
1 Übungsaufgabe #4: Linux-Upstream-Entwicklung

Anknüpfend an die letzte Übungsaufgabe gilt es in der dritten Aufgabe *echte* Defekte im linux-next Kernel zu finden und zu beseitigen. Die Defekte sollen sorgfältig analysiert und entsprechend repariert werden. Daraus resultierende Patches werden auf der Mailingliste und in den Rechnerübung bewertet und korrigiert und so für eine Upstream-Abgabe aufbereitet. Dieser Prozess wird einem echten Kreuzgutachten („Peer-Review“) nachempfunden, wie er im Abgabe-Prozess der Linux-Upstream-Entwicklung angewandt wird. Die aufbereiteten Patches sollen dann an die jeweiligen Maintainer eingesandt werden. Nach der Übung sollen im Kontext einer Tafelübung die Erfahrungen diskutiert werden.

1.1 Beseitigung von Defekten im Mainline-Kernel

Um Defekte im Mainline-Kernel zu finden bedarf es in der Regel keiner großen Mühe. Dies liegt in erster Linie an der Komplexität des Linux-Kerns. So zählt Linux v5.0 über 26 Millionen Zeilen Programmcode. Als Faustregel gilt, dass sich häufiger Fehler in selten abgearbeiteten Programmpfaden befinden, als in oft ausgeführten. Deshalb eignet sich vor allem die Analyse von neuem Programmcode, der *nicht* in Hauptkomponenten des Linux-Kerns (z. B. experimentelle Treiber) eingeht, um Defekte zu entdecken.

1.2 Linux-Staging

Für eine leichtere Integration von Linux Treibern in den Kern wurde vor einigen Jahren ein Bereich im Kernel erstellt, der es Entwicklern erleichtern soll, ihre Treiber in den Linux-Kern zu integrieren („Upstream“). Dieser Bereich wird als *Staging*-Bereich bezeichnet. Er ist Teil des linux-next Kernels und bietet die Möglichkeit, unausgereiften Programmcode aufzunehmen, der aus unterschiedlichen Gründen noch nicht für den Hauptbereich des Linux-Kerns akzeptiert wird (z. B. Stabilität, Programmcode-Qualität). Der Staging-Bereich beinhaltet demnach Programmcode, der sich für die Erstellung erster eigener Patches eignet. Im Idealfall finden sich durch die Aufnahme in den Staging-Bereich weitere Entwickler, die daran interessiert sind, den existierenden Programmcode des Treibers von Fehlern zu bereinigen und so aufzubereiten, dass er zu einem späteren Zeitpunkt entsprechend der Abgaberichtlinien (vgl. `Documentation/Submit{Checklist,tingDrivers,tingPatches}`) in den Hauptbereich des Linux-Kerns zu integrieren.

Der Staging-Bereich umfasst eine Vielzahl an Treibern die sich über eine mehr als eine halbe Million Zeilen Programmcode erstrecken. Damit diese Treiber bei der Übersetzung des Linux-Kerns kompiliert werden, muss die Konfigurationsvariable `STAGING` gesetzt sein. Um etwaige Warnungen bei der Übersetzung zu erhalten, sollten die Module des Staging-Bereichs mit `make V=1 M=drivers/staging` übersetzt werden. Analog kann mit dem Kommando `make V=1 M=drivers/staging clean` sichergestellt werden, dass lediglich zuvor übersetzter Programmcode aus dem Staging-Bereich gelöscht wird.

1.3 Kernel-Janitors

Das Kernel-Janitors-Projekt sammelt Beispiele und Hinweise, wie häufig auftretende Fehler im Linux-Kern beseitigt werden (→ <http://kernelnewbies.org/KernelJanitors/ToDo>). Diese Informationen sollen als Grundlage für die gezielte Suche nach subtilen Fehlern im Staging-Bereich genutzt werden. Des Weiteren sind auf der Hauptseite von Linux-Kernel-Newbies weitere Hinweise darüber zu finden, wie der Prozess für die Abgabe von Programmcode-Änderungen für den Linux-Kern funktioniert und abläuft.

Für die Abgabe von solchen Patches wird die Janitor-Mailingliste (<http://marc.info/?l=kernel-janitors>) verwendet. Im Gegensatz zu der Haupt-Mailingliste LKML (<http://marc.info/?l=linux-kernel>) ist der Umgangston auf dieser Liste weniger rau und eignet sich daher besser für die Abgabe von ersten Kernel-Patches. Selbes gilt für die Mailingliste des „Linux Driver Project“ (<http://driverdev.linuxdriverproject.org/mailman/listinfo/devel>), auf welcher Patches für den Staging-Bereich eingesandt, diskutiert und akzeptiert bzw. abgelehnt werden.

Aufgaben:

- Programmcode z.B. des Linux-Staging-Bereich des aktuellen Staging-Baums (<https://git.kernel.org/pub/scm/linux/kernel/git/next/linux-next.git>) analysieren und Defekte finden
- Patches für gefundene Fehler erstellen und ausreichend testen:
 - Patch auf Sourcen des (aktuellen!) Linux-Kernel anwenden
 - Kompilieren und sicherstellen, dass der betroffene Code übersetzt wurde
 - Überprüfung des Patches mit `scripts/checkpatch.pl`

-
- Patches an `linux-kernel@i4.cs.fau.de` senden und Rückmeldung der Betreuer abwarten (→ konstruktive Kritik, Vermeidung von Doppelmeldungen)
 - Von Betreuern genehmigte Patches an die jeweiligen Upstream-Entwickler (`scripts/get_maintainers.pl`) einsenden
 - Einreichung an Entwickler bitte immer auch im CC an `linux-kernel@i4.cs.fau.de`
 - Kritik, Anregungen, Vorschläge von Upstream adäquat behandeln, mit Ziel den Patch aufgenommen zu bekommen
 - Vorstellung der Ergebnisse in der Tafelübung (Diskussionsrunde)

Bearbeitung bis 21.06.2019, Vorstellung der Ergebnisse in der Tafelübung am 27.06.2019