Dude, Where's My File?

or, learning about find from Chapter 17 of 'The Linux Command Line'

Nicholas LiCalzi

$ Search Methods

$ Search Methods

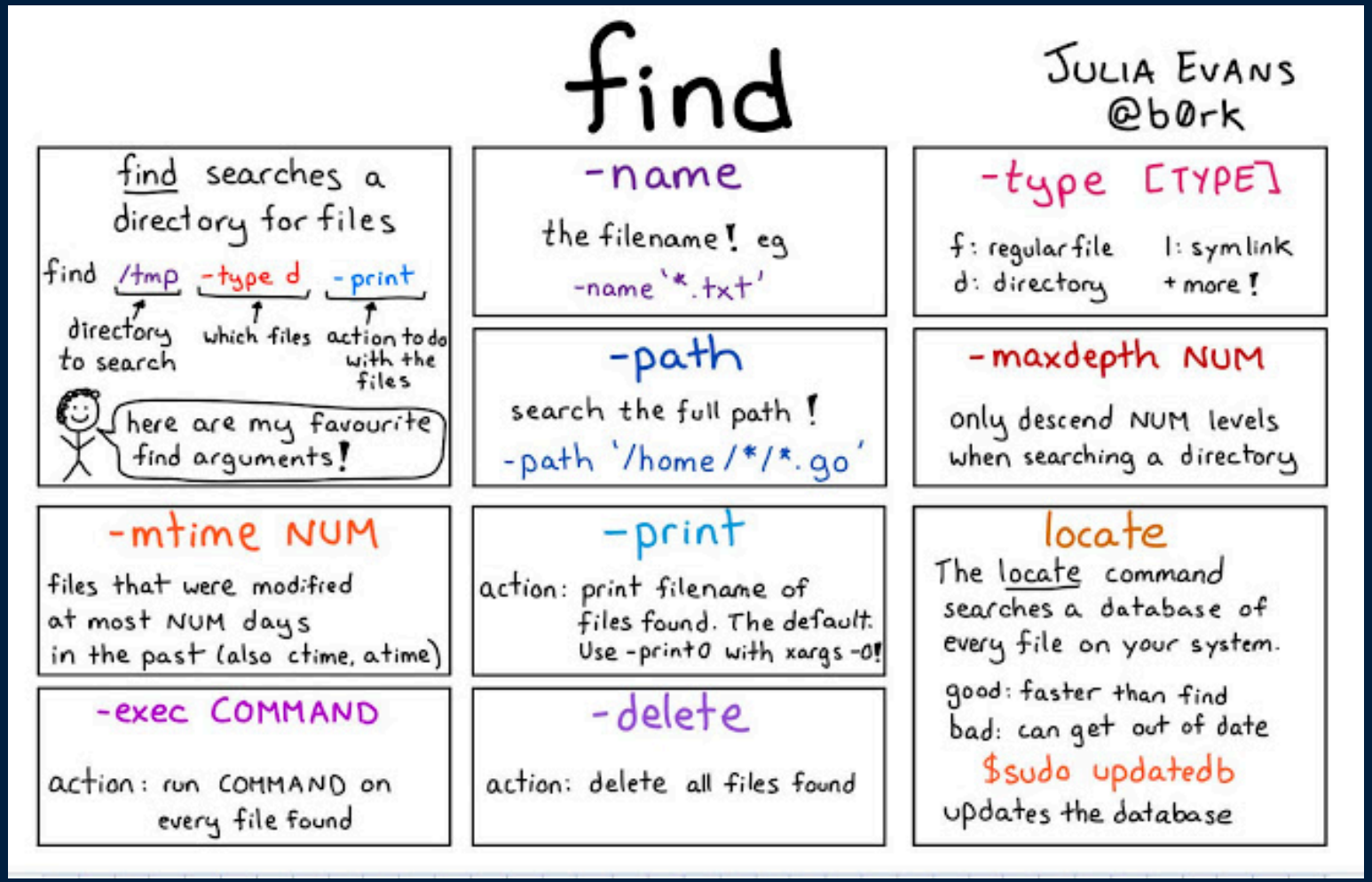
$ find

find is used to find files and directories based on their attributes, doing so through the application of *options*, *tests*, and *actions*.

At its most basic, we can call something like:

```
find /usr/
```

$ find


find
Julia Evans
@b0rk
find searches a directory for files
find /tmp -type d -print
directory to search
which files
action to do with the files
here are my favourite find arguments!
-name
the filename! eg
-name '*.txt'
-type [TYPE]
f: regular file
l: symlink
d: directory
+ more!
-path
search the full path!
-path '/home/*/*.go'
-maxdepth NUM
only descend NUM levels when searching a directory
-mtime NUM
files that were modified at most NUM days in the past (also ctime, atime)
-print
action: print filename of files found. The default. Use -print0 with xargs -0!
locate
The locate command searches a database of every file on your system.
good: faster than find
bad: can get out of date
$sudo updatedb
updates the database
-exec COMMAND
action: run COMMAND on every file found
-delete
action: delete all files found

$ find

- Tests
  - searching by Name, Time, Type, and Size
- Options
  - varying search Depth
- (Pre-defined) Actions
  - -delete, -ls, -print, and -quit
- Extending our power with `xargs` and other tricks

Searching by Name

using find's *Tests*

$ find [-name]

Compare the *basename* of the file to a passed-in *glob* pattern, finding matches.

```
find tmp -name ‘*.mp3’
# => Print all files ending in .mp3 in tmp
```

Note: -name is case-sensitive, but -iname will match in a case insensitive manner

Searching by Time

using find's *Tests*

$ find [-mtime] [n] [smhdw]

Find files that were last modified *n* time units ago.

```
find . -mtime 1    # modified exactly 1 day ago
find . -mtime -1h  # modified <1h ago
find . -mtime +1h  # modified >1h ago

find /usr -type f -mtime +30 -mtime -60
# files in /usr modified between 30 and 60 days ago
```

```
$ find [-ctime] [n]
```

Find files with statuses changed *n* time units ago.

```
$ find [-atime] [n]
```

Find files that were last accessed *n* time units ago.

Searching by Type

using find's *Tests*

$ find [-type] [dfl...]

- Using the -type test will return files that are of the type matching the argument used:
  - d: directory, f: file, l: symbolic link, etc…

```
find ~ -type d | wc -l
# => Print the count of directories in home

find /tmp -type d -empty
# => Find all empty directories in tmp (good deletion candidates?)
```

Searching by Size

using find's *Tests*

$ find [-size] [+/-][ckMG]

All three of the following commands will return identical results– names of files up to 5MB:

- `find / -size +5000000c # c: bytes`
- `find / -size +5000k    # k: KBs`
- `find / -size +5M       # M: MBs`

Limiting Search

using find's *Options*

$ find [-depth]

Setting a -depth will direct find to process a directory's files before the directory itself.

$ find [-(max/min)depth] [levels]

Set either the maximum or minimum *levels* that find will descend into a directory tree.

```
find tmp -maxdepth 1 -name '*.mp3'
# => find files in tmp/ that match '*.mp3'
```

Ok, I found my file… now what?

Ok, I found my file… now what?

using find's *Actions*

$ find [action]

- -delete
  - Delete the files we've found.
  - find ~ -type d -empty -delete # delete empty dirs from home (~)
- -ls
  - Equivalent to calling ls -dils (longform output)
- -print
  - Default action, printing the results to stdout
- -quit
  - Quits immediately, useful for stopping search once we've found what we want.
  - find /tmp/foo /tmp/bar -print -quit # only prints /tmp/foo

Extending Search Functionality

Some handy tricks with find (|, xargs, etc.)

$ xargs

xargs accepts input from stdin and converts it into an argument list for a specified command:

```
find ~ -type f -name 'foo*' -print | xargs ls -l
# => find all files having names like ‘foo*’ in home, and list them in long format
```

$ find | xargs

```
find . -name '*.rb' | xargs wc -l | sort -hr
   # =>
    # 1467 total
    # 322 user/base.rb
    # 261 user/general.rb
    # 251 user/collections.rb
```

$ find, now with error handling!

Oftentimes, find will end up returning some type of error for a given file (like “Permission Denied”). We can send those to the bitbucket and keep our output clean!

```
find [paths] [expression] [actions] 2>/dev/null
```

$ find and *permissions*

```
find / -perm /a=x
# => find all executable files


find / -type f -perm 0777 -exec chmod 644 {} \;
# => find all files with 777 permissions (read,
write, execute for owner, group, and others) and
modify them to have 644 permissions (Ow: RWE, G: R,
Ot: R)
```

```
$ -quit
```

Thanks for your time!