The UNIX Philosophy in 2023

Jason W. Eckert

jason.eckert@trios.com

jasoneckert.github.io

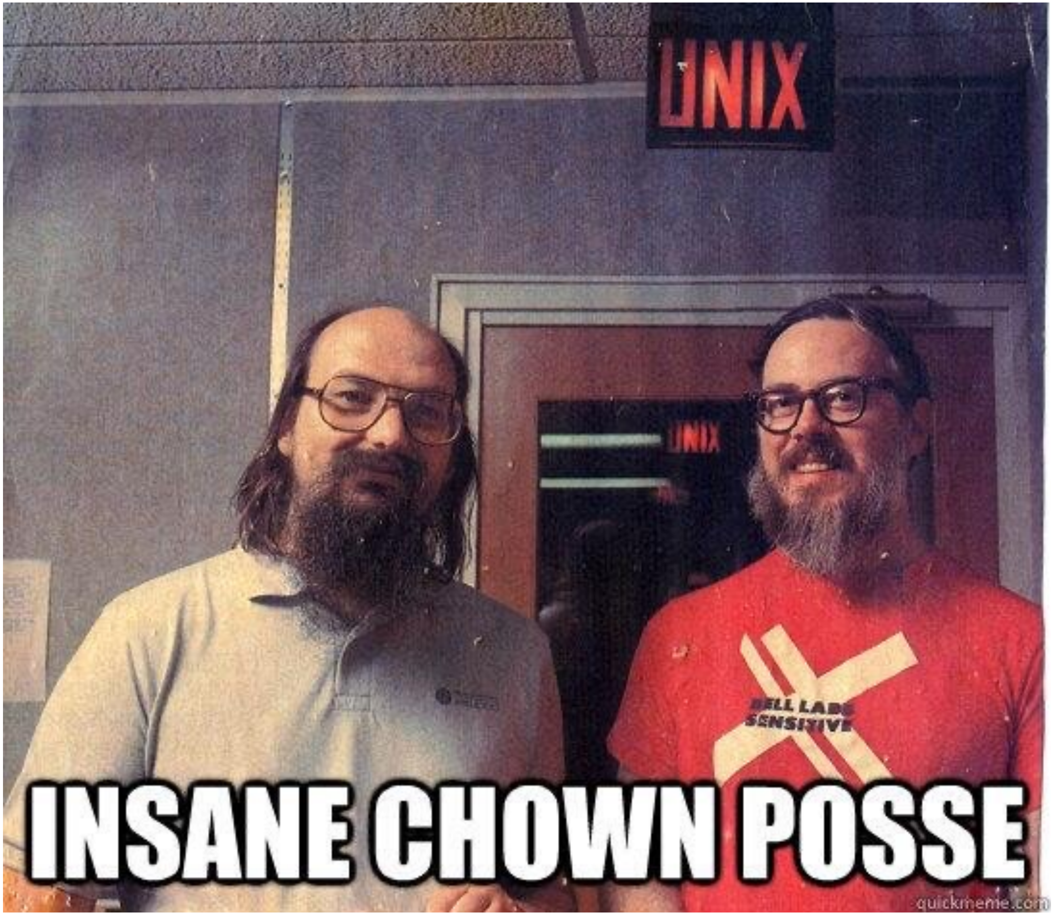
Wait, isn’t this a Linux User Group?

- Today, UNIX is considered a philosophy as well as a group of operating systems that evolved from the original UNIX operating system that share a similar structure and command set

Wait, isn’t this a Linux User Group?

- Today, UNIX is considered a philosophy as well as a group of operating systems that evolved from the original UNIX operating system that share a similar structure and command set:
  - BSD (FreeBSD, NetBSD, OpenBSD)
  - macOS
  - Linux

Brief UNIX Background



Brief UNIX Background

- Epoch (1969) – the first UNIX
  - Ken Thompson & Dennis Ritchie (AT&T)
  - AT&T sold the source to others (Xenix, HP-UX, AIX, IRIX, SCO, Sun Solaris)
  - Berkeley created BSD (SunOS, NeXTSTEP→macOS, Ultrix, OSF/Tru64, 386BSD, Free/Net/OpenBSD)

Brief UNIX Background

- Epoch (1969) – the first UNIX
  - Ken Thompson & Dennis Ritchie (AT&T)
  - AT&T sold the source to others (Xenix, HP-UX, AIX, IRIX, SCO, Sun Solaris) = **System III/IV/V UNIX**
  - Berkeley created **BSD** (SunOS, NeXTSTEP→macOS, Ultrix, OSF/Tru64, 386BSD, Free/Net/OpenBSD)

Brief UNIX Background

- Epoch (1969) – the first UNIX
  - Ken Thompson & Dennis Ritchie (AT&T)
  - AT&T sold the source to others (Xenix, HP-UX, AIX, IRIX, SCO, Sun Solaris) = **Sys III/IV/V UNIX**
  - Berkeley created **BSD** (SunOS, NeXTSTEP→macOS, Ultrix, OSF/Tru64, 386BSD, Free/Net/OpenBSD)
  - **Linux** was instead derived from the FSF GNU project (designed to create an open source UNIX)

Brief UNIX Background

- Epoch (1969) – the first UNIX
  - Ken Thompson & Dennis Ritchie (AT&T)
  - AT&T sold the source to others (Xenix, HP-UX, AIX, IRIX, SCO, Sun Solaris) = **Sys III/IV/V UNIX**
  - Berkeley created **BSD** (SunOS, NeXTSTEP→macOS, Ultrix, OSF/Tru64, 386BSD, Free/Net/OpenBSD)
  - **Linux** was instead derived from the FSF GNU project (designed to create an open source UNIX)

- Want details? Google "*Ultimate UNIX Timeline*"

What is the UNIX Philosophy?

Better applications with Unix Philosophy

JeffD • Apr 14 '20

DEV

Unix Philosophy in WordPress Plugins

// John James Jacoby

LOOP CONF

FEBRUARY 21–24, 2018

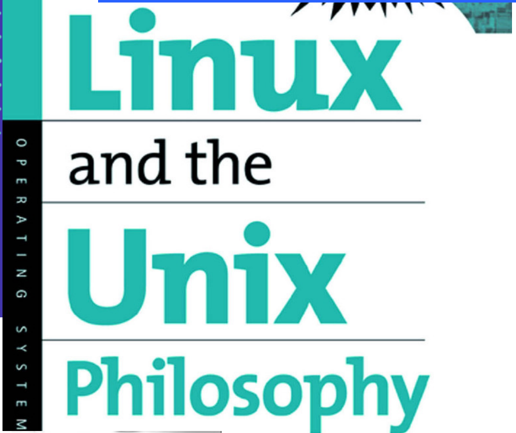
Steve Springer
stevesdevnotes.hashnode.dev

Unix design philosophy and music are great teammates

2 min read

Unix Philosophy

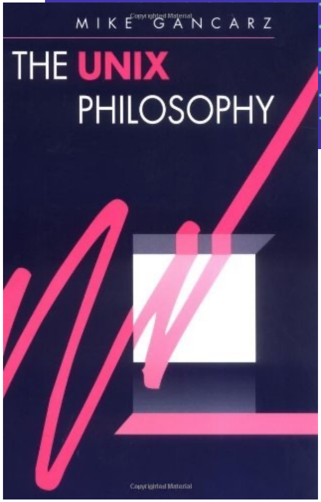
5 Tenants of The Unix Philosophy



M

Linux and the Unix Philosophy

OPERATING SYSTEM

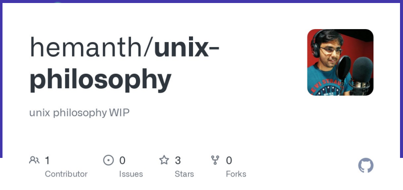


Copyrighted Material

MIKE GANCARZ

THE UNIX PHILOSOPHY

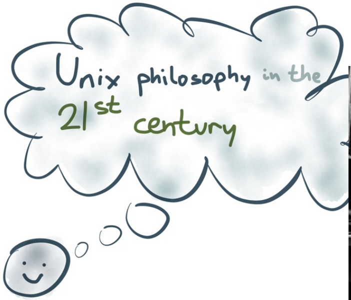
Copyrighted Material

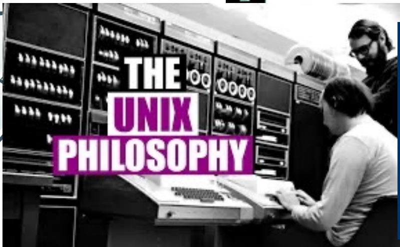


hemanth/unix-philosophy

unix philosophy WIP

1 Contributor | 0 Issues | 3 Stars | 0 Forks





Hatena Blog

本読んだやつ 読み物系

UNIX という考え方（The UNIX Philosophy）を読んだ

魂の生命の領域

What is the UNIX Philosophy?

- Most common today:

  In simplicity there is power.

What is the UNIX Philosophy?

- Most common today:

  In simplicity there is power.

  ```
  curl –L git.io/unix
  ```

What is the UNIX Philosophy?

- Most common today:

  In simplicity there is power.

  ```
  curl –L git.io/unix
  ```

- Changed over time, because the virtues of UNIX emerged as it grew:
  - Simple, portable, and programmer-focused
  - Focus on networking
  - Collaborative development

Early UNIX Philosophy

- First version by Ken Thompson in 1973:
  1. Write ***programs that do one thing*** and do it well
  2. Write ***programs to work together***
  3. Write programs that handle ***text streams***, because that is a universal interface

UNIX Features by 1980

- Ritchie and Thompson list the following:
  1. A hierarchical file system incorporating demountable volumes
  2. Compatible file, device, and inter-process I/O
  3. The ability to initiate asynchronous processes
  4. System command language selectable on a per-user basis
  5. Over 100 subsystems including a dozen languages
  6. High degree of portability

1980s UNIX Philosophy

- As defined by Brian Kernighan:
  1. Everything is a **file**
  2. Small, single-purpose programs (**modularity**)
  3. Ability to chain programs together to perform complex tasks (**piping**)
  4. Avoid captive user interfaces (most UNIX programs are **non-interactive**)
  5. Configuration data stored in **text**

1980s UNIX Philosophy

- David Tilbrook on UNIX:

"The one thing that has to be stated about UNIX is that it wasn't a great advance in computing; if anything, it was a great ***simplification****. It put into the realm of the user those things that were just inconceivable prior to that."*

1980s UNIX Philosophy

- Sun expanded the UNIX philosophy to the ***network*** (John Gage, 1984)
  - The network is the computer (NFS)
  - Store things once on the network/Internet
  - Use the network for collaboration

Sun microsystems

~~The Network is the Computer~~

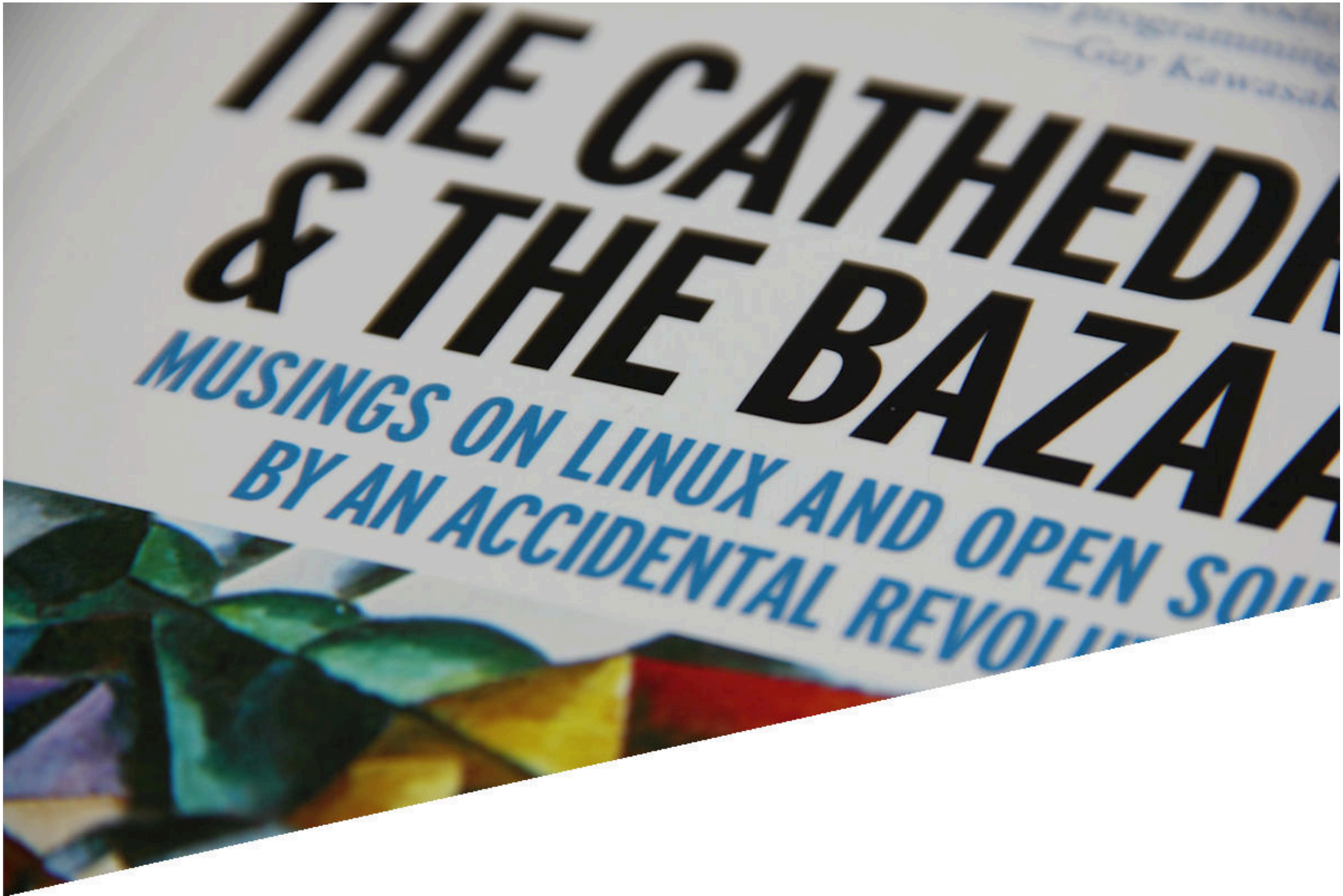
~~The Dot in Dotcom~~

The Internet is the Server!

UNIX (late 1980s - early 1990s)

- When AT&T bought a chunk of Sun in 1987 and announced that Sun OS 5 would be SysV, it shook up the UNIX world:
  - DEC, IBM, etc. formed **OSF** and OSF/1
  - Sun/AT&T formed **UNIX International** group
- Then came **Linux** (open source UNIX) and the **386BSD** legal battle (led to Free/NetBSD)
  - UNIX’s future = open source?
  - Linux is UNIX’s future?

1990s UNIX Philosophy



1990s UNIX Philosophy

- Eric S. Raymond (ESR) expanded it to reflect the focus on open source development:
  1. ***Modularity:*** Write simple parts connected by clean interfaces (no complex, unreadable code)
  2. ***Clarity:*** Clarity is better than cleverness (others must work on the code)
  3. ***Composition:*** Design smaller programs that can be connected with other programs (avoid complex monolithic programs)

1990s UNIX Philosophy

4. ***Separation:*** Let policies be changed without destabilizing mechanisms (consequently reducing the number of bugs)
5. ***Simplicity:*** Design for simplicity; add complexity only when you must
6. ***Transparency:*** Design for visibility to make inspection and debugging easier
7. ***Failure:*** When you must fail, fail noisily and ASAP

1990s UNIX Philosophy

8. ***Diversity:*** Distrust all claims for “one true way”
9. ***Extensibility:*** Design for the future, because it will be here sooner than you think
10. ***Parsimony:*** Write small, easily-replaceable code that can be thrown away if needed. Don't be afraid to throw away large chunks of code if it sucks.

Early 2000s

Early 2000s

- UNIX philosophy was widely regarded as an **outdated guideline** (although ESR’s points were key to projects)
- This was due to perception at the time:
  - It doesn't scale well
  - Monolithic can be good for some things
  - Security and UI needs don’t fit well into it
  - Microsoft & FUD contributed to this!

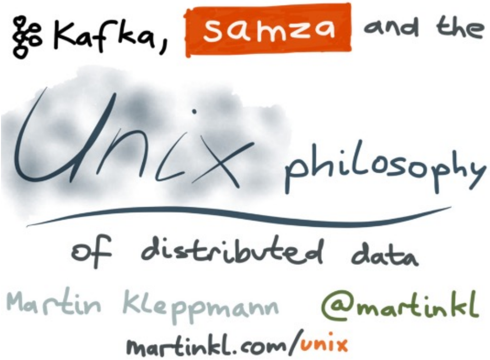
By the Mid-2010s

- Linux and open source rules the world/cloud
- Open source & UNIX philosophy widely regarded as **compulsory** (education too!)
  - Building small, focused applications collaboratively in a cloud and microservices environment
  - Network (HTTP) APIs, YAML/JSON config, XML, etc.

Revisiting the Unix philosophy in 2018 (Red Hat blog, Michael Hausenblas)

Where are we today?

- UNIX philosophy is widely known and 'quoted'
- Focus is on smaller, simpler, reusable components
- Text is still king
- Everything is stored on the network (cloud)
- Open source rules the world (many licenses, formal processes/norms)



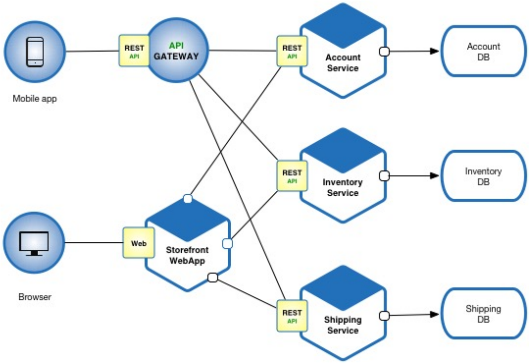
Kafka, samza and the Unix philosophy of distributed data

Martin Kleppmann @martinkl

martinkl.com/unix

Where are we today?

- UNIX philosophy is widely known and ‘quoted’
- Focus is on smaller, simpler, reusable components
- Text is still king
- Everything is stored on the network (cloud)
- Open source rules the world (many licenses, formal processes/norms)



Where are we today?

- UNIX philosophy is widely known and ‘quoted’
- Focus is on smaller, simpler, reusable components
- Text is still king
- Everything is stored on the network (cloud)
- Open source rules the world (many licenses, formal processes/norms)

Where are we today?

- UNIX philosophy is widely known and ‘quoted’
- Focus is on smaller, simpler, reusable components
- Text is still king
- Everything is stored on the network (cloud)
- Open source rules the world (many licenses, formal processes/norms)

Where are we today?

- UNIX philosophy is widely known and ‘quoted’
- Focus is on smaller, simpler, reusable components
- Text is still king
- Everything is stored on the network (cloud)
- Open source rules the world (many licenses, formal processes/norms)

Where are we today?

- LPI did a more modern take on how pervasive open source development has become – search YouTube for:

The future's hiring Linux Professional Institute

Where are we today?

- Today’s UNIX philosophy hasn’t hasn’t strayed too far from Ken Thompson’s original definition in 1973:
  - We write programs that do one thing and do it well
  - We write programs to work together
  - And we write programs that handle text streams, because that is a universal interface
- We’ve since emphasized simplicity and added the network and open source dimensions only