

Fehlerfinden und Qualitätssicherung im Linux-Kernel

Paul Menzel (Max-Planck-Institut für molekulare Genetik)

10. März 2018

Wer bin ich?



- ▶ Systemarchitekt beim Max-Planck-Institut für molekulare Genetik
- ▶ Diplom-Wirtschaftsmathematiker (TU Berlin)
- ▶ FLOSS-Befürworter

Präsentation

Folien in Markdown mit Pandoc nach LaTeX-Beamer umgewandelt,
verfügbar auf GitHub.

https://github.com/paulmenzel/fehlerfinden_und_qualitätssicherung_im_linux_kernel

Problem

- ▶ Hersteller/OEMs entwickeln nur für Microsoft Windows.
- ▶ Wenige Ausnahmen: Google Chromebooks und Apple
- ▶ Anderen Anbieter passen nur an
- ▶ Mehr Motivation: Vortrag *Hilf mit, damit Linux nur besser und nie schlechter wird* von Thorsten Leemhuis, Sonntag 10 Uhr
- ▶ Hohe Änderungsraten und Testlabore erst im Kommen
- ▶ <https://intel-gfx-ci.01.org/>

Arbeit mit Entwicklern

- ▶ Linux-Kernel in Distributionen meist nicht aktuell
- ▶ Linux-Kernel-Entwicklern bevorzugen Meldungen bezüglich der aktuellen Version
 - ▶ Zweig *master* in Linus' Git-Depot
 - ▶ Entsprechender Entwicklungszweig

Neuer Linux-Kernel

Pakete

- ▶ Ubuntu: <http://kernel.ubuntu.com/~kernel-ppa/mainline/>
- ▶ Fedora:
https://fedoraproject.org/wiki/Kernel_Vanilla_Repositories
- ▶ Debian: manchmal in *experimental*:
<https://packages.debian.org/linux-image-4.16>
- ▶ andere Distributionen ähnlich

Selber bauen (Debian)

```
$ git clone \
https://git.kernel.org/pub/scm/linux/kernel/git/
torvalds/linux.git
$ cd linux
$ cp -a /boot/config-4.15.0-1-amd64 .config
$ make help
$ make olddefconfig
$ make localmodconfig
$ make bindeb-pkg -j`nproc`
$ sudo dpkg -i ../linux*.deb
```


Problemstellen

Bautests, Funktionen, ACPI, Energiemanagement (Power Management), Grafik, Audio, Netzwerk, Dateisystem, Geschwindigkeit, ...

Bautests

Mehr oder weniger gelöst: nur sehr selten Fehler beim Bauen.
Danke Intel!

Geschwindigkeit

- ▶ „Pet peeve“ (vergleiche mit Google Chromebooks, Apple-, MS-Windows-Geräte)

Ziel

- ▶ Schneller Start des Linux-Kernels
- ▶ Schnelles Einschlafen und Aufwachen (ACPI S3)

[sleepgraph.py](#)/[bootgraph.py](#)

- ▶ <https://01.org/suspendresume>
- ▶ <https://github.com/01org/pm-graph>

System analysis tool source code and binary, and a blog which gives updates on actual changes being made to the kernel using the tool

Praxis

- ▶ Füge `initcall_debug` zu Linux-Kommandozeile hinzu

```
$ sudo vim /etc/default/grub
```

```
$ sudo update-grub
```

```
$ systemctl restart
```

- ▶ Installiere `systemd-bootchart` mit `sudo apt install systemd-bootchart`
- ▶ `man systemd-bootchart`
- ▶ Füge `init=/lib/systemd/systemd-bootchart` zu Linux-Kernel-Kommandozeile hinzu

systemd-bootchart

- ▶ SVG-Datei unter `/run/log/`
- ▶ Übersicht, welche Linux-Kernel-Funktionen, wie viel Zeit

sleepgraph.py/bootgraph.py

- ▶ Kopie von letzter Veröffentlichung in Linux-Kernel-Quellen
- ▶ Nutze Git-Depot direkt:

```
$ git clone https://github.com/01org/pm-graph
$ cd pm-graph
$ sudo ./sleepgraph.py -c config/suspend-callgraph.cfg
```

- ▶ Standardmäßig eigenständiges Aufwachen nach 15 Sekunden
- ▶ Tiefe über Parameter maxdepth erhöhen

Beispiel mit Dell XPS 13 9370

kselftest

- ▶ <https://www.kernel.org/doc/Documentation/kselftest.txt>
- ▶ <https://kselftest.wiki.kernel.org/>

The kernel contains a set of “self tests” under the tools/testing/selftests/ directory. These are intended to be small tests to exercise individual code paths in the kernel. Tests are intended to be run after building, installing and booting a kernel.

```
$ make kselftest
```


Einschub: Fehlerberichte

Einschub: Fehlerberichte

- ▶ LKML: Linux Kernel Mailing List
- ▶ <https://bugzilla.kernel.org/>
- ▶ Datei MAINTAINERS
- ▶ <https://01.org/linuxgraphics/documentation/how-report-bugs>

Grafik

Grafik

- ▶ Komplex, viele Ebenen
- ▶ Framebuffer, DRM, Mesa, GPU-Dekodieren/-Kodieren

Programme

- ▶ <https://cgit.freedesktop.org/drm/igt-gpu-tools/>
- ▶ <https://piglit.freedesktop.org/>

Sanitizers

UndefinedBehavior/AddressSanitizer

- ▶ `CONFIG_UBSAN`
- ▶ `CONFIG_KASAN`

Ausblick

kselftest

- ▶ Mehr Tests

Fazit

1. Jeder kann mitmachen.
2. Distributionen sollten Programme paketieren und ausliefern.
3. Nicht nur Linux-Kernel auch andere Ebenen und Anwendungen mit Tests

Fragen