

Subcontractors – find

David Morgan

find

- searches for files in a directory tree
- described by an expression
- expression consists of elements
 - options
 - tests
 - actions
- each element returns boolean result
- find evaluates as many elements of its expression as needed to know expression's outcome

Most common use

for <all files in a set of files>

if <something about the file> do <something with the file>

next

but the operation details are more complex than that

find example

expression

```
find . -maxdepth 1 -size +1000000c -print
```

an option a test an action

find files 1) in the current directory (no subdirectory search)
2) bigger than a million bytes
3) and print their names

Some example elements

find .	options*	tests	actions
	↓	↓	↓
	maxdepth mount <i>etc</i>	name atime +n size +n executable type empty false <i>etc</i>	print ls exec ok <i>etc</i>

* find's options, not shell command options

Some example elements

find .	options	tests	actions
	↓	↓	↓
what it returns:	true always	true or false	true or false
what it does:	influence find behavior	nothing	their particular action

Operational logic

"[evaluates] the given expression from left to right... until the outcome is known (the left hand side is false for and operations, true for or), at which point find moves on to the next file name."

- "find" man page

Operational logic

```
[root@frausto test]# ls
Afile1 Afile2 Bfile1 Bfile2
[root@frausto test]#
[root@frausto test]# find . -print
.
./Afile2
./Bfile1
./Bfile2
./Afile1
[root@frausto test]# find . -name "A*" -print
./Afile2 ← because -name "A*" is false for B* files
./Afile1
[root@frausto test]# find . -print -name "A*"
.
./Afile2
./Bfile1 } ← because printing happens before -name "A*" evaluation
./Bfile2 }
./Afile1
[root@frausto test]# find . -print -name "A*" -print
.
./Afile2
./Afile2 ← from the 2nd -print
./Bfile1 (2nd print doesn't happen for B* files)
./Bfile2
./Afile1
./Afile1
[root@frausto test]# █
```

exec action –

arbitrary response for qualifying files

- needs to be terminated with `;` ;
- uses `{ }` as placeholder for current file
- need to escape these from shell

a “finder” script command:

```
find . -type f -exec grep -l "$1" {} \;
```

print names of all files in current directory containing
a given string