

# Lösungen von Übungsblatt 2

## Betriebs- & Kommunikationssysteme

Tutorium: Cleve, Jonas; Freitag; 8:00 am,10:00 am

Boyan Hristov

12. Mai 2016

## Begriffe

**Protection Rings** Das sind die von der Betriebssystem konfigurierte Sicherheitsschichten, damit jedes Programm nur eine begrenzte Menge von Operationen ausführen kann und kein Zugang zu Operationen von unterste Niveau hat (kein Zugriff zu Hardware). So werden diese gefährliche Operationen nur mit Hilfe von Schnittstellen (Syscalls) ausgeführt. Ein Programm von Ring  $n$  hat nur Zugriff zu Operationen von den selben Ring oder von äußeren. Ring 0 ist das Kernel und Hardwaretreiber, 1 ist für allgemeinere Betriebssystemoperationen und 3 ist für Anwendungssoftware. Zwei ist für Anwendungssoftware mit „extended permissions“.

**Server in Server-Client Architektur** Der Server ist ein Programm, der anhand von Abfragen (Requests) unterschiedliche Dienste für das Client ausführt. Das wird geschafft durch Kommunikation zwischen beide. Der Server muss nicht unbedingt ein anderer Rechner sein, es geht in dem Begriff um das Prinzip von Fragen (Client) und Beantworten (Server).

## Implementierungen von Systemaufrufen

### I Subroutine call in Betriebssystem

Das wird in schlichten Betriebssysteme benutzt, wobei keine Protection Rings existieren und jedes Programm in Ring 0 ausgeführt wird. Hier gibt es keine Hardwaresicherheit, die Systemaufrufe werden so implementiert, dass es nur ein Sprung zu andere Adresse gibt und nichts weiteres. (Microkernel)

### II System call

Diese werden durch Interrupts implementiert, wobei das Kern analysiert, was das Grund dieses Interrupts war, führt die Operation aus und springt zurück dann zur User-Mode via return. (monolitische Kernel)

### III System call 2

Eine Sprünge nach Ring 0 wird realisiert, wobei aber ein anderer Prozess ausgeführt wird und dann wieder ins Programm gesprungen wird via return. (Microkernel)

### IV System call 3

Ähnlich zu Server-Client Architektur wird hier eine Aufgabe zu korrespondierenden Systemprozess geschickt, verarbeitet in Kernel und zurückgegeben in User space, wieder mit Prozesswechsel. (Microkernel)

## Tanenbaum–Torvalds-Debatte

Das Thema der Debatte war monolitische Kernen gegen Microkernel. Linus Torvalds, als Gründer von Linux, was auf monolithischen Kernel basiert ist, meinte, dass MINIX, die microkernelbasierte Betriebssystem von Andrew Tanenbaum, vererbte Designfehler hatte und dass Microkernel nur im Prinzip besser sind, aber nie gut realisiert worden. Andrew meinte, dass monolitische Kernel stärkere Rechner brauche, wegen ihrer größeren

Operationsanzahl und viel unsicherer sind, wegen ihres komplizierteren Models. Tanenbaum erklärte weiter, dass deswegen Mikrokernel weniger systemabhängig sind und deswegen in der Zukunft mehr verwendbar sind.

Ich bin mit Andrew einverstanden, da die Technologien zu letzter Zeit sich schon ziemlich viel entwickelt haben und wir sowieso fast nie die volle Kraft unseren Rechners nutzen. Heutzutage ist die Anzahl von unterschiedliche Geräte so groß, dass eine flexiblere Betriebssystem toleriert werden soll. Natürlich sind Mikrokernel nicht überall besser, aber ich bin sowieso kein großer Fan von Linux :D.