

Using pbcopy & pbpaste from Command Line on Mac

pbcopy: takes standard input and places it in the clipboard buffer

pbcopy is simple enough to use, basically direct something into it, and it'll copy it into the clipboard buffer. This can then be accessed either through pbpaste, or the standard Finder's paste command (command-v). Here's how to use it:

```
$ pbcopy < file.txt
```

That's it, now the contents of file.txt will be in your clipboard, ready to paste where ever. But pbcopy is much more powerful than that, and you can direct the output of commands and programs to be copied as well. Here's an example with the command 'ps':

```
$ ps aux | pbcopy
```

This pipes the results of the `ps` command into the clipboard, once again it can be pasted anywhere. If you want to filter your results a bit, you can throw `grep` into that too. An example using the `grep` command is:

```
$ ps  
aux |  
grep  
root |  
pbcopy
```

This pipes the results of the `'ps aux'` command, but filters specifically for `'root'`, and copies only those results into the clipboard. Cool huh?

pbpaste: takes data from the clipboard buffer and writes it to the standard output

pbpaste is equally simple to use, it will retrieve whatever you have placed in the clipboard buffer and spit it out. In it's most simple form, simply type:

```
$ pbpaste
```

This will print whatever data you copied from the pbcopy command or the Finder's copy command (command-c). You can easily route this output into a file via the command line to access later if you want though, with this simple command:

```
$ pbpaste > pastetest.txt
```

Filtering what is pasted is very useful though, and the command structure is much like what we saw earlier with pbcopy. We'll filter for 'rcp' but of course you can filter for anything you want

```
$ pbpaste | grep rcp
```

What you'll see pasted is only what matches your search for 'rcp' inside the data in the clipboard.



There's much more use to be had with pbcopy and pbpaste, but that should give you a general idea of how to use it and perhaps give you

some ideas to implement them in your own productivity. The screenshot simply demonstrates that you can copy and paste from the command line into a Finder app (in this case, TextEdit) without losing any formatting.

And yes you can do use the pbcopy and pbpaste commands at the terminal and then interact with

them again from the GUI of MacOS by using the standard Mac copy and paste keyboard shortcuts of Command+C and Command+V. It goes the other direction too, a copy from the GUI can be pasted with pbpaste at the command line.