i option edits files in place (i.e., it saves the changes to the file). The s/foo/bar/g is the substitution command: s for substitute, foo is the search term, bar is the replacement term, and g is for global replacement (replace all occurrences in the file).

1.1 Excluding Directory's

1.1.1 Single Directory

```
find . -type d -path ./excluded_dir -prune -o -type f -exec grep
↪  -l 'foo' {} \; -exec sed -i 's/foo/bar/g' {} +
```

Where

- -type d -path ./excluded_dir -prune: This part of the command checks if the current file is a directory (-type d) and matches the path ./excluded_dir. If it does, -prune is used to exclude it from further processing.
- -o: This is the logical OR operator. It separates the directory exclusion part from the rest of the command.
- The rest of the command (-type f -exec grep -l 'onedark' -exec sed -i 's/onedark/onedark-fk/g' +) remains the same

1.1.2 Multiple Directory's

```
find . -type d \( -path ./excluded_dir1 -o -path ./excluded_dir2
↪  \) -prune -o -type f -exec grep -l 'onedark' {} \; -exec sed
↪  -i 's/onedark/onedark-fk/g' {} +
```

Changing File Names for each file in sub-directory's with a specific name

Suppose we have some codebase and we want to change every filename that is ‘foo.cpp‘ to ‘bar.cpp‘

```
find . -type f -name foo.cpp -exec sh -c
↪  'mv "${0}" "$(dirname "${0}")/bar.cpp" ' {} \;
```

Where

- find . -type f -name 'foo.cpp': This finds all files named foo.cpp in the current directory and its subdirectories.
- -exec sh -c 'mv "$0" "$(dirname "$0")/bar.cpp"' \;: For each file found, this executes a shell command. The is replaced by the path of each found file.
  - mv "$0": Moves the found file.
  - "$(dirname "$0")/bar.cpp": This constructs the new file path. dirname "$0" gives the directory of the found file, and /bar.cpp appends the new file name to this path.