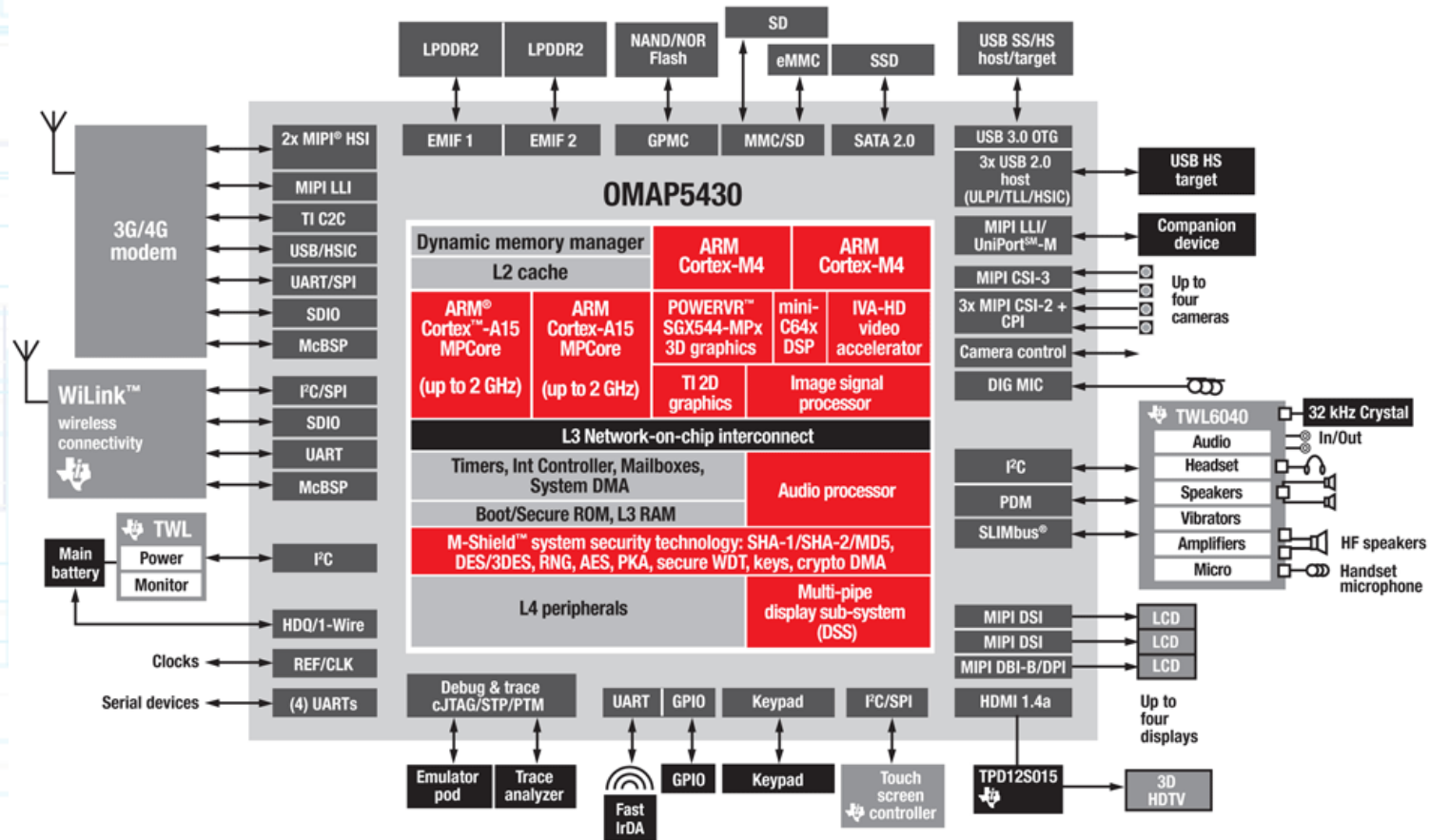


Eingebettete Betriebssysteme

Bachelorstudiengang
Technische Informatik
Hochschule Pforzheim

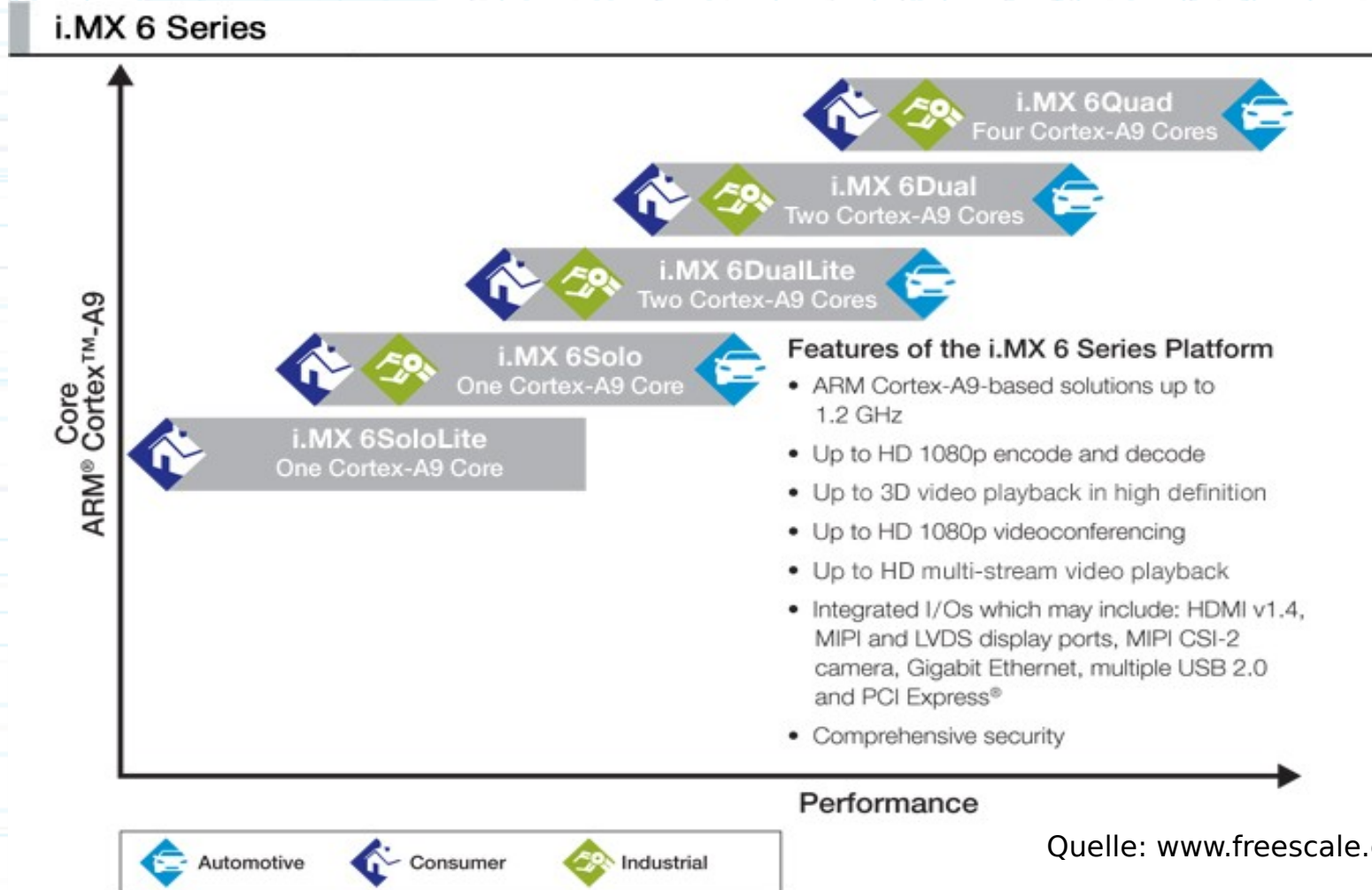
Dipl.-Ing.(FH) Marc Jüttner

TI OMAP5430



Quelle: ti.com

Skalierung von SoCs



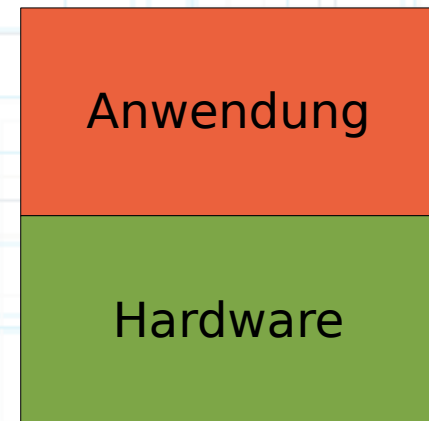
Quelle: www.freescale.com

Skalierung von SoCs

- Unterschiedliche Taktraten
- Unterschiedliche Konfiguration
 - Zahl der CPUs
 - Zahl der Beschleuniger
 - Verfügbare Peripherie

Softwarearchitekturen

- Direkter Hardwarezugriff
 - Geringer Footprint
 - Hochoptimiert
 - Schwer portierbar
 - Hohe Kosten
 - Geringe Wiederverwendbarkeit



Softwarearchitekturen

- Indirekter Hardwarezugriff
 - Betriebssystem-API
 - Kein/wenig direkter Hardwarezugriff
 - Abstrakt und komplex
 - Portierbar
 - Wiederverwendbare Software

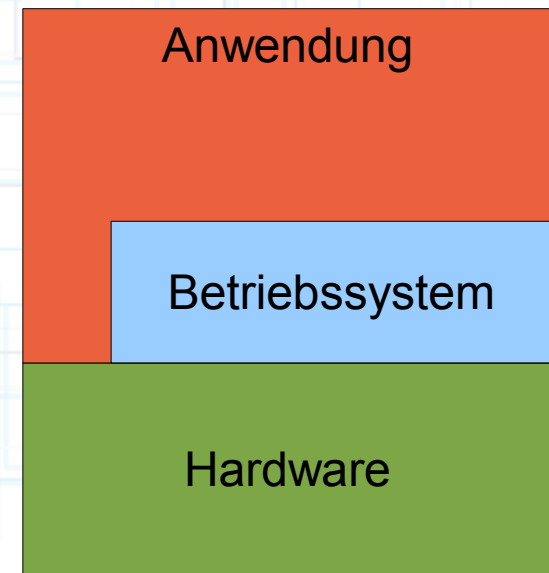


Abbildung von SoCs

- Komplexität von SoCs erfordert eine strukturierte Abbildung in Software
- Anforderungen
 - Nutzung für verschiedene SoCs
 - Erweiterbarkeit
 - Einheitliche Softwareschnittstellen

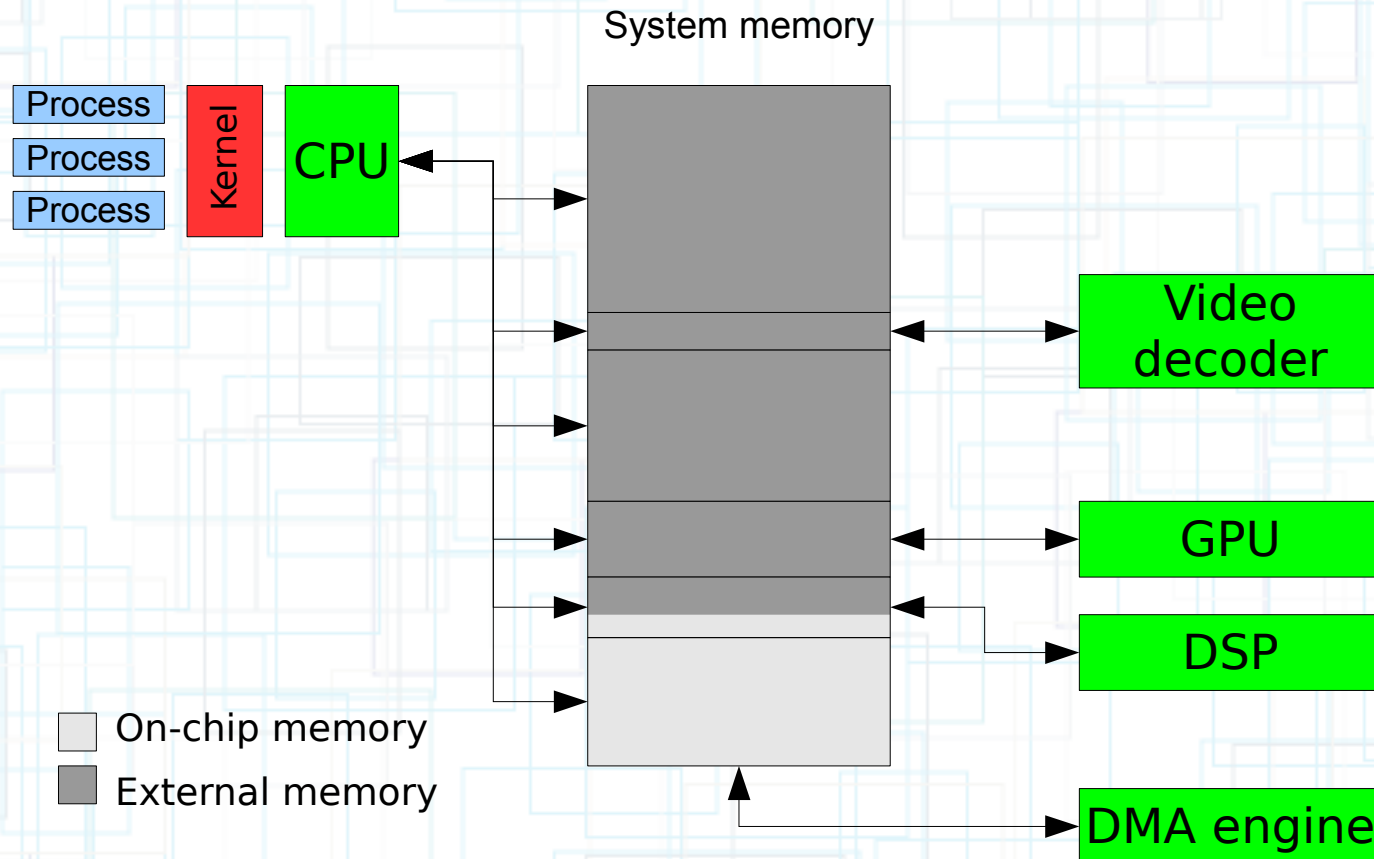
Abbildung von SoCs

- Infrastruktur
 - CPUs
 - Controller
 - Bussysteme
- Laufzeitverhalten
 - Power management
 - Offloading

Speicherverwaltung

- Verwaltung des physikalischen Speichers
- Zuweisung von Speicher an Prozesse und Kernel
- Verwalten freigegebenen Speichers
- Segmentierung/Paging
- Speicherzuweisung an alle CPUs und SoC-Peripherie

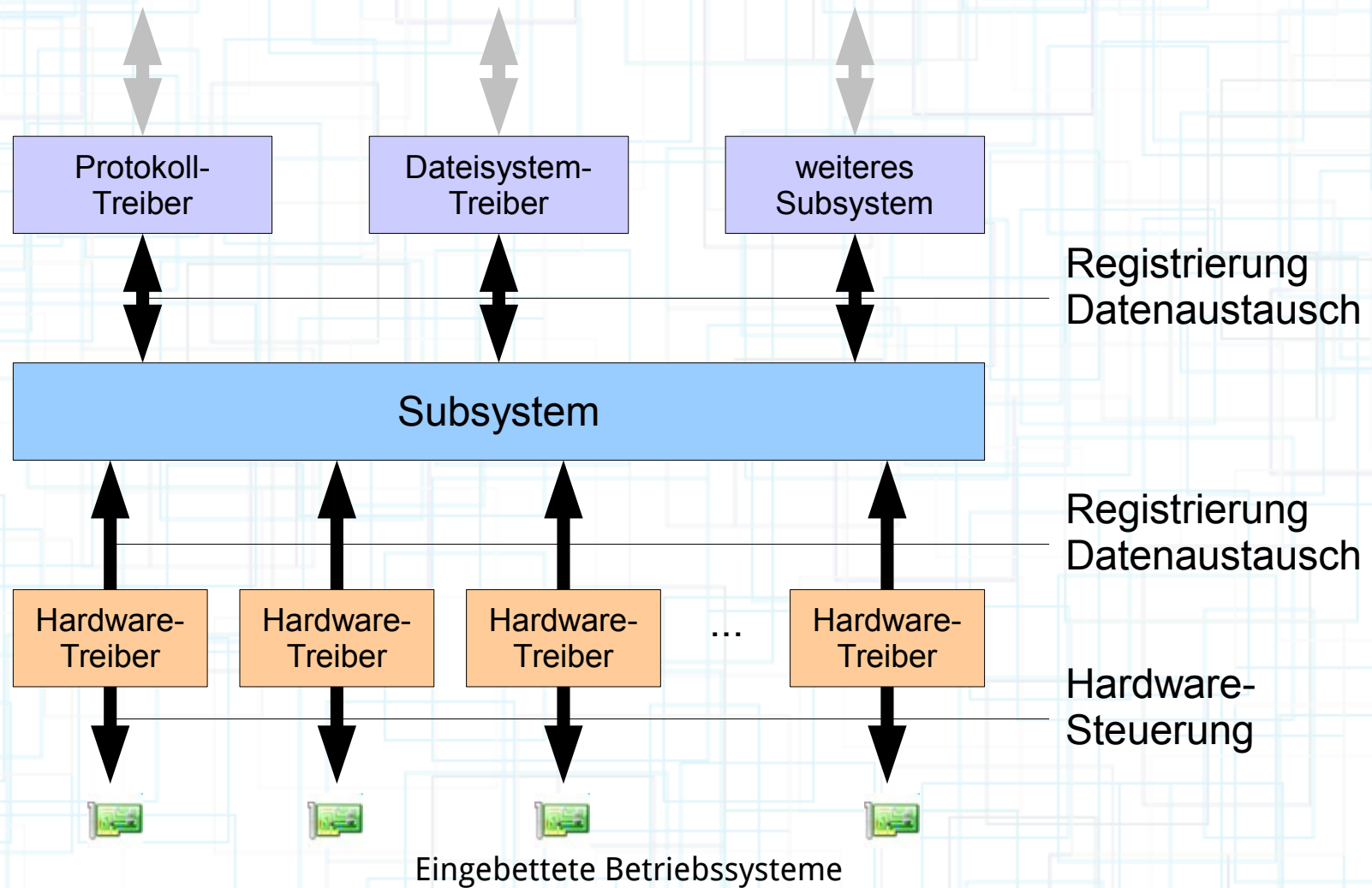
Speicherverwaltung



Arbitrierung

- Welcher Treiber für welches Gerät?
- Dynamische Busverwaltung
- Hotplug
- Coldplug

Subsystemarchitektur



Gerätetreiber

- Treiber für SoC-Peripherie und externe Peripherie erforderlich
 - Erkennung
 - Konfiguration
 - Betrieb

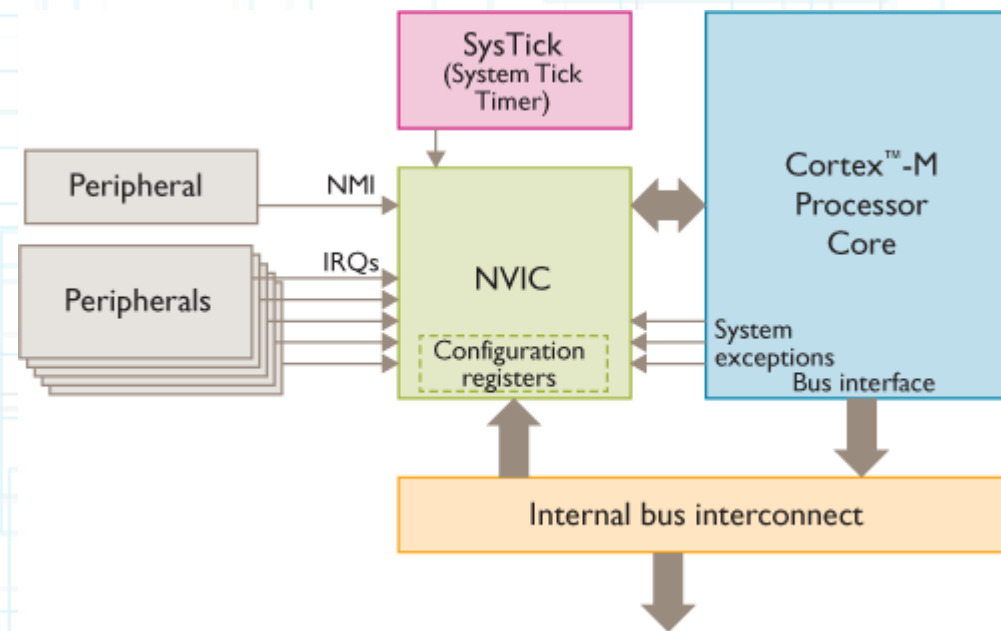
Gerätetreiber: Aufgaben

- Ein Treiber soll mehrere Geräte verwalten können
- Zuordnung von Anforderungen zur Hardware
- Zugriffsverwaltung
- Abstraktion

Interruptbehandlung

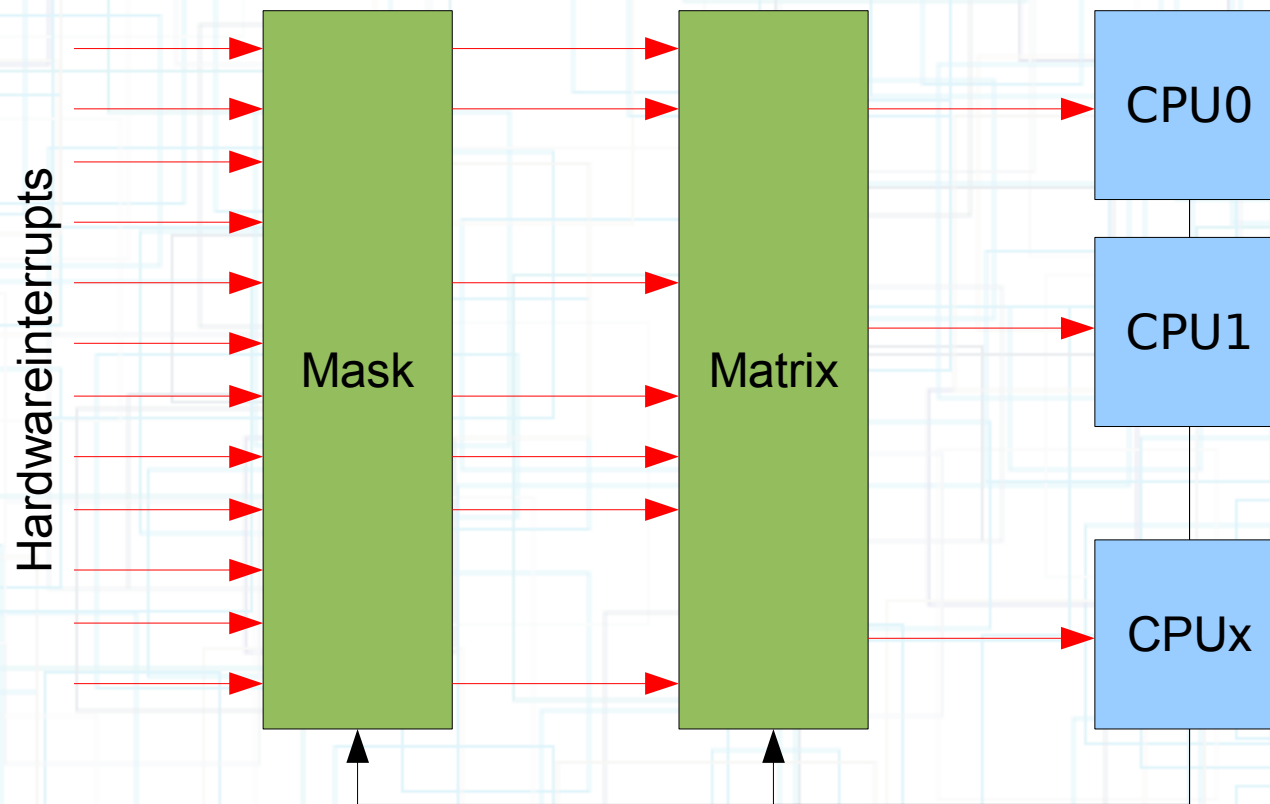
- Viele Interruptquellen auf einem SoC
 - Unter Umständen mehrere CPUs
 - Zuweisung von Interrupts an freie CPUs
- Vermittlung zwischen Interruptquelle und Interruptbehandlung
- Priorisierung
- Maskierung

Interruptcontroller

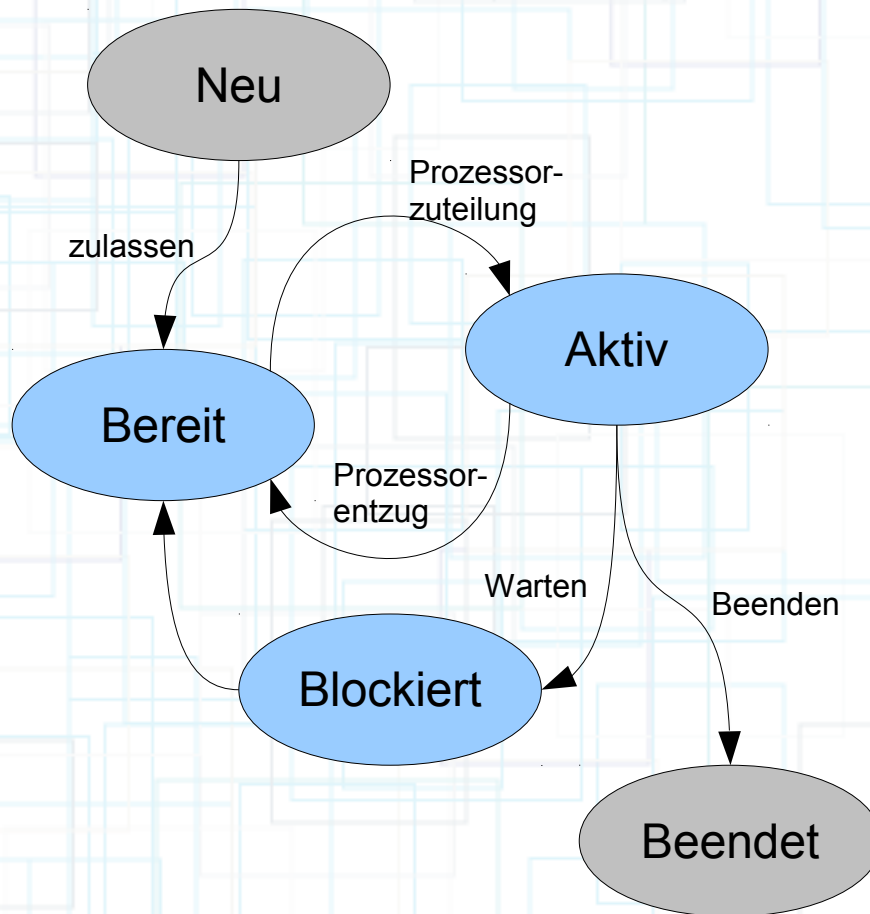


Quelle: arm.com

Interruptbehandlung



Multitasking



- Verwaltung von Prozesszuständen
- Anderweitige Nutzung der CPU während ein Prozess wartet

Betriebssysteme

- Software zur Bereitstellung grundlegender Systemfunktionalität

Definition nach DIN44300

Die Programme eines digitalen Rechensystems, die zusammen mit den Eigenschaften dieser Rechenanlage die Basis der möglichen Betriebsarten des digitalen Rechensystems bilden und die insbesondere die Abwicklung von Programmen steuern und überwachen.

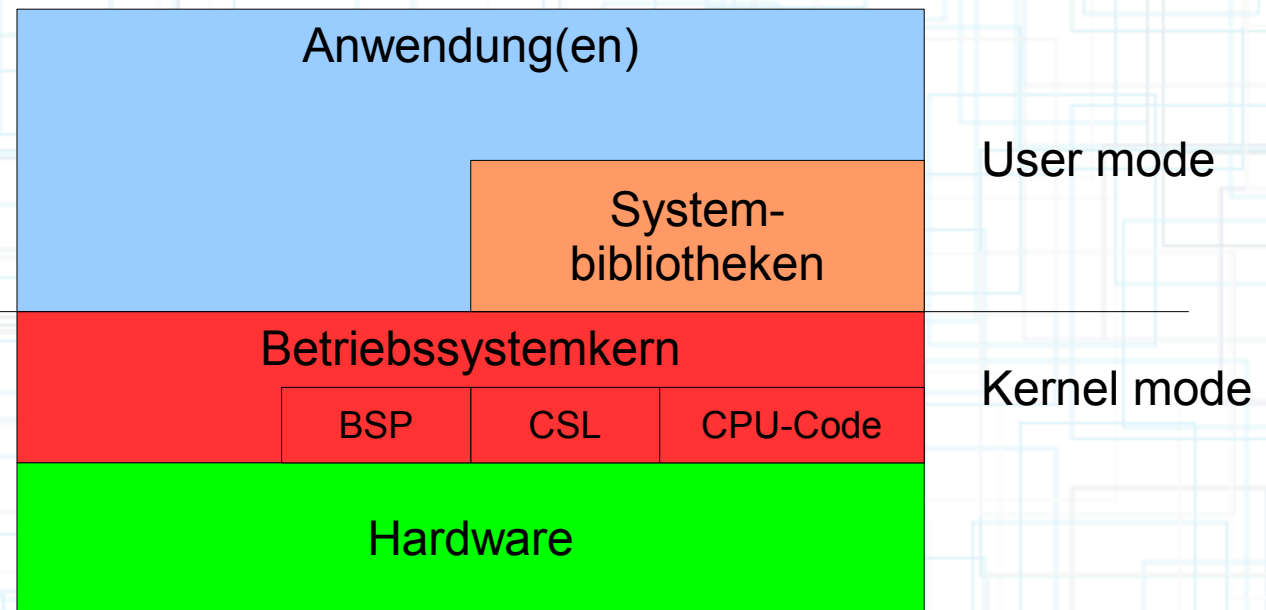
Aufgaben

- Verbergen der Komplexität der Maschine vor dem Anwender
- Bereitstellen
 - einer Benutzerschnittstelle
 - einer API
- Ressourcenverwaltung
 - Prozessor(en), Speicher, Medien, Geräte, Rechenzeit, Rechte, ...

Anforderungen

- Sicherheitsaspekt
 - Anwendungen können die Hardware nicht unkontrolliert manipulieren
- Integrität
 - Fehlfunktionen einzelner Anwendungen beeinträchtigen nicht das Gesamtsystem
- Portierbarkeit
- Leichte Bedienbarkeit

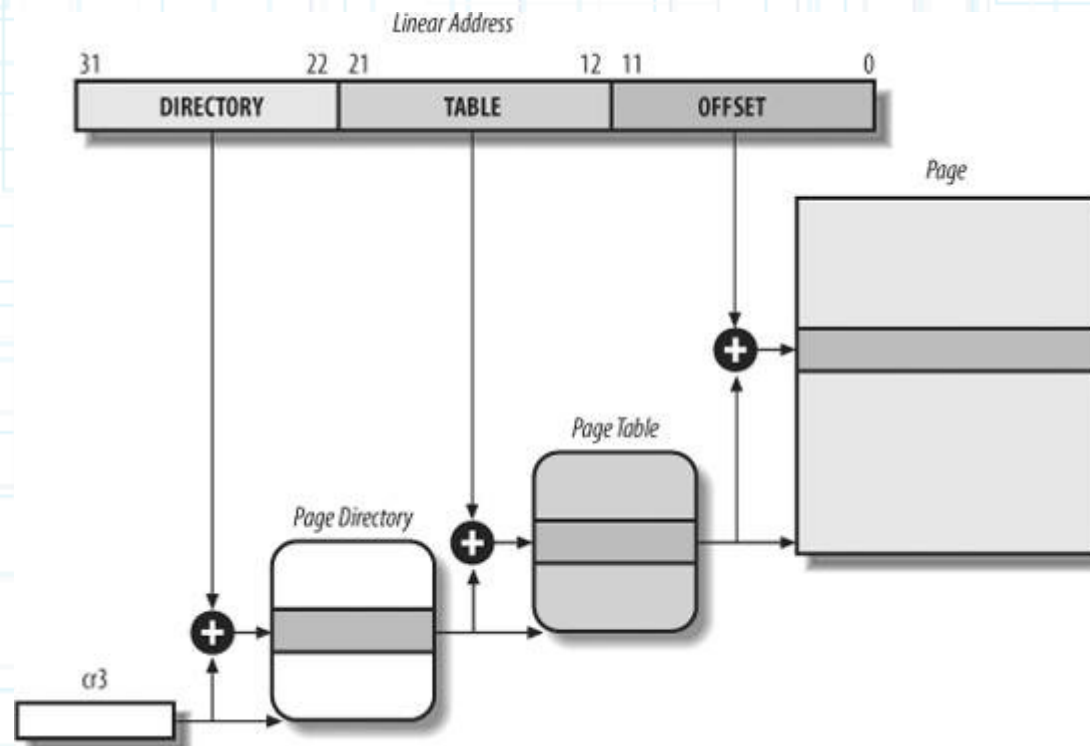
Struktur des BS



Umsetzung

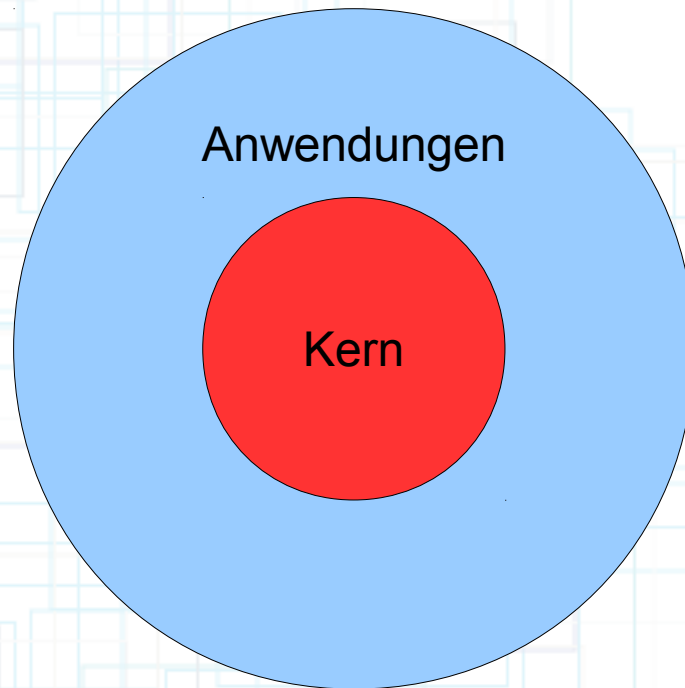
- Wie kann eine Privilegientrennung realisiert werden?
 - Getrennte Befehle für Speicher- und Peripheriezugriff
 - Virtueller Speicher mit MMU-Unterstützung

Virtueller Speicher



Quelle: Understanding the Linux Kernel

Privilegienebenen



- Betriebssystemkern auf höchster Ebene
- Anwendungen auf niedrigster Ebene
- Höhere Sicherheit bei Anwendungen mit Benutzereingriff

Kernkonzepte

- Implementierung des Betriebssystemkerns kann auf verschiedene Weise erfolgen
- Abhängig von den Anforderungen
- Unterschiedliche Nutzung der Prozessorunterstützung

Monolithischer Kernel

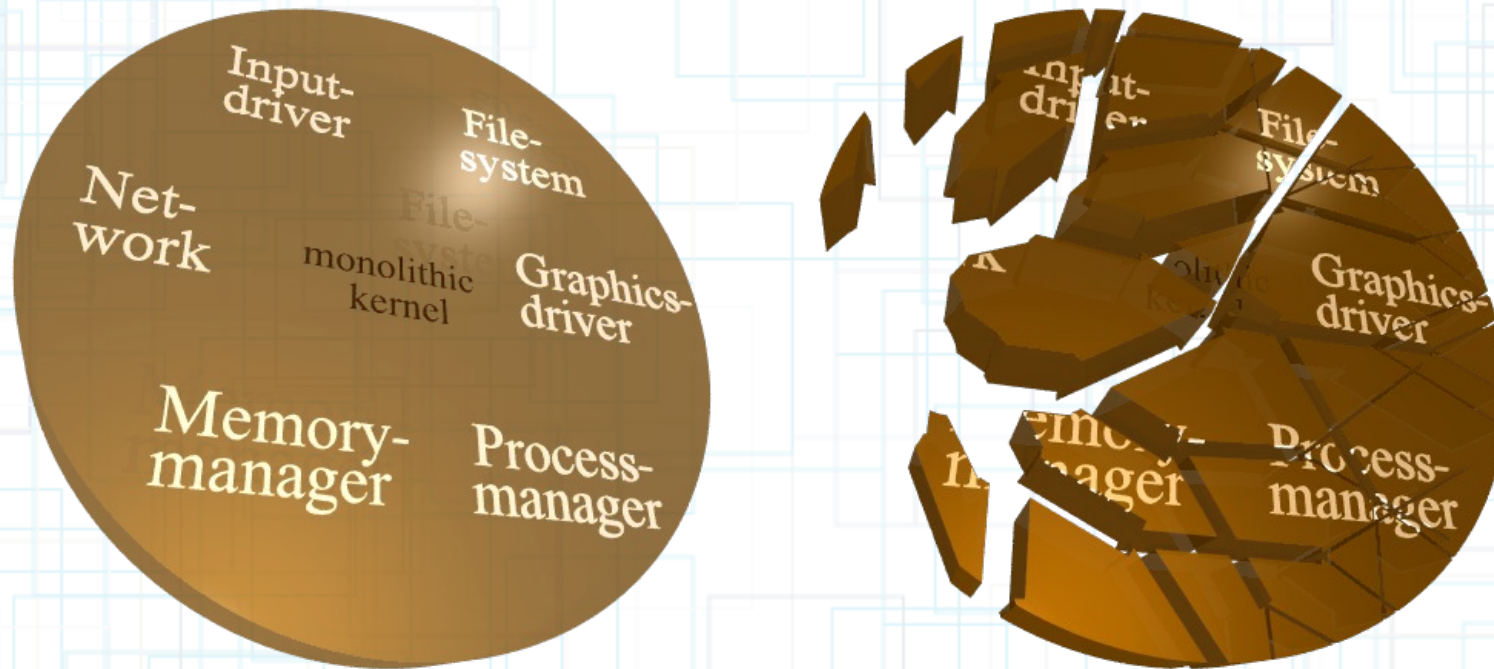
- Alle Kernelbestandteile sind zu einem monolithischen Binärprogramm zusammengefasst
 - Scheduler, Speichermanager, IO
 - Treiber, Subsysteme
- Kernel läuft mit vollen Privilegien
- Anwendungen laufen ohne Privilegien

Monolithischer Kernel

- Schnelle Ausführung
- Direkter Zugriff von überall
- Anfällig für Fehlfunktionen einzelner Teile



Veranschaulicht...

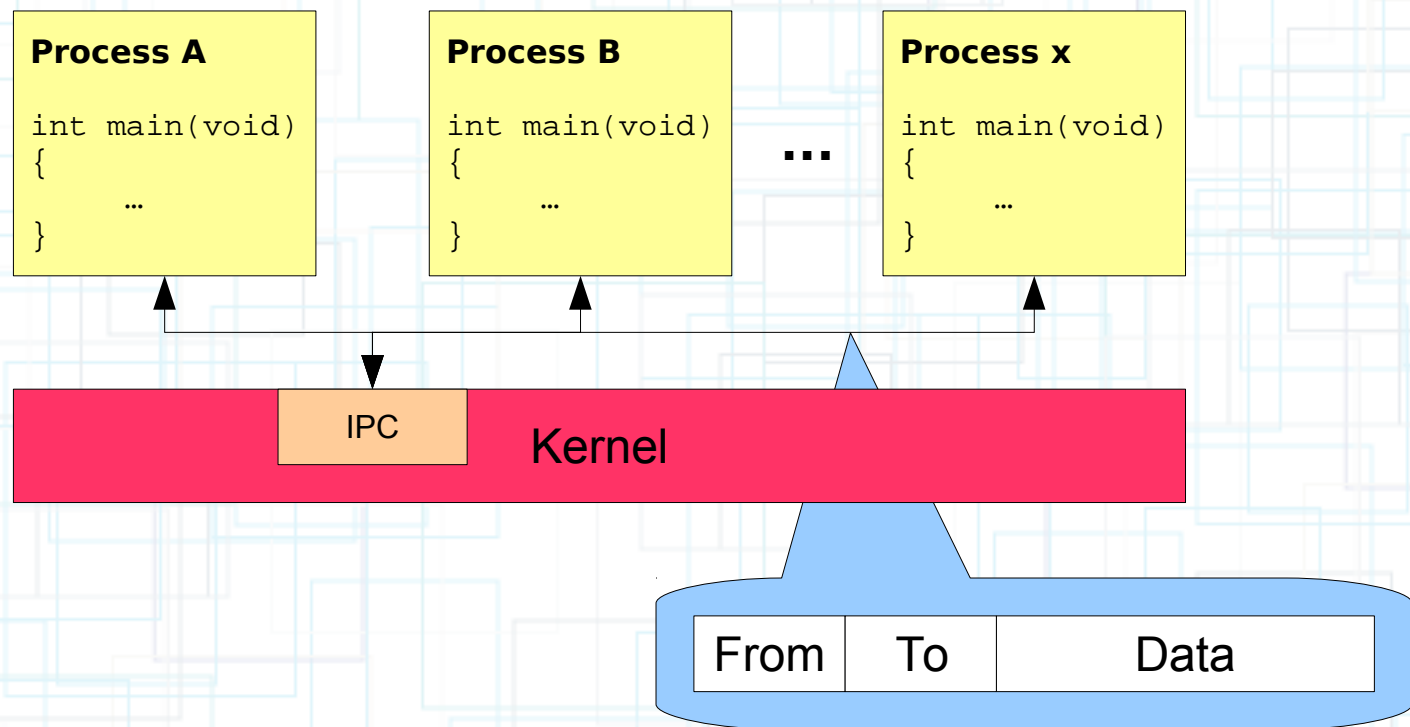


Quelle: TU Dresden Operating System – tudos.org

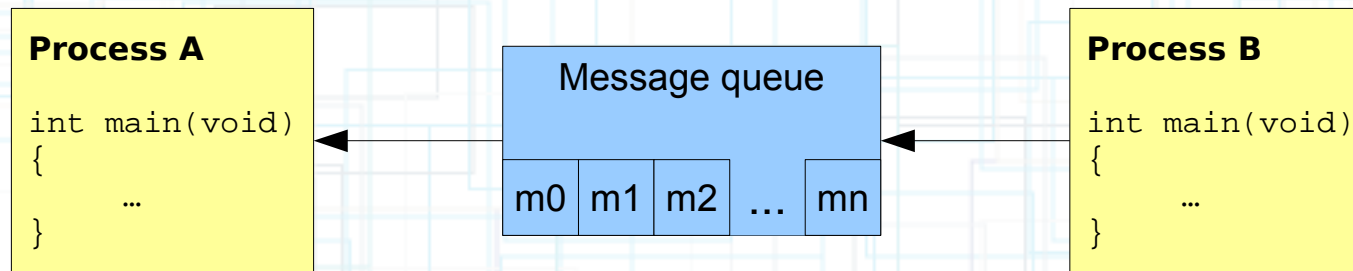
Mikrokernel

- Nur die wichtigsten Kernelteile laufen mit vollen Privilegien
 - Scheduler, Speichermanager, IO
 - Interprozesskommunikation (IPC)
- Alle Treiber und Anwendungen laufen ohne Privilegien
 - IPC-Nutzung erforderlich

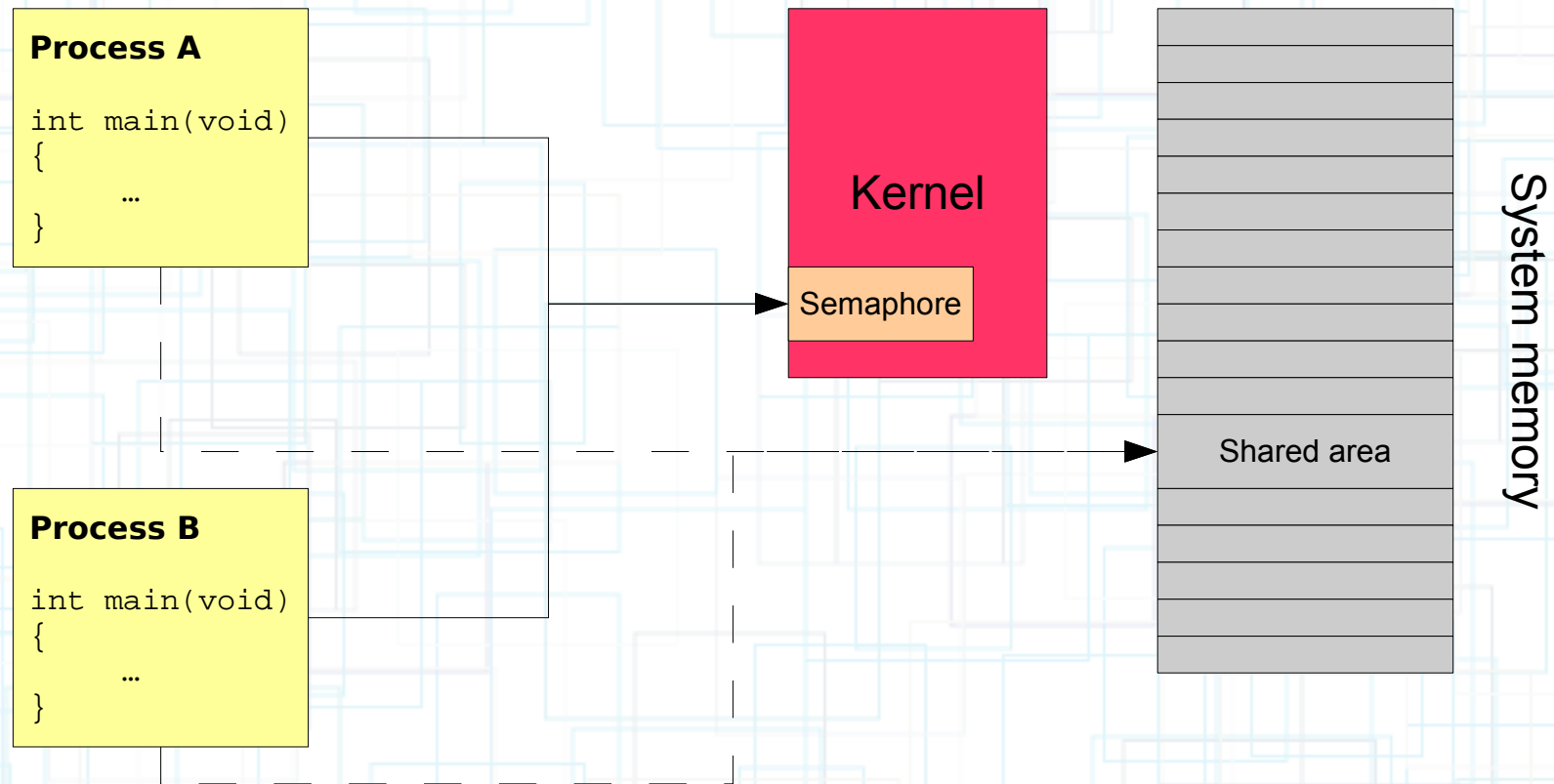
Exkurs: IPC



Exkurs: Message Queues



Exkurs: Shared Memory

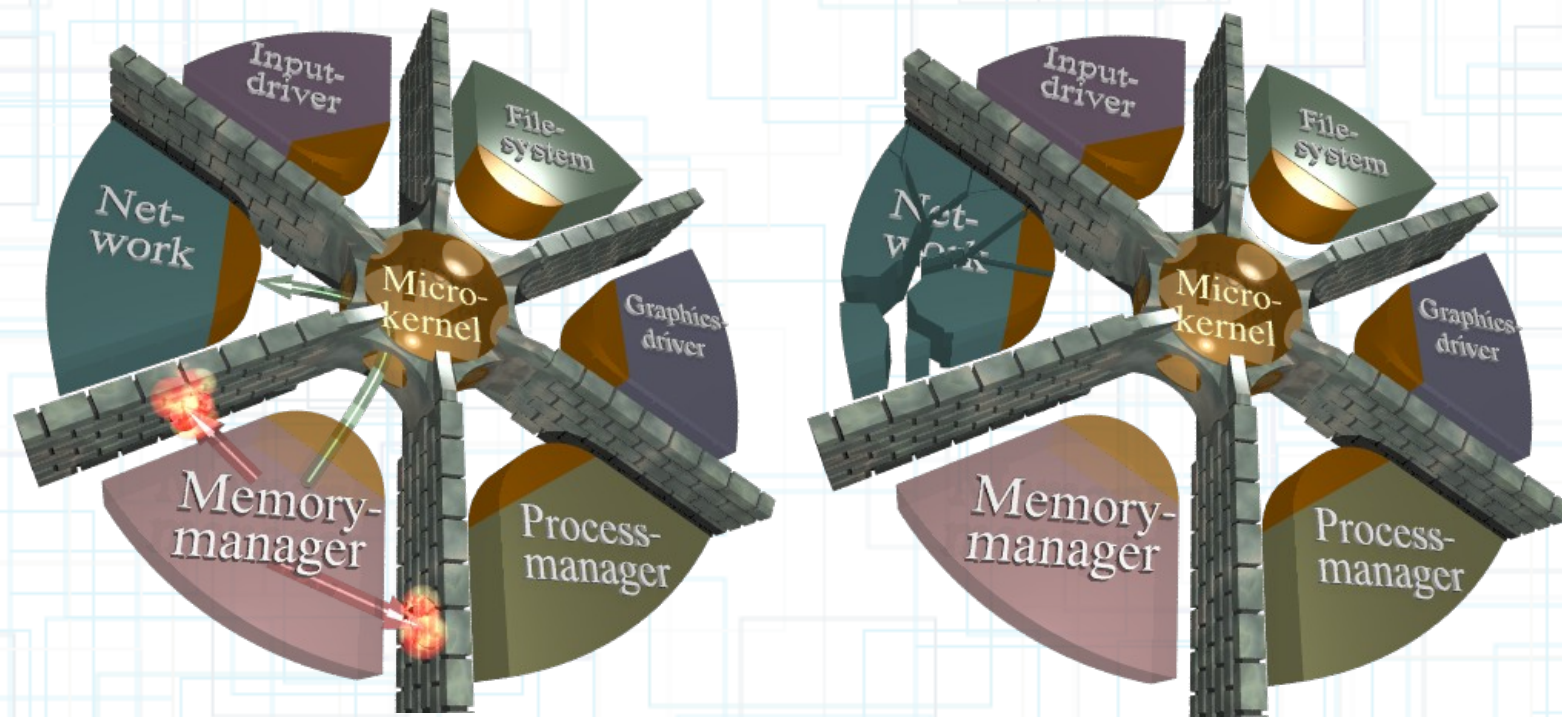


Mikrokernel

- Absolute Kapselung
- Fehlertolerant
- Langsamere Kommunikation zwischen Modulen



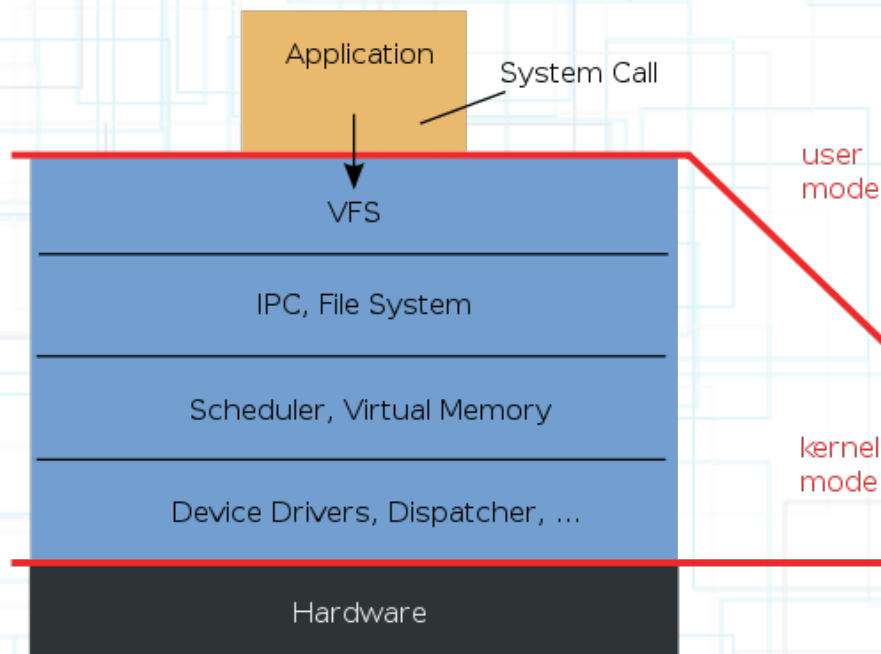
Veranschaulicht...



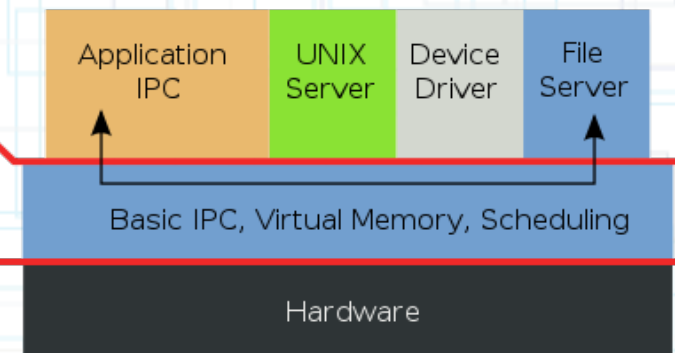
Quelle: TU Dresden Operating System - tudos.org

Vergleich

Monolithic Kernel based Operating System

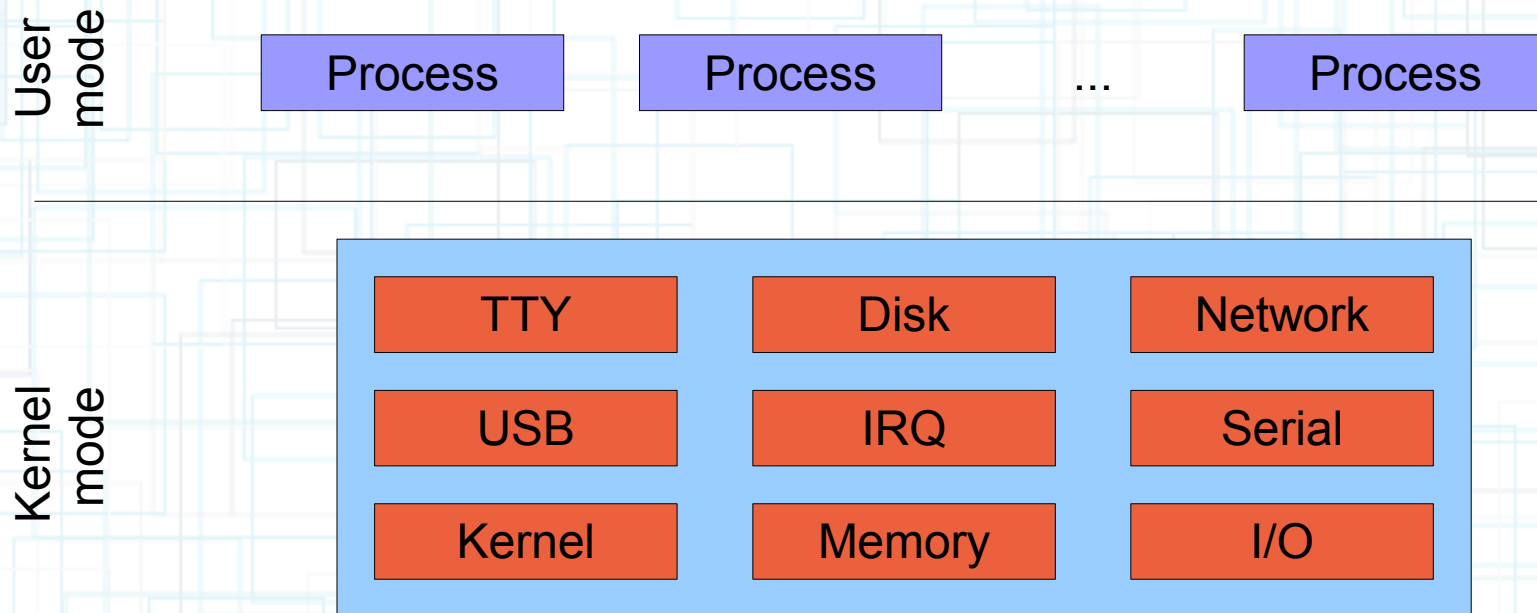


Microkernel based Operating System

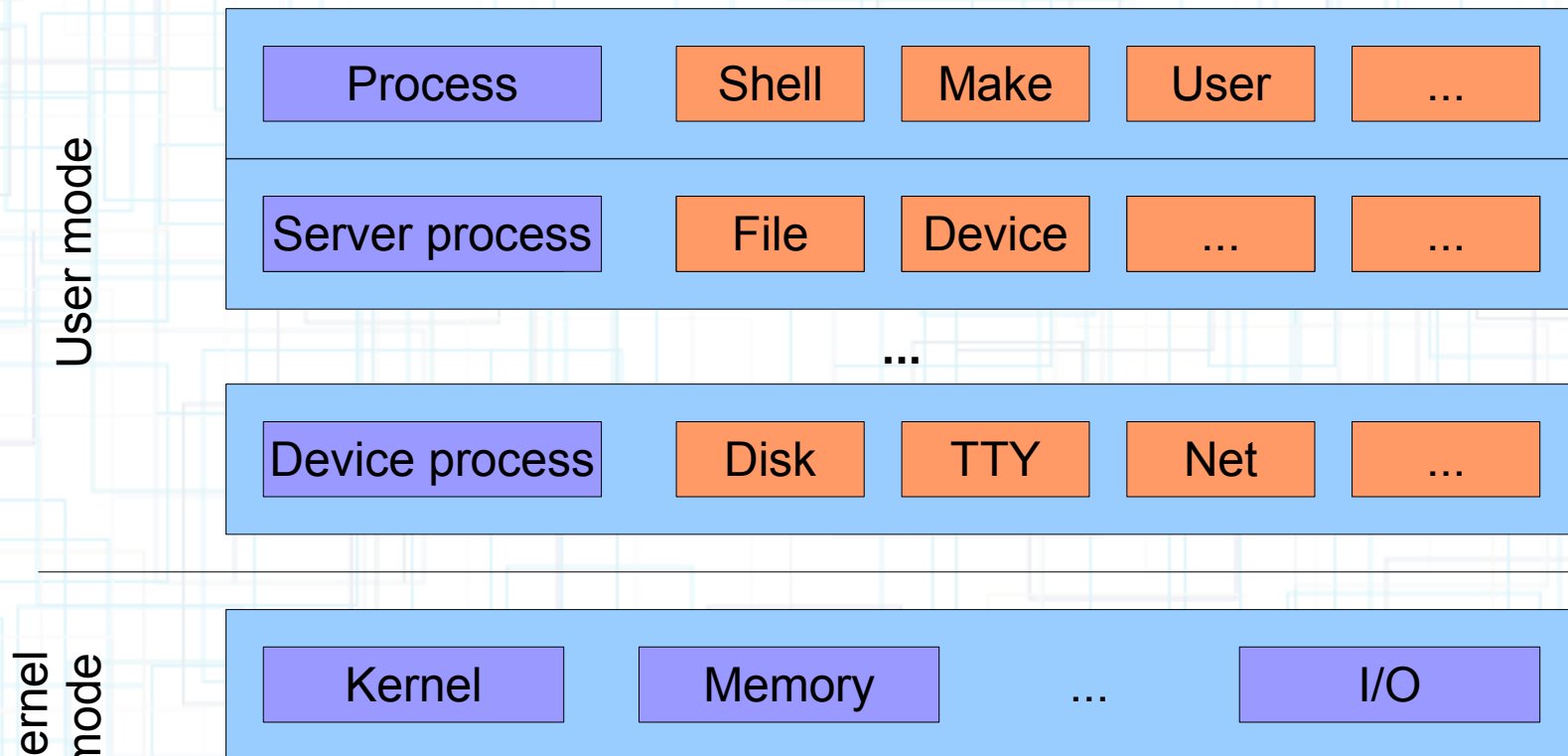


Quelle: Wikipedia

Prozesse beim „Monolith“



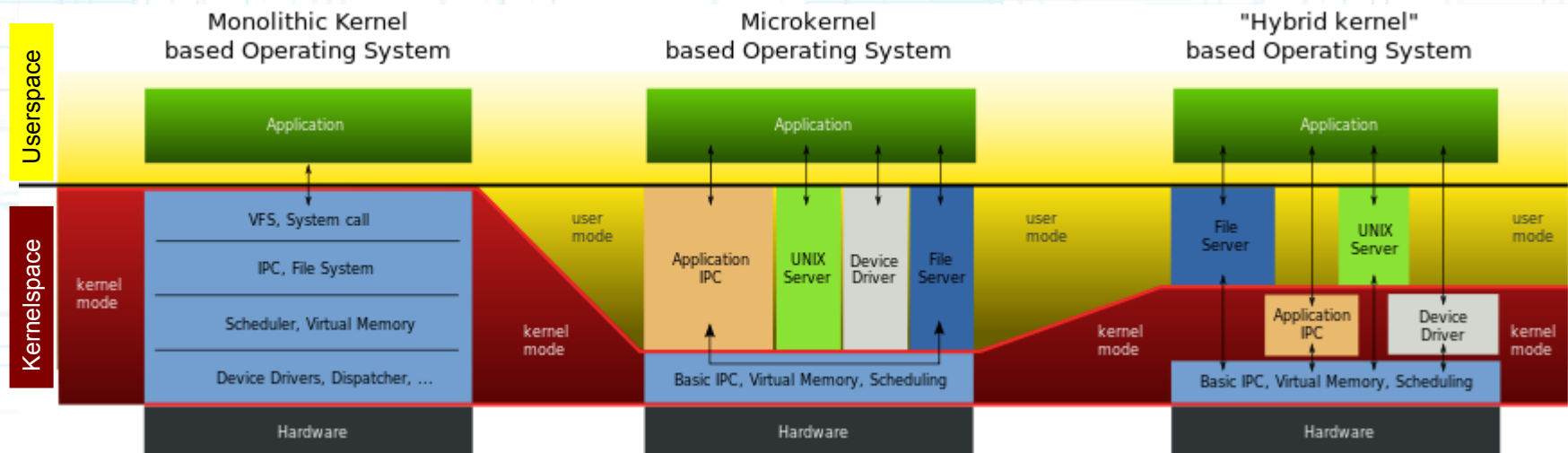
Prozesse beim Mikrokern



Kompromiss: Hybridkernel

- Kombination der Vorteile
- Reduktion der Nachteile
- Zeitkritische Teile privilegiert
- Sicherheitskritische Teile nichtprivilegiert

Gesamtvergleich

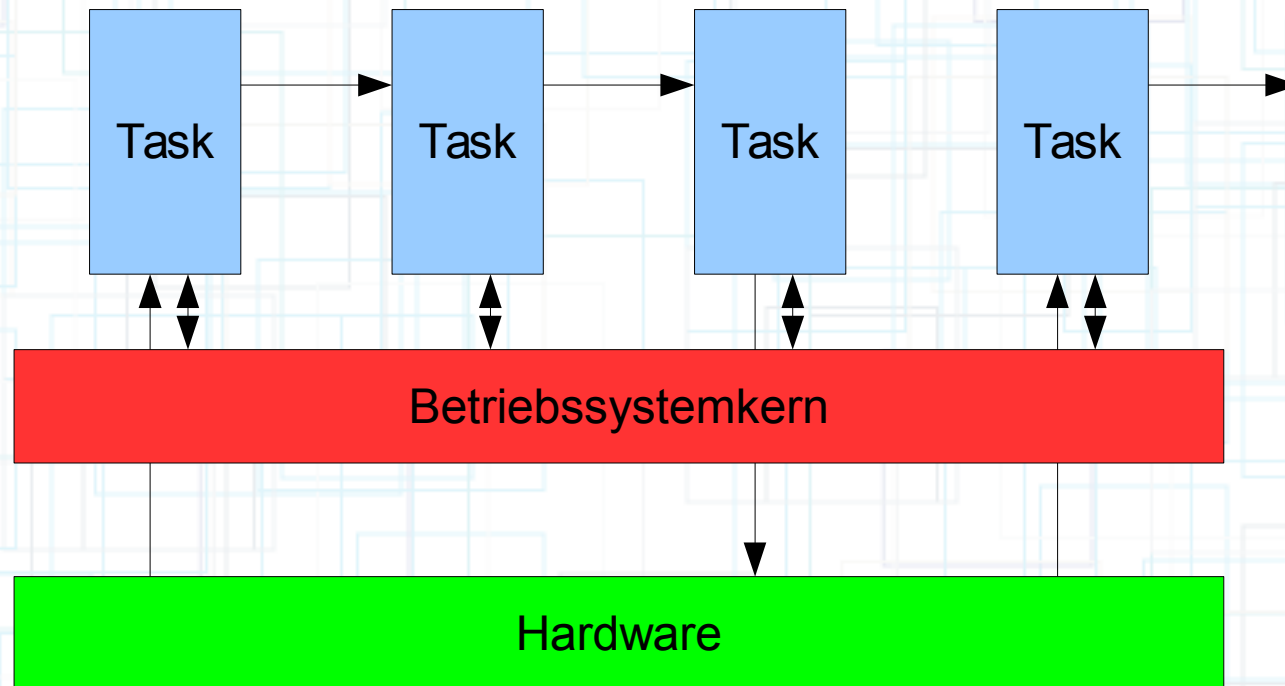


Quelle: Wikipedia

RTOS

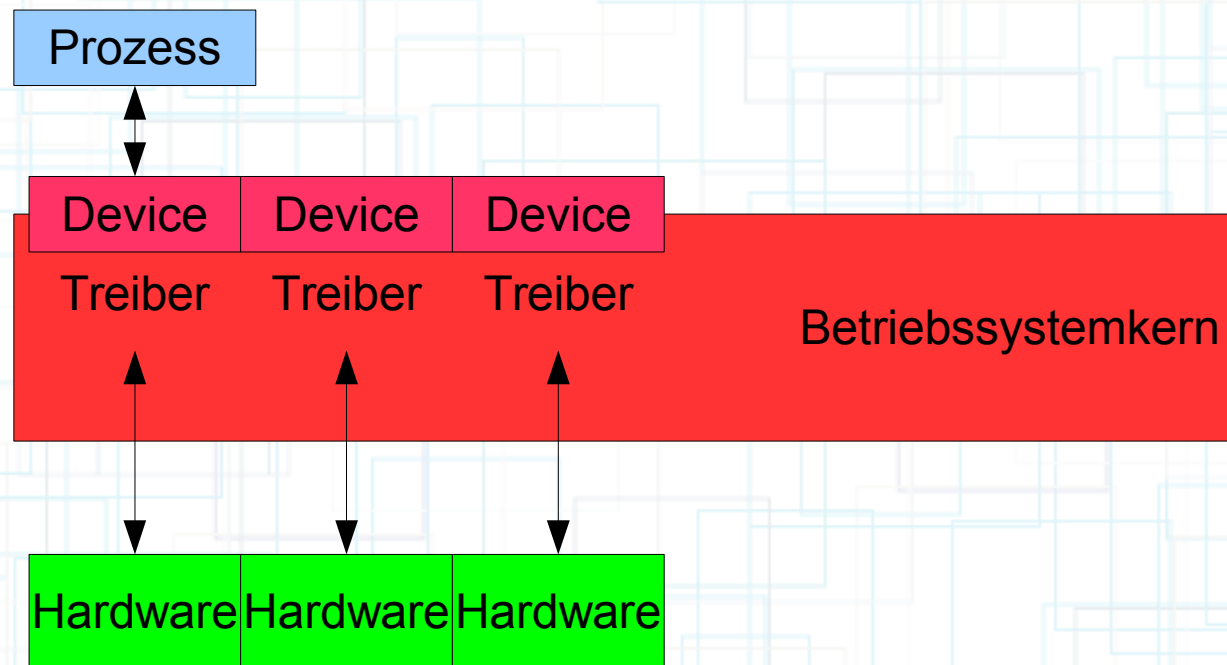
- Realtime Operating System
 - Taskorientiert
 - Ereignisgesteuert: Ereignisse werden von Tasks verarbeitet
 - Oft sehr anwendungsspezifisch
- Kern und Funktion häufig monolithisch integriert im Kernel Mode
 - Nicht immer!

RTOS-Struktur



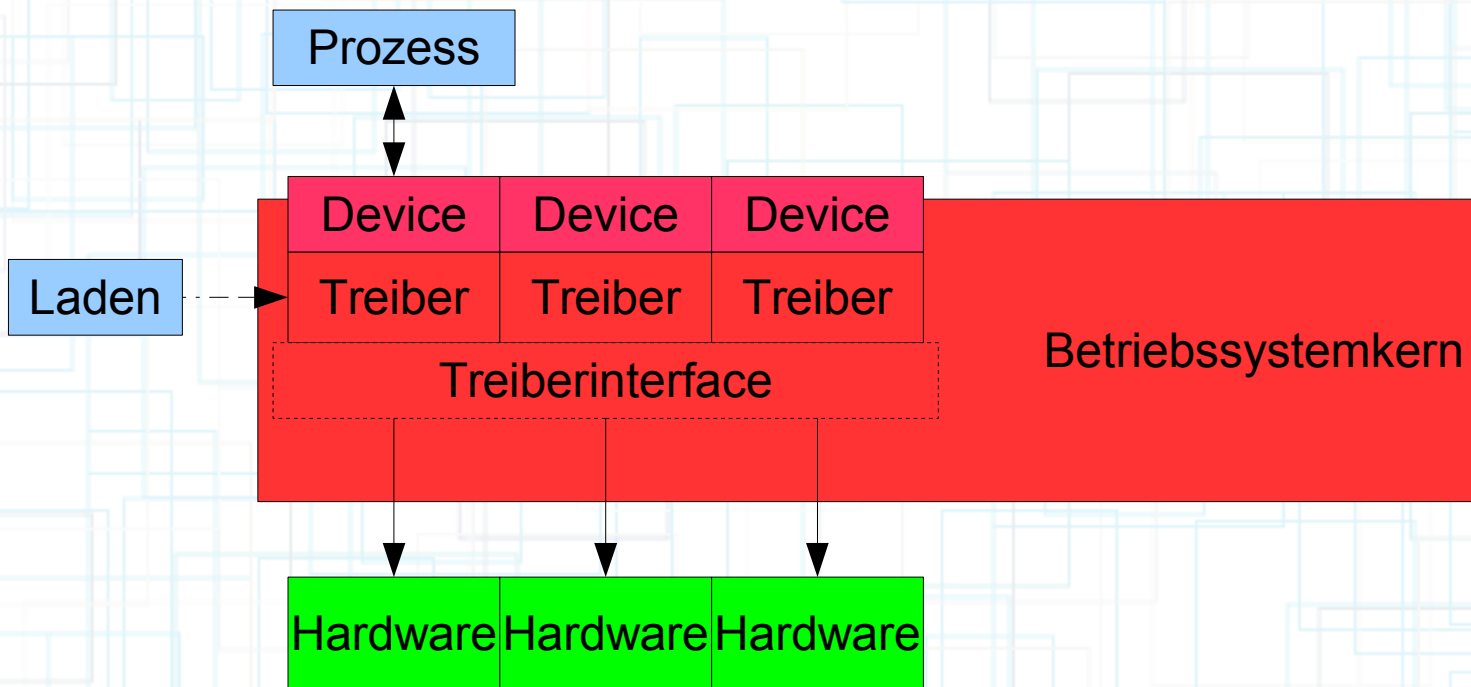
Kerneltreiber

- Treiber als Bestandteil des Kernels



Kernelmodule

- Treiber als ladbare Kernelkomponente



Usermode-Treiber

