Titbits of Linux, not original ofc

Arghya Dutta

Last updated: April 26, 2017

1 Command line

- **touch file.txt**

  creates an empty file called file.txt in the current folder.

- **cat file.txt**

  prints the content of file.txt onto the command line.

- **find ∼ -name *.jpg**

  Let's break this down. First, you have the command name, then you have the directory to search (∼ in this case). Following that is a flag that specifies a certain way for the command to behave (typically anything following a dash is a flag; there are a few exceptions). In this case, the flag tells find to search specifically for things with the name that follows, which in this case is *.jpg. The "*" is a wildcard, telling find to match any file as long as it ends with ".jpg".

- **grep "blog" temp.txt**

  This command will return every line in temp.txt that contains the word blog. What's interesting about grep is that it gives you some options for what you search on, allowing you to insert wildcards to increase the range of what gets matched.

  grep "bl.g" temp.txt

  The "." is a wildcard; it matches any character just once; for instance, output of this command might look something like:

  blog blag bl&g

  There are way more wildcards than just "." though. To whet your appetite, take a look at this site.

  "grep" provides a few more options too. To search a file in current directory for a string that's case insensitive, add the "-i" flag. The following command could match a line containing cats, cATs, CATS, etc.

  grep -i "cats" ./cats.txt

- **ps aux**

  "ps aux" will list all the processes running

  You can also use "grep" to search through your processes with "ps aux".

  ps aux — grep terminal

2 vim