

# Die L4-Mikrokern-Familie

Hauptseminar Ansätze für Betriebssysteme der Zukunft

*Michael Steil*

18.04.2002

# Aufbau des Vortrags

1. Mikrokerne: Idee und Geschichte
2. L4: ein schneller Mikrokernel
3. L4Linux: Linux auf einem Mikrokernel
4. Weiterentwicklungen, Zukunft

# Geschichte der Kernarchitektur

- Monolithische Kerne
- Mikrokerne der ersten Generation
- Mikrokerne mit Kompromissen
- Mikrokerne der zweiten Generation

## Windows

Der schwere Ausnahmefehler OE ist an Adresse 0028:C0026822 in der VXD-Datei VFAT(01) + 0000798E aufgetreten. Die aktuelle Anwendung wird beendet.

- \* Drücken Sie eine beliebige Taste, um die Anwendung abubrechen.
- \* Drücken Sie erneut Strg+Alt+Entf, um den Computer neu zu starten. Nichtgespeicherte Daten gehen dabei verloren.

Weiter mit beliebiger Taste.

# Monolithische Kerne

- Alles zu einem Binary gebunden
- Wenig Struktur, wenig Schichtung, wenig Interfaces

System calls					Interrupts and Traps		
Terminal handling		Sockets	File naming	Map-ping	Page Faults	Signal handling	Process creation
Raw tty	Cooked tty	Network protocols	File systems	Virtual Memory		Process scheduling	Process dispatching
	Line disciplines	Routing	Buffer Cache	Page Cache			
Character Devices		Network device drivers	Disk device drivers				

4.4BSD Kern

# Vor- und Nachteile

- Vorteile
  - (Zunächst) einfache Entwicklung
  - Hohe Geschwindigkeit
- Nachteile
  - Mangelnde Robustheit
  - Mangelnde Modularität und somit schlechte Wartbarkeit
  - Mangelnde Flexibilität

# Mikrokerne

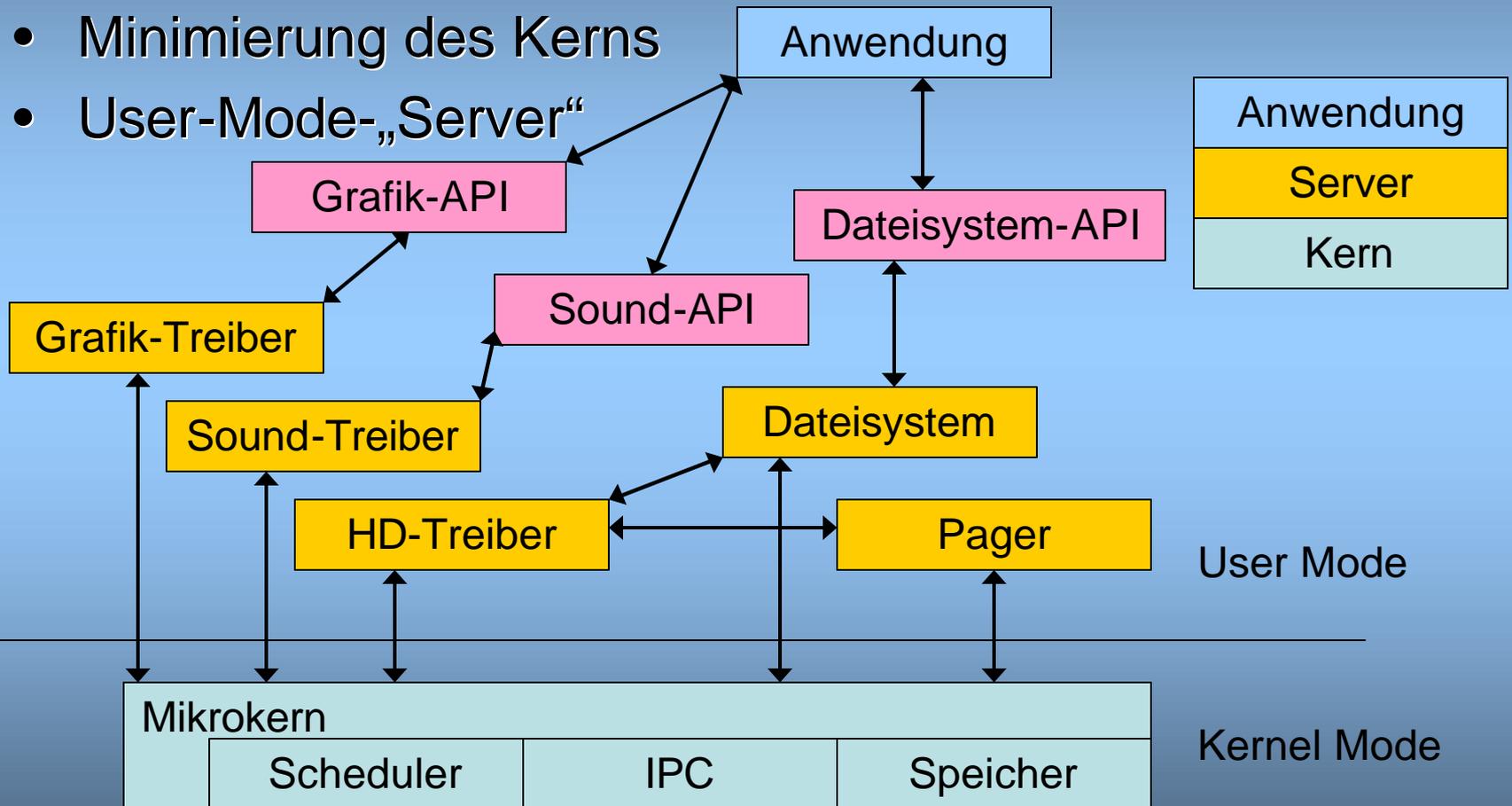
*„Traditionell wird das Wort ‚Kern‘ zur Bezeichnung desjenigen Teils des Betriebssystems verwendet, den alles andere an Software verpflichtend gemeinsam hat.*

*Die grundsätzliche Idee des Mikrokern-Ansatzes ist es, diesen Teil zu minimieren, d.h. was immer möglich ist außerhalb des Kerns zu implementieren.“*

J. Liedtke

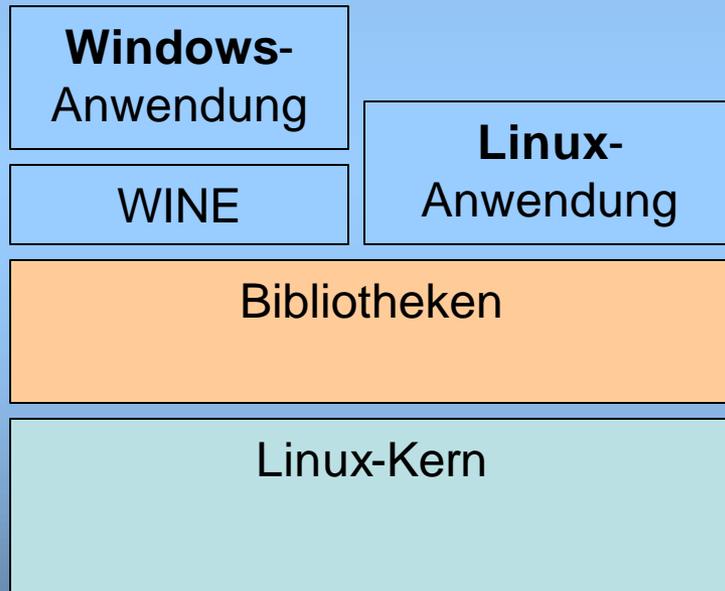
# Mikrokerne

- Minimierung des Kerns
- User-Mode-„Server“

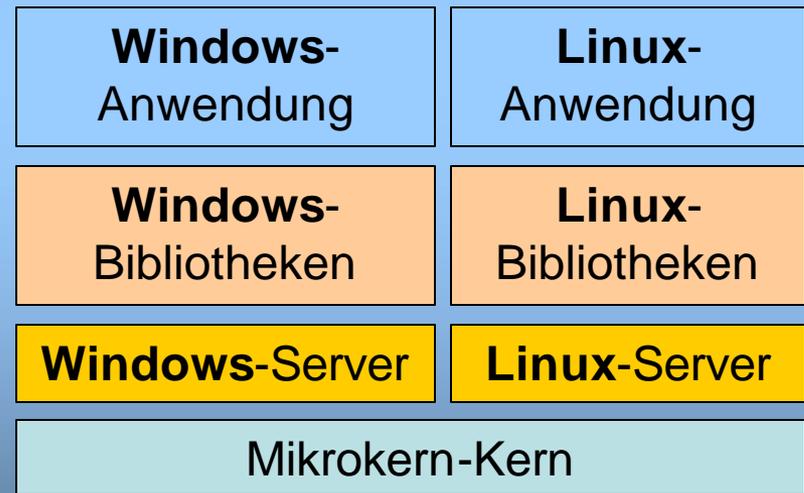


# OS Personalities

- Sammlung von Servern, die ein BS ausmachen
- Mehrere Personalities gleichzeitig möglich



Linux mit WINE



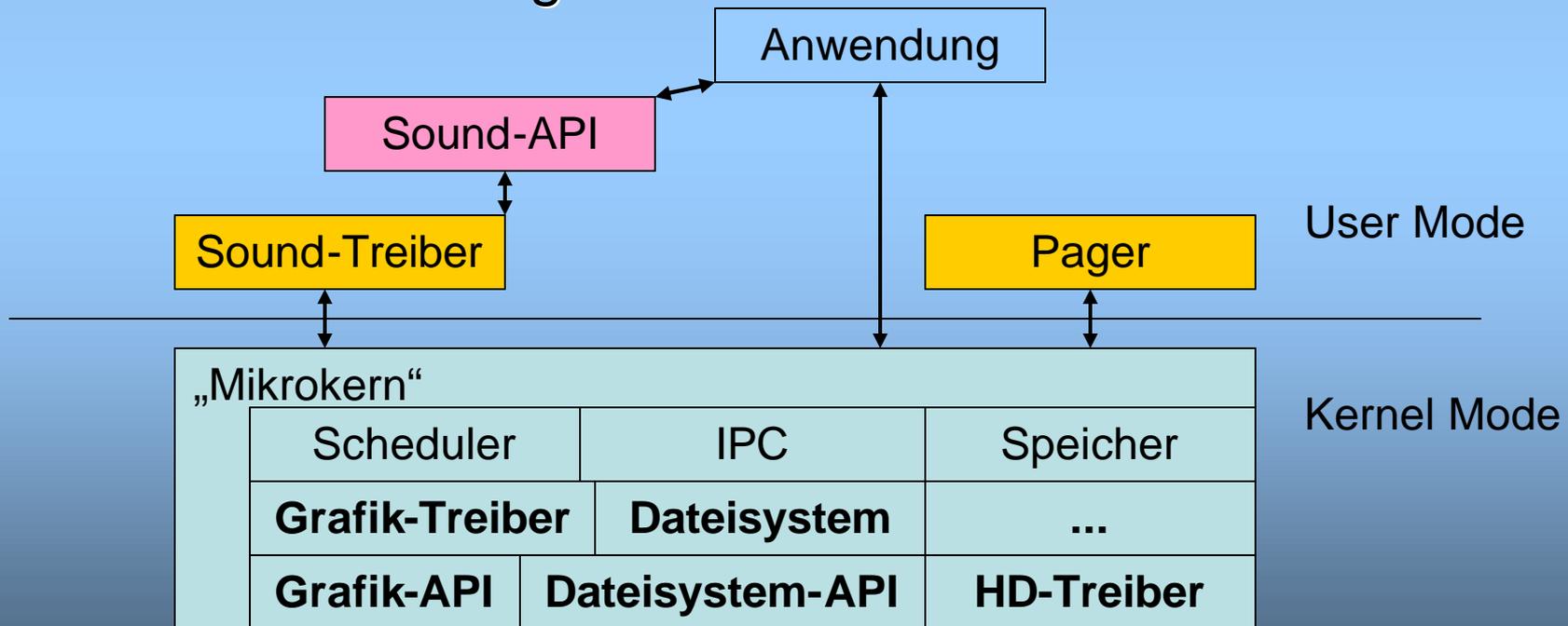
Mikrokern mit zwei Personalities

# Vor- und Nachteile

- Vorteile
  - Höhere Robustheit
  - Höhere Modularität
  - Höhere Flexibilität
  - Höhere Sicherheit
  - Geringerer Speicherbedarf
  - Bessere Reaktionszeit auf Interrupts
  - Einfacherer Umgang mit SMP
- Nachteil
  - Massive Performanceprobleme aufgrund der gesteigerten Interprozeßkommunikation

# Mikrokerne mit Kompromissen

- IPC-Performance war nicht signifikant zu steigern
- Reintegration zeitkritischer Server in den Kern
- Vorteile nur eingeschränkt erhalten



# Die zweite Generation

- Eigenschaften:
  - Höhere Geschwindigkeit
  - Höhere Flexibilität
- Hauptvertreter
  - L4 
  - QNX Neutrino 

# L4

- Mikrokern der zweiten Generation
- Entwickelt von Prof. Jochen Liedtke bei der GMD, bei IBM und an der Universität Karlsruhe
- Viele Implementierungen:
  - L4/x86 The logo for L4/x86 features a stylized airplane silhouette above the text "L4/x86" in a black box.
  - L4/Alpha, L4/MIPS
  - Fiasco (Dresden) The logo for Fiasco features a stylized tower or antenna structure above the text "FIASCO" in a black box.
  - Hazelnut (Karlsruhe) The logo for L4Ka features the text "L4Ka" in a black box, with "L4" in red and "Ka" in black.
  - weitere

# Geschwindigkeitsanalyse

## Potentielle Performance-Bremsen

- Umschalten zwischen Kernel- und User-Mode
- Kontextwechsel
- IPC

Konzeption nicht für Performance-Probleme  
verantwortlich,

**Implementierungen** verantwortlich!

# Geschwindigkeitssteigerung

## Designprinzipien:

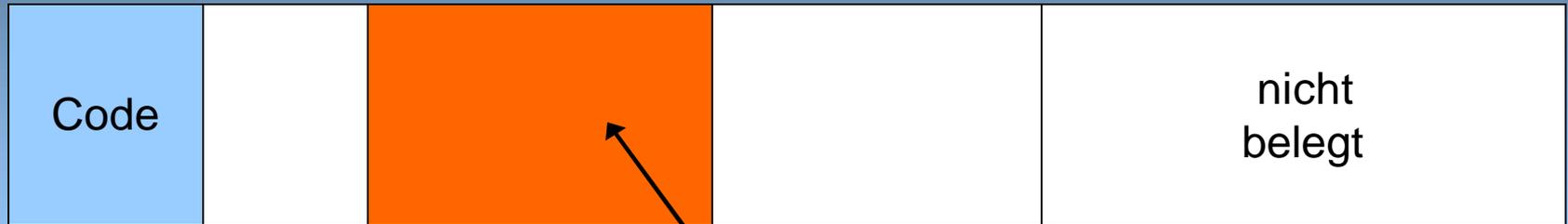
- Bottom-Up-Konstruktion
- *Minimale* Abstraktion der Maschine
- Kern notwendigerweise in Assembler geschrieben

# Architektur von L4

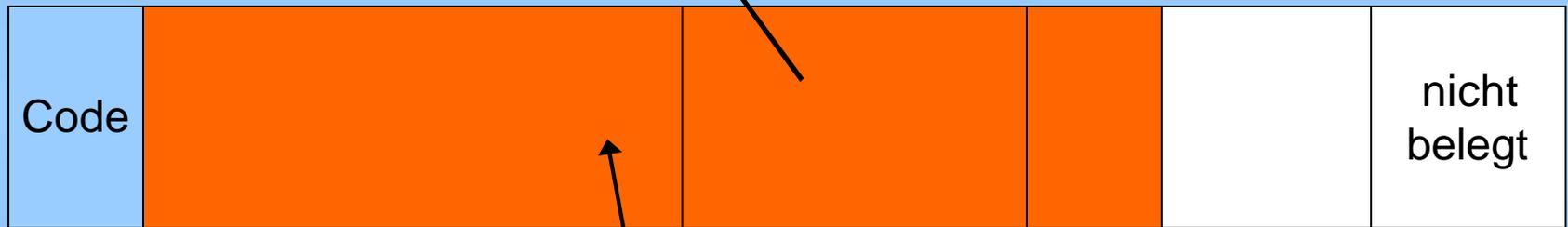
- Kern kümmert sich nur um:
  - Maschinenverwaltung
  - Grundlegende Speichervergabe
  - Thread-Verwaltung, Scheduling
  - Interprozeßkommunikation
- Nur sieben Systemaufrufe

ipc	thread_schedule
id_nearest	lthread_ex_regs
fpage_unma	task_new
p	
thread_switc	
h	

# Speicher



Speicher einer Anwendung



Speicher des Pagers/Speichermanagers



Physischer Speicher

# Threads, Tasks, Scheduling

## Konzepte

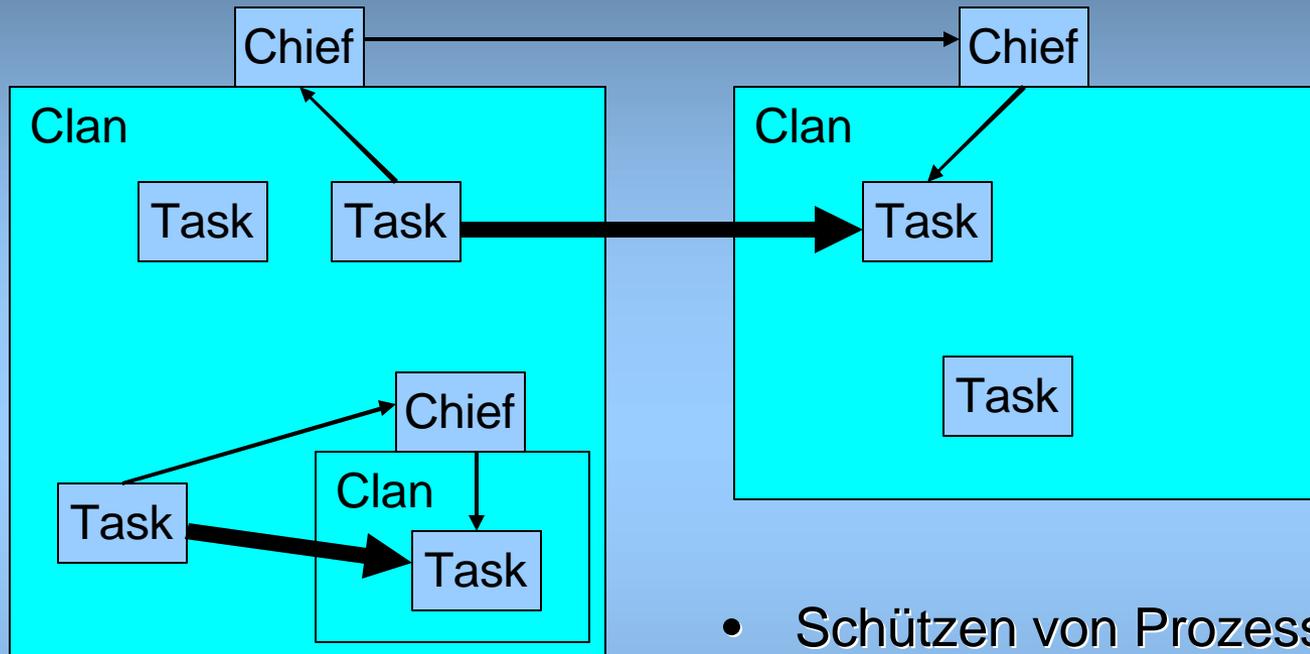
- Task: Adreßraum mit Threads
- Maximale Anzahl an Tasks und Threads
- Einfaches Scheduling im Kern
- Externer Scheduler möglich

# Interprozeßkommunikation

## Konzepte

- Synchron und ungepuffert
- Kombination über „call“ und „reply&receive next“
- Komplexe Nachrichten
- Kurze Nachrichten über Register

# Clans & Chiefs



- Schützen von Prozessen
- Sandbox
- Kompatibilität
- Angriffe aus dem Netz

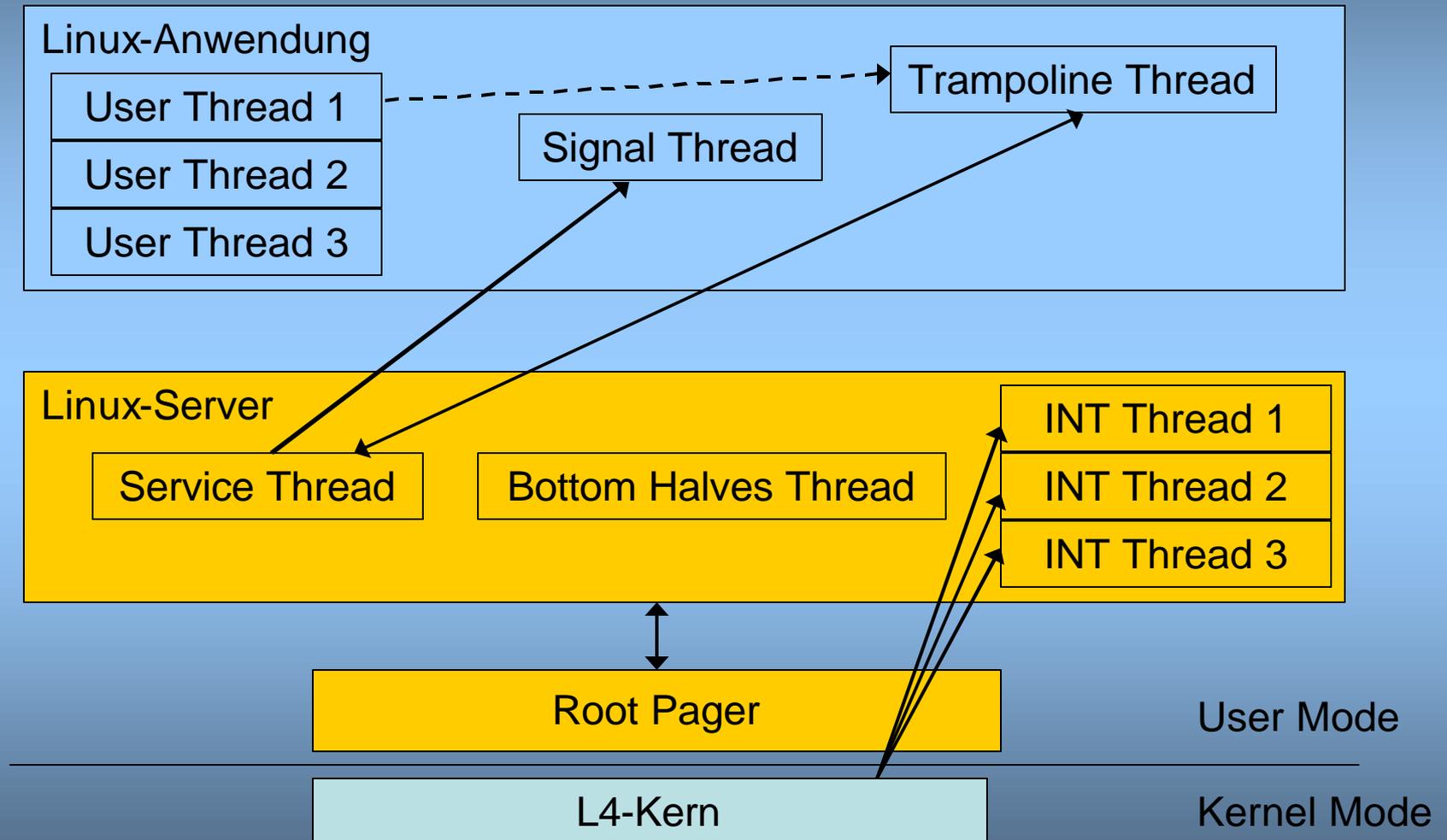
# Wie funktioniert dann...?

- Speicherverwaltung
- Scheduler
- Echtzeit
- Treiber
- Systeminformationen
- Simulation einer Umgebung

# L4Linux

- DROPS-Projekt
  - Echtzeit-Betriebssystem auf L4-Basis
  - UNIX Personality nötig
- Diplomarbeit von M. Hohmuth 1996
- Portierung von Linux 2.0 auf L4
- Ziele
  - vollständig im User-Mode
  - Binärkompatibilität
  - Test von L4

# L4Linux



# Wieso?

- Leistung schlechter als echtes Linux, aber besser als MkLinux
- Zunächst keine der drei Hauptvorteile
- Aber neue Möglichkeiten:
  - Parallele Personality
  - Treiber direkt auf L4
  - Höhere Geschwindigkeit

# Zusammenfassung

- Mikrokerne
  - haben viele Vorteile
  - sind in der Regel langsam
- L4 ist
  - ein Mikrokern der zweiten Generation
  - fast so schnell wie monolithische Kerne
- Linux-Personality auf L4
  - existiert und funktioniert
  - ist nur unbedeutend langsamer

# Zukunft

- Tod von Jochen Liedtke 2001
- Erfahrungen mit L4,  
fortschreitende Prozessorentwicklung
- 64-Bit-Version
- Geschwindigkeit von C++-  
Implementierungen