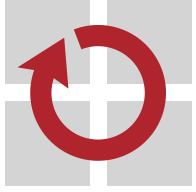
Praktikum angewandte Systemsoftwaretechnik (PASST)

Blockpraktikum

08. Februar 2021

Dustin Nguyen, Tobias Langer, Jonas Rabenstein, Phillip Raffeck

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



Lehrstuhl für Verteilte Systeme
und Betriebssysteme

FRIEDRICH-ALEXANDER
UNIVERSITÄT
ERLANGEN-NÜRNBERG

TECHNISCHE FAKULTÄT

Organisatorisches

- Projektwahl und Gruppenbildung: 2-3er Gruppen
- Projektvorstellung
  - 10 min. Präsentation im Plenum + 10 min. Diskussion
  - Vorstellung, 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: **15.03.2021 – 26.03.2021**

Zielsetzung (1/2)

Erfolg im Praktikum wird am Erreichen der Zielsetzungen gemessen:

- Gelerntes anwenden
- Selbständige Projektdurchführung im Team
- Entwicklungsprozesse in F(L)OSS-Projekten praktisch anwenden
  - Verwenden entsprechender Werkzeuge (git, Patche, ...)
  - Einbinden der Entwicklergemeinschaft (Features an Upstream)

Ziel

Benutzbare Software für euch, uns und den Rest der Welt

Zielsetzung (2/2)

Bewertet wird:

- Lösungsfindung und Umsetzung der Lösung
- Kollaboration zwischen euch
- (Kommunikation und Zusammenarbeit mit Upstream)
- Projekt wird veröffentlicht (Publish or it didn’t happen!)
  - Entsprechend: Repo mit Doku, Lizenz, …

Notenfindung (Wiederholung)

Teilnote	A1	A2	A3	A4	A5	Blockpraktikum
Gewichtung	1	1	1	1	1	7.5

- 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 (1/8)

Softwaremaßschneiderung: Docker-Containern

- Docker-Container als minimale Linux-Distros
  - Anwendung + benötigte Shared Libraries
  - **Problem**: Shared Libraries sind nicht vollumfänglich notwendig
- Framework zur Maßschneiderung vorhanden
  - Anwendung auf Docker-Container
  - Erweiterung und Anpassung wo notwendig
  - Analyse des erreichbaren Grads der Maßschneiderung

Andreas

Themen für das Blockpraktikum (2/8)

CrazyFlies 1: Externe Positionsbestimmung

- Triangulierung per Kamera
- WIFI AP plus WIFI Clients
- Kalibriermodus mittels OpenCV
- Tracking Modus: Objekterkennung im Client
- Tracking Modus: Sensorfusion im Server
- Zusätzlich: Synchronisationsansatz entwickeln
- Hardware: Raspi, ESP32-CAM, ESP32 NodeMCU

Themen für das Blockpraktikum (3/8)

CrazyFlies 2: Autonome Positionsbestimmung

- Positionsbestimmung vom System aus Objekterkennung (Marker)
- Kamerakalibrierung
- Sensorfusion mittels Kalman Filter
- Hardware: Webcam, Laptopkamera, Handy

Tim

Themen für das Blockpraktikum (4/8)

JITTY

- Hot-Plugging-Support
  - USB-Geräte zur Laufzeit erkennen
- Nutzung IO-MMU
  - Absichern von Geräte-DMA-Zugriffen
- SMP unter Raspberry Pi
  - Unterstützung für IPIs und Locks
- Dynamische Speicherverwaltung
  - mehr als 1GB Hauptspeicher
  - Ein-/Ausblenden physikalischer Speicher

Volkmar

Themen für das Blockpraktikum (5/8)

Virtueller Temperatursensor: Virtuelle Lehre

- Motivation: Einsteigerprojekt PASST-Temperatursensor
  - Temperatursensor als Lehrprojekt
  - Umsetzung als USB Kernelmodul
  - Äußere Umstände erschweren Präsenzlehre
- Virtueller Temperatursensor für virtuelle Lehre
  - Emulation des Temperatursensors
  - Kommunikation per USB-over-IP Server
  - Kerneltreiber sollte nach wie vor funktionieren

Gradual Typing in Großprojekten (Simon) (6/8)

Platin: nicht alles, was glänzt, ist ein Edelstein

- **Ruby**-Programm zur statischen Laufzeitanalyse
- Nutzung einiger „spannender“ Sprachkonstrukte:
  - Enten-Typisierung (engl. duck typing)
  - Spezialisierte Listenimplementierung mittels `module_eval`
- **Aber**: statische Typisierung ist sehr sinnvoll: Fehlersuche, Codevervollständigung, Typprüfung

⇝ Sorbet: ein externer gradueller Typprüfer für Ruby

- Benötigt Typannotationen für kritische Codestellen:
  - Von Hand: Manuelle Annotation :/
  - ggf. aus Programmmitschnitten (die zu erzeugen wären…)
  - ggf. mittels Ruby-Parser aus den Quelldateien (inkl. YARD)
- **Vorkenntnisse:** Rubykenntnisse sind erforderlich!

Gradual Typing in Großprojekten (Simon) (6/8)

Platin: nicht alles, was glänzt, ist ein Edelstein

- **Ruby**-Programm zur statischen Laufzeitanalyse
- Nutzung einiger „spannender“ Sprachkonstrukte:
  - Enten-Typisierung (engl. duck typing)
  - Spezialisierte Listenimplementierung mittels `module_eval`
- **Aber**: statische Typisierung ist sehr sinnvoll: Fehlersuche, Codevervollständigung, Typprüfung

⇝ ~~Sorbet: ein externer gradueller Typprüfer für Ruby~~

- Benötigt Typannotationen für kritische Codestellen:
  - Von Hand: Manuelle Annotation :/
  - ggf. aus Programmmitschnitten (die zu erzeugen wären…)
  - ggf. mittels Ruby-Parser aus den Quelldateien (inkl. YARD)
- **Vorkenntnisse:** Rubykenntnisse sind erforderlich!

Gradual Typing in Großprojekten (Simon) (6/8)

Platin: nicht alles, was glänzt, ist ein Edelstein

- **Ruby**-Programm zur statischen Laufzeitanalyse
- Nutzung einiger „spannender“ Sprachkonstrukte:
  - Enten-Typisierung (engl. duck typing)
  - Spezialisierte Listenimplementierung mittels `module_eval`
- **Aber**: statische Typisierung ist sehr sinnvoll: Fehlersuche, Codevervollständigung, Typprüfung

⇝ Ruby 3.0: neu: eingebauter graduelle Typprüfer für Ruby

- Benötigt Typannotationen für kritische Codestellen:
  - Von Hand: Manuelle Annotation :/
  - ggf. aus Programmmitschnitten (die zu erzeugen wären...)
  - ggf. mittels Ruby-Parser aus den Quelldateien (inkl. YARD)
- **Vorkenntnisse:** Rubykenntnisse sind erforderlich!

tt-eCos (Phillip) (7/8)

Zeitgesteuertes Betriebssystem . . .

- feste Ablauftabelle
- periodische Abarbeitung
- studentische Arbeit

. . . mit Problemen

- Todeslinienabfrage
- überlappende Einplanung von Aufgaben
- Übernahmeprüfung nichtperiodischer Last

Themen für das Blockpraktikum (8/8)

Eigene Ideen und Vorschläge

Themen für das Blockpraktikum

- Crazy Flies
- JITTY
- Maßschneiderung von Docker-Containern
- Virtueller Temperatursensor
- Gradual Typing
- tt-eCos

Fragen?