

Unix: explained

markus schnalke <meillo@marmaro.de>

What is Unix?

Unix

- Operating system
 - Kernel (Systemcalls)
 - Userland
 - Filesystem
- Philosophy

Background and History

Historical background

The late 60s and early 70s

Operating systems are complex

Brooks's "The Mythical Man-Month"

A lot of incompatible hardware

Limited computing power

Textual input and output (line printers)

Timeline

1968:

- MULTICS fails

1969:

- Thompson, Ritchie, McIlroy, Ossanna still think about operating systems
- The pushing idea is a file system (mainly Thompson)
- The PDP-7

1970:

- Kernighan suggests the name UNIX
- Deal: a PDP-11 for a document preparation system

Timeline

1971:

- 1st Edition

1972:

- C

1973:

- Rewrite of Unix in C

1974:

- First publication

First paper 1974: Abstract

UNIX is a general-purpose, multi-user, interactive operating system for the Digital Equipment Corporation PDP-11/40 and 11/45 computers. It offers a number of features seldom found even in larger operating systems, including:

- (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;
- and (5) over 100 subsystems including a dozen languages.

This paper discusses the nature and implementation of the file system and of the user command interface.

CACM July 1974, vol. 17, n. 7

Timeline

1975:

- Release of 6th Edition

1976/1977:

- Lions's Book
- BSD

1979:

- Release of 7th Edition

1982/1983:

- System III, System V

Operating System

Kernel

- Very simple system call interface
- Very simple process creation concept
- Compatible file, device, inter-process I/O
- Described in detail in the Lions's Book

Userland

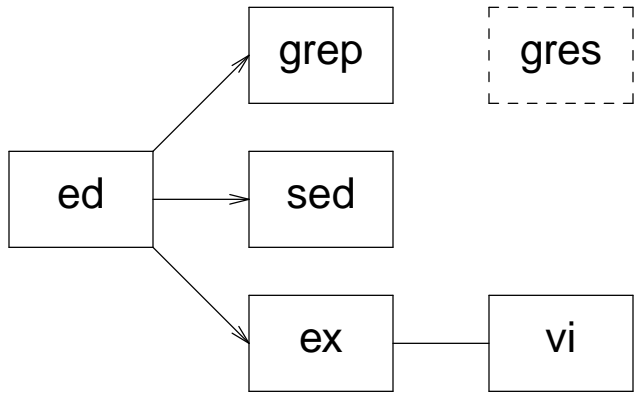
- Tools
- Pipes and filters
- Users are programmers
- No special-treatment of the shell

Command invocations in 1974:

15%:	ed
10%:	ls
6% (each):	rm, cc, cat, <i>user-commands</i>
3% (each):	who, mv, stat
2% (each):	?roff, if, db, sh, pr
1% (each):	ps, as, echo, cp, pr, date, ...

cd ? ;-)

ed



Filesystem

In the 1974 paper: “The most important role of UNIX is to provide a file system.”

- Hierarchical
- Demountable volumes
- Regular files, directories, special files
- Inodes

Philosophy

What is the Unix philosophy?

“The Unix philosophy is a set of cultural norms and philosophical approaches to developing software based on the experience of leading developers of the Unix operating system.” (wikipedia)

How the inventors of Unix write software

Common things in classic Unix tools

Difficult to define

Explanations

- Doug McIlroy (1978)
- Mike Gancarz: “The Unix Philosophy” (1994)
- Eric S. Raymond: “The Art of Unix Programming” (2003)

- Richard Gabriel: “Worse is Better” (1989)

Doug McIlroy

This is the Unix philosophy:

- Write programs that do one thing and do it well.
- Write programs to work together.
- Write programs to handle text streams, because that is a universal interface.

Mike Gancarz: “The Unix Philosophy”

- Small is beautiful.
- Make each program do one thing well.
- Build a prototype as soon as possible.
- Choose portability over efficiency.
- Store data in flat text files.
- Use software leverage to your advantage.
- Use shell scripts to increase leverage and portability.
- Avoid captive user interfaces.
- Make every program a filter.

Plus ten lesser tenets

Small is beautiful

Small software is easier to understand, write, maintain

Less lines of code contain less bugs

Monsters are large

Make each program do one thing well

Programs with many functions are large

One thing is easier to understand

Often straight forward to implement

Toolchests

Reusable

Build a prototype as soon as possible

Shows the quality of the design

Shows the problems of the software

The best way to shape a software

Users find bugs

Incremental development

Choose portability over efficiency

(Originates in a lot of incompatible hardware in history)

Use is most important

Availability

Only needs to be fast enough

Store data in flat text files

(originally: “Store numerical data in flat ASCII files”)

Binary data is machine-dependent

Human readable data is:

- As generic as possible
- Is very likely supported
- Many tools work on it (Unix toolchest)
- Directly editable by humans

Processing needs only to be fast enough

Use software leverage to your advantage

What do we have computers for?

Make best use of computing power

Reduce development effort

Toolchests and a powerful shell

Use shell scripts to increase leverage and portability

Was very important in history

High level languages

Prototyping

Quick hacks

Users are “programmers”

Avoid captive user interfaces

Don't assume the user to be human

Allow non-interactive use

Automate

How does it scale?

Bloat

Make every program a filter

Programs transform data

Combine programs

Have one common interface

Toolchests

Case studies

Case studies

- `ls`
- `cat -v`
- `readline`
- `grep -R`
- `find -printf`
- `sloccount`
- `parted`
- `uzbl, surf, dwb, ...`
- `MH, nmh, mmh`

Final thoughts

Final thoughts

- What is the value and cost of a feature?
- What is the value and cost of a thousand features?
- What is the value and cost of bad features?

- How do we measure complexity?
- How do we strive for elegance?

- What can we learn from Unix?

References

Literature

- Kernighan and Pike: “The Unix Programming Environment” 1984.
- Doug McIlroy: “The Unix and the Echo” in The UNIX Programming Environment, p. 78-79.
- Pike and Kernighan: “Program design in the UNIX environment” (aka. “cat -v Considered Harmful”), 1983.
<http://harmful.cat-v.org/cat-v/>
- Kernighan and Pike: “The Practice of Programming”, 1999.
- Mike Gancarz: “The Unix Philosophy”, 1994.
- Eric R. Raymond: “The Art of Unix Programming”, 2003.
- P.J. Plauger: “Signal and Noise in Programming Languages”, Proc. of ACM’75, p. 216.

Literature

- Ritchie and Thompson: “The UNIX Time-Sharing System”, CACM, 1974, vol. 17, no. 7, p. 365-375.
- John Lions: “Commentary on the Six Edition UNIX Operating System”, 1977.
- Steven R. Bourne: “The UNIX System”, 1983.
- (Maurice J. Bach: “The Design of the UNIX Operating System”, 1986.)
- Fred Brooks: “The Mythical Man-Month” (1975) and “No Silver Bullet” (1986).

This talk was prepared using tools of the Heirloom project:

<http://heirloom.sf.net>

The slides macros are based on

<http://repo.cat-v.org/troff-slider/>

The slides are available on my website

<http://marmaro.de/docs/>

2013-01-20 at upLUG