locate and find

CIS 2230 Linux System Administration
Lecture 7b
Steve Ruegsegger

Finding Files with locate

- The `locate` command is a simple and fast way to find files
- Syntax:

```
$ locate <string>
```

- Search operation:
  - The “string” is a sub-string anywhere in the *filename*
  - Basically, the string is ‘globbed’
  - That is, ‘string’ really becomes ‘*string*’
- How does it work?
  - The locate command searches a **database** of filenames for that string
  - The database needs to be updated regularly
  - Ubuntu sets up anacron to do this (we'll study this later).
  - So, `locate` will not find files created *after* the last db update!

Finding Files with locate

- For example, to find files relating a work item "`mod005`":

```
$ locate mod005
```

- Search options:
  - **-i** option makes the search *case-insensitive*
  - **-r** treats the pattern as a *regular expression*, rather than a simple string
  - To 'remove' the wildcarding and search exactly for a filename (see man page):

```
$ locate -b '\string'
```

```
$ locate mod005
/home/steve/Documents/dd12_analysis/Vejle_dd12_ TB14_Vmin_mod005.ppt
/home/steve/Documents/p_clk_mcma_5/Vejle_HTS_mcma_5_PEMmod005_FAreport.pdf
/home/steve/Documents/p_clk_mcma_5/mcma_5_PEM_mod005_100x_view.xml

$ locate -r 'mod005.*pdf$'
/home/steve/Documents/p_clk_mcma_5/Vejle_HTS_mcma_5_PEMmod005_FAreport.pdf
```

Finding Files More Flexibly: find

- `$ find` is very different than `locate` in how it works
- It does not use a db, but actually searches the hard-drive file-by-file

```
$ man find
```

- Command format:

```
$ find [find_opts] [start-dir] [expressions]
```

  - *find_opts* – deals with links
  - *start-dir* – where to start the search – just one directory
  - *expressions* → 3 types (next page)

find opts

```
FIND(1)                                                                 FIND(1)

NAME
       find - search for files in a directory hierarchy

SYNOPSIS
       find [-H] [-L] [-P] [-D debugopts] [-Olevel] [path...] [expression]
```

- Soft-links
  - -P – do not follow links (default)
  - -L – follow all links
  - -H – only follow links in [path...]

Finding Files More Flexibly: find

```
$ find [find_opts] [start-dir] [expressions]
$ find [find_opts] [start-dir] [exp_options] [tests_pairs] [actions]
```

1. *exp_options* – testing options
2. *test_pairs*
   - If the “test expression” is TRUE, then the file is returned/selected in the find
   - In word “pairs”: `-key value`
   - The key always begin with a single hyphen (even though they have long names)
   - Be sure to quote globs
   - e.g. `-name "*.doc"  -mtime +3`
3. *actions*
   - What to do with “found” files – files that are TRUE for the test_pairs
   - The default 'action' is to simply “list” the files found

find Expression Test_Pairs

- Useful keys are:
  - `-name pattern`: selects files whose name matches the shell-style wildcard (glob) pattern
  - `-type d, -type f:` select directories or plain files
  - Times: `-[cma]time n`
    - accessed (`-atime`), modified (`-mtime`) , creation (`-ctime`)
    - `n` is the number of 24-hour periods (exactly)
    - `+n` means greater than n days ago
    - `-n` means less than n days ago

- *e.g.* "find pdf files in my home dir which have been read within the last 3 days"

```
$ find $HOME -name "*.pdf" -atime -3
```

- Much more in `man`

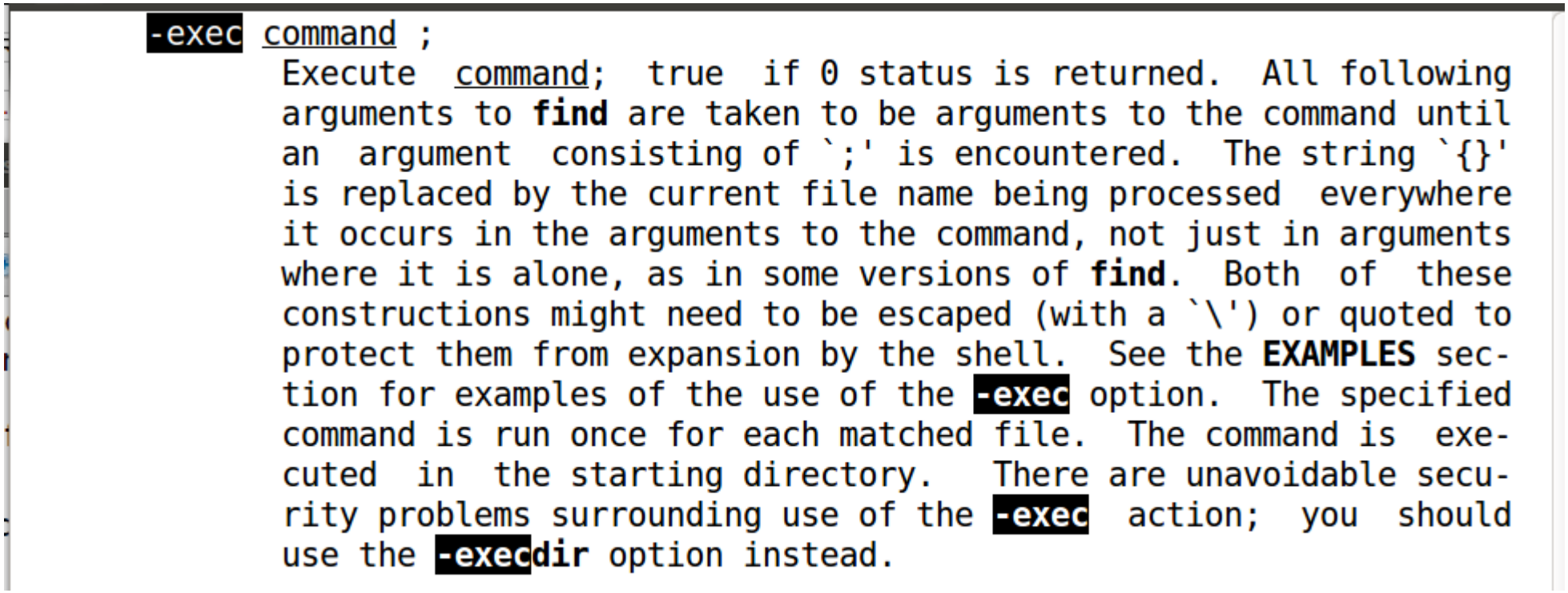
find Actions

`find` has arguments which let you specify an action for each file found

1. Print filenames
   - `-print` (the default action)
   - `-ls` = prints like 'ls -l'
2. Delete the files found
   - `-delete`
3. You can execute any linux command with `-exec ;`
   - Very helpful when you need to execute a command on one file at a time
   - The syntax is a little funny
     1. `{}` = the file
     2. Must end `-exec` clause with ";" → need to escape it "`\;`"

```
$ man find / -exec
```

rtfm



-exec command ;
Execute command; true if 0 status is returned. All following arguments to **find** are taken to be arguments to the command until an argument consisting of `;' is encountered. The string `{}' is replaced by the current file name being processed everywhere it occurs in the arguments to the command, not just in arguments where it is alone, as in some versions of **find**. Both of these constructions might need to be escaped (with a `\') or quoted to protect them from expansion by the shell. See the **EXAMPLES** section for examples of the use of the **-exec** option. The specified command is run once for each matched file. The command is executed in the starting directory. There are unavoidable security problems surrounding use of the **-exec** action; you should use the **-execdir** option instead.

find -exec examples

- backup all files whose name starts with 'manual':

```
$ find . -name 'manual*' -exec cp {} ../bak \;
```

The command `cp {} ../bak` is run for each file, where '{}' is replaced by the filename found

- copy a file to all users ~/.vnc dir

```
$ find /home -type d -maxdepth 1 -exec cp \
 /etc/skel/.vnc/xstartup {}/.vnc \;
```

Each dir in home represents a user and they need a new xstartup script in their .vnc subdir which is copied from /etc/skel

- compress all user's text files over 1Mb

```
$ find /home  -name '*.txt' -size +1M -exec gzip {}  \;
```

- Print the first line of all Matlab *.m files

```
$ find /home  -name '*.m' -exec head -1 {} \;
```

locate vs find

- `locate` limitations:

  (-) only finds files by name

  (-) only if in the database (since last update)

  (+) if you know part of filename and if it's in the db, then it's fast

- `find` has some advantages and disadvantages compared to locate

  (-) find is slower because it actually looks through the file system in real time

  (+) find has a wide number of criteria (not only name)

  (+) however, it can then find any file, not just ones in a db

  (+) can run any command with the results using –exec \;

Let's time them

- `locate`

```
steve@steveprecise:~$ time locate mod005
/home/steve/Documents/PE/vejle/dd12_analysis/Vejle_dd12_ TB14_Vmin_mod005.ppt
/home/steve/Documents/PE/vejle/packaging_HTSqual/Vejle_HTS_mcma_5_PEMmod005_FAreport.pdf
/home/steve/Documents/PE/vejle/packaging_HTSqual/mcma_5_PEM_mod005_100x_view.xml

real    0m1.946s
user    0m1.888s
sys     0m0.032s
```

- `find` – of course, the time depends on the starting directory given

```
steve@steveprecise:~$ time find /home/steve -name '*mod005*'
/home/steve/Documents/PE/vejle/packaging_HTSqual/Vejle_HTS_mcma_5_PEMmod005_FAreport.pdf
/home/steve/Documents/PE/vejle/packaging_HTSqual/mcma_5_PEM_mod005_100x_view.xml
/home/steve/Documents/PE/vejle/dd12_analysis/Vejle_dd12_ TB14_Vmin_mod005.ppt

real    0m23.537s
user    0m0.236s
sys     0m0.644s
```

End of L7

- After 'just' 3 weeks of linux, I consider you now like typical 'users'
  - We've learned in 3 weeks what most 'users' eventually learn 'on the job'
- Key concepts:
  - Unix philosophy: free, everything is a file, small and with 1 task
  - The shell interprets your commands to the kernel, which commands the hardware
  - commands: cd, cat, pwd, locate, find, man, cp, rm, mv, history, env
  - globbing: *, ?
  - command options: ls -la, ls -lrt
  - file structure: ., .., root dir, home dir, abs. & rel. paths, 'dot-files'
  - locate & find
- The next step is to get you past 'Typical user' to 'Super user'!
  - On your way to being a sys admin...