

3.33pt

# Praktikum angewandte Systemsoftwaretechnik

## Blockpraktikum

Alexander Würstlein

Lehrstuhl Informatik 4

2018-02-08

# 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: 2018-03-05 P 2018-03-16
- Beginn: Mo 2018-03-05 10:00 Uhr (s.t.), 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 „Energietransparenz“ für Cortex-M4-Boards (Peter W.)
  - Latenzen messen für den Übergang zwischen Power-States
  - Automatisierung mittels Red Pitaya
  - Erweiterung auf Energieverbrauchsmessung des Übergangs
- 2 RedPitaya-Frontend zur Energiemessung (Heiko, Peter W., Stefan)
  - USB/Netzwerk-Oszi und Funktionsgenerator, programmierbar
  - Spannung über Messwiderstand aufsummiert = Energie
  - Summieren zw. passiven o. aktiven Trigger-Ereignissen, tabellieren, ausgeben, plotten

- ③ Dateisystem mit möglichst konstanten, vorhersagbaren Antwortzeiten (Stefan)
  - ⇒ Folien separat

# Themen für das Blockpraktikum (Forts.)

- 4 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 (?!)
- 5 Erweiterungen von FAUmaschine (Volkmar)
  - virtuelle Maschine mit Fehlerinjektion und externer Steuer-/Skriptbarkeit
  - Erweiterung z.B. um
    - virtuelles Peripheriegerät (z.B. Balanciertisch), physikalisch simuliert und graphisch dargestellt, zum Entwickeln und Testen von Echtzeitsystemen
    - EFI-Unterstützung einbauen



- ⑥ Testen von FAUmaschine (Volkmar)
  - Vergleich mit realer Hardware
  - Power-on-self-test (POST) aufbohren
- ⑦ Energiemessungen auf X86 mittels RAPL (Volkmar)
  - sysfs-Interface zur aktuellen Leistungsaufnahme des Systems
  - Energieverbrauch von möglichst feingranularen Programmen und Codeschnipseln
  - Evaluation von Auflösung und Rauschen

- 8 Implementierung eines Schedulers für Jobcluster zum automatisierten Testen (Florian F., Tobias)
  - Verteilung von Testfällen auf heterogene Hardware
  - Überwachung und dynamische Anpassung des Jobablaufs
  - Ressourcenplanung, -beschränkung und „einfrieren“ von Jobs
- 9 USB-Stack auf EZS-Board integrieren (Simon, Peter U.)
  - auf ARM STM32 Cortex M4, bestehenden USB-Stack in ECOS integrieren
  - Datentransfer zu Linux-Host
  - Vergleich und Auswahl passender Software zur graphischen Darstellung auf dem Host

- 10 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?

## 11 Benchmarks für Branch-Prediction-Hinweise (Florian S.)

- `if (unlikely(a < 19))`
- geeignete Testfälle für Linux mit/ohne `likely & Co`
- verschiedene Architekturen (x86 Intel/AMD, ARM),  
`static_branch_likely()`

# Themen für das Blockpraktikum (Forts.)

- 12 WeeChat um Jabber-Plugin erweitern, aufbohren (Florian S., Simon)
  - bestehendes Python-Plugin/Bibliothek, taugt nix
  - GPG- oder Omemo-Unterstützung
  - weitere Funktionalität
- 13 Test-Runner für BFT-Systeme (Michael E.)
  - byzantinische Fehlertoleranz, Konsensfindung zwischen mehreren Parteien
  - zu testende Software auf Rechner verteilen, synchron starten
  - Aufzeichnung von Performance-Daten vor/während/nach Ausführung
  - mehrere Software-Varianten/Konfigurationen testbar

# Themen für das Blockpraktikum (Forts.)

- 14 Verbesserungen der Kaffeekasse (Simon, Florian S.)
  - XMPP-Bot für Kontostand, Überweisungen
  - Authentifizierung und Verschlüsselung für den Bot
  - Uhr, Animation, Nachrichten auf Wartebildschirm
- 15 Dateisystem in OOSTuBs (Sebastian M., Bernhard, Volkmar)
  - POSIX-Interface zum System/Anwendungen anbieten
  - Datenformat zB. Minix oder FAT auf RAMdisk
  - IDE-Festplatten-Unterstützung

# Themen für das Blockpraktikum (Forts.)

- 16 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)
  - Aufbau auf bestehende Vorarbeiten
- 17 Postfix-Transport für synchrone SMTP-Dialoge (arw)
  - Serverkommunikation  $A \rightarrow B \rightarrow C$
  - B reicht jeden Befehl sofort an C weiter, ggf. Ablehnung dann sofort an A
  - reduziert Annahme von Spam

- 18 „Offline“-Löser für Paketabhängigkeiten (arw)
  - dpkg/apt entscheidet nichtdeterministisch bei jedem Aufruf über zu installierende Pakete
  - Admin hätte gerne Determinismus, identische Installationen im ganzen Netz, Caching
  - Erweiterung von bestehendem Tool pkgsync und/oder apt



- 19 Dateisystem-Quota-Support in Linux-Desktop-Umgebungen (arw)
  - gebräuchliche graphische Dateimanager um Quota-Support erweitern
  - Quota-Systray-Applet
- 20 „SmartACLs“ (arw)
  - BPF-Skripte in xattr als Dateisystem-ACLs in Linux
  - „Zugriff Mo - Fr 0800-1700 von ungeraden PIDs erlaubt“

# Themen für das Blockpraktikum (Forts.)

- 21 PDF-Viewer Zathura, zeigt Annotationen (Simon, Christian E.)
  - Annotationen bearbeitbar, erstellbar machen
- 22 qutebrowser erweitern (Christian E., Simon)
  - Python
  - sinnvollen Passwort-Manager einbauen
  - besseren AdBlocker implementieren (momentan Hostnamen-basiert)

# Themen für das Blockpraktikum (Forts.)

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

# Eure Aufgabe

- 1 Themen-Kandidaten aussuchen
- 2 mit Betreuern reden (<https://www4.cs.fau.de/People/>)
- 3 bis Fr 2018-02-16: 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