

1. Kinds of terminal program

Kind	Needs
One-shot shelF list	args, streams, exit status
Suite shelF add loan	+ help, usage errors, completions
REPL shelF repl	line editing, history, key-at-a-time input
TUI shelF tui	raw mode, alt screen, mouse/resize, guaranteed restore
Daemon client	local IPC + service unit

A TUI should be a view over a tool that also works without one.

2. Which stack?

Lang	Start / size	Cross-compile	TUI lib
C / C++	fast, small	zig cc / clang cross toolchain	ncurses 6.6, FTXUI v7.0.3
Rust	fast, no GC	rustup target , cross, cargo-zigbuild	Ratatui 0.30.2
Go 1.27.3	fast, runtime linked in	G00S/G0ARCH, CG0_ENABLED=0	Bubble Tea v2.1.0
Swift 6.4.1	native; Linux needs runtime or Static SDK	Swift SDKs (musl)	weak (SwiftTUI dormant)
Obj-C	very fast	Apple; GNUstep elsewhere	ncurses
.NET 10 (10.0.13)	Native AOT: fast	per RID; AOT not cross-OS	TerminalGui v2.5.0, Spectre 0.57.2
JVM / Kotlin	JVM startup; GraalVM 25 native	bytecode portable; jlink/jpackage	JLine 4.4.8, Lanterna, TamboUI 0.6.0, Mosaic
Python 3.15.1	interpreter; Pylinstaller 6.22.3	bundle per OS	Textual 8.2.8, Rich 15.0.0
Node 24	runtime; SEA	build per OS	Ink 8.0.0 (React 19.3+, Node 22+)

3. Terminal stack, canonical vs raw

emulator (draws grid, sends keys) ↔ **PTY** master/slave ↔ kernel **line discipline** ↔ shell / program (fd 0,1,2). Windows: Windows Terminal / conhost + **ConPTY**.

Canonical (ICANON): line-buffered, echo, edit keys, EOF on Ctrl-D.

Raw/cbreak: tcgetattr, clear ICANON|ECHO (ISIG, IEXTEN for raw), VMIN=1, tcsendp; always restore (atexit, signals, panic). Size: TIOCGWINSZ, SIGWINCH.

4. Escape sequences (ECMA-48 / xterm)

Seq	Effect
CSI 1;31 m / 0 m	SGR bold red / reset; 30-37 fg, 40-47 bg, 90-97 bright
38;5;n 38;2;r;g;b	256 / truecolor fg (48 = bg)
CSI r; c H, n A/B/C/D	cursor position (1-based) / move
CSI 2 J, 2 K	erase screen / line
CSI ?25 l/h	hide / show cursor
CSI ?1049 h/l	alternate screen on / off
?1002 h + ?1006 h	mouse drag + SGR encoding CSI < b;x;y M/m
?2004 h	bracketed paste: CSI 200~ ... CSI 201~
?1004 h	focus in/out CSI I/O
?2026 h/l	synchronized output (community; query ?2026\$p)
OSC 8;;URL ST text	hyperlink; ST = ESC \
OSC 8;; ST	

Colour policy: no colour if not a TTY or NO_COLOR non-empty; truecolor if COLORTERM=truecolor|24bit; 256 if TERM ends ~256color; FORCE_COLOR forces on; TERM=dumb none. Prefer the 16 base colours (user palette). Never print untrusted text raw (strip ESC/OSC).

5. POSIX essentials

- fork+exec+waitpid, or posix_spawn (preferred); pipe, dup2, 0_CLOEXEC
- fds: 0 stdin, 1 stdout (data), 2 stderr (diagnostics); 2>&1
- sigaction (not signal), SA_RESTART; handlers set a flag / self-pipe only
- SIGPIPE on closed pipe then EPIPE; SIGWINCH resize; SIGINT/SIGTERM/SIGHUP stop; SIGTSTP suspend
- Exit: 0 ok; 126 found, not executable; 127 not found; 128+N killed by signal N (130 = Ctrl-C); keep 0-255

6. Other TUI skeletons

- Bubble Tea v2** charm.land/bubbletea/v2: Init() Cmd, Update(Msg), View() tea.View (tea.NewView(s)), tea.KeyPressMsg, tea.NewProgram(m).Run()
- Textual**: class ShelfApp(App), compose() yields widgets, CSS, reactives, app.run(); test with Pilot
- TerminalGui v2**: using IApplication app = Application.Create(); app.Init(); app.Run(window);
- ncurses**: setlocale, initscr, cbreak, noecho, keypad, KEY_RESIZE, endwin via atexit; Windows: PDCursesMod v4.5.4
- FTXUI**: ScreenInteractive::FullScreen(), screen.Loop(component)

7. Windows console essentials

- SetConsoleMode out: ENABLE_PROCESSED_OUTPUT | ENABLE_VIRTUAL_TERMINAL_PROCESSING; in: ENABLE_VIRTUAL_TERMINAL_INPUT; raw: clear ENABLE_LINE_INPUT, ENABLE_ECHO_INPUT; ENABLE_WINDOW_INPUT for resize
- UTF-8: SetConsoleOutputCP(CP_UTF8) + SetConsoleCP (console-wide: restore on exit) or manifest activeCodePage UTF-8 (1903+)
- argv: **one command-line string**, split by the program. MSVC rule: 2n backslashes + quote = n backslashes, quote toggles; 2n+1 = n + literal quote; backslashes elsewhere literal. CommandLineToArgvW differs slightly
- Ctrl handler: SetConsoleCtrlHandler: CTRL_C_EVENT, CTRL_BREAK_EVENT, CTRL_CLOSE_EVENT...; no SIGTERM. Exit 0-255 (259 = still active)
- ConPTY (1809+): CreatePseudoConsole + two pipes + attribute list

8. Platform differences

	Windows	macOS	Linux
Path / case	\ or /, drives; case-insens.	/; APFS case-insens.	/; case-sens.
Config dir	%APPDATA%, %LOCALAPPDATA%	~/Library/Applicat ion Support	XDG ~/ .config
PATH / exe	;;, PATHEXT, no shebang	;, +X, #!	;, +X, #!
Stop	console ctrl events	signals	signals, signalfd
Text	UTF-16 wide API, CRLF	UTF-8, LF	bytes, LF
Exe path	GetModuleFileNam e	_NSGetExecutablePa th	/proc/sel f/exe

BSD userland: sed -i '', date -j -f, no grep -P on macOS. CI matrix: ubuntu / macOS / windows-latest.

9. Argument parsing per language

Lang	Use
C	getopt / GNU getopt_long (optstring, struct option), argp; MSVC: wmain
C++	CLI11 v2.7.2 (subcommands, validators), cxxopts, Boost.Program_options
Swift	swift-argument-parser 1.8.2: @main ParsableCommand / AsyncParsableCommand, @Argument/@Option/@Flag
Obj-C	NSProcessInfo; NSUserDefaults argument domain -key value; getopt_long
.NET	System.CommandLine 2.0.13: RootCommand, Option<T>, Argument<T>; dotnet run app.cs
Go	flag / FlagSet, urfave/cli v3.14.0, Cobra v1.10.2
Rust	clap 4.6.7 derive (Parser, Subcommand, ValueEnum, env); anyhow, color-eyre
JVM	Clickt 5.1.0 (+ Mordant 3.1.0), picocli 4.7.7; not kotlinc-cli (obsolete)
Python	argparse, Click 8.5.0, Typer 0.27.3

10. TUI loop and library row

```
enter raw + alt screen + hide cursor
loop { read event (key|mouse|paste|resize)
  update(state, msg) // TEA: Model, Update, View
  draw(state) -> cell buffer -> diff -> escapes }
restore on EVERY exit path (error, panic, signal, suspe nd)
```

Lib	Lang	Model
Ratatui	Rust	immediate, you own loop
Bubble Tea v2	Go	Elm (retained msgs)
Textual	Python	retained DOM + CSS
TerminalGui v2	.NET	retained views
ncurses / FTXUI	C / C++	windows / DOM
Ink 8	Node	React components

11. Ratatui skeleton (0.30.2)

```
fn main() -> std::io::Result<()> {
  ratatui::run(|t| loop {
    t.draw(|f| {
      let [list, status] = Layout::vertical(
        [Constraint::Fill(1), Constraint::Length(1)]), a
      reas(f.area());
      f.render_widget(List::new(items).block(Block::bor
        dered()), list);
      f.render_widget("q: quit", status);
    });
    if event::read().? .is_key_press() { break Ok(()); }
  })
}
```

init = raw + alt screen + panic hook; restore undoes. Constraints: Length, Percentage, Ratio, Min, Max, Fill. Windows: filter KeyEventKind::Press. Async: Tokio + EventStream + action channel.

Test: TestBackend::new(w,h) + Terminal + assert_buffer_lines; snapshots with insta 1.49.0. No 1.0: pin 0.30.

12. Prompts, line editing, progress

Lang	Line edit / prompts	Progress / colour
C/C++	GNU Readline 8.3, libedit, linenoise, replxx	manual SGR
Rust	rustyline 19.0.0, readline; inquire, dialoguer	indicatif 0.18.6, anstream, owo-colors
Go	Huh v2.0.3	Bubbles, Lip Gloss, mpb
Python	prompt_toolkit 3.0.53, questionnaire	Rich, tqdm, colorama
.NET	Spectre prompts	Spectre Progress/Status
Node	@inquirer/prompts, @clack/prompts	ora 9.4.1, chalk 6.0.1
JVM	JLine 4.4.8	Mordant

Password: clear ECHO (Windows ENABLE_ECHO_INPUT). Non-TTY: no prompts, no spinners.

13. Completions and services

- bash complete/compgen; zsh compsys (fpath before compinit, _shelF; fish ~/.config/fish/completions/shelF.fish; PowerShell Register-ArgumentCompleter; Nushell extern
- Generate: clap_complete, Cobra completion cmd, swift-argument-parser completion script, dotnet completions script, argcomplete. Dynamic: Cobra __complete
- Shell state change (cd): eval "\$(_shelF_init zsh)"
- systemd**: unit Type=exec(notify; systemctl enable --now shelF; journalctl -u shelFd -f
- launchd**: launchctl bootstrap gui/\$(id -u) ~/Library/LaunchAgents/x.plist, bootout
- Windows**: sc.exe create "Bookshelf Daemon" binpath= ..., sc.exe failure, .NET Worker Service
- IPC: Unix domain socket / Windows named pipe; lock or PID file

14. Native binaries and cross-compile

```
rustup target add x86_64-unknown-linux-musl
cargo build --release --target x86_64-unknown-linux-musl
cargo zigbuild --release --target universal2-apple-darwin
G00S=linux G0ARCH=arm64 CG0_ENABLED=0 go build
swift build -c release --arch arm64 --arch x86_64
swift build -c release --swift-sdk x86_64-swift-linux-musl
dotnet publish -c Release -r win-x64 -p:PublishAot=true
zig cc -O2 -target x86_64-linux-musl shelF.c -o shelF
clang -arch arm64 -arch x86_64 shelF.c -o shelF; lipo -archs shelF

musl = static; glibc = symbol versions (target old glibc, e.g. .2.17);
Windows CRT /MT vs /MD; macOS --mmacosx-version-min. Native
AOT / GraalVM: build per target OS. Reproducible:
SOURCE_DATE_EPOCH. Release tools: dist 0.33.1, GoReleaser.
```

15. Signing, notarization, packaging

```
codesign --force --options runtime --timestamp --sign
"Developer ID Application: ..." shelF
ditto -c -k --keepParent shelF.zip
xcrun notarytool submit shelF.zip --keychain-profile sh
elf-notary --wait
pkgbuild --root root --identifier com.example.shelF she
lF.pkg
xcrun stapler staple shelF-1.0.0-signed.pkg # bare Ma
ch-O: no staple
signtool sign /fd SHA256 /tr URL /td SHA256 shelF.exe
signtool verify /pa /v shelF.exe
sha256sum *.tar.gz > SHA256SUMS; gpg --armor --detach-s
ign SHA256SUMS
cosign sign-blob SHA256SUMS --bundle SHA256SUMS.sigstor
e.json --yes
```

OS	Channels
macOS	Homebrew tap, signed .pkg/.dmg, zip
Windows	winget (wingetcreate), Scoop, Chocolatey, MSI/MSIX
Linux	./deb/.rpm (nfpm, cargo-deb), apt/dnf repo, Snap, AppImage, Nix

Gatekeeper quarantine on downloads; SmartScreen reputation. curl | sh installers are a risk.

16. Testing PTY and TUIs

- Real PTY: expect, pexpect 4.9.0, rexpect / expectrl, portable-pty; Windows ConPTY in CI
- Snapshots: Ratatui TestBackend + insta; Textual Pilot + pytest-textual-snapshot 1.1.0; Bubble Tea teatest
- tmux: new-session -d -x 100 -y 30, send-keys, capture-pane -p
- Demos / visual regression: VHS v0.12.1 (vhs x.tape), asciinema rec -c "shelF tui"

17. Accessibility and i18n

- Provide a plain mode (no animation, no box drawing, words not colour alone); honour NO_COLOR / --color; explicit flag beats env
- Use 16 ANSI colours; check contrast on light and dark themes
- Screen readers read linear text: keep shelF list complete without the TUI; print the final result after leaving the alt screen
- Localize with gettext, Project Fluent, .NET resources, String Catalogs; plural rules; Unicode normalization; wide characters (UAX #11)