

Praktikum angewandte Systemsoftwaretechnik

Organisatorisches

Alexander Würstlein, Moritz Strübe, Rainer Müller

Lehrstuhl Informatik 4

Oktober 2013

Beteiligte Personen, Lehrstuhl Informatik 4

Organisatoren des Praktikums



Moritz Strübe



Alexander Würstlein



Rainer Müller

Ziele in PASST

- Erlernen und Vertiefen von Kenntnissen der Softwareentwicklung
 - Im Umfeld des Linux-Kerns
 - Innerhalb des Linux-Kerns
- Vermittlung des notwendigen Werkzeugwissens
 - Kernel konfigurieren und bauen
 - Verwendung von virtuellen Maschinen als Testplattform
 - Versionsverwaltung mit **git**
 - Verwendung von GDB innerhalb des Linux-Kerns
- Hintergrundwissen
 - Open-Source-Entwicklungsprozesse kennenlernen
 - Finden und Beheben von Bugs

Aufbau der Veranstaltung

Vorlesungszeit

Tafelübung (2,5 ECTS)

- Vorstellung der Aufgaben
- Hilfestellung zur Lösung
- Präsentationen und Diskussionen

Aufbau der Veranstaltung

Vorlesungszeit

Tafelübung (2,5 ECTS)

- Vorstellung der Aufgaben
- Hilfestellung zur Lösung
- Präsentationen und Diskussionen

Programmierteil (5 ECTS)

- Bearbeitung der Aufgaben
- Vorbereitung auf Blockpraktikum
- Rechnerübungen mit konkreter Hilfestellung

Aufbau der Veranstaltung

Vorlesungszeit

Tafelübung (2,5 ECTS)

- Vorstellung der Aufgaben
- Hilfestellung zur Lösung
- Präsentationen und Diskussionen

Programmierteil (5 ECTS)

- Bearbeitung der Aufgaben
- Vorbereitung auf Blockpraktikum
- Rechnerübungen mit konkreter Hilfestellung

Semesterferien

Blockpraktikum (2,5 ECTS)

Gruppenweises Bearbeiten einer umfangreichen Aufgabe (Projektcharakter)

Möglicher Termin: 3. bis 16. März 2014 (KW 10&11)

Programmierteil

Aufgaben

- Sechs *gemeinsame* Aufgaben
- Programmieren, Kurzvorträge und Praktikumsvorbereitung
- Bearbeitung in **Zweiergruppen**
- Etwa zwei bis drei Wochen pro Aufgabe
- Abgabe durch Vorführen in der Rechnerübung am Abgabetag

Programmierteil

Aufgaben

- Sechs *gemeinsame* Aufgaben
- Programmieren, Kurzvorträge und Praktikumsvorbereitung
- Bearbeitung in **Z**weiergruppen
- Etwa zwei bis drei Wochen pro Aufgabe
- Abgabe durch Vorführen in der Rechnerübung am Abgabetag

Rechnerübungen

- Termine:
 - Mo 12 – 14 Uhr
 - Mi 14 – 16 Uhr (?)
 - ?
- Betreuer zu Beginn vor Ort
- Sonst auf Abruf in Büro

Aufgaben in der Vorlesungszeit

Aufgabe 1: Umgebung einrichten Einrichten einer Debian-Installation in einer VM mit selbst kompiliertem Kern. Verwendung von GDB und KGDB.

Aufgabe 2: Kernel-Vergleiche Kernel, Werkzeuge und Entwicklungsprozesse verschiedener Linux-Distributionen vergleichen und dazu vortragen.

Aufgabe 3: OOPS! Kernel-Bugs finden und reparieren Einen komplexen Bug im Linux-Kern lokalisieren und reparieren.

Aufgabe 4: Patches bauen und einsenden Erzeugen eines Patches zur Behebung eines Kernel-Problems, "Einsenden" und akzeptiert bekommen und davon berichten.

Aufgabe 5: Temperatursensor Entwicklung eines Gerätetreibers für einen (selbstzulötenden!) USB-Sensor in den Kern.

Aufgabe 6: Treiber für PCI Gerät Einen komplizierteren Treiber für einen Logic Analyzer auf FPGA-Basis entwickeln.

Mögliche Themen für das Blockpraktikum

- ① Debugging von AVR-Mikrocontrollern (AVaRICE) (Morty)
 - AVR-Debugging unter Linux ist langsam
 - verbessern durch Reverse-Engineering der USB-Kommunikation des Debuggers unter Windows
- ② Optimierung des Cache-Verhaltens von Linux (Morty, arw)
 - Implementierung von `MADV_DONTNEED` in `madvise(2)`
 - verhindert, dass einmaliges Lesen z. B. durch Backupsoftware den Plattencache überschreibt

Mögliche Themen für das Blockpraktikum (Forts.)

- ③ Schwachstellen in USB-Treibern finden und beheben (Rainer)
 - Kreative Dinge mit dem facedancer11 (emuliert beliebige USB-Clients)
 - Beispiel: Xorg crasht(e) bei „%n%n%n%n“ als Geräteiname
- ④ Logic Analyzer auf PCI Express (arw)
 - bestehende PCI-Logic-Analyzer-Karte auf PCI Express portieren
 - weitere Features implementieren

Mögliche Themen für das Blockpraktikum (Forts.)

- 5 LAOS: Latency Aware Operating System (Gabor)
 - einfachere Syscalls
 - Scheduling-Entscheidung nicht unnötigerweise treffen
 - verlässliches Scheduling beim Timer-IRQ
 - Initialisierung globaler System-Objekte
- 6 Eigene Hardware bauen (arw)
- 7 Entwicklung eines Gerätetreibers
 - Ihr kennt/habt Hardware, die nicht unter Linux funktioniert?
 - Entwickelt einfach euren eigenen Treiber

Mögliche Themen für das Blockpraktikum (Forts.)

8 Eigene Ideen und Vorschläge

Benotung der Veranstaltung

- Punkte auf Aufgaben und Vorträge
 - 40 % Punkte Semesterteil
 - Punkte pro Aufgabe (15 bis 30 Punkte)
 - Gewichtungsfaktor: Aufwand, Schwierigkeit
 - 60 % Punkte Blockpraktikumsteil
 - Vorbereitung und Anfangsvortrag
 - Projektdurchführung
 - Abschlussvortrag
- Punkte in Note nach Notenschlüssel

Bewertungskriterien

- Aufgabenerfüllung
- Codequalität: Funktion, Lesbarkeit, Kommentare, Coding Style
- interne Interaktion: Zusammenarbeit und Aufgabenteilung innerhalb der (Zweier-)Gruppe
- externe Interaktion: Kommunikation und Zusammenarbeit mit Betreuern und Externen (Upstream)
- Vortrag und Vorstellung: Gliederung, Verständlichkeit, Inhalt, Stil, Behandlung von Fragen
- **sofern anwendbar** und **unterschiedlich gewichtet** je nach Aufgabe