CS 45, Lecture 12

Recent Unix Tools

Akshay Srivatsan, Ayelet Drazen, Jonathan Kula

Winter 2023

Announcements

- Assignment 5 is due today at 11:59 PM, contact us if you need an extension.
  - Please either make a private post on Ed or email all three of us together, if you email just one of us we may miss it.
- Assignment 6 will go out today or tomorrow.
- Final Project guidelines will go out soon (and we'll talk about it in a second).

Final Projects

Task:

- Pick a tool or concept related to this class (either one we've covered or one we didn't cover but you're interested in).
- Do research on what it's for/how it works/how you use it.
- Write a short guide on how/when to use the tool.
- Make a few slides describing the tool and giving example use cases.

Logistics:

- Due on March 20, 2023 (Monday of Finals Week).

Outline

Contents

1 Overview

What's wrong with the old tools?

For the most part, they're okay, but…

- They only work on text files.
- They're (mostly) single-threaded.
- They don't take advantage of new discoveries and inventions.
- Their interface is so standardized that they can't innovate.

Types of Tools

upgrades: modernized versions of the traditional tools you know

swiss army knives: bundles of many related tools

- Kind of like Git; Git contains `add`, `commit`, `merge`, etc.

The point of this lecture is **not** that you become an expert in these tools!

We're showing you these tools so you know that they exist, because we think they're useful (and/or really cool).

2 Upgrades

ripgrep

- Ripgrep is an upgraded version of `grep` and `find`
- Its main selling points are that it's fast and it's "ergonomic".

Example 2.1. <only@+>[ripgrep for text] Searching for a file in the current directory (or subdirectories) containing the text "hello":

```
rg hello
```

Example 2.2. <only@+>[ripgrep for regex] Searching for a file in the current directory (or subdirectories) containing the regular expression `/hello.*!/`:[1]

```
rg 'hello.*!'
```

Example 2.3. <only@+>[ripgrep in a file] Searching for lines of `student_hobbies.txt` containing the string `akshay01`:

```
rg akshay01 student_hobbies.txt
```

[1] Surrounding a regular expression with slashes (like you do when using `sed` is a common way of distinguishing it from an ordinary string. Some languages (like JavaScript) even support it as part of their syntax!

fd

- fd is a user-friendly alternative to `find`
- Its main selling point is that it's much more intuitive than `find`

Example 2.4. <only@+>[fd for files named "grep"] Search the current directory and all subdirectories for every file with "grep" in its name:

```
fd grep
```

Example 2.5. <only@+>[fd for symbolic links] Search for every symbolic link in the current directory or its subdirectories:

```
fd --type symlink
```

Example 2.6. <only@+>[fd all large files] Search for every file greater than or equal to 500 MB in size and print out a helpful message:

```
fd --size +500MB --exec echo You should delete {/} in directory {//}
```

exa

- exa is a modern alternative to `ls`
- Its main selling point is that it has more features than `ls`, like viewing your current git status (and it's colorful!)

Example 2.7. <only@+>[exa: sort files by size] List all the files in the current directory, ordered by size

```
exa --sort=size
```

Example 2.8. <only@+>[exa: git status] List every file in the current directory's git status:

```
exa --long --git
```

Example 2.9. <only@+>[exa: tree] Show a tree of files in the current directory and all subdirectories:

```
exa --tree
```

fish

- fish is a modern alternative to `bash` and `zsh`.
- Its main selling points are that it has a lot more conveniences, like autosuggestions and a nicer scripting language.
- It is **not** backwards compatible with `sh`!

[Demo Time]

3 Swiss Army Knives

3.1 Images

Images

- Images exist.[citation needed]
- Sometimes, we want to modify those images.
- Images are, notably, **not** text files, so our usual Unix commands won't work.
- *ImageMagick* is a set of tools for working with images.

ImageMagick

ImageMagick is broken into subcommands:

convert is the one you usually want (and the default), it modifies a file

mogrify modifies a file in-place (overwriting the original)

display opens an image in a window

compare diffs two images

ImageMagick Convert

Example 3.1. <only@+>[magick: png to jpg] Convert a `jpg` file to a `png` file:

```
magick convert input.jpg output.png
```

Example 3.2. <only@+>[magick: compress] Compress an image:

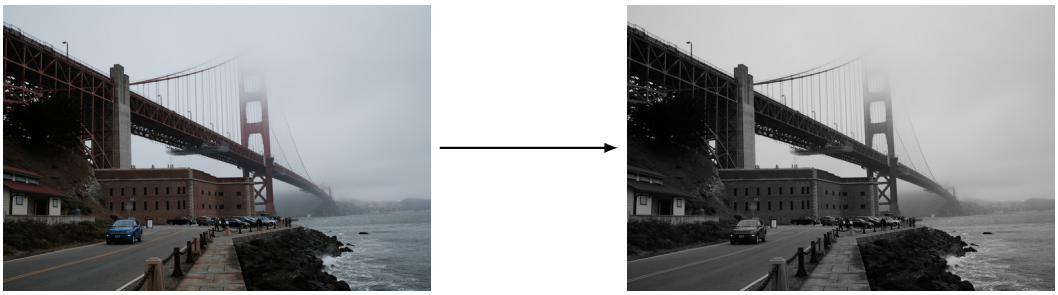
```
magick convert input.jpg -quality 50 output.jpg
```

Example 3.3. <only@+>[magick: resize] Resize an image:

```
magick convert input.jpg -resize 320x240 output.jpg
```

Example 3.4. <only@+>[magick: grayscale] Make an image grayscale:

```
magick convert input.jpg -colorspace gray output.jpg
```



Example 3.5. <only@+>[magick: brightness] Brighten an image:

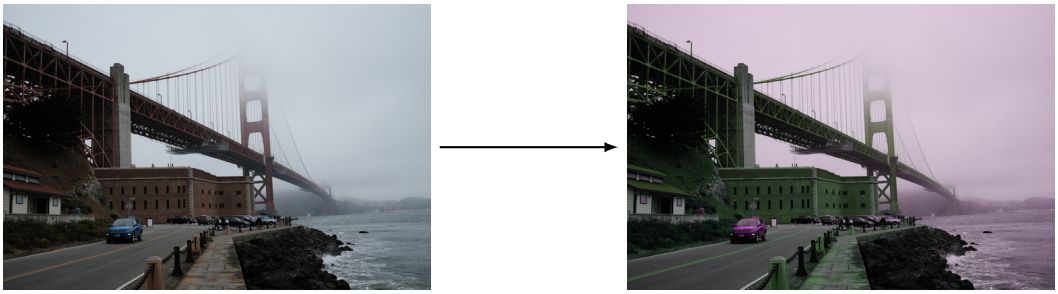
```
magick convert input.jpg -modulate 200,100,100 output.jpg
```

Example 3.6. <only@+>[magick: saturation] Saturate an image:

```
magick convert input.jpg -modulate 100,200,100 output.jpg
```

Example 3.7. <only@+>[magick: hue] Hue an image:

```
magick convert input.jpg -modulate 100,100,150 output.jpg
```



Example 3.8. <only@+>[magick: rotate] Rotate an image:

```
magick convert input.jpg -rotate 180 output.jpg
```

Example 3.9. <only@+>[magick: negate] Get a negative image:

```
magick convert input.jpg -negate output.jpg
```

Example 3.10. <only@+>[magick: crop] Crop an image:

```
magick convert input.jpg -crop 320x240+0+0 output.jpg
```

Example 3.11. <only@+>[magick: caption] Caption an image:

```
magick convert input.jpg -pointsize 56 -gravity south -fill white -annotate +0+0 "Karl the
Fog" output.jpg
```

There's a bunch of other things ImageMagick can do! Their website has a full list.

3.2 Documents

Documents

- Sometimes we need to work with formatted documents.
- We *could* use Google Docs or MS Word, but what if we need to work with a lot of documents?
- There's a million incompatible document formats (Text, RTF, Word, ODT, PDF, HTML, Markdown, AsciiDoc, LaTeX, EPUB, DocBook, etc.), and trying to find the right tool for each one is hard.
- *Pandoc* is "a universal document converter" that can work with all these formats!

Pandoc

Pandoc is really just for converting between formats:

Example 3.12. <only@+>[pandoc] Converting between formats:

```
pandoc input.md -o output.docx
```

Example 3.13. <only@+>[pandoc: multiple files] Combining files and converting between formats:

```
pandoc title.md body.md epilogue.md -o output.docx
```

Example 3.14. <only@+>[pandoc: HTML fragment] Converting a Word Doc into an HTML fragment:

```
pandoc input.docx -o fragment.html
```

Example 3.15. <only@+>[pandoc: HTML page] Converting a Word Doc into an HTML website (e.g., a blog post):

```
pandoc input.docx --standalone --metadata title="My Website" -o
fragment.html
```

Example 3.16. <only@+>[pandoc: PowerPoint] Converting a Markdown file into a slideshow:

```
pandoc input.md -o output.pptx
```

[Demo Time (Again)]

Example 3.17. <only@+>[pandoc: PDF Slides] Converting a Markdown file into a PDF slideshow:

```
pandoc input.md -to beamer -o output.pdf
```

Once you've converted a file with Pandoc, you can edit it using whatever program you'd normally use.

3.3 Videos

FFmpeg

- Dealing with videos is a pain.
- There are video container formats: Matroksa, MPEG-4, QuickTime, Audio Video Interleave, WebM, Ogg
- There are video encoding formats: HEVC, H.264, AV1, VP9, VP8
- There are audio encoding formats: AAC, MP3, Opus, FLAC
- There are audio/video settings: bitrate, fps, resolution, sample rate
- *FFmpeg* is a tool to record, convert, and stream audio/video.
- *FFmpeg* also has a million different options and settings… ask a search engine if you ever need to use it.

FFmpeg Examples

FFmpeg examples from my command history:

Example 3.18. <only@+>[ffmpeg: record] Recording a video (on Linux):

```
ffmpeg -f v4l2 -framerate 30 -video_size 1280x720 -i /dev/video4 recording.mkv
```

Example 3.19. <only@+>[ffmpeg: container] Change a video container:

```
ffmpeg -i input.webm -vcodec copy -acodec copy screen.mkv
```

Example 3.20. <only@+>[ffmpeg: encoding] Reëncoding a video:

```
ffmpeg -i input.webm -vcodec h264 -acodec copy screen.mkv
```

Miscellanea

- If you choose to research a tool for your final project, your slides might look like today's:
  - What problem does this tool solve?
  - What does the tool do?
  - How do you use the tool?
- If you have fewer than six points by now (according to the guide from Lecture 1), come talk to us.