

Praktikum angewandte Systemsoftwaretechnik (PASST)

Blockpraktikum

31. Januar 2019

Tobias Langer, Stefan Reif, Michael Eischer,
Bernhard Heinloth und Florian Schmaus

Lehrstuhl für Informatik 4
Friedrich-Alexander-Universität Erlangen-Nürnberg



Lehrstuhl für Verteilte Systeme
und Betriebssysteme



FRIEDRICH-ALEXANDER
UNIVERSITÄT
ERLANGEN-NÜRNBERG

TECHNISCHE FAKULTÄT

- Projektwahl und Gruppenbildung: 2–3er Gruppen
- Projektvorstellung
 - 10 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: **25.02.2019 – 08.03.2019**

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%

1. Linux mit LLVM/CLANG verheiraten
LTS-Linux-Kernel mittels der LLVM-Werkzeugkette unter Verwendung von Bindezeitoptimierung (*Link Time Optimization* übersetzen.)
 - Konfigurationsuntermenge identifizieren
 - Bauprozess anpassen



Bernhard

2. PASST Temperatursensor goes IoT „Smarte“ Temperaturerfassung

- Portierung des Temperatursensors auf ESP8266
- Entwicklung MQTT Schnittstelle



Tobias

3. Maßschneidung von ELF-Binärdateien mit DyninstAPI und/oder BOLT
Eliminierung von unbenötigtem Code in Binaries
 - Binary Rewriter zur Entfernung ungenutzter Funktionen nutzen & erweitern
 - Evaluation der Effektivität
4. Funktionstracing in Zephyr OS
Optimierung von Betriebssystemkonfigurationen
 - Entwicklung einer Tracing-Infrastruktur für Funktionsaufrufe



Andreas

5. Compilerbasierte Maßschneidung von Shared Libraries

Eliminierung von unbenötigtem Code bei der Kompilierung

- Entwicklung eines GCC/Clang-Plugins zur Entfernung von ungenutztem Code
- Evaluation der Effektivität



Andreas

6. Cacheanalyse

Erweiterung des Analysetools `platin`

- bisher nur **must**-Analyse für 2-Wege LRU-Caches
- Persistenzanalyse, andere Cacheorganisationen,
...



Phillip

7. Annotationsprache zur Spezifikation von Programmverhalten

Entwicklung einer Annotationsprache für `platin` basierend auf `Dhall`

- Einblick in Compilerbau
- Entwicklung in Ruby



Simon

8. Eigene Ideen und Vorschläge

Fragen?