



a web browser that adheres to the Unix Philosophy

markus schnalke <meillo@marmaro.de>

better:

thoughts on sane web browsers

Recap: The Unix Philosophy (1)

Gancarz:

- small is beautiful
- make each program do one thing well
- use software leverage to your advantage
- avoid captive user interfaces
- make every program a filter
- ...

Recap: The Unix Philosophy (2)

Mcllroy:

- write programs that do one thing and do it well
- write programs to work together
- write programs to handle text streams

modern web browsers

- ... are not small (huge amounts of code)
- ... do not do one thing (include lots of stuff)
- ... do not use software leverage (do not use available tools)
- ... have captive user interfaces (do not fit into the Unix UI)
- ... are no filters (what about html2text?)
- ... do not work together (everything's already included)
- ... do not handle text streams (have no interfaces anyway)

⇒ **They do completely conflict with the Unix Philosophy!**

Three problems to solve

- 1) user interface
- 2) size, simplicity
- 3) software leverage, combination, filters

Problem 1: user interface

already addressed (e.g. by vimperator)

pretty easy to implement

but/only a user-side problem

Problem 2: size, simplicity

suckless community's limit: 10k SLOC

but:

- gecko (xulrunner-1.9: 2.6m SLOC)
- webkit (webkit-1.1: 390k SLOC)
- khtml (gtkhtml-2.8: 70k SLOC)

(now imagine 0.1–10 bugs/KLOC)

simplicity is not possible because of today's web (→ digression)

digression: today's web

... is broken!

- state in a state-less technology (deep-links, back-button)
 - misused technologies (flash)
 - totally overloaded, much too complex
- ⇒ simple render engines are not possible anymore
- ⇒ web browsers have no chance – they are essentially complex

Problem 3: software leverage, combination, filters

this is the point to put hands on

– why should the bookmark management be **inside** the browser?

– why does the browser need an **own** download manager?

⇒ better: use external programs that are available!

How could it look like

- take one of the bloated render engines (black box)
- wrap it into a small program with software leverage in mind
- have interfaces to refer to **external** programs
- add a good user interface

Examples (1)

uzbl

`http://uzbl.org`

by Dieter Plaetinck (Dieterbe)

since 2009-04

started as `http://bbs.archlinux.org/viewtopic.php?id=67463`

2700/1100 SLOC (2300/280 SLOC in May)

is quite active (various branches)

impressive work was achieved in short time

Examples (2)

surf

`http://surf.suckless.org`

by Enno Boland (Gottox)

since 2009-06

500 SLOC

minimalistic reimplementaion of uzbl !?

like uzbl was in the very beginning

5 steps to improve the software world

- 1) understand: the Unix Philosophy!
 - 2) realize: a lot of modern software does not comply
 - 3) realize: it's mostly the same kinds of problems
 - 4) realize: it can be done better
 - 5) do it better!
- ⇒ write, help, use sane software!

software used:

- Debian GNU/Linux
- LaTeX beamer, latexmk, fbgs
- vim, sloccount, mercurial

Thanks for your attention

The slides are available on <http://marmaro.de/docs>

2009-08-13