Praktikum angewandte Systemsoftwaretechnik

Blockpraktikum

Alexander Würstlein, Moritz Strübe, Rainer Müller

Lehrstuhl Informatik 4

Januar, 2014

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: 2014-03-05 P 2014-03-18
- Beginn: (Ascher-)Mi 2014-03-05 10:00 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 Ablauftabelle ab
   - Erweiterung: eine Ablauftabelle pro Prozessor $\rightarrow$ 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

Themen für das Blockpraktikum (Forts.)

3. Debugging von AVR-Mikrocontrollern (AVaRICE) (Morty)
   - AVR-Debugging unter Linux ist langsam
   - verbessern durch Reverse-Engineering der USB-Kommunikation des Debuggers unter Windows
4. Optimierung des Cache-Verhaltens von Linux (Morty, arw)
   - Implementierung von `MADV_DONTNEED` in `madvise(2)`
   - verhindert, dass einmaliges Lesen z. B. durch Backupsoftware den Plattencache überschreibt
5. USB-over-IP (Morty)
   - 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.)

6. 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
7. Logic Analyzer auf PCI Express (arw)
   - bestehende PCI-Logic-Analyzer-Karte auf PCI Express portieren
   - evtl. weitere Features implementieren

Themen für das Blockpraktikum (Forts.)

8. basteln mit Galileo-Boards (Gabor, arw)
   - x86-basierter Arduino-Ersatz von Intel (10 Stück spendiert)
   - eigene Projekte
9. Messungen der Echtzeit-Eigenschaften der Galileo-Boards (Florian, Tobias)
   - benutzt Linux mit einer Laufzeitumgebung: Jitter, Interrupt-Latenz, etc messen
   - evtl. Vergleiche mit anderen Systemen, Code ohne Arduino-Laufzeitumgebung
   - Benchmarks der CPU-Leistung, Speicherdurchsatz
10. Linux-Schnittstelle für das SDS 200A USB-Oszilloskop (morty)
    - günstiges USB-Oszi, was Linux-Unterstützung gebrauchen kann

Themen für das Blockpraktikum (Forts.)

11. Konsistenz- und Paritätsprüfung bei Lesezugriffen auf Linux MD-RAID (Daniel Danner, arw)
    - traditionellerweise wird die Prüfsumme oder zweite Kopie nur beim Schreiben und im Fehlerfall verwendet
    - für erhöhte Konsistenzanforderungen wäre aber eine ständige Konsistenzprüfung wünschenswert
12. Visualisierung von Abhängigkeiten in Gerrit Code-Review (Daniel Danner, arw)
    - Abhängigkeiten zwischen Patches, die eine erforderliche Reihenfolge haben
    - Graph oder topologische Sortierung würde Merges erleichtern

Themen für das Blockpraktikum (Forts.)

13. paralleles rsync für lokale Dateien (arw)
    - ermöglicht dem Betriebssystem besseres I/O-Scheduling, den Platten TCQ
    - würde den Fall „viele kleien Dateien“ stark beschleunigen
14. 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

Themen für das Blockpraktikum (Forts.)

15. Eigene Hardware bauen (arw)
16. Entwicklung eines Gerätetreibers
    - Ihr kennt/habt Hardware, die nicht unter Linux funktioniert?
    - Entwickelt einfach euren eigenen Treiber
17. Eigene Ideen und Vorschläge

Eure Aufgabe

1. Themen-Kandidaten aussuchen
2. mit Betreuern reden (`https://www4.cs.fau.de/People/`)
3. bis zum Semesterende: Thema aussuchen
4. mit Betreuer(n) Aufgabenstellung diskutieren
5. ins Thema einlesen
6. Blockpraktikum vorbereiten: Problemvorstellung, Lösungsansatz, erwartete Ergebnisse, Zeitplan
7. bis zum Praktikumsbeginn: Anfangspräsentation erstellen