Contributing to the Linux Kernel

The father of all open source projects. . .

Levente Kurusa

Department of Computing
Imperial College London

February 11, 2016

About me

- Red Hat intern during 2014 and 2015 summers
- Fedora Ambassador
- Involved in Linux kernel for 4 years now (and counting. . .)
- . . . and other low-level projects

Common misbeliefs

- It is hard to contribute **FALSE**



Common misbeliefs

- It is quite difficult to debug **FALSE**

netconsole, serial outputs, GPIO, gdb stub, the list goes on...

Common misbeliefs

- It could be dangerous for my hardware **FALSE**

Unless you are writing ancient hardware, chances are small.

Common misbeliefs

- The community isn't so welcoming **TRUE**

Linus Torvalds

Mauro, SHUT THE * UP!

It's a bug alright - in the kernel. How long have you been a maintainer? And you *still* haven't learnt the first rule of kernel maintenance?
[...]
WE DO NOT BREAK USERSPACE!

A really easy way of contribution

- $ make randconfig

Example error

error: ’KVM_USER_MEM_SLOTS’ undeclared here (not in a function)

Contributor statistics

- "Many projects would consider 400 patches a major release, and here they are behind two dots."

  - *Avi Kivity (KVM inventor)*

- New 'major' version every 8 (or so...) weeks
- Used by 95% of super computers

Surely, there must be something going on to keep the maintainers sane.

Coding style

- The Linux Kernel enforces a strict policy on coding style.
- Information can be found in *Documentation/CodingStyle*
- Linus Torvalds himself is particularly picky about the style

Coding style

```
if (!dev)
```

Coding style

```
if (!dev)
  return -ENODEV;
```

Why is coding style important?

- Allows for faster (and more automatic) spotting of bugs
- "Programs are meant to be read by humans and only incidentally by computers."

  H. Abelson & G. Sussman

Eudyptula Challenge

Series of increasing-difficulty challenges designed to introduce you to the Linux Kernel.

Reward at the end!

`http://eudyptula-challenge.org/`

Building it

`$ make localmodconfig`

- Configures your kernel into your currently running configuration
- You can install it using `$ make install`

$ make -j`nproc`

- Builds the kernel

$ make clean

- Clean the build directory

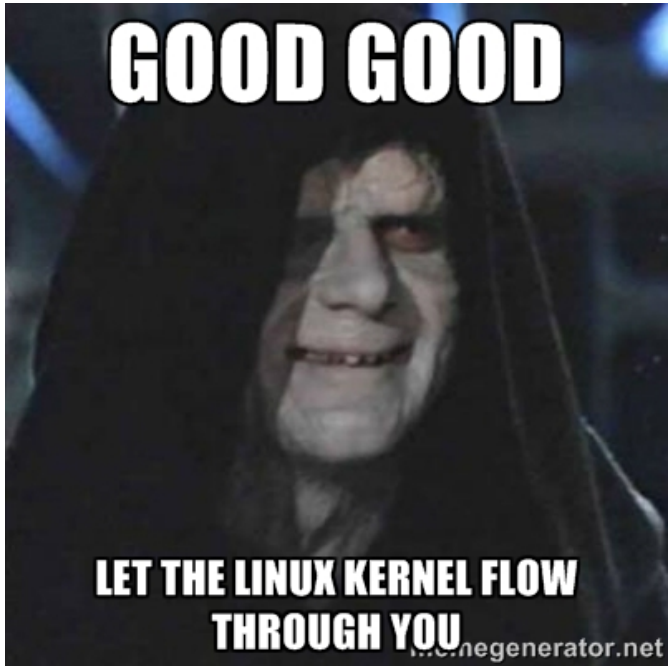
Submitting a simple patch

1. Figure out what to modify
2. Figure out a *reasonable* way to modify it
3. Apply `scripts/checkpatch.pl` and fix suggestions
4. Send it *properly*

Proper patch sending

1. $ git send-email
2. Preferred way to send patches to most git-managed Open Source projects
3. Keeps the tabs as tabs
4. Generates MBOX format

Questions?



Levente Kurusa
Email: levex@linux.com
Twitter: @ilevex