

# Praktikum angewandte Systemsoftwaretechnik (PASST)

## Versionskontrollsysteme / Aufgabe 2

---

09. November 2020

Dustin Nguyen, Tobias Langer, Jonas Rabenstein, Phillip Raffeck

Lehrstuhl für Informatik 4  
Friedrich-Alexander-Universität Erlangen-Nürnberg



Lehrstuhl für Verteilte Systeme  
und Betriebssysteme



FRIEDRICH-ALEXANDER  
UNIVERSITÄT  
ERLANGEN-NÜRNBERG

TECHNISCHE FAKULTÄT

# Rückblick

---

- Virtualisierung bspw. mit **QEMU/KVM**
  - Leichteres Debugging
  - Kürzere Umlaufzeiten
- Konfigurieren und bauen des Linux-Kernels
  - Konfiguration über **Kconfig**
- Debugging mit **KGDB**
  - Remote Debugging
  - Linux bietet spezielle Debugoptionen an

# Motivation

---

- Free/Libre Open Source Software erlaubt Änderungen am Sourcecode
  - eigenständige Behebung von Fehlern
  - implementieren eigener Featurewünsche

- Free/Libre Open Source Software erlaubt Änderungen am Sourcecode
  - eigenständige Behebung von Fehlern
  - implementieren eigener Featurewünsche
  - ...idealerweise Teilen der Änderungen

- Free/Libre Open Source Software erlaubt Änderungen am Sourcecode
  - eigenständige Behebung von Fehlern
  - implementieren eigener Featurewünsche
  - ...idealerweise Teilen der Änderungen
- Kollaboration erfordert einheitliches Entwicklungsmodell
  - Diskussion über Änderungen & langfristige Richtungsentscheidungen
  - Erhalten von Qualitätsstandards
  - Umgang mit Lizenzfragen

Im Anschluss an diese Aufgabe solltet Ihr...

- die Werkzeuge zur Entwicklung des Linux-Kernels beschreiben
- den Entwicklungsprozess des Linux-Kernels beschreiben
- die **Git** Grundlagen beschreiben
- eigene Patches für den Linux-Kernel in den Entwicklungsprozess einbringen und akzeptiert bekommen

...können.



# Agenda

Free/Libre Open Source Software

Linux Entwicklungsmodell

Arbeiten mit Git

Arbeiten & Navigieren im Linux-Kernel

Zusammenfassung

Aufgabe 2

# Free/Libre Open Source Software

---

## The four essentials freedoms<sup>1</sup>

„The freedom to...

0. ... run the program as you wish, for any purpose.
1. ... study how the program works, and change it so it does your computing as you wish.
2. ... redistribute copies so you can help your neighbor.
3. ... distribute copies of your modified versions to others.“

<sup>1</sup>Richard Stallman - „The Free Software Definition“

- Streitpunkt über Bedeutung der Begriffe
- Diskussionsfrage:
  - Wie umgehen mit Änderungen und Redistribution?
  - Wie umgehen mit Änderungen die anderen Lizenzen unterliegen?
  - Aber auch: Stellt die GPL Freiheit sicher oder schränkt sie diese ein?
- Free/Libre Open Source Software als „Kompromiss“
  - Libre als bewusste Abgrenzung zum Gratisbegriff
  - „Think free as in free speech, not free beer.“

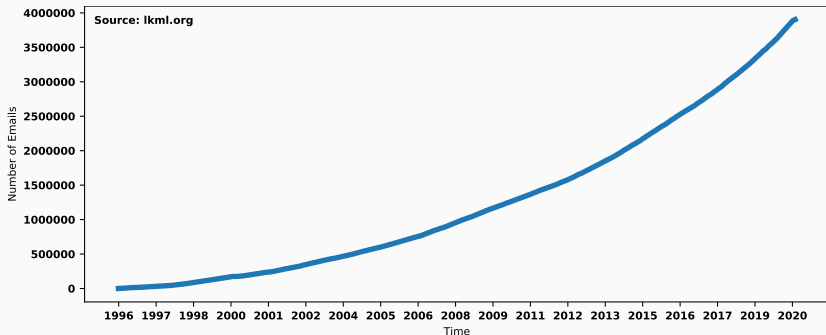
# Linux Entwicklungsmodell

---

- „The Cathedral and the Bazaar“ (Eric S. Raymond)
  - Essay über Methoden der (Open-Source) Software-Entwicklung
  - Basiert auf Beobachtungen des Entwicklungsprozesses des Linux-Kernels
- **Hauptaussagen**
  - „Every good work of software starts by scratching a developer’s personal itch.“  
Kapitel 2: „The Mail Must Get Through“
  - „Release early. Release often“  
Kapitel 4: „Release Early, Release Often“
  - „If you treat your beta-testers as if they’re your most valuable resource, they will respond by becoming your most valuable resource.“  
Kapitel 5: „Is A Rose Not A Rose?“

# Linux-Upstream-Entwicklung (Fortsetzung)

- Gesamte **relevante** Kommunikation nur via Email
- Zentraler Kommunikationskanal:  
Linux-Kernel-Mailing-List (LKML)



- Ca. 5700 Abonnenten
- Weitere Verbreitung über RSS-Feeds, News u.Ä.

# Kollaboration am Linuxkernel

- Änderungen durchlaufen Kreuzgutachten („Peer-Review“)
  - Gewachsene Hierarchie mit „Benevolent Dictator For Life“
  - Meistens siegt der überlegene Ansatz
- Festgelegter Prozess - Unabhängig ob Fix oder Feature

## Einflussreichste Aktoren



Linus Torvalds



Andrew Morton



Alan Cox



Greg Kroah-Hartman



- Quelltext in Linux werden normalisiert bearbeitet
- Documentation/CodingStyle
- automatisierter Test: `scripts/checkpatch.pl`

⇒ Erleichtert das Lesen und Verständnis

- Quelltext in Linux werden normalisiert bearbeitet
- Documentation/CodingStyle
- automatisierter Test: `scripts/checkpatch.pl`

⇒ Erleichtert das Lesen und Verständnis  
... und vermeidet Auseinandersetzungen

**Verwendet `checkpatch.pl`!**

Verstöße gegen die Richtlinien führen häufig zur Ablehnung des Patches.

- Patches enthalten generell nur Änderungen & deren Beschreibung
- Problem: Lizenzfragestellungen
  - Ist Sender des Patches auch sein Autor?
  - Wie dürfen die Änderungen genutzt werden?

⇒ Konvention

- Patches müssen von Autoren abgesegnet werden
- Spezielle Notation für Art des Beitrags

Signed-off-by: Random J Dev <random@developer.example.org>

### **Bedeutung**

(Auszug, Details in `Documentation/SubmittingPatches`)

- Der Autor bestätigt das Werk ganz oder in Teilen selbst geschrieben zu haben.
- Der Autor bestätigt das Recht zur Veröffentlichung zu haben
- Der Autor erlaubt die Verwendung und den Vertrieb der Änderung unter den Bedingungen der ursprünglichen Version

- Co-developed-by: Wer hat an dem Patch mitgewirkt (Erfordert zusätzliches Signed-off-by)
- Reported-by: Wer hat das Problem (richtig) gemeldet
- Reviewed-by: Wer hat die Änderung begutachtet
- Tested-by: Wer hat den Patch getestet
- Acked-by: Patch ist zur Kenntnis genommen worden, nicht notwendigerweise aber getestet

# Beispiel für Absegnungen

Subject: [PATCH] ACPICA: Fix possible fault in return package object repair code

Fixes a problem that can occur when a lone package object is wrapped with an outer package object in order to conform to the ACPI specification. Can affect these predefined names: \_ALR,\_MLS,\_PSS,\_TRT,\_TSS,\_PRT,\_HPX,\_DLM,\_CSD,\_PSD,\_TSD

[https://bugzilla.kernel.org/show\\_bug.cgi?id=44171](https://bugzilla.kernel.org/show_bug.cgi?id=44171)

The bug got introduced by commit 6a99b1c94d053b3420eaa4a4bc in v3.4-rc6, thus it needs to get pushed into 3.4 stable kernels as well.

Reported-by: Vlastimil Babka <caster@gentoo.org>  
Tested-by: Vlastimil Babka <caster@gentoo.org>  
Tested-by: marc.collin@laboiteaprog.com  
Signed-off-by: Bob Moore <robert.moore@intel.com>  
Signed-off-by: Lin Ming <ming.m.lin@intel.com>  
CC: stable@vger.kernel.org

---

```
drivers/acpi/acpica/nspredef.c | 2 +-  
1 files changed, 1 insertions(+), 1 deletions(-)
```

## „Should I just stop attempting to make these trivial fixes?“

**LKML Post**

[lkml.org/lkml/2004/12/20/255](http://lkml.org/lkml/2004/12/20/255)

„Should I just stop attempting to make these trivial cleanups/fixes/whatever patches? are they more noise than gain? am I being a pain to more skilled people on lkml or can you all live with my, sometimes quite ignorant, patches?“

## „Should I just stop attempting to make these trivial fixes?“

LKML Post

[lkml.org/lkml/2004/12/20/255](http://lkml.org/lkml/2004/12/20/255)

„Should I just stop attempting to make these trivial cleanups/fixes/whatever patches? are they more noise than gain? am I being a pain to more skilled people on lkml or can you all live with my, sometimes quite ignorant, patches?“

- Oft wird der Nutzen von kleineren Patches unterschätzt
- Contributor fehlt der Mut den Patch einzureichen
- Prozess erfordert insbesondere viel **Ausdauer**



# Should I just stop attempting to make these trivial fixes?

## Nicht schüchtern sein

- FLOSS Projekte in der Regel froh über Beiträge
- Einreichungen werden häufig nicht sofort **akzeptiert**.
- Maintainer sind auch nur Menschen

# Should I just stop attempting to make these trivial fixes?

## Nicht schüchtern sein

- FLOSS Projekte in der Regel froh über Beiträge
- Einreichungen werden häufig nicht sofort **akzeptiert**.
- Maintainer sind auch nur Menschen
- Auch Maintainer haben eine persönliche Schmerzgrenze!

## Checkliste: Einsenden von Patches in den Linuxkern (1/2)

- ✓ Patches im unified diff Format
- ✓ Sinnvolle und nachvollziehbare Beschreibungen der Änderungen
- ✓ Nur **eine** logische Änderung pro Patch (/Commit)
- ✓ Richtige(n) Adressat(en) finden
- ✓ Keine Dateianhänge, kein MIME, kein HTML

## Checkliste: Einsenden von Patches in den Linuxkern (2/2)

- ✓ Basisversion genau angeben
- ✓ Knappe Betreffszeile
  - Betreffszeile maximal 75 Zeichen mit Kurzbeschreibung
  - Zusätzlich ausführliche Patchbeschreibung und Intention
- ✓ Unnötige Diskussionen über Geschmack (etc.) vermeiden
  - ↪ Siehe auch [bikeshed.org](http://bikeshed.org)

### Weitere Ressourcen

- Andi Kleen: „On submitting kernel patches”  
[halobates.de/on-submitting-patches.pdf](http://halobates.de/on-submitting-patches.pdf)
- [Documentation/SubmittingPatches](#)

# Arbeiten mit Git

---



<https://imgs.xkcd.com/comics/git.png>

# Repositories einrichten und Änderungen einspielen

- Initialisieren einen Repos im aktuellen Verzeichnis

```
$ git init
```

- Initiales Klonen der Quellen

```
$ git clone \  
git.kernel.org/pub/scm/linux/kernel/git/torvalds/linux.git
```

- Vormerken einer (versionierten) Datei als Kandidat für nächsten Commit („staging“)

```
$ git add Datei
```

- Einspielen von eigenen Änderungen in Datei oder aller Änderungen

```
$ git commit Datei
```

```
$ git commit -a
```

- Dateizustände (neu, unbekannt, geändert, vorgemerkt) anzeigen  
`$ git status`
- Die neuste Änderung untersuchen  
`$ git show`
- Commit-Historie betrachten  
`$ git log`



## ■ Anzeigen von Änderung zw. Commits, Working Tree, ...

```
$ git diff 61831aa9cf08 0fd83bfd71a5
diff --git /arch/x86/include/asm/kvm_host.h b/arch/x86/include/asm/kvm_host.h
index 7ba99c0759cf..d79b40cd8283 100644
--- a/arch/x86/include/asm/kvm_host.h
+++ b/arch/x86/include/asm/kvm_host.h
@@ -1664,8 +1664,8 @@ static inline bool kvm_irq_is_postable(struct kvm_lapic_irq *irq)

/* We can only post Fixed and LowPrio IRQs */
-     return (irq->delivery_mode == APIC_DM_FIXED ||
-            irq->delivery_mode == APIC_DM_LOWEST);
+     return (irq->delivery_mode == dest_Fixed ||
+            irq->delivery_mode == dest_LowestPrio);
```

## ■ Anzeigen von Änderung zw. Commits, Working Tree, ...

```
$ git diff 61831aa9cf08 0fd83bfd71a5
diff --git /arch/x86/include/asm/kvm_host.h b/arch/x86/include/asm/kvm_host.h
index 7ba99c0759cf..d79b40cd8283 100644
--- a/arch/x86/include/asm/kvm_host.h
+++ b/arch/x86/include/asm/kvm_host.h
@@ -1664,8 +1664,8 @@ static inline bool kvm_irq_is_postable(struct kvm_lapic_irq *irq)

/* We can only post Fixed and LowPrio IRQs */
-     return (irq->delivery_mode == APIC_DM_FIXED ||
-            irq->delivery_mode == APIC_DM_LOWEST);
+     return (irq->delivery_mode == dest_Fixed ||
+            irq->delivery_mode == dest_LowestPrio);
```

## ■ Veränderte Dateien

# Anzeigen von Änderungen

## ■ Anzeigen von Änderung zw. Commits, Working Tree, ...

```
$ git diff 61831aa9cf08 0fd83bfd71a5
diff --git /arch/x86/include/asm/kvm_host.h b/arch/x86/include/asm/kvm_host.h
index 7ba99c0759cf..d79b40cd8283 100644
--- a/arch/x86/include/asm/kvm_host.h
+++ b/arch/x86/include/asm/kvm_host.h
@@ -1664,8 +1664,8 @@ static inline bool kvm_irq_is_postable(struct kvm_lapic_irq *irq)

/* We can only post Fixed and LowPrio IRQs */
-   return (irq->delivery_mode == APIC_DM_FIXED ||
-           irq->delivery_mode == APIC_DM_LOWEST);
+   return (irq->delivery_mode == dest_Fixed ||
+           irq->delivery_mode == dest_LowestPrio);
```

### ■ Veränderte Dateien

### ■ Gelöschte Inhalte

# Anzeigen von Änderungen

## ■ Anzeigen von Änderung zw. Commits, Working Tree, ...

```
$ git diff 61831aa9cf08 0fd83bfd71a5
diff --git /arch/x86/include/asm/kvm_host.h b/arch/x86/include/asm/kvm_host.h
index 7ba99c0759cf..d79b40cd8283 100644
--- a/arch/x86/include/asm/kvm_host.h
+++ b/arch/x86/include/asm/kvm_host.h
@@ -1664,8 +1664,8 @@ static inline bool kvm_irq_is_postable(struct kvm_lapic_irq *irq)

/* We can only post Fixed and LowPrio IRQs */
-   return (irq->delivery_mode == APIC_DM_FIXED ||
-           irq->delivery_mode == APIC_DM_LOWEST);
+   return (irq->delivery_mode == dest_Fixed ||
+           irq->delivery_mode == dest_LowestPrio);
```

■ **Veränderte Dateien**

■ **Gelöschte Inhalte**

■ **Neu hinzugefügte Zeilen**

# Anzeigen von Änderungen

## ■ Anzeigen von Änderung zw. Commits, Working Tree, ...

```
$ git diff 61831aa9cf08 0fd83bfd71a5
diff --git /arch/x86/include/asm/kvm_host.h b/arch/x86/include/asm/kvm_host.h
index 7ba99c0759cf..d79b40cd8283 100644
--- a/arch/x86/include/asm/kvm_host.h
+++ b/arch/x86/include/asm/kvm_host.h
@@ -1664,8 +1664,8 @@ static inline bool kvm_irq_is_postable(struct kvm_lapic_irq *irq)

/* We can only post Fixed and LowPrio IRQs */
- return (irq->delivery_mode == APIC_DM_FIXED ||
-        irq->delivery_mode == APIC_DM_LOWEST);
+ return (irq->delivery_mode == dest_Fixed ||
+        irq->delivery_mode == dest_LowestPrio);
```

### ■ Veränderte Dateien

### ■ Gelöschte Inhalte

### ■ Neu hinzugefügte Zeilen

### ■ Zeile, Einrückung der Modifikation

# Anzeigen von Änderungen

## ■ Anzeigen von Änderung zw. Commits, Working Tree, ...

```
$ git diff 61831aa9cf08 0fd83bfd71a5
```

```
diff --git /arch/x86/include/asm/kvm_host.h b/arch/x86/include/asm/kvm_host.h
```

```
index 7ba99c0759cf..d79b40cd8283 100644
```

```
--- a/arch/x86/include/asm/kvm_host.h
```

```
+++ b/arch/x86/include/asm/kvm_host.h
```

```
@@ -1664,8 +1664,8 @@ static inline bool kvm_irq_is_postable(struct kvm_lapic_irq *irq)
```

```
/* We can only post Fixed and LowPrio IRQs */
```

```
- return (irq->delivery_mode == APIC_DM_FIXED ||
```

```
-        irq->delivery_mode == APIC_DM_LOWEST);
```

```
+ return (irq->delivery_mode == dest_Fixed ||
```

```
+        irq->delivery_mode == dest_LowestPrio);
```

### ■ Veränderte Dateien

### ■ Gelöschte Inhalte

### ■ Neu hinzugefügte Zeilen

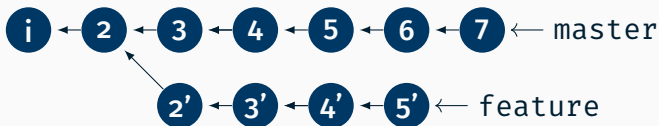
### ■ Zeile, Einrückung der Modifikation

### ■ Name der geänderten Funktion

# Nicht-lineare Commit Historie

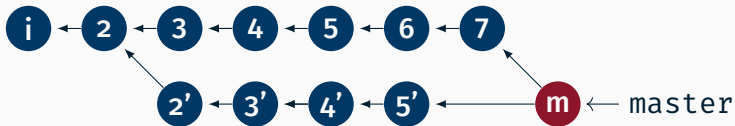
## ■ Erstellen von Verzweigungen

```
$ git branch feature
```



## ■ Zusammenführen von Verzweigungen

```
$ git merge feature
```



- Einspielen von entfernten Änderungen

```
$ git pull
```

- Weitere entfernte Repositories registrieren

```
$ git remote add next git://git.kernel.org/.../linux-next.git
```

- Registrierte Repositories auflisten

```
$ git remote -v
```

- Alle Remotes nachladen (aktueller Branch wird nicht verändert)

```
$ git remote update
```

oder

```
$ git fetch --all
```



- Commit Historie ist selten linear
  - Merge Commits
  - Ein Commit kann mehr als einen Elterncommit haben
  - Historie ist ein gerichteter azyklischer Graph
- Spezielle Notation zum Zugriff auf Vorgängercommits

# ~n zeigt auf nten Vorgängercommit

```
$ git show HEAD~1
```

# Ersten "Großelterncommit", alternativ HEAD~~

```
$ git show HEAD~2
```

# Austauschen von Änderungen als Patches

- Die letzten **2** Änderungen als Patch formatieren
  - # bei einem Merge den ersten Vorgänger wählen
  - \$ git format-patch HEAD~2
- Sendeziel für Patchversand via E-Mail vorgeben
  - \$ git config sendemail.to=linux-kernel@i4.cs.fau.de
- Patchset mit den letzten **3** Änderungen via E-Mail senden
  - \$ git send-email --compose HEAD~3

# Arbeiten & Navigieren im Linux-Kernel

---

- **GDB/KGDB** nicht für alle Fehlertypen der beste Ansatz
  - Was tun, wenn der Kernel sehr früh *oopst*?
  - Wie vorgehen, die serielle Schnittstelle fehlerhaft ist?
  - Wenn alles „wegoptimiert“ ist?
- Lösung: `printk()` Ausgaben auf der Konsole und im Kernel-Log

## Debuggen des Linux-Kernels (2/2)

Beispiel aus `linux-5.1/init/main.c`:

```
pr_notice("Kernel command line: %s\n", boot_command_line);
```

mit

```
#define pr_notice(fmt, ...) \
printk(KERN_NOTICE pr_fmt(fmt), ##__VA_ARGS__)
```

Meldungen *nachlesen*:

```
$ dmesg
```

```
...
```

```
[0.000000] Kernel command line: root=/dev/vda1 console=ttyS0
```

```
...
```

# Debuggen mit `printk()`

- alle Ausgaben haben eine Priorität (<n> am Stringanfang)
- Ausgabe erst bei Kernel Log-Level  $n + 1$
- Log-Level wird standardmäßig auf 7 gesetzt
- Anpassen über Kommandozeile (`debug`, `loglevel`) und `klogd(8)`

## Mögliche Prioritäten (`include/linux/printk.h`)

```
#define KERN_EMERG    "<0>" /* system is unusable          */
#define KERN_ALERT    "<1>" /* action must be taken immediately */
#define KERN_CRIT     "<2>" /* critical conditions             */
#define KERN_ERR      "<3>" /* error conditions                */
#define KERN_WARNING  "<4>" /* warning conditions              */
#define KERN_NOTICE   "<5>" /* normal but significant condition */
#define KERN_INFO     "<6>" /* informational                   */
#define KERN_DEBUG    "<7>" /* debug-level messages            */
```

## ■ cscope

- Symbolsuche, insbesondere für Funktionen
- Aufrufer und Aufgerufene

```
$ make cscope # cscope Index erzeugen
```

```
$ cscope -d # Index für modifizierte Dateien nicht neu erzeugen
```

```
... # Verlassen mit CTRL-d
```

## ■ ctags

- Ebenfalls Symbolsuche
- Autocompletion

# Zusammenfassung

---



- Free/Libre Open Source Software
  - Charakteristika von Freie/Libre und Quelloffener Software
  - Der Linux-Kernel als Vorzeigeprojekt der Open Source Community
  - Entwicklungsprozess des Kernels
- Debugging des Linux-Kernels
  - `printk`, wenn der **GDB/KGDB** versagt
  - Es existieren viele Werkzeuge zur Navigation in Sourcecode

## Aufgabe 2

---

- Vorgegebene Kernelquellen mit injizierten Fehlern:  
`/proj/i4passt/kernel/linux-borked.git`
- Vorgegebene Kernelconfig:  
`/proj/i4passt/kernel/linux-borked.config`
- Verschiedene Fehlertypen, *nicht nur* Systemabstürze
  - „Normale“ Systembenutzung um alle Fehler zu finden
  - **KGDB** nicht immer das optimale Werkzeug
- Insgesamt zwölf verschiedene Fehler
- Patches an `linux-kernel@i4.cs.fau.de`
  - Achtung, Hannover liest mit!

Bearbeitungszeit bis 23. November

**Fragen?**