Open Source
The Brave New World

Petr Baudiš ⟨pasky@ucw.cz⟩

Stellenboch 2011

Outline

The Brave New World. . .

Open Source

- Open source code
- Rights to modify and redistribute the code
- (Obligation to allow others do the same)

Why? + Contents

- You can improve the world!
- You can improve yourself!
- Or at least figure out why it doesn't work.

Why? + Contents

- You can improve the world!
- You can improve yourself!
- Or at least figure out why it doesn't work.

- How did it all happen and what's next
- How to survive in the open source community

Why? + Contents

- You can improve the world!
- You can improve yourself!
- Or at least figure out why it doesn't work.

- How did it all happen and what's next
- How to survive in the open source community

- Fork me on Github! `https://github.com/pasky/oss-lec`

Outline

History

- Antikythera (150–100 B.C.) — ancient hi-tech clock, astronomically precise Moon movement, documentation on the case
- UNIX (1970s A.D.) — distributed on tapes, of course including the source code
- Closed source software — on the rise as the software is decoupled from the hardware
- 386BSD again opened (rewritten) UNIX code, but GNU and Linux appears in the meantime

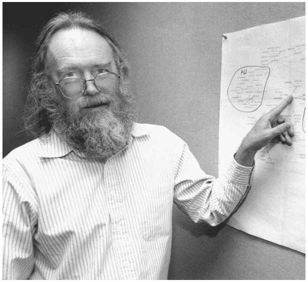
The Internet

- Communication (USENET, e-mail, IRC) allows world-wide cooperation of programmers (similar to the Wikipedia effect 30 years later)
- The *hacker culture* emerges — access to the source code is an important attribute

The Internet

- Communication (USENET, e-mail, IRC) allows world-wide cooperation of programmers (similar to the Wikipedia effect 30 years later)
- The *hacker culture* emerges — access to the source code is an important attribute

- The internet (based on ARPAnet) is completely open system
- Protocol specs are published as *Requests for Comment*, open standards process
- Jon Postel: “Be conservative in what you send, liberal in what you accept.”



The GNU and Free Software Foundation

- Richard M. Stallman (MIT AI labs): Can't even tweak the firmware for my printer?
- Founds GNU in 1983, FSF in 1985
- *Free software* that anyone can modify if he retains this right for others too ("copyleft"). Unlimited usage and commerce.
- *General Public Licence* (GPL), *Lesser GPL*, *GFDL*.
- GNU: Basic tools, text editor, compiler, now also an image editor etc.
- Kernel by Linus Torvalds ⇒ GNU/Linux (but. . .)

The GNU and Free Software Foundation

- Richard M. Stallman (MIT AI labs): Can't even tweak the firmware for my printer?
- Founds GNU in 1983, FSF in 1985
- *Free software* that anyone can modify if he retains this right for others too ("copyleft"). Unlimited usage and commerce.
- *General Public Licence* (GPL), *Lesser GPL*, *GFDL*.
- GNU: Basic tools, text editor, compiler, now also an image editor etc.
- Kernel by Linus Torvalds ⇒ GNU/Linux (but. . .)
- Non-free software can be even immoral — **political and social agenda.**

Linus Torvalds, 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 (same physical layout of the file-system
(due to practical reasons) among other things).
I've currently ported bash(1.08) and gcc(1.40), and things seem
to work. This implies that I'll get something practical within a
few months, and I'd like to know what features most people would
want. Any suggestions are welcome, but I won't promise I'll
implement them :-)
Linus (torvalds@kruuna.helsinki.fi)
PS. Yes - it's free of any minix code, and it has a
multi-threaded fs. It is NOT portable (uses 386 task switching
etc), and it probably never will support anything other than
AT-harddisks, as that's all I have :-(.
```

Early Linux

- *Sadly, a kernel by itself gets you nowhere. To get a working system you need a shell, compilers, a library etc. . . . Most of the tools used with linux are GNU software and are under the GNU copyleft.*
- The Tanenbaum–Torvalds debate:
    - A: *. . . designing a monolithic kernel in 1991 is a fundamental error. Be thankful you are not my student. You would not get a high grade for such a design :-)*
    - L: *Your job is being a professor and researcher: That's one hell of a good excuse for some of the brain-damages of minix.*
    - A: *I think it is a gross error to design an OS for any specific architecture, since that is not going to be around all that long.*
    - L: *An acceptable trade-off, and one that made linux possible in the first place.*

Open Source Initiative

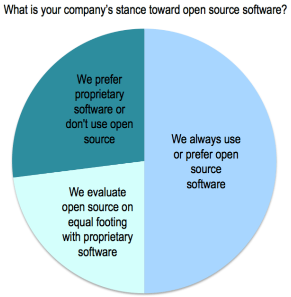
- Free software reduces individual freedom “for the good of the society” — the modified version must be licenced the same way
- Alternative — BSD / MIT / X11 etc. licences; short and sweet, do anything you want with the sources
- Open source encompasses these licences as well
- Open Source Initiative (Bruce Perens, Eric S. Raymond) — let’s stop moralizing and be pragmatic!

Creative Commons

- Software licences don't fit other content well — or other blueprints
- Easy to understand, few variants for content creators:
  - BY (attribution)
  - NC (non-commercial)
  - SA (share alike; copyleft again!)
  - ND (no derivative works)
- *Free culture* — pictures, music, writings, other creations; **Wikipedia** is the "poster child"

The Present

- The Internet relies on OSS from a large part — both infrastructure and services
- Linux in a range of embedded devices (routers, MP3 players, Android)
- Large companies, academia, sometimes home computers
- Open software common on Windows too (Firefox, VLC, LibreOffice)
- Not just software: Project Guttenberg, Wikipedia, Thingiverse
- Software patents, controversial trademarks, web and (A)GPLv3.



Outline

Project infrastructure

- *The place where the source code lives*
- Project homepage — description, news, download, documentation, development
- Communication — mailing list or web forum, wiki, IRC
- Development — bugtracker, version control system
- Index — distributions, ~~FreshMeat~~ FreshCode (. . . , Code Search, Koders.com)

- Forges: Sourceforge / Savannah, Google Code
- VCS Hosting: Github / Gitorious, Bitbucket, Launchpad

A Patching Cookbook

- Get the sources. (Web download, `apt-get source`, ...)
- Find the right place, grok the conventions, keep the coding style. (Doxygen, `HACKING`, ...)
- Build it. (Install dependencies, development libraries, `apt-get build-dep`, ...; `./configure; make; make install`)
- Document it. Write a testcase if unit testing is used.

A Patch Submission Cookbook

- Create a patch or patch series.
  (`diff -u` or version control system)
- Send a patch to the mailing list.
  (Beware whitespace damage, line wrapping.)
- Github: Commit changes, fork, push, pull request.

A Patch Submission Cookbook

- Create a patch or patch series.
  (`diff -u` or version control system)
- Send a patch to the mailing list.
  (Beware whitespace damage, line wrapping.)
- Github: Commit changes, fork, push, pull request.

- Noone replied in a few days? Resubmit and persist.
- Respond to comments and bugreports. Ignore rudeness.
  Be prepared to make major changes in your implementation (but argue first).
- "Anyone" can comment, the maintainer(s) have the last word.
- Copyright assignment may be required.

Opensourcing a Project Cookbook

- Make sure people can easily build and run it.
- Don't postpone for code cleanup!
- Pick a licence. When in doubt, GPL or BSD.
- Write a basic `README` and homepage — both what it's *about*, why is it *special* and how to *build and use it*. Brief is fine!
- Advertise in an interest group, expect more work at the beginning.
- Review, but be liberal in what you accept.

Outline

The Hackerspaces

- The Internet enabled world-wide cooperation of programmers
- Historically, the academia or large tech companies enabled local cooperation
- Wider tech accessibility, fragmented community

The Hackerspaces

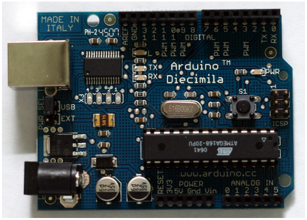
hackerspaces
build! unite! multiply! hackerspaces.org

brmlab
hackerspace prague

- The Internet enabled world-wide cooperation of programmers
- Historically, the academia or large tech companies enabled local cooperation
- Wider tech accessibility, fragmented community

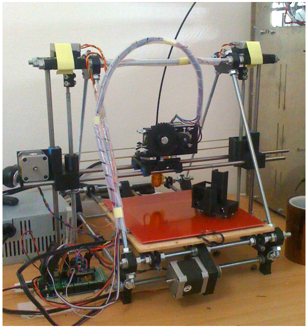
- **Hackerspace** or **makerspace**
- Independent, community-driven, hacker-run
- DIY-based, Open Source culture
- Critical mass, idea polling, base for larger projects

Open Source Hardware



- Arduino microcontroller board!
- Wearable computing, lights and home automation, robots, quadcopters (hackaday.com)
- DIY Bio: OpenPCR, simple hacks for DNA analysis, OpenEEG
- FPGA-based ICs (OpenSPARC, etc.)
- USRP and GNU Radio — hack the EM spectrum
- Global Village Construction Kit
- RepRap / Maker Bot 3D printing!

Open Source Things



- 3D printing popular lately
- Plastic 3D printing (horizontal layers, ABS or PLA)
- RepRap costs $<$ 10000 Rands, partially replicable

- thingiverse.com: Repository of *things* — download CAD file and print!
- Fun items — whistles, action figures, charms, toys
- Practical accessories like knobs, hooks, doorhandles, simple tools, glasses
- Parts for commercial equipment or DIY projects

Thank you!

Petr Baudiš ⟨pasky@ucw.cz⟩

Faculty of Mathematics and Physics
Charles University in Prague

brmlab
hackerspace prague