

ProgOS - Vorbesprechung

Roland Kammerer, Benedikt Huber

Institut für Technische Informatik
Technische Universität Wien

6. März 2012

Überblick

1. Lehrziele
2. Pintos
3. Modus der LVA
4. Projekte
5. Zeitlicher Überblick
6. Unterlagen/Source Code
7. TODOs für die LVA/Ausblick

Teil I

Lehrziele

Lehrziele der LVA

- ▶ Einblick in die praktische Umsetzung von Betriebssystemkonzepten
- ▶ Praktische Fertigkeiten in der Implementierung von Betriebssystemkomponenten
- ▶ Verständnis für Aspekte und Probleme unterhalb der Abstraktionsebene von Hochsprachen
- ▶ Verständnis der Betriebssystemschicht (zwischen HW (z.B.: Microcontroller) und Low-level Userspace SW (z.B.: Betriebssysteme UE))

Teil II

Pintos

Warum Pintos?

- ▶ Saubere Implementierung
- ▶ Source Code vorhanden
- ▶ Hervorragende Dokumentation (auch im Source)
- ▶ Überschaubare Projektgröße
- ▶ Ausreichender Designspace bei Projekten
- ▶ Ausführung und Debugging mit Hilfe von Emulatoren (prinzipiell auf x86 startbar)
- ▶ Wird erfolgreich in der Lehre eingesetzt (z.B.: Stanford)

Pintos Prinzipien

- ▶ Zuerst lesen, dann Programmieren
 - ▶ Dokumentation ist vorhanden
 - ▶ Jede Zeile Code ist schwer verdient (it's kernel programming)
- ▶ Maximaler kreativer Freiraum
 - ▶ *Anforderungen* werden spezifiziert
 - ▶ Implementierung / Weg zur Implementierung nicht vorgeschrieben
- ▶ Entwicklung ist Test-driven
 - ▶ Testfälle sind bekannt und machen Anforderungen konkret
 - ▶ Studierende können leicht eigene Testfälle hinzufügen

Stanford CS 140 vs. TU ProgOS

- ▶ Wesentlich kleinere Untermenge an Projekten (Stanford vs. TU):
 - ▶ Userspace bis auf Teile von Projekt 0 implementiert
 - ▶ Projekt 1 (Threads/Scheduling): alarm clock als Projekt 0, kein mlfqs scheduler
 - ▶ Projekt 2 (User Programs): Nur argument parsing und stack setup als Projekt 0
 - ▶ Projekt 3 (Virtual Memory): Kein Swapping (kein frame table / swap table)
 - ▶ Projekt 4 (File Systems): Gestrichen

Gemeinsamkeiten:

These projects are hard. CS 140 has a reputation of taking a lot of time, and deservedly so. We will do what we can to reduce the workload, such as providing a lot of support material, but there is plenty of hard work that needs to be done. - Stanford CS 140

Teil III

Modus der LVA

TISS, myTI, TI-Lab

- ▶ Sie: Anmeldung im TISS
- ▶ Wir: TISS-Daten werden automatisiert in myTI übernommen
- ▶ Sie: Gruppenfindung (3 Personen) im myTI
- ▶ Sie: Account in betreuter Übungszeit im TI-Lab abholen
- ▶ Zutrittskarte für TI-Lab abholen falls nicht vorhanden
 - ▶ TI Studierende haben/bekommen permanente Karten (Foto-Upload im myTI notwendig)
 - ▶ Alle anderen bekommen einen Leihkarte

Aufgaben der Gruppe/Gruppenmitglieder

- ▶ Gruppen aus 3 Personen
- ▶ 3 Projekte
 - ▶ gemeinsames Designdokument pro Aufgabe
 - ▶ *Verpflichtendes Analysegespräch* für jedes Projekt
 - ▶ Finales Designdokument pro Aufgabe
 - ▶ Abgabe eines git-tar-balls pro Aufgabe (pro Gruppenmitglied)
- ▶ Abgabegespräch am Ende des Semesters (Review der Lösungen und der Designdokumente)

Arbeiten an Projekten

- ▶ Self-hosted git Repository im TI-Lab (entsprechendes Skript wird von uns bereit gestellt)
- ▶ Persönlicher Tutor pro Gruppe
 - ▶ Ansprechperson für die Gruppe
 - ▶ Besprechung des initialen Desingdokuments (nach zirka 2 Wochen)
- ▶ Bei Fragen:
 - ▶ www.informatik-forum.at (Vorteil: Fragen/Antworten sind für alle Teilnehmer an zentraler Stelle beantwortet)
 - ▶ Tutor
 - ▶ Bei organisatorischen Fragen:
progos-team@vmars.tuwien.ac.at

Benotung

	Implementierung	Designdokument
Projekt 0	10%	10%
Projekt 1	30%	10%
Projekt 2	30%	10%

- ▶ Implementierung: Requirements müssen erfüllt sein + automatisierte (bekannte) Tests

Teil IV

Projekte

Zu implementierende Projekte

- ▶ Projekt 0 (Intro): alarm clock, commandline parsing/stack setup
- ▶ Projekt 1 (Scheduling): Priority Scheduler
- ▶ Projekt 2 (Virtual Memory Management): Paging (ohne Swapping)

Teil V

Zeitlicher Überblick

Zeitlicher Ablauf der LVA

Time Table

Vorträge	
6. März	Vorbesprechung
13. März	Einführung in Pintos
Übung	
12. März	Übungsbeginn
20. April	Abgabe Projekt 0
18. Mai	Abgabe Projekt 1
22. Juni	Abgabe Projekt 2
25/26. Juni	Abgabegespräche

- ▶ Bitte aktuelle Informationen auf der LVA - Homepage beachten (<http://ti.tuwien.ac.at/rts/teaching/courses/progos>)

Teil VI

Virtuelle Maschine und Source Code

- ▶ Tools für die LVA (compiler, qemu, patched bochs, git,...)
- ▶ Weitere Programme müssen selbst installiert werden (z.B.: kein default Editor installiert)
- ▶ <http://pan.vmars.tuwien.ac.at/progos>
- ▶ **Bitte** torrent file verwenden (hat eingebauten Webseed)
 - ▶ VM ist aber auch als direkter Download verfügbar (eingeschränkte Download Geschwindigkeit)
 - ▶ Sonst mit USB-Stick zu Tutor (\approx 900 MB)
- ▶ Username/Passwort in der VM: progos/progos
- ▶ Datenaustausch zwischen Host/Guest per SSH/SCP
- ▶ Tastaturlayout in der VM:
 - ▶ Ixkeymap ist installiert (Einstellungen werden nicht gespeichert)
 - ▶ `echo „setxkbmap -variant nodeadkeys de“ >> ~/.profile`

Source Code zur LVA

- ▶ Projekte werden mit git verwaltet⇒

1. Anmeldung im TISS
 2. Gruppenfindung via myTI
 3. Ein Gruppenmitglied hosted das Repository im TI-LAB
(entsprechendes Skript wird zur Verfügung gestellt)
 4. Gruppenmitglieder haben Zugriff auf den Source Code
- ▶ Tar-ball ist im myTI verfügbar
 - ▶ Limitierte Anzahl an Arbeitsplätzen (mit allen Tools) im TI-Lab vorhanden

Teil VII

TODOs für Studierende

Vorraussetzungen

- ▶ **Vorraussetzungen für die LVA**
 - ▶ Solide Kenntnisse der Programmiersprache C
 - ▶ Command Line + Tools (z.B.: ssh/scp, gcc, make,...)
 - ▶ git
 - ▶ bochs
 - ▶ qemu
 - ▶ gdb/debugging: Sehr wichtig!

Referenzen und Ausblick

▶ Referenzen

▶ git:

- ▶ <http://progit.org/>
- ▶ <http://whygitisbetterthanx.com/>

▶ gdb:

- ▶ <http://www.gnu.org/software/gdb/documentation/>

▶ progos/pintos:

- ▶ <http://pan.vmars.tuwien.ac.at/progos/doc/progos>

▶ Ausblick auf nächsten Vortrag:

- ▶ Getting Started
- ▶ Tour durch Pintos