

# **Konzepte von Betriebssystem-Komponenten**

## **Microkernel-Betriebssysteme Mach, L4, Hurd**

**14. November 2002**

**Holger Ruckdeschel**

**[holger.ruckdeschel@informatik.stud.uni-erlangen.de](mailto:holger.ruckdeschel@informatik.stud.uni-erlangen.de)**

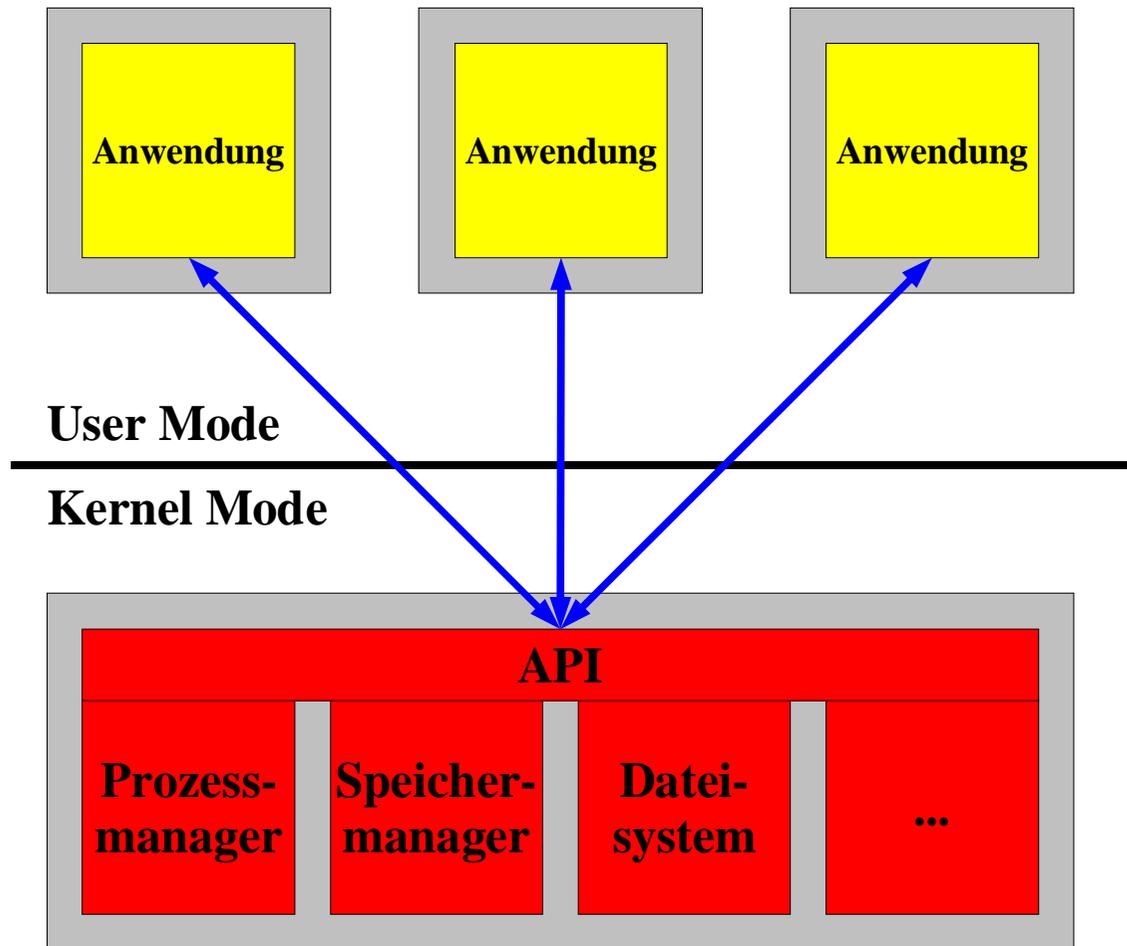
# Übersicht

- **Gegenüberstellung: Monolithischer Kernel - Microkernel**
- **Der Mach Microkernel**
- **Der L4 Microkernel**
- **GNU Hurd**
- **Geschwindigkeitsvergleich: Mach - L4**

# Monolithischer Kernel

- **Klassischer Ansatz (Unix)**
- **Alle BS-Komponenten in Kern integriert**
- **Gemeinsamer Adressraum**

# Monolithischer Kernel



# Monolithischer Kernel

## ■ Vorteile

- effiziente Kommunikation im Kern
- Hardwarezugriffe jederzeit und unmittelbar durchführbar

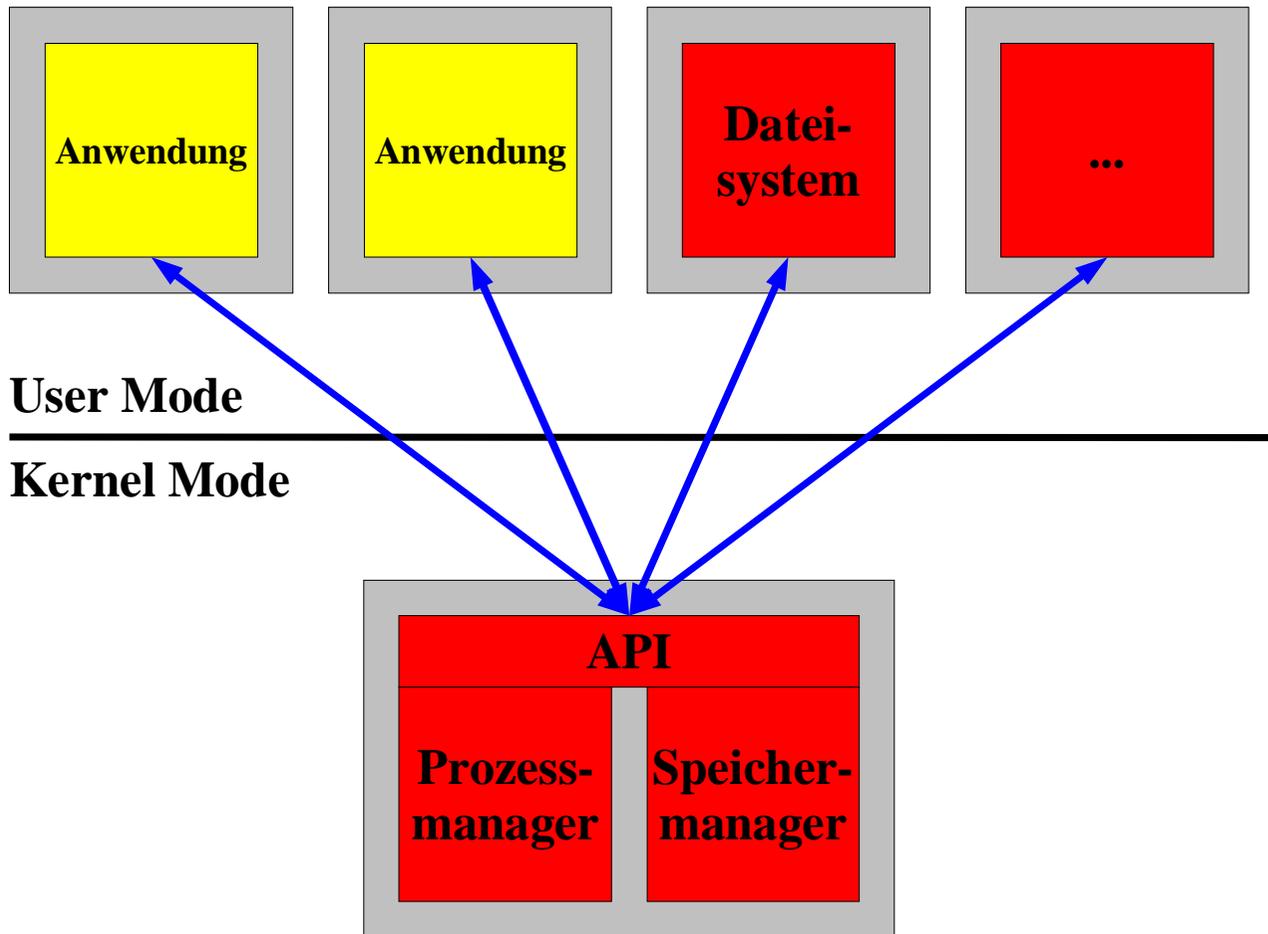
## ■ Nachteile

- kein Zugriffsschutz zwischen den Komponenten
- undurchsichtige Struktur

# Microkernel

- **Auslagerung von BS-Komponenten in Server**
- **BS-Komponenten in getrennten Speicherbereichen**
- **Kernel bietet lediglich**
  - einfaches Prozessmanagement
  - einfaches Speichermanagement
  - effiziente Wege zur Inter-Prozess-Kommunikation (IPC)

# Microkernel



# Microkernel

## ■ Vorteile

- Schutz der BS-Komponenten voreinander
- hohe Flexibilität

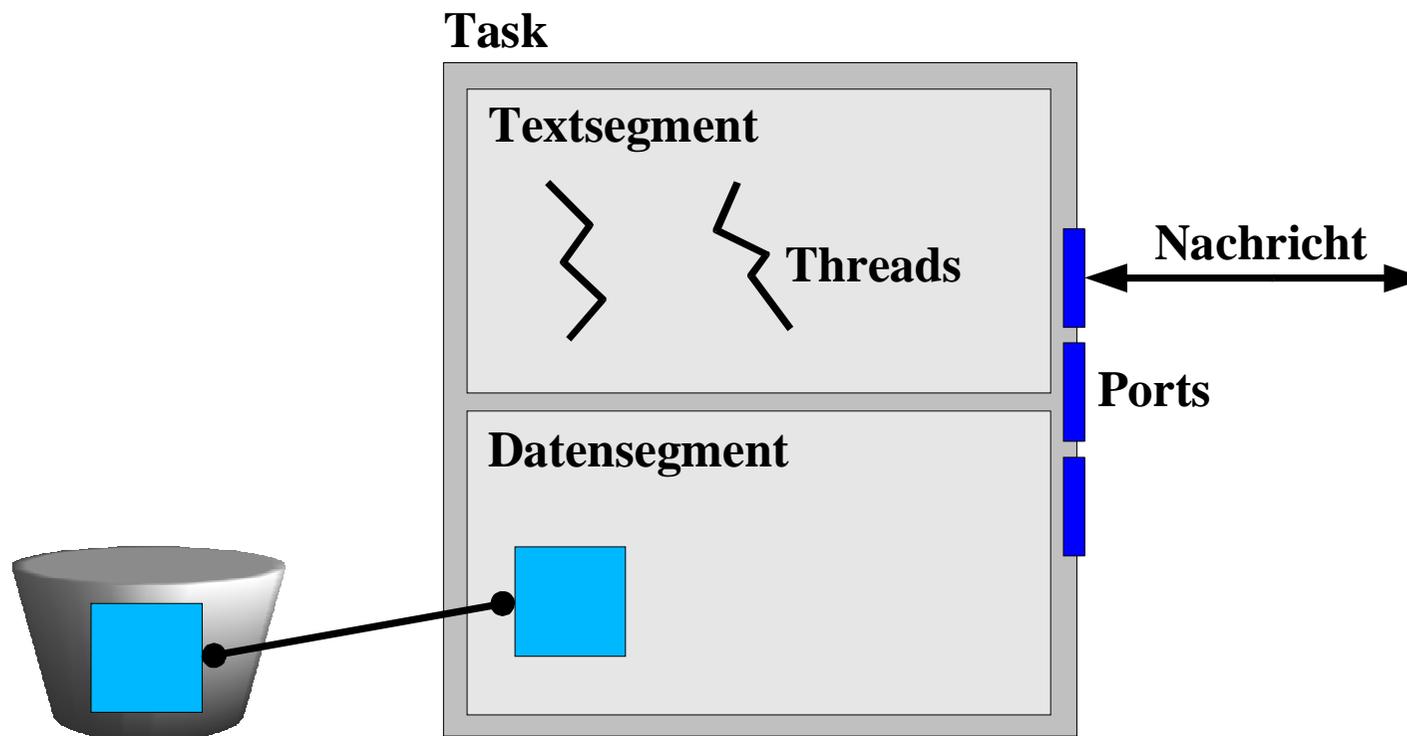
## ■ Nachteile

- Kommunikation ist teuer

# Mach

- **Ausgangspunkt 4.2BSD**
- **Schrittweiser Tausch von BSD- gegen Mach-Code**
- ➔ **Mach Release 2**
  - **4.3BSD kompatibel**
  - **monolithisch**
- **Auslagerung von BSD-Komponenten in Server**
- ➔ **Mach Release 3**
  - **Unix-frei**
  - **Microkernel**
  - **Geräteschnittstelle**

# Mach - Konzepte



nach: Silberschatz

# L4

- **Von Anfang an als Microkernel entworfen**
- **Erste Version: L4/x86**
- **Heute: verschiedene Architekturen**
  - Alpha
  - MIPS
  - StrongARM
  - PowerPC
  - IA64

# L4 - Konzepte

- **Flexpages**
- **Virtuelle Adressräume**
  - bestehen aus Flexpages
  - Ein- und Ausblenden von Flexpages mit `grant`, `map`, `flush`
- **Threads**
- **Tasks**
- **Clans**
  - Gruppe von Tasks
  - mit privilegiertem Chief
- **kein eigenes I/O-System**

# L4 - I/O

- **Keine I/O-Schnittstelle im Kernel**
- **Gerätetreiber im User-Mode**
- **Einblenden der I/O-Adressräume in Tasks**
- **Weiterleitung von IRQs als IPC-Nachrichten**

# GNU Hurd

- **Microkernel-Betriebssystem auf Basis von GNU Mach**
- **Kernel enthält Gerätetreiber**
- **Server für**
  - **Dateisysteme (ext2, isofs u.a.)**
  - **Netzwerkprotokolle (IP, ...)**
  - **...**
- **spezielle Bibliotheken (z.B. libext2fs)**
- **spezielle Utilities (z.B. settrans)**

# GNU Hurd

## ■ Beispiel für Hurd-Systemkommando: Dateisystem mounten

```
settrans -c /mountpoint /hurd/ext2fs /dev/hd0s1
```

- ext2fs-Server läuft als normaler Prozess
- Jeder Benutzer kann eigene Server schreiben und starten

## ■ Zum Vergleich: Linux

```
mount /dev/hda1 /mountpoint -t ext2
```

- Befehl direkt an Kernel
- Kernel muss Dateisystem bereitstellen

# Geschwindigkeitsvergleich

## ■ Vergleich von

- **Linux (monolithisch)**
- **MkLinux (auf Mach aufsetzend)**
- **L<sup>4</sup>Linux (auf L4 aufsetzend)**

# Geschwindigkeitsvergleich

## ■ Low-Level Benchmark: `getpid()`-Aufruf

<b>Linux</b>	<b>1,68 <math>\mu</math>s</b>
<b>L4Linux</b>	<b>3,95 <math>\mu</math>s</b>
<b>MkLinux</b>	<b>110,60 <math>\mu</math>s</b>

## ■ High-Level Benchmark: Übersetzen des Linux-Servers

<b>Linux</b>	<b>476 s</b>
<b>L4Linux</b>	<b>506 s</b>
<b>MkLinux</b>	<b>605 s</b>