



Command-Line Alchemy: Turning Scripts into Superpower

no right answers?

```
cat access_log | grep 200.199.198.197
```

```
grep 200.199.198.197 access_log
```

```
grep -F 200.199.198.197 access_log
```

welcome to my world

[A] `bash -c "exit 0"`

[B] `bash -c "exit 1"`

[C] `false`

[D] `true`



\$POSIXLY_CORRECT

not posix compatible

[-z \$POSIXLY_CORRECT]

- A ‘tool’ that I wrote Oct 2025
- How do you crop images?
 - GIMP
 - vs gthumb
 - vs gthumb with keyboard shortcuts
 - vs imagemagick
 - vs imagecrop-interactive

- And of course, getting offsets for scripting!
- Extracted images from a GoPro [..] but many images have weird artifacts [..] is there a fix somewhere?

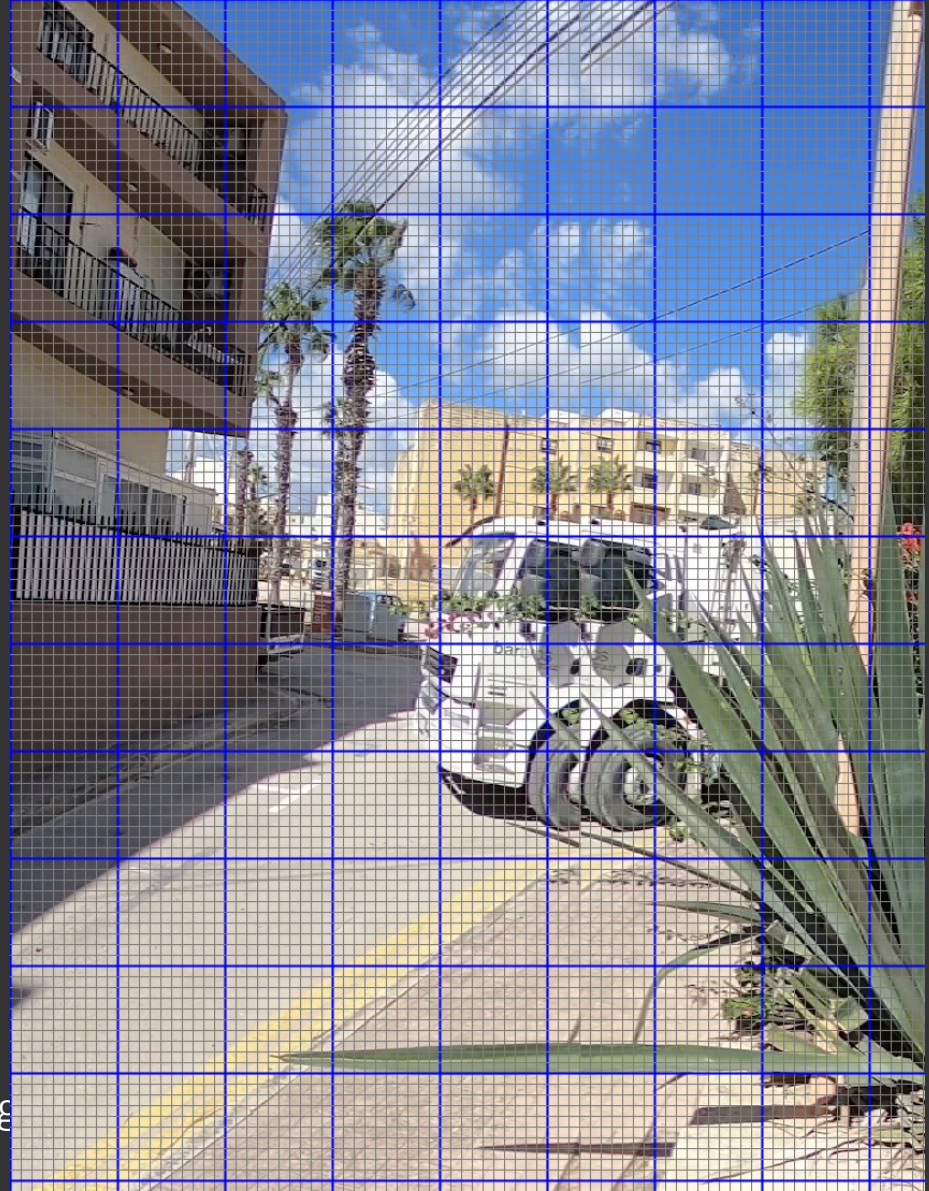
Source: @richlv@mastodon.social

BSides Riga 2026

<https://kirils.org>



```
for x in *.png; do
convert $x \(
input.png -gravity east
-crop 320x0+0+0 +repage \)
-geometry +52+0
-composite -gravity east
-crop +52+0 +repage
out-$x; done
```



- Write programs to handle text streams, because that is a universal interface.
- Write programs that do one thing and do it well.
- Write programs to work together.
- When a program has nothing surprising to say, it should say nothing.

Source: <https://cscie2x.dce.harvard.edu/hw/ch01s06.html>

```
cat access_log | grep -F 127.0.0.1 | cut -d : -f 2|sort|uniq -c
```

I want to testrun this without grep and readd it later. What's the best way to achieve that?

Wrong answers:

```
cat access_log # | grep -F 127.0.0.1 | cut -d : -f 2|sort|uniq -c
```

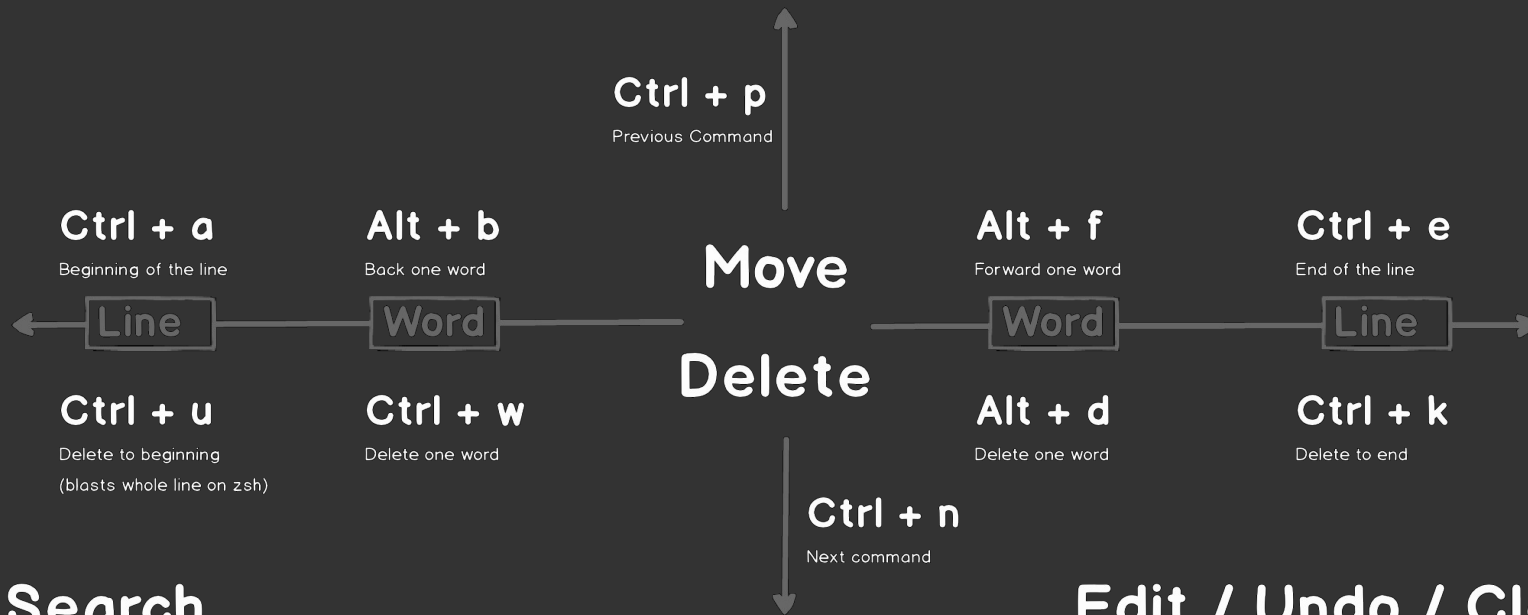
```
cat access_log | : grep -F 127.0.0.1 | cut -d : -f 2|sort|uniq -c
```

back to the basics

cmdchallenge.com

Navigating the Command Line

github.com/dwmkerr/effective-shell



Search

Ctrl + r **Ctrl + s**

Search Backwards Search Forwards

Ctrl + g ← , →

Cancel Edit

Edit / Undo / Clear

Ctrl + x,e **Ctrl + l** **Ctrl + _**

Edit Clear Undo

- file descriptors -- stdin, stdout, stderr --, redirects
 - not redirects: << and <<<
- pipes, fifo, process substitution, command substitution
 - | is not ||
- signals
- job control, subshells
 - & is not &&
- globbing, brace expansions
- variables: local, shell, env
- arithmetic

shellscripts

<https://github.com/oki/shellscripts/>

I think that's about enough of this

More funky
stuff here

