

# Praktikum angewandte Systemsoftwaretechnik

## Blockpraktikum

Alexander Würstlein

Lehrstuhl Informatik 4

2017-02-09

# Organisatorisches

- Projektwahl und Gruppenbildung: 2–3er Gruppen
- Projektvorstellung
  - 20 min. Präsentation im Plenum + 10 min. Diskussion
  - Problemvorstellung, Ansatz, erwartete Ergebnisse, Zeitplanung
- 2 Wochen Vollzeit
  - Bei Bedarf tägliches *Jour Fixe*
  - Zwischentreffen
- Abschlusspräsentation
  - 20 min. Präsentation im Plenum + 10 min. Diskussion
  - Ergebnisse, Erfahrungen, Fazit
- Termin: 2017-03-01 P 2017-03-14
- Beginn: Mo 2017-03-01 10:00 Uhr, 0.031-113 (Aquarium)

# Zielsetzung

Erfolg im Praktikum wird am Erreichen der Zielsetzungen gemessen:

- Gelerntes anwenden
- Selbständige Projektdurchführung und Gruppenarbeit
- Softwareentwicklungsprozesse in OSS-Projekten praktisch anwenden
  - durch Verwendung entsprechender Werkzeuge (git, Patche, ...)
  - durch Einbindung der Entwicklergemeinschaft (Features an Upstream)
  - Endziel: benutzbare Software für euch, uns und den Rest der Welt

Bewertet wird:

- Lösungsfindung und Lösung
- Kollaboration zwischen euch
- Kommunikation und Zusammenarbeit mit Upstream
- Projekt wird veröffentlicht (Publish or it didn't happen!)

# Notenfindung (Wiederholung)

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

- Semesterbegleitender Teil macht 40% der Punkte aus
- erreichbare Punktezahlen und damit Gewichtung entsprechend dem Umfang der Aufgaben
- Blockpraktikum umfasst die restlichen 60%

# Themen für das Blockpraktikum

## ① USB-over-IP (arw)

- in den letzten Semestern erstellte Verbesserungen (IPv6, Crypto)
- Erweiterung um z.B. komfortablere Userspace-Tools, ACLs und fein-granulare Authentifizierung, ...
- Windows-Treiber (?!)

## ② Erweiterungen von FAUmachine (Volkmar)

- virtuelle Maschine mit Fehlerinjektion und externer Steuer-/Skriptbarkeit
- Erweiterung z.B. um
  - USB-IP-Integration oder USB-Geräte-Emulation
  - virtuelles Peripheriegerät (z.B. Balanciertisch), physikalisch simuliert und graphisch dargestellt, zum Entwickeln und Testen von Echtzeitsystemen
  - ...

- ③ Implementierung eines Schedulers für Jobcluster zum automatisierten Testen (Florian, Tobias)
  - Verteilung von Testfällen auf heterogene Hardware
  - Überwachung und dynamische Anpassung des Jobablaufs
  - Ressourcenplanung, -beschränkung und „einfrieren“ von Jobs

- ④ automatisches modulares Testsystem für EZS-Übungsaufgaben (Florian, Tobias, Peter Ulbrich)
  - Hardware in the loop, generiert Eingaben und prüft Verhalten des zu testenden Systems
  - Entwicklung von Übungsaufgaben, Testfällen
  - Entwurf und Herstellung der Hardware, evtl. auch mit zu regelndem physikalischem System (Balanciertisch)
  - Teilaspekt davon, in Zusammenarbeit mit Mitarbeitern, weiteren Gruppenmitgliedern

- ⑤ Gerrit um Abhängigkeitsgraph erweitern
  - bestehende Vorarbeit, inzwischen aktualisierungsbedürftig
  - aber von Upstream nicht angenommen, in Form bringen
  - neue Features
- ⑥ Logic Analyzer auf PCI Express (arw)
  - bestehende PCI-Logic-Analyzer-Karte auf PCI Express portieren
  - evtl. weitere Features implementieren



## 7 USB-Serial in Userspace (Rainer)

- Serielle USB-Geräte (FTDI, USB RS232) im Kernel (z.B. Linux: `/dev/ttyUSB*`)
- auf Betriebssystemen wie Mac OS X schlecht gewartet
- mit libusb aber auch als Benutzerprogramm umsetzbar
- Aufgabe: pty-Unterstützung im Kernel mit mehr ioctls, Erweiterung von socat um USB-Serial

## 8 Entwicklung auf Intel Xeon Phi (Rainer)

- PCIe-Erweiterungskarte mit vielen Kernen, Betriebssystem Linux
- gedachter Anwendungsbereich: HPC
- Implementierung von 512-bit SIMD für GLM (OpenGL Mathematics)
- Eigene Ideen?

## 9 fanotify() erweitern (arw)

- Schnittstelle zur Beobachtung von Dateisystemereignissen fuer Virens Scanner
- kann momentan (fast) nur Ereignisse bei `open()`
- unsere Anwendungsfälle Backup und Quotacheck brauchen aber noch `(un)link`, `chown`, ...

- 10 SDDM erweitern: Login-Manager für grosse Multiuser-Umgebungen (arw)
  - Nutzerfreundlichkeit verbessern: Tastenbelegung, Timeouts, Konfigurierbarkeit, Knöpfe
  - PAM-Einbindung: benutzerspezifische Meldungen beim Login
  - administrative globale Hinweise
- 11 Zwei-Faktor-Authentifizierung per U2F im Display-Manager (arw)
  - U2F: Challenge-Response ECDSA mit einem USB-Stick
  - Einbindung und Anpassung best. libs in Display-Manager (KDM, SDDM)
  - PAM-Modul (?)

# Themen für das Blockpraktikum (Forts.)

- 12 Eigene Hardware bauen
- 13 Entwicklung eines Gerätetreibers
  - Ihr kennt/habt Hardware, die nicht unter Linux funktioniert?
  - Entwickelt einfach euren eigenen Treiber
- 14 Eigene Ideen und Vorschläge

# Eure Aufgabe

- 1 Themen-Kandidaten aussuchen
- 2 mit Betreuern reden (<https://www4.cs.fau.de/People/>)
- 3 bis Do 2017-02-16: Thema aussuchen, Mail an [i4passt@lists.cs.fau.de](mailto:i4passt@lists.cs.fau.de)
- 4 dann: mit Betreuer(n) Aufgabenstellung diskutieren
- 5 ins Thema einlesen
- 6 Blockpraktikum vorbereiten: Problemvorstellung, Lösungsansatz, erwartete Ergebnisse, Zeitplan
- 7 bis zum Praktikumsbeginn: Anfangspräsentation erstellen