

# Praktikum angewandte Systemsoftwaretechnik

## Organisatorisches

Alexander Würstlein

Lehrstuhl Informatik 4

2016-10-20

# Beteiligte Personen, Lehrstuhl Informatik 4

## Organisatoren des Praktikums



Alexander Würstlein

# 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 (2,5 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 (2,5 ECTS)

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

## Semesterferien

### Blockpraktikum (5 ECTS)

Gruppenweises Bearbeiten einer umfangreichen Aufgabe (Projektcharakter)

Möglicher Termin: 27. Februar bis 10. März 2016 (KW 9&10)  
Festlegung vor Semesterende

# Programmierteil

## Aufgaben

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

# Programmierteil

## Aufgaben

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

## Rechnerübungen

- Termine:
  - Mi 16 – 18 Uhr
  - Fr 12 – 14 Uhr
- Betreuer zu Beginn vor Ort
- Sonst auf Abruf in Büro (0.052)

# 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 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%n“ als Gerätename
- ② Logic Analyzer auf PCI Express (arw)
  - bestehende PCI-Logic-Analyzer-Karte auf PCI Express portieren
  - weitere Features implementieren

# Mögliche Themen für das Blockpraktikum (Forts.)

- ③ paralleles rsync für lokale Dateien (arw)
  - ermöglicht dem Betriebssystem besseres I/O-Scheduling, den Platten TCQ
  - würde den Fall „viele kleinen Dateien“ stark beschleunigen
- ④ USB-Serial in Userspace (Rainer)
  - Serielle USB-Geräte (FTDI, USB RS232) im Kernel (z.B. Linux: `/dev/ttyUSB*`)
  - auf Betriebssystemen wie MacOS schlecht gewartet
  - mit libusb aber auch als Benutzerprogramm umsetzbar
  - Aufgabe: Erweiterung von socat, aufbohren der pty-Unterstützung um USB-Serial

# Mögliche Themen für das Blockpraktikum (Forts.)

- ⑤ Erweiterung von Time-Triggered eCos auf Mehrkernbetrieb (Florian, Tobias)
  - tt-eCos: zeitgesteuertes Echtzeitbetriebssystem für eingebettete Einprozessorsysteme
  - arbeitet Ablaufabelle ab
  - Erweiterung: eine Ablaufabelle pro Prozessor → zeitsynchron!
  
- ⑥ Kexec für Xen Gastdomänen (Klaus)
  - Austausch des Kernels einer paravirtualisierten VM
  - Portierung von Patch für Kernel 2.6.18 auf aktuellen Kernel
  - Anpassungen an Bootcode und Paging notwendig

# Mögliche Themen für das Blockpraktikum (Forts.)

- 7 LAOS: Latency Aware Operating System (Gabor)
  - einfachere Syscalls
  - Scheduling-Entscheidung nicht unnötigerweise treffen
  - verlässliches Scheduling beim Timer-IRQ
  - Initialisierung globaler System-Objekte

# Mögliche Themen für das Blockpraktikum (Forts.)

- 8 Eigene Hardware bauen (arw)
- 9 Entwicklung eines Gerätetreibers
  - Ihr kennt/habt Hardware, die nicht unter Linux funktioniert?
  - Entwickelt einfach euren eigenen Treiber
- 10 Eigene Ideen und Vorschläge
- 11 Themenliste ist **vorläufig** und wird sich ändern

# 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

# Notenfindung

Teilnote		A1	A2	A3	A4	A5	A6	Blockpraktikum
Gewichtung		1	1	2	2	2	2	15

# 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