

Betriebssysteme Übungen

Barry Linnert

Wintersemester 2017/18

„Was du mir sagst, vergesse ich.
Was du mir zeigst, daran erinnere ich mich.
Was du mich tun lässt, das verstehe ich.“

Konfuzius, chinesischer Philosoph, 551–479 v. Chr.

- Ziel: Wir bauen unser eigenes Betriebssystem!
 - Entwicklungsprozess erleben
 - ohne Black-Box-Bestandteile
 - schrittweises Vorgehen mit Erfolgserlebnissen

- Bauen eines neuen, kleinen Betriebssystems
 - Start von Null
 - vollständige Kontrolle
 - beliebige Modularisierung
 - nachvollziehbarer Entwicklungsprozess
 - kaum Vorgaben/Einschränkungen, dafür erhöhte Komplexität
- Begrenzung der Komplexität
 - Konzentration auf grundlegende Konzepte
 - Verzicht auf Optimierungen und spezialisierte Algorithmen
 - Nutzung einer relativ einfachen Zielplattform
 - roter Faden gegeben durch Aufgaben, schrittweise Steigerung der Komplexität
 - jeder Zwischenschritt hat eine erkennbare neue Funktionalität

- QEMU mit ARM-Prozessor
 - www.qemu.org
 - toolchain für Linux wird gestellt
 - lauffähig im Linux-Pool
- Abgaben müssen mittels toolchain im Linux-Pool laufen
- Abgabe über KVV
- schrittweises Realisieren der wesentlichen Funktionen
- Aufgaben bauen aufeinander auf
- Code wird nach Abschluss einer Aufgabe gestellt
- eigene Entscheidung: Benutzen des eigenen oder des gestellten Codes für weitere Aufgaben

- Bearbeitung und Abgabe allein oder in Zweiergruppe
- Gruppen können jederzeit verändert werden
- keine „Kooperation“ mit anderen Gruppen
- Übernahme von fremden Code ist unzulässig!
- keinen Code veröffentlichen (KVV, github, ...)

1. Serielle Schnittstelle
2. Prozessormodi, Stacks und Ausnahmebehandlung
3. Interrupts; Timer
4. Kontextwechsel, einfaches Scheduling von Prozessen
5. Systemrufe, Gerätetreiber
6. Speichervirtualisierung mit Memory Management Unit
7. Dynamik und Speicherschutz
(Änderungen vorbehalten)

- C-Kenntnisse werden vorausgesetzt
 - C ist nicht Java!
 - keine Nachhilfe

- Gerät heißt »taskit Portux MiniPC«
 - ist eingebettetes System als »system on a chip«
 - SoC enthält neben Prozessorkern fast alle funktionalen Einheiten
 - enthält zusätzlich Speicher, Stromversorgung, Schnittstellen
- Ausstattung
 - Prozessorarchitektur: ARM,
Befehlssatz: ARMv4T,
Prozessorfamilie: ARM9,
Prozessorkern: ARM920T,
Prozessor: AT91RM9200
 - 16 MiB Flash, 64 MiB Arbeitsspeicher
 - Ethernet 100 MBit/s, 4seriell, USB Host und Target
 - Bootloader U-Boot im Flash installiert
 - 3 farbige Leuchtdioden, Textdisplay mit 4 * 16 Zeichen
 - stabile Reset-Taste

