

Exzerpt zu: BUCH

Linux-Kernel-Handbuch:

Leitfaden zu Design und Implementierung von Kernel 2.6

1. Preview:

- * Vorstellung und Einstieg in den Kernel
- * Prozessmanagment/Scheduling
- * Systemcalls, Interrupts und Interrupt-Handler
- * Kernel-Synchronisierung, Methoden der Kernel-Synchronisierung
- * Timer und Zeit-Management
- * Memory-Management
- * Das virtuelle File-System, der Block-I/O-Lager
- * Der Process Address Space (Prozess-Adressraum)
- * Der Page-Cache und Page-Writeback
- * Module
- * kobjects und sysfs
- * Debugging
- * Portabilität
- * Patches, Hacking und die Community

2. Question:

- * Wie sieht die Implementierung und das Design einzelner Systeme aus?

3. Read:

4. Reflect:

- * design der Module als runtime linkable
 - > schnell wie monolit Kernel aber trotzdem modular
- * VFS eine zentrale Schnittstelle
 - > Details bleiben versteckt
 - > einfacher Einbau neuer Dateisysteme
- * I/O
 - > mehrere Moeglichkeiten

5. Recite: kurze Inhaltsangabe

Nach dem Vorwort beschreibt der Autor die Geschichte des Kernels und vergleicht den linux kernel mit anderen unix-artigen und geht kurz auf die Entwicklergemeinschaft ein. Installation wird obligatorisch behandelt, Besonderheiten wie inline asm kleiner Statck, Synchronisation werden kurz besprochen

Im prozessmanagment wird auf das design und die Implementierung eingegangen (Virtualisierung von cpu und speicher) das Konzept des cpu-scheduler wird vorgestellt. Die wichtigste Schnittstelle die syscalls werden ausfuerhlich beschrieben. Desweiteren wird ueber das Locking und preempting design gesprochen.

Methoden zur Sychnronisierung -- wie atomare Operationen und spinlock, semaphore. Auf System Timer wird eingegangen.

Design des memory-management. Beziehung zwischen Caches, Slabs, ... Slaballocator.

Die Vereinheitlichte Schnittstelle VFS fuer alle Dateisysteme. Eine Abstraktionsschicht fuer alle. Vorgestellt werden Design und Implementierung (verschiedene Objekte und deren Zusammenhang)

I/O scheduler Funktionsweise der implementierten disk-scheduler und deren Verwendung

Adressraum: kernel-threads, pagetable

Implementierung und Desing der Kernel-Module. RuntimeLinking, parameter.

6. Review: Abschließende Beurteilung

Von einem kernel Insider geschrieben bietet das Buch ein sehr gutes Nachschlagewerk. Sehr praktisch erklart um einfachen Zugang zum Kernel zu erhalten