

A brief history of Unix

Emanuele Rocca

Berlin Unix Appreciation Society

Berlin Unix Appreciation Society!

whoami

ema

- ▶ Debian Developer since 2003
- ▶ Member of the Italian Linux Society
- ▶ Lead DevOps Engineer at ZenMate

Outline

- ▶ 1969 - 1974
- ▶ 1974 - 1980
- ▶ No BSD, TCP/IP, UNIX wars, FOSS
- ▶ Love, Ken

1969 - 1974



Multics

- ▶ Multiplexed Information and Computing Service
- ▶ Early time-sharing operating system
- ▶ Enabled communal computing

The decline and fall of Multics

- ▶ Large project with thousands of pages of technical specifications
- ▶ Bell Labs pulled out of the project
- ▶ K. Thompson, D. Ritchie, M. D. McIlroy, J. F. Ossanna were the last Bell Labs researchers actually working on Multics
- ▶ They started looking for alternatives

What they wanted to preserve

- ▶ Not just a good environment in which to do programming
- ▶ A system around which a fellowship could form

The essence of communal computing is not just to type programs into a terminal instead of a keypunch, but to encourage close communication

What they wanted to preserve

A system on which to play Space Travel

No hardware

- ▶ Throughout 1969, Thompson and Ritchie lobbied for the purchase of a machine for which they promised to write an operating system

We were asking the Labs to spend too much money on too few people with too vague a plan

- ▶ In 1956 the Dwight D. Eisenhower Administration ends a seven-year antitrust suit against AT&T: they have to **stay out of the computer business**

Initial design

- ▶ Designed a file system on blackboards and scribbled notes
- ▶ Later to become the heart of Unix

PDP-7

- ▶ They found a little-used PDP-7 computer
- ▶ Rewrote Space Travel
- ▶ Introduction to the clumsy technology of preparing programs for the PDP-7

Hello world!

- ▶ Thompson implemented the *paper* file system
- ▶ Notion of processes
- ▶ Small set of user-level utilities, including a shell
- ▶ All the programs were written using another system
- ▶ Files were transferred to the PDP-7 on paper tape
- ▶ Once an assembler was completed the system was able to support itself!
- ▶ Unix was born (but not baptised yet)

The first running Unix code

- ▶ Lightweight development
- ▶ Informal methods
- ▶ Brainstormed by three people
- ▶ Implemented by Ken Thompson in two days

IO Redirection

- ▶ `ls >destfile`
- ▶ Appeared quite early
- ▶ Inspired by an idea from Multics (like much else in Unix)

IO Redirection on Multics

```
iocall attach user_output file destfile  
list  
iocall attach user_output syn user_i/o
```


IO Redirection on Multics

- ▶ Very clumsy
- ▶ Used often during the Multics days
- ▶ Straightforward to integrate into the Multics shell
- ▶ Nobody thought of doing that

IO Redirection Multics vs. Unix

- ▶ The Multics project was just too big!
- ▶ Bell Labs implemented the IO system
- ▶ MIT implemented the shell
- ▶ Bell Labs did not consider making changes to the shell: it was *their* program
- ▶ MIT may not even have known of the usefulness of iocall
- ▶ On Unix, when the right idea finally surfaced, it was a matter of an hour or so to implement it

Poor PDP-7

- ▶ the PDP-7 was obsolete
- ▶ it sat in a corner enumerating all the closed Knight's tours on a 6x8 chess board. That was a three-month job
- ▶ A new system was necessary, together with a good excuse for buying it

Let's buy a PDP-11 and do text processing

- ▶ In early 1970 the acquisition of a PDP-11 was proposed
- ▶ The proposal this time did not mention writing an operating system
- ▶ A system for editing and formatting text

The PDP-11



Figure 1: Ken (seated) and Dennis at a PDP-11. They can now enjoy 24K of memory.

roff on the PDP-11

- ▶ Transliterated the roff text formatter into PDP-11 assembler language
- ▶ Offered to the Patent department as a text-processing server, but they were already considering a commercial system
- ▶ Advantage 1: it could print most of the math symbols they required
- ▶ Advantage 2: roff was quickly hacked to produce line-numbered pages, which the commercial system could not handle

Unix at work!

- ▶ The experiment was successful
- ▶ The Patent department adopted Unix
- ▶ They become the first of many groups at Bell Labs to ratify work on Unix

1971: The Unix Programmer's Manual

- ▶ Published on November 3, 1971
- ▶ During the system's first two years one literally had to work beside the originators to learn it
- ▶ Authorship was attributed to individuals to know who to query in case of issues
- ▶ From time to time, a snapshot was taken as a new printed edition
- ▶ Available online on Ritchie's homepage:
<http://cm.bell-labs.com/cm/cs/who/dmr/1stEdman.html>

1971: The Unix Programmer's Manual

OSSANNA

UNIX PROGRAMMER'S MANUAL

K. Thompson

D. M. Ritchie

November 3, 1971

Figure 2: Printed version belonging to Joe Ossanna

cat(1)

11/3/71

CAT (I)

NAME `cat -- concatenate and print`

SYNOPSIS `cat file1 ...`

DESCRIPTION `cat` reads each file in sequence and writes it on the standard output stream. Thus:

```
cat file
```

is about the easiest way to print a file. Also:

```
cat file1 file2 >file3
```

is about the easiest way to concatenate files.

If no input file is given `cat` reads from the standard input file.

FILES

SEE ALSO `pr`, `cp`

DIAGNOSTICS none; if a file cannot be found it is ignored.

BUGS

OWNER `ken`, `dmr`

1972: pipes

- ▶ Appeared at the insistence of M. D. McIlroy
- ▶ Explained one afternoon on a blackboard

The one-input one-output model of command execution seemed too confining. What a failure of imagination!

- ▶ Initial syntax was re-using the same character as I/O redirection: `sort input >pr>opr>`

pipes

- ▶ Enthusiastically received
- ▶ The term “filter” was soon coined
- ▶ Many commands were changed to make them usable in pipelines
- ▶ No one had imagined that anyone would want the sort utility sort its standard input if given no explicit arguments
- ▶ The present notation was soon introduced to avoid ambiguity

pipes

- ▶ Other systems also had similar concepts
- ▶ They required software to be coded in specific ways

The genius of the Unix pipeline is precisely that it is constructed from the very same commands used constantly in simplex fashion. The mental leap needed to see this possibility and to invent the notation is large indeed.

High-level languages

- ▶ Every Unix program running on the PDP-7 was written in assembly
- ▶ Thompson decided they needed Fortran to offer a real computing service
- ▶ He sat down to write a Fortran
- ▶ The intent to handle Fortran lasted about a week

B

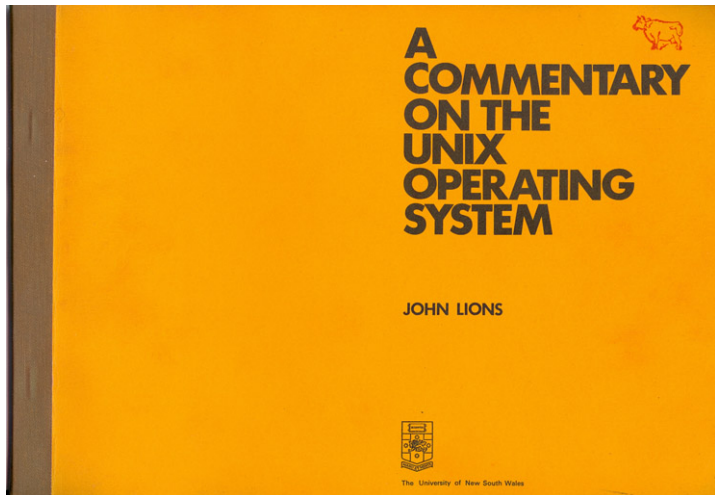
- ▶ He produced the definition of and a compiler for a new language called B
- ▶ BCPL stripped of any component they could do without
- ▶ Reasonable language to write “systems programs”: compilers, assemblers, and the like
- ▶ B version of the multi-precision “desk calculator” program **dc** was one of the earliest programs to run on the PDP-11

C

- ▶ B was slow
- ▶ In 1971, work began on what was to become the C language
- ▶ In 1973 the operating system kernel was rewritten in C (WAT?)
- ▶ The internal structure of the system became much more rational and general

Much of the success of Unix follows from the readability, modifiability, and portability of its software that in turn follows from its expression in high-level languages

1974 - 1980



The UNIX Time-Sharing System, 1974

Dennis M. Ritchie, Ken Thompson, *The UNIX time-sharing system*,
Communications of the ACM, v.17 n.7, p.365-375, July 1974

- ▶ Unix gets public exposure for the first time
- ▶ The paper describes Unix unprecedentedly simple design
- ▶ Research labs and universities all over the world want to try it out

Love, Ken

- ▶ Unix could not be turned into a product (antitrust case)
- ▶ Bell Labs was required to license its nontelephone technology to anyone who asked
- ▶ Ken Thompson quietly began answering requests by shipping out tapes and disk packs
- ▶ According to legend, each package had a note signed “love, ken”

Licenses

- ▶ AT&T licensed Version 5 (1974) to educational institutions
- ▶ Version 6 (1975) was the first one to be sold to commercial firms
- ▶ \$20000 for companies, \$200 for educational institutions
- ▶ The system was offered to all users under very restrictive terms
 - ▶ source only
 - ▶ no support
 - ▶ no refunds
 - ▶ no warranty
 - ▶ no maintenance

Contributions

- ▶ Yale: students modified the shell
- ▶ Purdue: performance, support for more users
- ▶ Berkeley: new shell, dozen of utilities

Flipping a finger at the system

- ▶ The pioneering programmers were shaggy hippies
- ▶ Use of minicomputers was less regulated than the mainframes
- ▶ Unix development rapidly took on a countercultural air
- ▶ Unix subverted all the technical assumptions and business practices that went with Big Computing
- ▶ They were simultaneously building the future and flipping a finger at the system

The Lions book

- ▶ 6th Edition allowed classroom use of the source code
- ▶ Lions' Commentary on UNIX 6th Edition, with Source Code by John Lions (1976)
- ▶ 7th Edition specifically excluded classroom use
- ▶ Thousands of computer science students around the world spread photocopies
- ▶ Many pioneers of Unix had a treasured multiple-generation photocopy

Minix

- ▶ When AT&T released Version 7, it began to realize that UNIX was a valuable commercial product
- ▶ Andrew S. Tanenbaum created Minix

Beginning of an industry

- ▶ Santa Cruz Operation, SCO, founded in 1978
- ▶ By 1980 an obscure software company in Seattle was also getting into the Unix game, shipping a port of the AT&T version for microcomputers called XENIX

OCTOBER 20, 1982

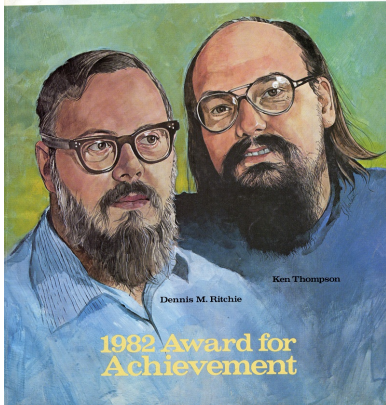
ANNUAL TECHNOLOGY UPDATE ISSUE

2- μ m processes, second-generation optical-projection and wafer-stepping systems,
and new computer-aided design tools usher in the 256-K dynamic RAM era / 116

88¢ DOLLARS A McGRAW-HILL PUBLICATION

International

Electronics



Dennis M. Ritchie

Ken Thompson

1982 Award for
Achievement

Lessons

- ▶ Limitations can be beneficial
- ▶ Third-system effect
- ▶ Communal computing

more

- ▶ Ritchie, Dennis M. *The Evolution of the Unix Time-Sharing System*, 1979.
- ▶ <http://cm.bell-labs.com/cm/cs/who/dmr/1stEdman.html>
- ▶ McIlroy, Douglas M. *A Research Unix reader: Annotated Excerpts from the Programmer's Manual*, 1986.
- ▶ Salus, Peter H. *A quarter century of Unix*. Reading, Mass.: Addison-Wesley, 1994.
- ▶ Raymond, Eric S. *The art of Unix programming*, Addison-Wesley Professional, 2003.