Linux Evolution

D. P. Bovet   M. Cesati

System Programming Research Group
DISP, University of Rome “Tor Vergata”

Rome, 26 March 2008

Once upon a time…

- August 1991: first public mention of a new OS for Intel 80386 PCs

```
From: torvalds@klaava.Helsinki.FI
                   (Linus Benedict Torvalds)
Newsgroups: comp.os.minix
Subject: What would you like to see most in minix?
Summary: small poll for my new operating system
Date: 25 Aug 91 20:57:08 GMT
Organization: University of Helsinki

Hello everybody out there using minix -

I'm doing a (free) operating system (just a hobby,
won't be big and professional like gnu) for 386(486)
AT clones.  This has been brewing since april, and
is starting to get ready.  I'd like any feedback on
things people like/dislike in minix, as my OS
resembles it somewhat [...]
```

Once upon a time…

- Linus Torvalds is an undergraduate student in computer science, with no serious experience in programming

- The OS project is initially focused on Intel 80386 PC's

- When asked about the portability of his project, Linus replied:

  > *Simply, I'd say that porting is impossible. It's mostly in C, but most people wouldn't call what I write C. It uses every conceivable feature of the 386 I could find, as it was also a project to teach me about the 386. As already mentioned, it uses a MMU, for both paging (not to disk yet) and segmentation. It's the segmentation that makes it REALLY 386 dependent* *[Linus Torvalds, comp.os.minix, 25 Aug 1991]*

- At the time, there was no plan for Linux's *world domination*!

Initial releases

Date	Version	Notes
Sep. 1991	0.01	Announced in comp.os.minix
Oct. 1991	0.02	“Do you pine for the nice days of minix 1.1, when men were men and wrote their own device drivers?”
Oct. 1991	0.03	“Pretty useable”
Dec. 1991	0.10	ISA+AT disks, no login, no SCSI. . .
Dec. 1991	0.11	VGA, EGA, floppy disk, multilang. keyboards
Dec. 1991	0.12	Swapping to disk, switch to GPL license
Mar. 1992	0.95	Symbolic links
Jul. 1992	0.96	Named pipes
. . .	. . .	(several tens of versions)
Mar. 1994	1.0.0	First “production” version

Fast forward to 2008

As a matter of fact, Linux has changed a bit since early times:

- Version 0.01: 76 files, 512 KB on disk, $<$ 8500 lines of code
- Version 2.6.24: $\sim$ 20 000 files, 250 MB on disk, 8 000 000 LOC

The crucial questions

- What have been the most evident progresses in the kernel?
- Who share the merits for the tremendous Linux evolution? Is it just a personal achievement of a prominent programmer like Linus Torvalds?
- How did it happen that a pet OS project of a second-year computer science student has become a successful operating system competing with those of the major IT companies?

Linux progresses: lots of versions

The management of the Linux versions has changed over the years

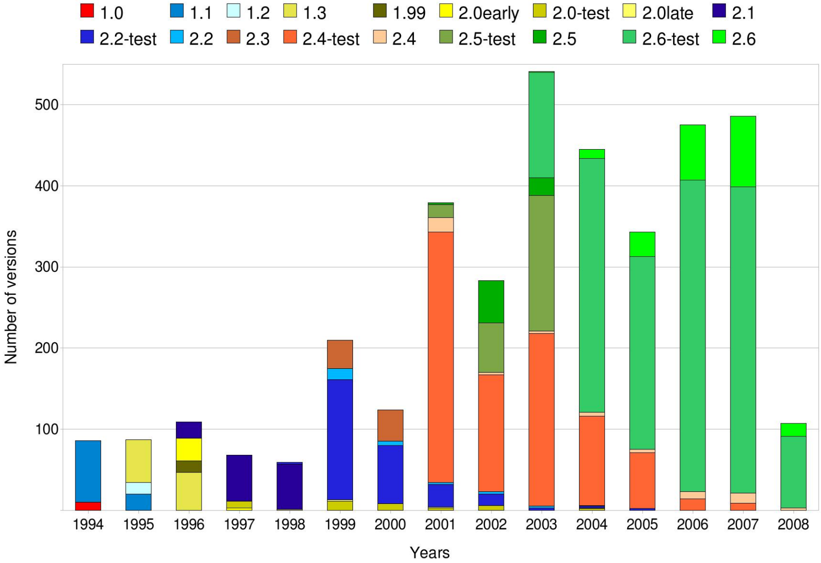
1994–2005

- *stable* versions: series 1.0, 1.2, 2.0, 2.2, 2.4, early 2.6
- *development* versions: 1.1, 1.3, 1.99, 2.1, 2.3, 2.5, . . .

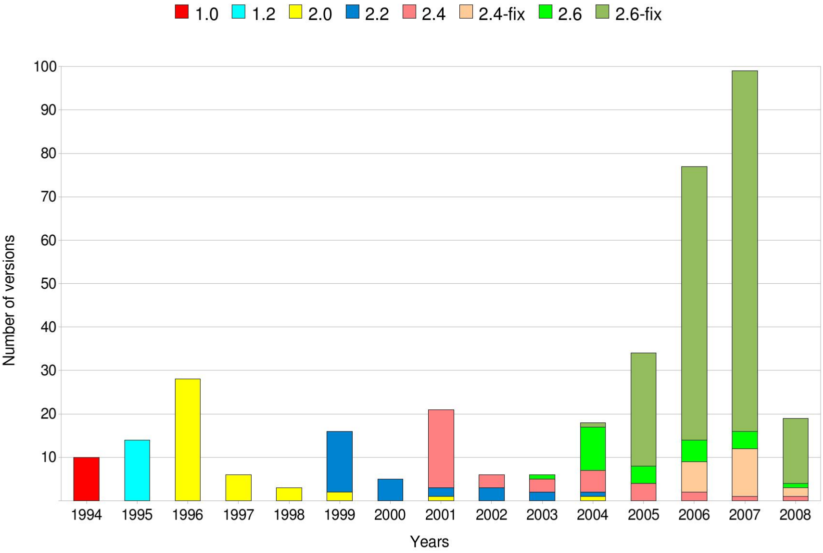
2005–2008

- *stable* versions: late 2.6
- *development* versions: 2.6-mm, 2.6-git, 2.6-rcX, . . .

Linux progresses: all versions



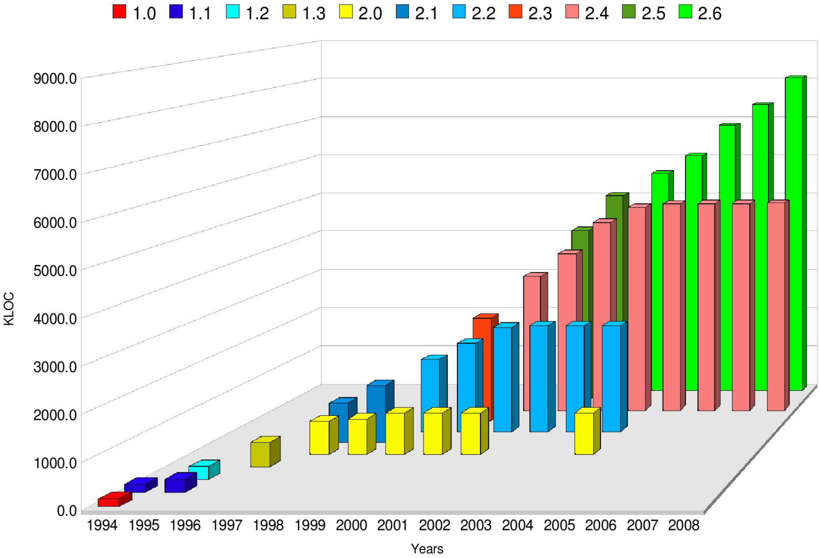
Linux progresses: stable versions



Linux progresses: the size of the source code

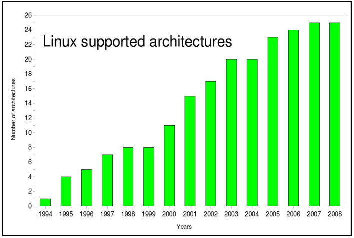
- Admittedly, the size of the source code, by itself, tells us nothing about the features of the kernel
- However, the size of the source code is related to:
  - the design complexity of the kernel
  - the effort in handling the project
- Any one of the following metrics gives equivalent results for the Linux kernel:
  - number of lines of code
  - number of functions
  - number of source files
  - size of the source code on disk

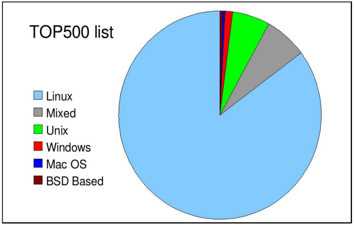
Linux progresses: number of lines of code



Linux progresses: supported architectures

- The most striking progress of Linux is perhaps the number of hardware architectures on which it runs
- It is one of the most portable operating systems (the only strong competitor is NetBSD)
- Bottom line: lots of embedded systems, like cellular phones and wireless access points
- Top line: mainframes and supercomputers: 426 out of the 500 most powerful supercomputers in the world runs (only) Linux





Linux progresses: supported hardware devices

- This is one of the most frequent criticisms to Linux: “many hardware devices have no support!”
- As a matter of fact, Linux is the OS with the broader range of supported hardware devices: each existing driver is architecture-independent and supported “forever”
- Problems exist only for devices that come without specifications *and* a vendor-provided open-source driver (anyway lately vendors are showing a positive attitude towards the Linux community)
- Most commercial operating systems integrate drivers only for devices that are widely adopted. In the other cases, the vendor usually provides a driver only for the version of the most adopted OS that is “current” at the time the product has been marketed
- The 64-bit revolution is coming: almost all drivers in Linux are 64-bit ready. What about other commercial OS?

The Linux keys to success

Is Linux “here to stay” beyond any reasonable doubt?

Yes! Because Linux has *not* been designed for that goal!

The top reasons for Linux success

1. Give me back your code: the GPL license
2. Quickly changing code base: flexibility
3. Delegating responsabilities: smart project management
4. Personalities of kernel hackers: professional programmers who love their craft
5. Contributions from the industry: essential support from lots of IT companies

All these reasons are strongly related to each other

Linux license: the GPL

- The success of Linux is rooted in the Richard Stallman's project GNU (*GNU's not UNIX*) for a free (as in speach) operating system
- The GNU project provided many essential tools for Linux, for instance the GCC compiler suite, the C standard library, and the system command tools
- The GNU GPL license is a key ingredient for Linux success

> ### The GPL in a nutshell
>
> Anyone can freely read and modify the source code of the kernel provided that, if the new product is distributed, the modified source code is made available to the final users and to the kernel developers

- However, Linux cannot be simply confused with the GNU project, as its success has also other important reasons

Linux changing rate

The kernel code is changing at an amazing rate

Only in 2007, the Linux code base has been modified as follows:

- 30 000 changesets applied
- 2 000 000 lines of code modified
- 750 000 lines of code added

(source: lwn.org)

Statistically, in every day of 2007 the kernel developers added over 2 000 lines of code and touched over 5 000 lines of code!

So. . . how many kernel developers are there?

How many kernel developers

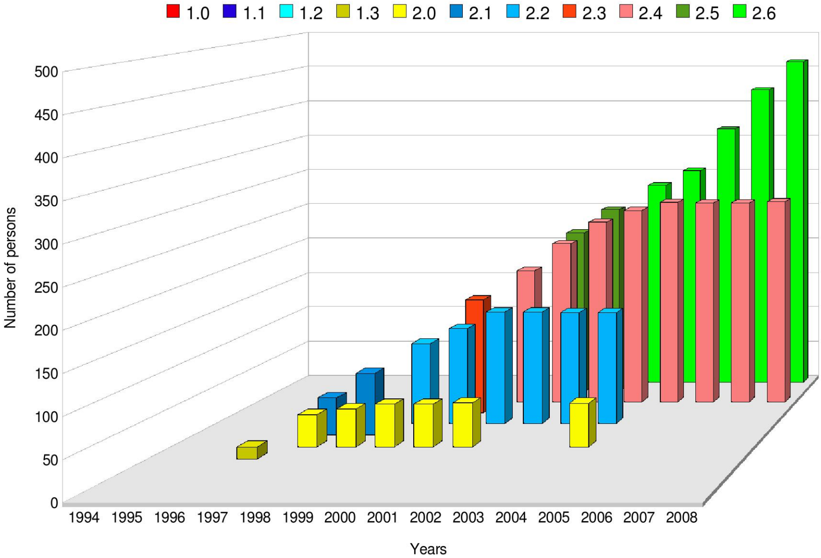
- The CREDITS file in the source code includes nowadays about 500 names, but it is largely incomplete
- The kernel changes made in year 2007 have been submitted by not less than 1 900 developers—but they may be a lot more because developers working in teams usually count as one
- It is generally assumed that the community of Linux developers is composed by 5 000 or 6 000 members
- In any case, the number of developers is huge:
  how can Linus Torvalds cope with all of them?

> “Open-source development violates almost all known management theories”
>
> *[Dr. Marietta Baba, Dean of the Dept. of Social Science, Michigan State University]*

Subsystem maintainers

- The kernel source code is highly modular: core components, bus drivers, network stacks, and device drivers are logically separated and have well-defined interfaces
- Linus Torvalds has appointed a number of *maintainers*: each of them is a kernel hacker who is in charge of a specific component or subsystem of the kernel
- Each maintainer loosely coordinates a number of kernel hackers working on his own component or subsystem
- In any case, changes affecting a given subsystem are only accepted by Linus if they come from the proper maintainer
- The *main maintainer*, Andrew Morton, integrates the most innovative changes in the 2.6-mm series of kernels. These kernels are downloaded and executed by about 2 000 users, so as to catch most bugs and regressions

How many maintainers



Who are the kernel developers

- Linus Torvalds and Andrew Morton do not analyze in depth each and every change of the kernel, but only the most critical ones (10%–20%)

- Linux's robustness thus largely depends on the quality of the work of maintainers and developers

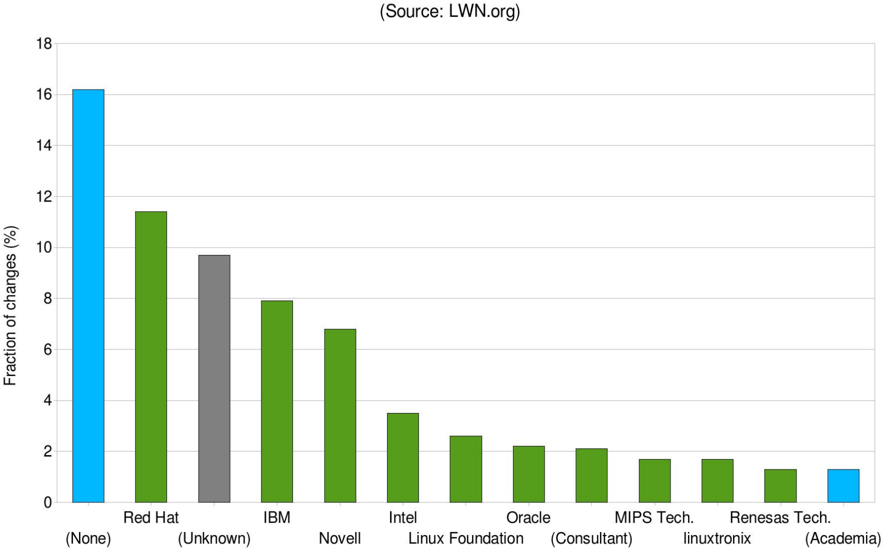
> "[. . .] it is strange: people who I believe are some of the best programmers in the world choose to work on free software" *[Andrew Morton, 2005]*

> "Even though we're all paid to do kernel work, the culture is largely unchanged from the earliest days of Linux. We work on it because we love our craft and because it allows us to contribute to our society in the way in which we are most able. Corny but true." *[Andrew Morton, 2005]*

Linux and the industry

- In the early days, Linux was in the hands of a community of computer geeks spending lots of spare time writing code
- This is not true any longer: nowadays many large and small IT companies support Linux by hiring top-level programmers and let them to work full-time on the kernel
- Why? Top reasons:
  - Supporting its own business core (e.g., Red Hat or Novell)
  - Ensuring that their hardware products work well with Linux
  - Customizing and specializing the kernel for its own internal or external requirements
- Each company has strong interest in getting its contributions merged into the official code base, so that its code will evolve together with the rest of the kernel

Contributions to Linux 2.6 in 2007



Who design Linux

- Changes in the Linux kernel are mostly driven by IT companies
- Thus, the main role of Linus Torvalds and Andrew Morton is planning the future evolution of Linux, right? Not at all!

"Linux is evolution, not intelligent design!" [Linus Torvalds, 2005]

"People like Linus Torvalds and I don't plan the kernel evolution. We don't sit there and think up the roadmap for the next two years, then assign resources to the various new features. That's because we don't have any resources. The resources are all owned by the various corporations who use and contribute to Linux, as well as by the various independent contributors out there. It's those people who own the resources who decide ..." [Andrew Morton, 2005]

Linux evolution

- The top reason for the success of Linux is that it is not driven by someone who has a technical, commercial, or political agenda.
- Rather, Linux is driven by the requirements of the real world entities that want to adopt it: IT companies and final users
- Linus Torvalds and other top-level developers are thus similar to referees, who ensure that each change in the kernel is technically sound and, even more important, potentially beneficial to the whole Linux community
- Thus, it is not really surprising that Linus Torvalds has a full-time job in the Linux Foundation, which is a nonprofit consortium supported by many large IT companies and dedicated to fostering the growth of Linux

Conclusions

- Nowadays Linux is no longer a pet project for young, brilliant computer geeks…
- But Linux kernel hacking is still a fascinating art, as in the early days
- Most of the current work on the kernel is done by professional programmers…
- But many of them were individual enthusiasts who spent their spare time hacking the kernel, and who were later hired by large companies to work full-time on Linux
- Linux is evolving right now, and it is here to stay!