

Published on *Tux Machines* (<http://www.tuxmachines.org>)

[Home](#) > [Blogs](#) > [Roy Schestowitz's blog](#) > The Command Line Challenge

The Command Line Challenge

By *Roy Schestowitz*

Created *17/01/2022 - 10:10pm*

Submitted by Roy Schestowitz on Monday 17th of January 2022 10:10:50 PM Filed under [Just talk](#) [1]

Authored by Andy Farnell



Cheapskates wonderful guide is currently running a "[One Week Command Line Challenge \[2\]](#)". Some of the students I teach now are so young (to an old beard like me) they think this is some "crazy new thing". Is there new hope and a new perspective to be explored here? Something other than *retro and cool*. Perhaps historical baggage, the narrative of

how "superior" graphical interfaces replaced "old" consoles is an obstacle to new visions for the next generation?

As a lifelong textual user interface (TUI) user this got me thinking. If you were to give me "The GUI Challenge" I'd be sunk! My world (dwm, emacs, w3m etc) feels so familiar, it's in my bones. After thirty or forty years on the command line if I were forced to use "normal computers" it would cripple my ability to do anything.

The command-line is super empowering, but particular. Put me on a Mac or Windows machine and I revert to a child-like flap, randomly clicking around on icons that look promising. I'd be twenty times less productive than my peers, yet, modesty be damned, I'm ten times *more* effective/productive at average computing tasks than other professionals when in my comfort zone - at the command-line. Isn't this true for us all, that we have our comfy shoes?

Of course this isn't about some innate inability to use graphical tools. I've mastered some jolly complex ones like Blender and Unreal editors (virtual world building), and ProTools or Ardour (for sound and music). One of the most complex I recall was a VLSI/CAD creator that used two four button mice (or mouse and ball).

So, is the command line challenge unfair? I am no more capable of quickly learning a new graphical paradigm than an entrenched GUI user is of adopting the keyboard and console. This probably applies at any age or ability level where you are comparing like-for-like paradigm switching.

No, the issue here is deeper and is about utility paradigms. How do people relate to computers as tools at the highest level - at the operating system level and above?

If you dig back in the Usenet and mailing-list archives, you'll find fascinating, passionate and intelligent debates on the merits of different interfaces going right back to Xerox-PARC. They are really separate computing cultures. There's a fair [historical summary](#) [3] here.

The above history ends in 2001. GUIs did not end there, the debate has moved further, and many new things have not been well analysed. Mobile, which essentially emulates button-based handheld appliances, cannot really be compared to GUI (in its traditional sense), even though it's technically a computer running a graphical interface.

It's only since about 2010 that the GUI function of abstracting (hiding away complexity) was subverted by wicked corporations to hide away deception and to effect control. This shift from the abstract to the abstruse and obstructive is what we sometimes call "Dark Computing Patterns", but really it goes deeper than that - visual computing is it's own realm of psychology, politics, semiotics, iconography and subterfuge that in many cases thoroughly bastardises the function of computers qua "tools".

The GUI/TUI debate can be framed in many ways; preference, freedom, extensibility, cognitive overhead, portability, control (tweakability), depth of understanding (legibility), and more.

For me, tool longevity and stability are important. I still use the same applications and skills I learned in 1980. Some people, foolishly I think, imagine that to be a bad/anti-progressive stance. One of the most underrated abilities in computer programming is knowing when something is finished. As is the ability to just *use* something instead of worshipping it as a digital artefact (cue NFT "first editions of brand apps").

By contrast many of my colleagues must re-learn their entire productivity stack every few months at the whim of corporate developers or seemingly random events in "the market". I literally hear them anthropomorphising:

"Oh, Slack won't let me do that now"

"Oh, Google ate my email"

"Sorry, something broke, can you resend it please?"

Their "computers" are chaotic mystery machines, magic fun fairs where superstitious ritual ministrations must be

performed. This sort of Scooby-Doo "clown computing" has no place in serious business, in my opinion. So, another hugely underrated quality that TUIs favour is *stability*.

Where did this mess come from? In the 1980s "home computers" created a culture of their own, and from there Apple and Microsoft, needed to counter a socially constructed but actually mythical "fear" of computers as nerdy and silly, but also "dangerous". Remember granny worrying that it would "blow up" if you typed the wrong thing?

Continuing a culture of sysadmins from the time-sharing Unix days, we created the "user" as a particular stereotype. To put it quite bluntly, we manufactured "users" to be idiots. Indeed, use of the word "users" instead of a more neutral term like "operators" is significant. The developer-user relationship today is a power relationship, and often an abusive one (in both directions).

In fact denigrating attitudes have their roots in the fragility of early software development. The "user" was an enemy who would always find ways to break our software and exhibit extraordinary "stupidity" by failing to understand our non-obvious interface puzzles. We used tropes like (P.E.B.K.A.C), lusers, and treated others with disrespectful and superior smugness.

Computing had its hashtag moment, and markets demanded that perceptions change. Microsoft solved the problem by erecting some soothing blue fire-hazard cladding around a crumbling DOS. Underneath, exposure to "The Registry" was like staring directly into the open core of Chernobyl.

At that point, enter Apple, who could play Good Cop, adding value by simply subtracting (or consolidating) features. For many, Steve Jobs was elevated to the man who "invented computers". For a certain generation, he did. The ancient science of HCI (human computer interaction) was beaten and disfigured into the designer denomination of UX/UI that emphasised intuition, feel, and experience, which in turn ushered in the age of performative productivity. This trajectory of form over function culminated in neurotic obsessions with \$2000 disposable thin laptops and the Onion's infamous [Apple Wheel](#) [4] parody that confused many as to whether it was a genuinely good idea.

Meanwhile the command line simply kept calm and carried on. Nothing changed in 30 years. Those who ran the servers, databases, scientific and technical applications never strayed far from the console, except where "presentation" demanded. However, through the mass media and advertising, digital technology became synonymous with these corporate veneers over actual *computers*, while Hollywood made the command-line a glowing green preserve of malcontents bent on destroying civilisation.

So, although the Command Line Challenge is fun - and I hope it inspires some people to go beyond their comfort zone - let's be aware that human factors, history and politics play a greater role behind the scenes. Yes, it's about mental models, rote motor skills and habits, rather than any intrinsic good or bad. But it's also about culture and popular ideas of what a computer "is".

The emphasis of Cheapskate's article is on TUI allowing the use of older computers. That's a very topical and important concern in the age of climate emergency. If readers don't know already about books like Gerry McGovern's [World Wide Waste](#) [5], I urge you to read more about e-waste. Making the connections between textual interfacing, more modest [tech-minimalist](#) [6] use, and a better society and healthier planet, isn't obvious to everyone.

There are many reasons people may prefer to return to the command line. I vastly prefer TUI's for another reason. As a teacher I deal in ideas not applications, so it's a way of imparting lasting concepts instead of ephemeral glitter. Commands are connections of action concepts to words, essential for foundational digital literacy. Almost everything I can teach (train) students to use by GUI will have changed by the time they graduate.

For younger people the difference is foundational. My daughter and I sit down together and do basic shell skills. She can log in, launch an editor, play music and her favourite cartoon videos. We use Unix talk to chat. It's slow, but great fun, because character based coms is very *expressive* as you see the other person typing. She's already internalising the Holy Trinity - storage, processing and movement.

To make this work I obviously customised bash, creating a kind of safe sandbox for her with highly simplified syntax. This week we are learning about modifier keys - shift is for SHOUTING and control is to CANCEL (you can't get around needing to teach CTRL-C). What we are really working on is her typing skills, which are the foundation of digital literacy in my opinion. I think at the age of 5 she is already a long way ahead of her school friends who paw at tablets.

In conclusion then, the TUI/GUI saga is about much more than interchangeable and superficial ways of interacting with computers. In it's essence it is about literacy, the ability to read and write (type). Behind, and ahead of it, are matters of cultural importance relevant to education, autonomy, democracy, self-expression, and the economy. So if you're a mouser or screen smudger, why not give Cheapskate's challenge a try? [?](#)

[Just talk](#)

Source URL: <http://www.tuxmachines.org/node/160388>

Links:

- [1] <http://www.tuxmachines.org/taxonomy/term/68>
- [2] <https://cheapskatesguide.org/articles/cmdline-challenge.html>
- [3] <https://kartsci.org/kocomu/computer-history/graphical-user-interface-history/>
- [4] <https://www.theonion.com/apple-introduces-revolutionary-new-laptop-with-no-keybo-1819594761>
- [5] <https://gerrymcgovern.com/books/world-wide-waste/>
- [6] <https://www.calnewport.com/books/digital-minimalism/>