The TTY

Joseph Hallett

January 23, 2023

University of BRISTOL

Whats all this about then?

Bit of computing history

- That appears an awful lot *under the hood*
- You remember how disks no longer have cylinders and platters?
- You remember that I keep using `ed` in these lectures?

A Teletype!

Back before computers had screens...

You'd connect via a *serial* port

- Every character you sent would be typed by the printer
- Including *newlines*, *linefeeds*...
- Little bell for attracting attention if you sent a certain character

You didn't want to have to waste paper and ink (expensive)

- So line editors wouldn't show you what you were editing
- This is the sort of thing that UNIX was written on
- This is the sort of thing we were using 100 years ago

But we’re long past that, right?

```
ls /dev/tty*
```

```
/dev/tty /dev/tty0 /dev/tty1 /dev/tty10 /dev/tty11 /dev/tty12 /dev/tty13 /dev/tty14
/dev/tty15 /dev/tty16 /dev/tty17 /dev/tty18 /dev/tty19 /dev/tty2 /dev/tty20 /dev/tty21
/dev/tty22 /dev/tty23 /dev/tty24 /dev/tty25 /dev/tty26 /dev/tty27 /dev/tty28 /dev/tty29
/dev/tty3 /dev/tty30 /dev/tty31 /dev/tty32 /dev/tty33 /dev/tty34 /dev/tty35 /dev/tty36
/dev/tty37 /dev/tty38 /dev/tty39 /dev/tty4 /dev/tty40 /dev/tty41 /dev/tty42 /dev/tty43
/dev/tty44 /dev/tty45 /dev/tty46 /dev/tty47 /dev/tty48 /dev/tty49 /dev/tty5 /dev/tty50
/dev/tty51 /dev/tty52 /dev/tty53 /dev/tty54 /dev/tty55 /dev/tty56 /dev/tty57 /dev/tty58
/dev/tty59 /dev/tty6 /dev/tty60 /dev/tty61 /dev/tty62 /dev/tty63 /dev/tty7 /dev/tty8
/dev/tty9 /dev/ttyACM0 /dev/ttyACM1 /dev/ttyACM2 /dev/ttyS0 /dev/ttyS1 /dev/ttyS10
/dev/ttyS11 /dev/ttyS12 /dev/ttyS13 /dev/ttyS14 /dev/ttyS15 /dev/ttyS16 /dev/ttyS17
/dev/ttyS18 /dev/ttyS19 /dev/ttyS2 /dev/ttyS20 /dev/ttyS21 /dev/ttyS22 /dev/ttyS23
/dev/ttyS24 /dev/ttyS25 /dev/ttyS26 /dev/ttyS27 /dev/ttyS28 /dev/ttyS29 /dev/ttyS3
/dev/ttyS30 /dev/ttyS31 /dev/ttyS4 /dev/ttyS5 /dev/ttyS6 /dev/ttyS7 /dev/ttyS8
/dev/ttyS9
```

```
ls /dev/pts/*
```

```
/dev/pts/0 /dev/pts/2 /dev/pts/ptmx
```

Pseudo Teletypes

It turns out the *abstraction* that a TTY provides is a useful one

- Device with streams for input and output
- Suitable for keyboard based interaction
- Everything is a file

So UNIX based OSs use it as the fundamental control system

- Connect to terminals with `getty`, `cu`, `screen` or SystemD
- Configure terminals with `stty` and `stdbuf`

Except...

Obviously this *shouldn't work*:

- How do you handle colors?
- How do you handle images?
- How do you handle the mouse?

Some (mostly) standard stuff

The following keys *usually* work

`C-c`	interupt
`C-d`	end-of-file
`C-l`	clear screen
`C-t`	display a progress report (rarely works)

If you don't bother to configure `readline`:

`C-a`	start of line
`C-e`	end of line
`C-k`	clear line

(same as *Emacs*... these work in Mac OS everywhere too)

Readline?

Turns out dealing with the forward and backward character IO stuff is a pain
Readline makes it easier by abstracting some of it away

- Gives you command history
- Gives you delete without seeing a bunch of random escape codes
- ...If when using an app you see a bunch of `^[[A` or weird try running the command in `rlwrap`

You can do more with readline

- Have a look at `~/.inputrc` and search for its manual (Linux doesn't always install it)
- Also C bindings worth looking at if you write a lot of CLI apps

```
$if Bash
  "\C-xs": "\C-adoas \C-e"
$endif
```

Escape codes?

So what are these weird `^[[0m` things you see?

- Terminals haven't looked like that teletype since the 1930s

They still *technically* exist

- Receipt printers

...but they're rare.
Most terminals accept there is a screen with more than one row of text displayed at once!

- And all accept a bunch of extra control codes for dealing with multi-line text

Which is standardised right?

...lol.

A little bit?

See `man termcap`

- Then consider a career doing something else

If anything our modern PTY and terminal emulators are *less* featureful than the ones from the 1930s.

- Support for overstruck non-existant
- Advanced formatting pretty much gone
- Loads of support for glitches in particular terminals
- *Meta* key?!
- Non 8-bit bytes!
- Underlining but no overstriking modes!

Don't ever try and do this yourself

If you need to emulate a GUI on the commandline use `ncurses`

If you need to set colors for output use a library

Or in the worst case, use `tput`

`tput setaf 1`	make foreground color 1 (red)
`tput setab 2`	make the background of text 2 (green)

Make sure to handle the case when you're *not* outputting to a TTY `;-)`
Check `tput colors` to see how many colors your terminal has

- But it lies.

If you really need to do it the escape codes look something like:

- `\e[34mThis text in blue!\e[0m and now back to default`

But if you go beyond the default 8 colors, it rapidly gets weird...

Sometimes you need to emulate a TTY in a TTY

Use `screen` for that

- Or `tmux` if you'd like something less featured but more modern

Conclusion

TTYs still exist.

- Even if they're a pain.
- Mostly things *just work*.
- Expect suffering if you try and go beyond *just working*.