

Praktikum angewandte Systemsoftwaretechnik

Blockpraktikum

Alexander Würstlein, Moritz Strübe, Sebastian Maier

Lehrstuhl Informatik 4

Januar 2015

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: 2015-03-16 P 2015-03-28
- Beginn: Mo 2015-03-16 10:15 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 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!
- 2 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

- ③ Portierung von eCos auf neue Hardwareplattform (Florian)
 - spezielle ARM-Variante
- ④ USB-over-IP (Morty, 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
 - (virtuelle) Energieverbrauchsmessung
 - SATA-Simulation
 - Injektion von CPU-Register-Fehlern
 - SMP-Tests
 - ...
- 6 Verfassen von Test- und Demoprogrammen für die Zuse Z23 (Volkmar, Klaus)
 - interessante, sehr alte Hardwarearchitektur
 - Finden von Hardwareproblemen
 - Demonstration von Funktionen

Themen für das Blockpraktikum (Forts.)

- 7 Verteilte symbolische Ausführung zur Analyse der Worst-Case Energy Consumption (Peter W., Christopher)
 - Verwendung der parallelen symbolischen Ausführungsmaschine Cloud9
 - Ausführung auf Amazon-EC2-Instanzen
 - Evaluationsszenario: Mälardalen WCET-Benchmarks
 - Vollständige symbolische Ausführung möglich?
 - Wie viel(e) Rechenzeit/Instanzen notwendig?
 - Integration eines SMT-Multi-Solvers (metaSMT, KLEE Multi-Solver)
- 8 Portierung der Echtzeitregelungsanwendung des I4Copters auf Erika/OSEK (Peter Ulbrich)
 - Erika/OSEK auf EasyRun Eval-Board portieren
 - schrittweise Portierung der Anwendungskomponenten
 - Funktionstest

- 9 Integration der I4Copter-Software in automatische Build- und Testumgebung (Peter Ulbrich)
 - Entwickeln von Testfällen
 - Versionsverwaltung auf Git umstellen und in Gerrit einbinden
 - automatische Builds und Tests mit Jenkins
- 10 Matlab-Frontend für RTSC (Peter Ulbrich)
 - Real-Time Systems Compiler
 - Matlab-Beschreibung einlesen
 - Zeitablauf der Signalverarbeitungsketten modellieren
 - Scheduling planen

Themen für das Blockpraktikum (Forts.)

- 11 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 Gerätename
- 12 Logic Analyzer auf PCI Express (arw)
 - bestehende PCI-Logic-Analyzer-Karte auf PCI Express portieren
 - evtl. weitere Features implementieren
- 13 Linux-Schnittstelle für das SDS 200A USB-Oszilloskop erweitern (morty)
 - günstiges USB-Oszi, was Linux-Unterstützung gebrauchen kann
 - reverse Engineering und erste Implementierung letztes Semester
 - erweitern, verbessern, verschönern

- 14 sicherer Bildschirmschoner und Konsolensperre für Linux (arw)
 - chvt im Kernel beschränken durch gezielte Sperrung von VTs
 - Freigabe durch privilegierten Prozess auf speziellem VT
 - Erreichbar durch „secure access key“
- 15 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

- 16 Reverse-Engineering des USB-Protokolls eines minipro 3 (Klaus)
 - Programmer für Cypress-Mikrocontroller
 - Windows-Treiber verfügbar
 - Linux-Treiber für OpenOCD gewünscht
- 17 Eigene Hardware bauen (arw)
- 18 Entwicklung eines Gerätetreibers
 - Ihr kennt/habt Hardware, die nicht unter Linux funktioniert?
 - Entwickelt einfach euren eigenen Treiber
- 19 **Eigene Ideen und Vorschläge**

Eure Aufgabe

- 1 Themen-Kandidaten aussuchen
- 2 mit Betreuern reden (<https://www4.cs.fau.de/People/>)
- 3 bis Do 2015-02-12: Thema aussuchen, Mail an i4passt@lists.cs.fau.de
- 4 mit Betreuer(n) Aufgabenstellung diskutieren
- 5 ins Thema einlesen
- 6 Blockpraktikum vorbereiten: Problemvorstellung, Lösungsansatz, erwartete Ergebnisse, Zeitplan
- 7 bis zum Praktikumsbeginn: Anfangspräsentation erstellen