Praktikum angewandte Systemsoftwaretechnik

Blockpraktikum

Alexander Würstlein

Lehrstuhl Informatik 4

Februar 2016

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: 2016-03-07 P 2016-03-18
- Beginn: Mo 2016-03-07 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. 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
2. 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 (?!)

Themen für das Blockpraktikum (Forts.)

3. 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
     - ...
4. Portierung der Echtzeitregelungsanwendung des I4Copters auf Erika/OSEK (Florian, Tobias)
   - Erika/OSEK auf EasyRun Eval-Board portieren
   - schrittweise Portierung der Anwendungskomponenten
   - Funktionstest

Themen für das Blockpraktikum (Forts.)

5. Integration der I4Copter-Software in automatische Build- und Testumgebung (Florian, Tobias)
   - Entwickeln von Testfällen
   - Versionsverwaltung auf Git umstellen und in Gerrit einbinden
   - automatische Builds und Tests mit Jenkins
6. Implementierung eines Schedulers für Jobcluster zum automatisierten Testen (Florian, Tobias)
   - Verteilung von Testfällen auf heterogene Hardware
   - Überwachung und dynamische Anpassung des Jobablaufs
   - Resourcenplanung, -beschränkung und „einfrieren“ von Jobs

Themen für das Blockpraktikum (Forts.)

7. 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
8. 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

Themen für das Blockpraktikum (Forts.)

9. Logic Analyzer auf PCI Express (arw)
   - bestehende PCI-Logic-Analyzer-Karte auf PCI Express portieren
   - evtl. weitere Features implementieren
10. Neue Kompressionsalgorithmen im Linux-Kern (Rainer)
   - Brotli: verspricht ähnliche Laufzeit wie deflate, aber bessere Kompression
   - Zstd: targeting real-time compression scenarios
   - anwendbar dann z.B. in vmlinuz, initramfs, squashfs, zRAM

Themen für das Blockpraktikum (Forts.)

11. 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
12. 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?

Themen für das Blockpraktikum (Forts.)

13. Zeitstempelsignaturen in Git (arw)
    - kryptographisch gesicherte Zeitstempel nach RFC3161
    - analog zu GPG-signierten Tags
    - beweisen Existenz von Daten zu spätestens dem signierten Zeitpunkt: Übungsabgaben, andere Deadlines
14. Rechnersteuerung für Jura-Kaffeemaschinen (stettberger, Jens)
    - individualisierte Kaffeeprofile bei RFID-Tag an Kasse
    - Protokoll/Bibliothek zur Ansteuerung und Kaffeekasse vorhanden
    - Integrieren, Testen

Themen für das Blockpraktikum (Forts.)

15. Änderungserkennung in Sourcecode per Hashes auf AST-Ebene (stettberger)
    - Buildsysteme erkennen Änderungen an Zeitstempeln
    - gemeinsames `#include <config.h>` erzeugt unnötige Rebuilds
    - besser: Beachtung der Verwendung und tatsächlichen Abhängigkeiten
16. Zuverlässiges redundantes Monitoring (arw)
    - typische Systeme wie Nagios überwachen Rechnerbetrieb, bei Ausfall des Monitoringsystems Alarm
    - Monitoring nicht verfügbar bis repariert, Fehlermeldungen gehen verloren: nicht ausreichend, wenn Fehlermeldungen kritisch sind
    - Entwurf und geeignete Implementierung, Test

Themen für das Blockpraktikum (Forts.)

17. Mac-Book-Powermanagement unter Linux fixen (Valentin)
    - suspend geht nicht
    - hibernate geht nicht
18. Bugs im Kernel finden und fixen (Valentin)
    - Bugs aus undertaker
    - Bugs aus coccinelle
    - Bugs aus checkkconfigsymbols

Themen für das Blockpraktikum (Forts.)

19. Eigene Hardware bauen
20. Entwicklung eines Gerätetreibers
    - Ihr kennt/habt Hardware, die nicht unter Linux funktioniert?
    - Entwickelt einfach euren eigenen Treiber
21. Eigene Ideen und Vorschläge

Eure Aufgabe

1. Themen-Kandidaten aussuchen
2. mit Betreuern reden (`https://www4.cs.fau.de/People/`)
3. bis Di 2016-02-16: Thema aussuchen, Mail an `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