Improve Your Bash Command Line Fu

@JarredOlson

Why should I care?

- The server you're running on won't always have a GUI
- Increase productivity
- Right tool for the job

I Am Devloper
@iamdevloper

Follow

Over 1 year, the average Vim user saves 11 minutes in productivity.

However, they lose 27 hours through evangelising Vim to non-users.

RETWEETS 3,349 FAVORITES 2,270

10:55 AM - 15 Apr 2015

Are we only going to talk about vim?

https://twitter.com/iamdevloper/status/588355053104267264

command name

“English name or alias”
Brief description of what the function does.

|

"Pipe"
Takes output from one command and uses it as input for another command

The End

|

"Pipe"
Takes output from one command and uses it as input for another command

man

“man page or manual entry”
Shows the documentation for a command line function

GREP(1) BSD General Commands Manual GREP(1)

NAME

grep, **egrep**, **fgrep**, **zgrep**, **zegrep**, **zfgrep** -- file pattern searcher

SYNOPSIS

grep [**-abcdDEFGHhIiJLlmnOopqRSsUVvwxZ**] [**-A** num] [**-B** num] [**-C**[num]] [**-e** pattern] [**-f** file] [**--binary-files**=value] [**--color**[=when]] [**--colour**[=when]] [**--context**[=num]] [**--label**] [**--line-buffered**] [**--null**] [pattern] [file ...]

DESCRIPTION

The **grep** utility searches any given input files, selecting lines that match one or more patterns. By default, a pattern matches an input line if the regular expression (RE) in the pattern matches the input line without its trailing newline. An empty expression matches every line. Each input line that matches at least one of the patterns is written to the standard output.

grep is used for simple patterns and basic regular expressions (BREs); **egrep** can handle extended regular expressions (EREs). See re_format(7) for more information on regular expressions. **fgrep** is quicker than both **grep** and **egrep**, but can only handle fixed patterns (i.e. it does not interpret regular expressions). Patterns may consist of one or more lines, allowing any of the pattern lines to match a portion of the input.

zgrep, **zegrep**, and **zfgrep** act like **grep**, **egrep**, and **fgrep**, respectively, but accept input files compressed with the compress(1) or gzip(1) compression utilities.

The following options are available:

-A num, **--after-context**=num
Print num lines of trailing context after each match. See also the **-B** and **-C** options.

-a, **--text**
Treat all files as ASCII text. Normally **grep** will simply print ``Binary file ... matches'' if files contain binary characters. Use of this option forces **grep** to output lines matching the specified pattern.

-B num, **--before-context**=num
Print num lines of leading context before each match. See also the **-A** and **-C** options.

-b, **--byte-offset**
:

man grep

tree

“tree”
Lists the contents of directories in a tree like format

cat

“cat or concatenate”
Reads files sequentially writing them to standard output

grep

“grep”
Searches any given input, selecting lines that match one or more patterns.

grep

- -i Case insensitive search
- -v Inverts the search match
- -r or -R Recursive search
- -B Before Context
- -A After Context
- -C Before and After Context

find

“find”
Recursively descends the directory tree to find matching files

find

- -type [‘f’ for file ‘d’ for directory]
- -name file or folder name to search for
- -size find files by size
- -exec executes command on every match found

sort

“sort”
Sorts lines of input

sort

- -n Numeric sort
- -r Reverses the sort
- -k Sort based on a given column

uniq

"unique"

Compares ADJACENT lines of input and outputs a unique list.

uniq

- -c Precede each output line with the count of the number of times that line occurred in the input.
- -d Only output duplicated lines.
- -u Only output unique lines.

cut

“cut”
Cut the selected portions of each line of input separated by a given delimiter

cut

- -d specify a delimiter
- -f fields to output

Real World Example 1

git

Redirecting Output

- > Will replace the contents of an existing file or create a new one with the output of the previous command
- >> Will append the output of the previous command to the end of an existing file or create a new one

Chaining Commands

- && Will execute chained commands if the preceding result is true or 0
- || Will execute chained commands if the preceding result is false or greater than or equal to 1
- ; Will execute chained commands regardless of preceding result.

reverse-i-search

“Control r”
Searches your recent command history

reverse-i-search

- control r to enter
- control r to go to previous match
- enter to execute selected command
- tab or right arrow to edit selected command

bash profile

“bash profile”
Allows you to create/set user defined functions, aliases, and variables.

Bash Profile

- ~/.bash_profile
- To have changes take effect you need to “source” it
  - source ~/.bash_profile

Bash Profile

Aliases

```
#Maven
alias mci="mvn clean install"
alias mci_no_tests="mci -Dmaven.test.skip=true"
```

```
#Git
alias gs="git status"
alias gd="git diff"
alias gsl="git stash list"
alias gsp="git stash pop"
alias git-url="git remote show origin | grep 'Fetch URL'"
alias git-rm-deleted="git ls-files -d | xargs git rm"
```

Bash Profile

Function

```
function md() {
        mkdir $1
        cd $1
}

function trash() {
        DATE_FOLDER=`date +"%Y_%m_%d_%H_%M_%S"`
        mkdir -p /tmp/TRASH/"$DATE_FOLDER"
        PWD=`pwd`

        for i in "$@"
        do
                folderNameWithSlashes=`echo $i | sed 's/\ /\\\ /g'`
                NEW_DIRECTORY=/tmp/TRASH/"$DATE_FOLDER"/"$PWD"/"$i"
                mkdir -p $NEW_DIRECTORY
                mv $i $NEW_DIRECTORY

        done
}
```

Automating/Efficiency

- Tie multiple commands together
- Add things to your bash profile that you do often
- Average person types 3.25 keystrokes a second.
  - Saving 1 minute a day = 4 hours a year

Real World Example 2

Large Log Files

http://ita.ee.lbl.gov/html/traces.html

How to get better

- https://github.com/JarredOlson/command-line-fu
- Try it
- Force yourself to use it
- Read the man page

Commands

- |
- man
- cat
- tree
- grep
- find
- tail
- head
- cut
- sort
- uniq
- wc
- du
- > and >>
- && || ;
- history
- reverse-i-search / Ctl R
- clear / Ctl L
- Bash Profile Variables
- Bash Profile Alias
- Bash Profile Functions

- https://github.com/JarredOlson/command-line-fu

The End