

# The Biggest Shell Programs in the World

Edit

New page

Matěj Cepl edited this page on Apr 23 · [32 revisions](#)

Help me fill out this page! It's freely editable.

What programs should be listed? I'm using "biggest" in the sense of **substantial**, not necessarily the raw number of lines.

- **Hand-written** shell scripts. There are many big autoconf-generated scripts, like the [70K-line coreutils script](#), but I don't consider them substantial in this sense.
- Shell programs that use **data structures and algorithms**.
  - As a counterexample, [bash-completion](#) is pretty sophisticated, but it's also repetitive because it has a relatively simple function for every command on a Unix machine.
- Shell programs that are over **5K lines** (though there may be exceptions). The biggest shell programs that aren't repetitive tend to be in the 10K+ line range. I haven't seen any programs over 100K lines.

## The List

- [akinomyoga/ble.sh](#) -- 61K LoC (46K SLoC) in total. Bash Line Editor---fish-like interactive line editor in pure bash! The main file `out/ble.sh` has 30K lines of code (23K SLoC), but there are 60K+ lines of code in the repo in total. There are many comments (in Japanese).
  - [How Interactive Shells Work](#) has a nice overview of how ble.sh works. It's very sophisticated, using `bind -x` to read raw bytes from the terminal, decoding those itself in multiple explicit state machines, maintaining a drawing buffer, updating the buffer, etc. It has timing and "fibers", etc.
  - [This comment on issue 663](#) has details on the shell parser! I think this is one of the most sophisticated uses of data structures in shell I've seen.
  - [We are trying to run ble.sh](#) under Oil. It mostly parses.

Pages 137

▶ [Home](#)▶ [Ad Hoc Protocols in Unix](#)▶ [Alternative Regex Syntax](#)▶ [Alternative Shells](#)▶ [Architecture of Various Inter...](#)▶ [Autocompletion Use Cases](#)▶ [BootstrappingCaseStudies](#)▶ [Can Oil Use Nix?](#)▶ [Capers](#)▶ [Carrots](#)▶ [CLAP Grammar Proposal](#)▶ [Command vs Expression Mode](#)▶ [Compact AST Representation](#)▶ [Compiler Engineer Job](#)▶ [Compiler Engineer Notes](#)

Show 122 more pages...

+ Add a custom sidebar



- First commit in 2015 with 8K LoC / 6 SLoC. The actual development has started in 2013.
- [bashdb](#), the bash debugger. ~14K lines of bash. Interesting history: [Implementing Debuggers](#)
- [drwetter/testssl.sh](#) -- 21K lines of bash in a single file! Appears to be hand-written. History: *Back in 2006 it started with a few openssl commands....* (Hit issue #606 parsing it.)
- [Simplenetes: Kubernetes in 17K lines of Shell](#). Amazing! But seems dormant. [Hacker News Thread](#).
  - Related to [my bold claim here](#)
- [rkhunter](#) ([official site](#)) -- 21K lines of Bourne shell written from 2003-2018!
- [romkatv/powerlevel10k](#) -- 12K lines of zsh scripts in the directory `internal/`. There are other 8K lines of configs and helper scripts. First commit in 2014.
- [dylananaraps/neofetch](#) -- 10K lines of bash 3.2. Displays system information. May also do something interesting with images (?)
  - first commit 2015
- [xwmx/nb](#) -- 24K LoC (19K SLoC) of bash in [nb](#) itself. An additional 86K LoC (55K SLoC) if we count bats tests as bash.
  - First commit in 2014, but active commit history starts in early 2016.
- [distrobox](#) -- over 7k lines of bash script. Use any linux distribution inside your terminal.
- [acme.sh](#) -- 7K lines of shell script. Issues and renews certificates.
- [inxi 2.3.56](#) [obsolete] -- 16K lines of bash. A fork of `infobash` in 2008 (889 lines then). `infobash` has started in 2005. From v2.9, `inxi` is replaced by the Perl implementation.
- [bashforth](#) -- At ~3800 lines this isn't huge, but it apparently implements a real programming language. It has a lot of whitespace and comments.
- [vegardit/bash-funk](#) -- 27K LoC (24K SLoC) in total. A Bash library. The first commit in May, 2017 (with 10K LoC / 8K SLoC).

## Shell-Like Languages / DSLs

- [modernish](#) is a portable shell dialect written in shell
- [bats](#) is a DSL for writing tests. Generates bash code.
- [bashible](#) is an Ansible-like DSL in bash. [comments](#)
- [clash](#) is an object oriented framework compatible with any modern POSIX shell.

- [bash Infinity](#) is a standard library and a boilerplate framework for bash.

## Smaller Programs

- Alpine, Aboriginal, Debian scripts -- see [blog post](#)
- Completion scripts are big, but often repetitive.
  - [\\_git](#) Zsh completion -- 8.3k lines of code
  - git-completion.bash
  - Docker completion
- [dyne/Tomb](#) is a ~3500 line zsh script.
- [Basalt](#) -- a full-featured package manager in pure Bash (est. ~2021). Although it's only a few thousand lines, there's already a [rich ecosystem](#) (15+ apps/libs) that includes many thousand more lines. Some *Bash libraries* include (at various stages of development):
  - [bash-core](#), a library for enhancing the `trap` and `shopt` builtins, adding `stacktraces`, and many essential amenities
  - [bash-object](#), a library for constructing arbitrarily nested data structures in pure Bash (nearly 200 tests)
  - [bash-json](#), a library for parsing and printing JSON in pure Bash

## Links

- [Oil's "Wild" Tests](#) parse over a million lines of shell. However most of these are small programs and distro package definitions like Alpine PKGBUILD and Gentoo ebuilds, which are repetitive.
- [Shell Programs That Run Under OSH](#)

+ Add a custom footer