Command Line (bash)

There's More than One Way to Groom a Cat(alog): Technologies for Data Analysis and Manipulation. OLAC Preconference October 26, 2017

Common commands:

- **ls** – list directory contents
- **cat** – output the whole file
- **cd** – change directory
- **pwd** – present working directory (where are you in directory tree?)
- **head** – output the first part of the file
- **tail** – output the last part of the file
- **less** – view the file screen by screen
- **grep** – output lines / files that match a pattern
- **wc** – count lines, words, or bytes
- **cp** – copy files
- **mv** – move/rename files or directories
- **mkdir** – create a new directory
- **rm** – delete files/directories
- **rmdir** – delete an empty directory
- **nano** – edit a file with the friendly nano text editor (see also **vim**, **emacs**)
- **python** – run a program written in python
- **man** – view the manual page for a command on the system
- **diff** – show the differences between two files or directories
- **sed** – stream editor (apply a regular expression to a file)
- **find** – identify files in a directory tree (such as those whose names match a regex)
- **sort** – sort the lines in a text file
- **curl**, **wget** – download files (like web pages)
- **gzip**, **gunzip** – file compression
- **rsync** – efficiently copy large directories
- **convert**, **magick** – image file manipulation
- **sox** – sound file manipulation

Shortcuts:

- ~ (tilde) means your home directory
- . (one dot) means your present working directory
- . . (two dots) means parent directory
- If you've started typing a command or filename, pressing **tab** might complete it for you!
- Do you want to type a very similar command to what you just typed? Press the **up** arrow (maybe multiple times!), edit the command, and run it again.
- You can use some wildcards ("globbing", not regex!), such as *.mrc for all .mrc files

Input / Output Redirection:

Many programs have output that goes to the screen, unless you redirect it by adding phrases like this to the end of the command:

- `> filename.txt` -- makes a new file called filename.txt containing the output
- `>> filename.txt` -- appends output to the end of filename.txt
- `| program` -- sends the output to the program as input

You can chain multiple commands together this way, like this:

```
grep 'a.b[cd]' infile.txt | sort | head -20

diff file1.txt file2.txt | grep -v '^Title' >> changes.txt

curl "https://www.zemkat.org/" | sed -e 's!z!Z!' > Zem.html
```

Tools and Resources

YAZ http://www.indexdata.com/yaz/

Free/open-source software for working with MARC. Functionality includes conversion between MARC formats (binary MARC, MARCXML, line-based MARC) and character encodings (MARC-8 and UTF-8/Unicode)

pymarc https://github.com/edsu/pymarc

Free/open-source python library for working with MARC records.

Further study and practice

Learn the Command Line (Codecademy) https://www.codecademy.com/learn/learn-the-command-line

Free interactive lessons on bash, covering navigation, viewing and changing the file system, redirecting input and output, configuring your environment

Codecademy also has courses in python and other programming languages.

man pages

Most unix command line environments will include “man pages”, describing what commands do, and documenting their (often extensive) options