

Konzepte von Betriebssystemkomponenten

Gerätetreiber

Mario Körner

26.01.2004



Übersicht

- Einordnung in die Betriebssystemarchitektur
- Schnittstelle zur Hardware
- Schnittstelle zum Betriebssystem am Beispiel Linux
 - Zeichenorientierte Treiber
 - Blockorientierte Treiber
 - Netzwerktreiber



Was ist ein Gerätetreiber?

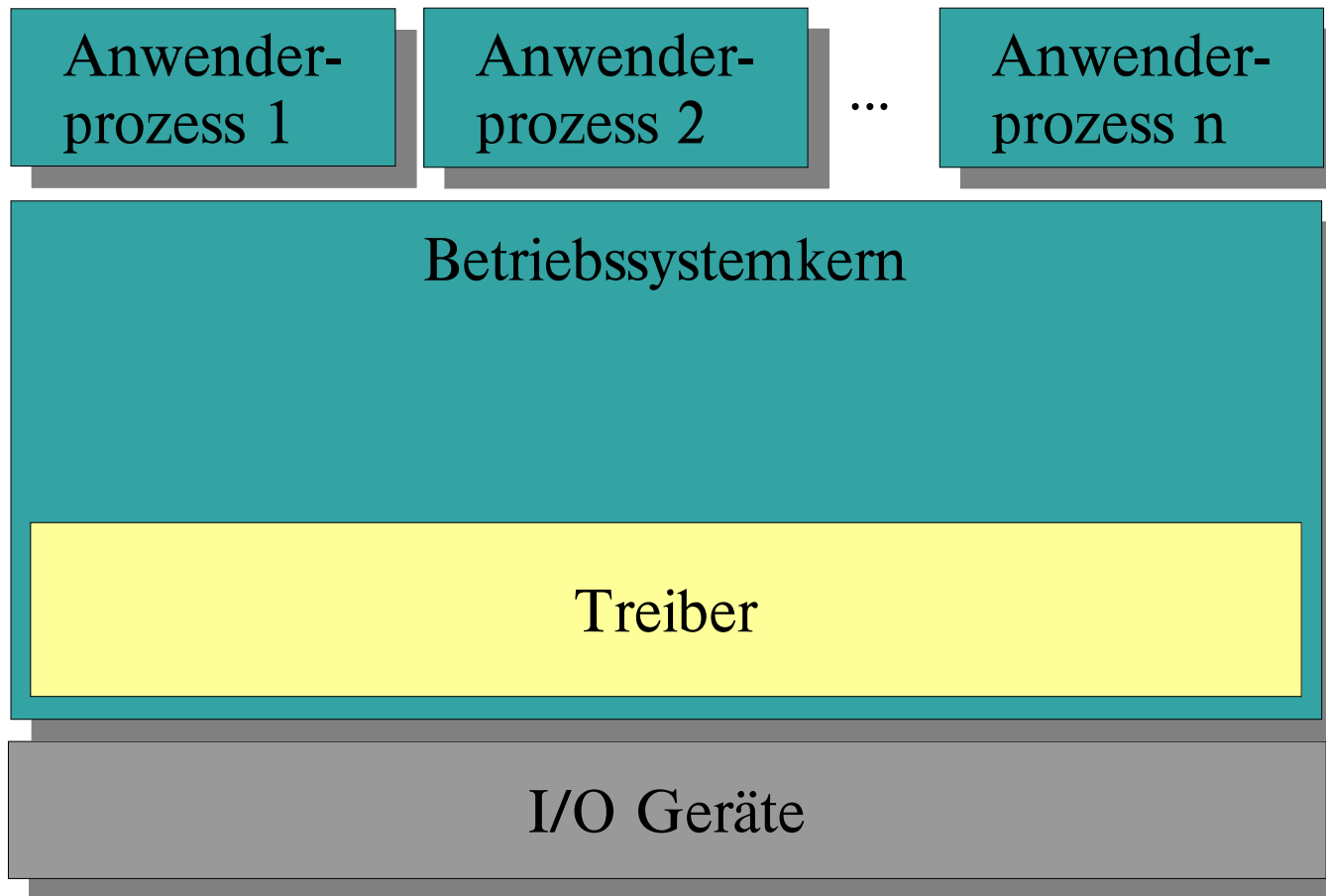
- Abgeschlossenes Softwaremodul mit definierter Schnittstelle
- Aufgaben:
 - Zugriff auf Hardware abstrahieren
 - Unabhängigkeit des Betriebssystems von der Hardware gewährleisten



Anforderungen

- Stabilität
- Performance
- Flexibilität

Rolle im Betriebssystem

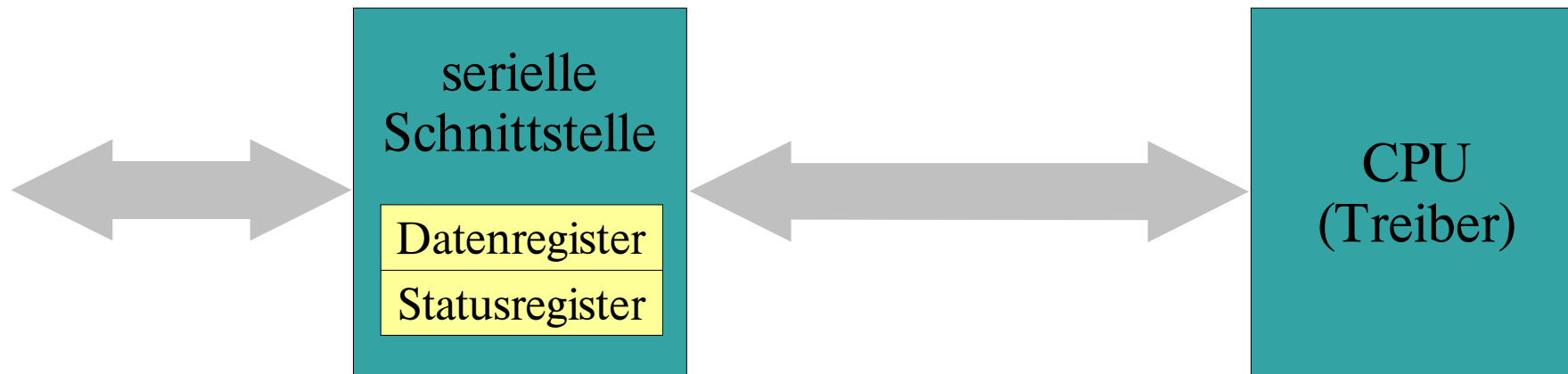




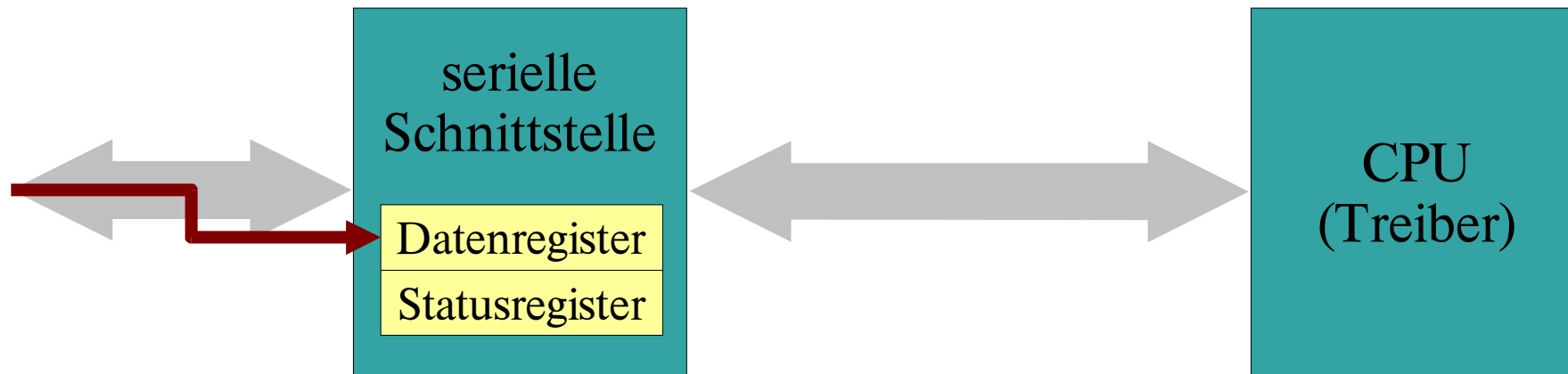
Schnittstelle zur Hardware

- Vom Hardwarehersteller vorgegeben
- Interrupts für asynchrone Nachrichten an die CPU
- Control-/Status-Register für Initialisierung, Konfiguration, Datentransport und Statusabfragen

Schnittstelle zur Hardware: Beispiel

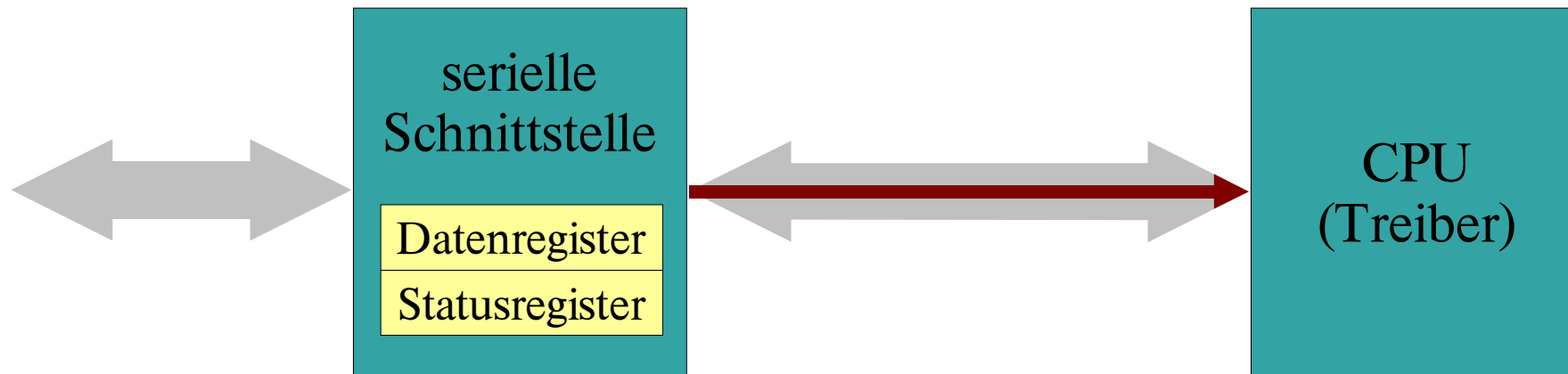


Schnittstelle zur Hardware: Beispiel



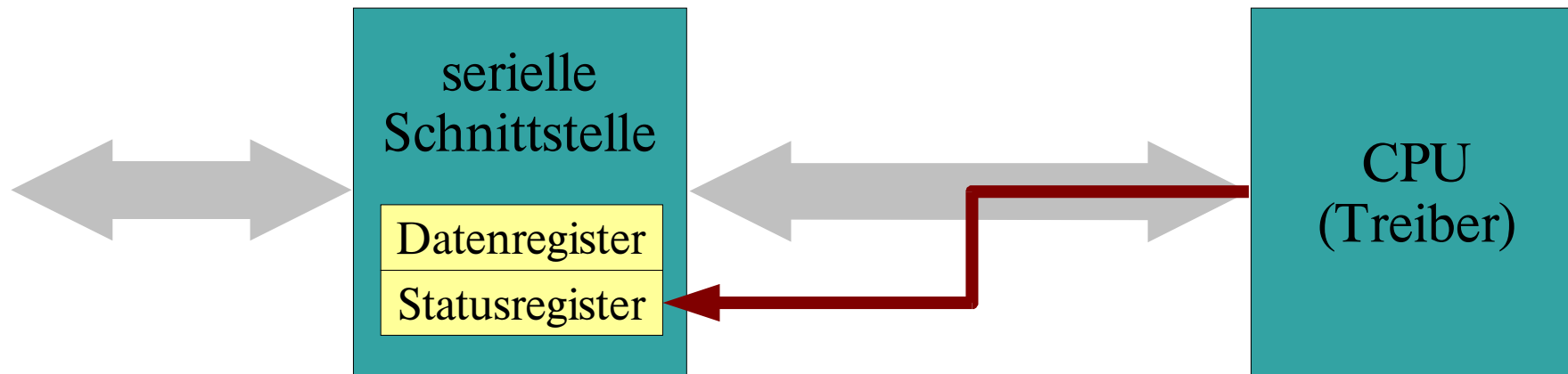
- serielle Schnittstelle empfängt Daten

Schnittstelle zur Hardware: Beispiel



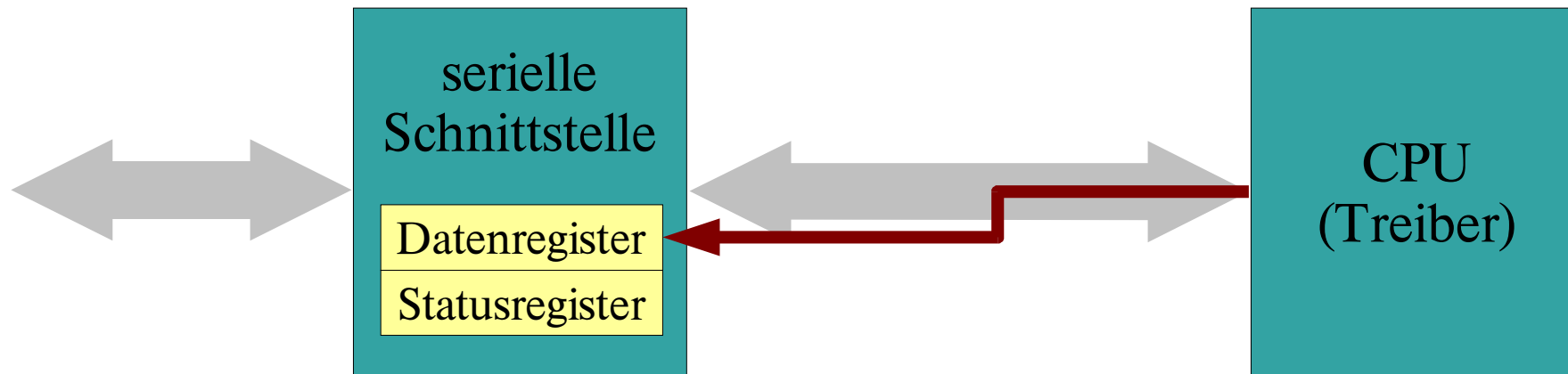
- serielle Schnittstelle meldet ein Ereignis an die CPU

Schnittstelle zur Hardware: Beispiel



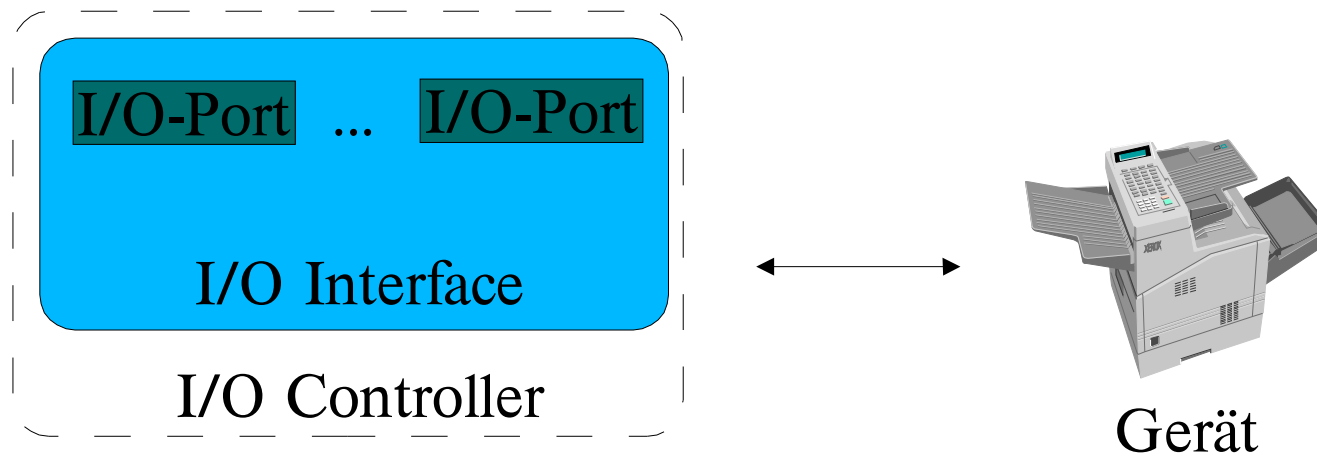
- Treiber fragt das Statusregister ab, um den Grund des Interrupt festzustellen

Schnittstelle zur Hardware: Beispiel

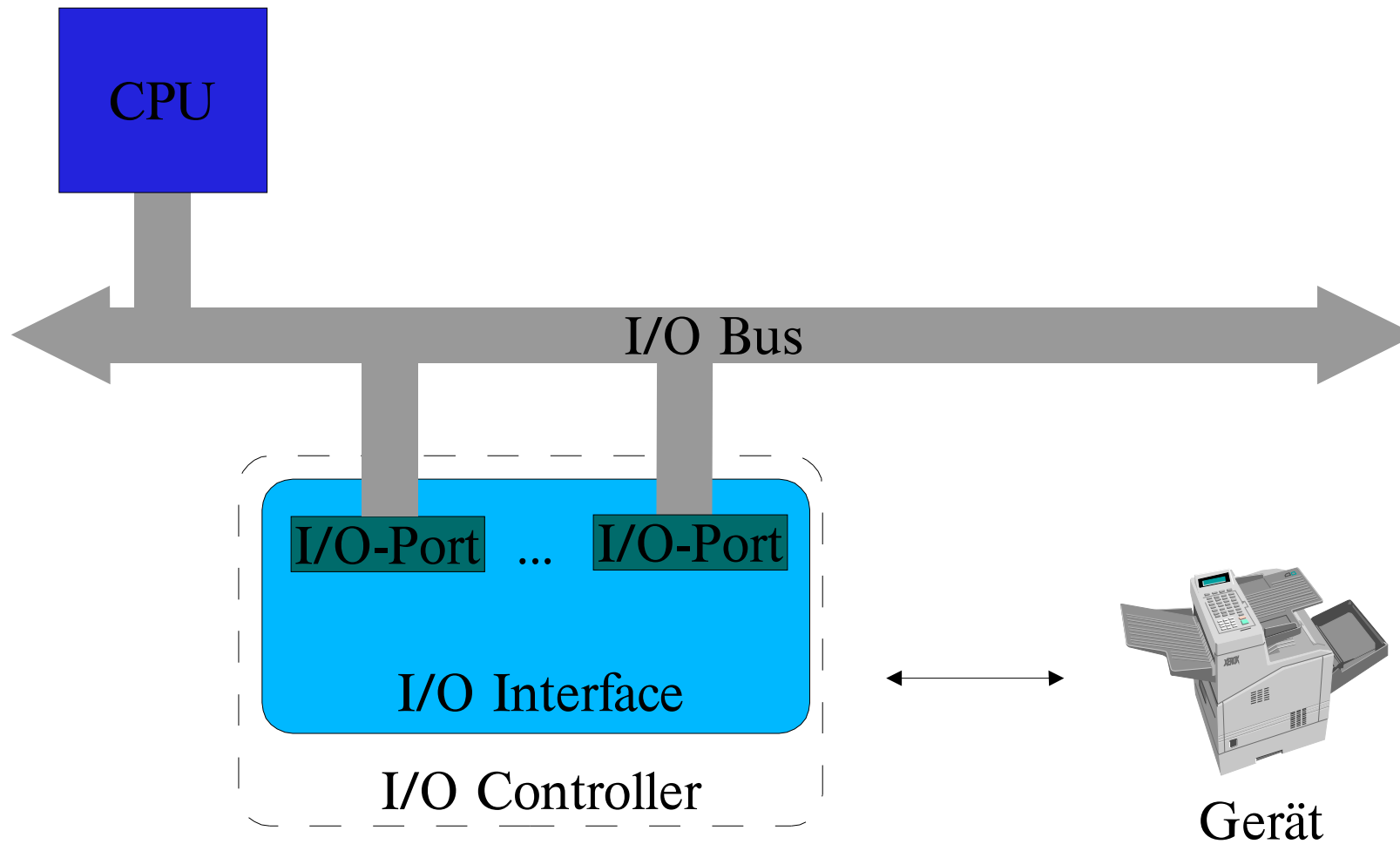


- Treiber holt die empfangenen Daten vom Datenregister ab

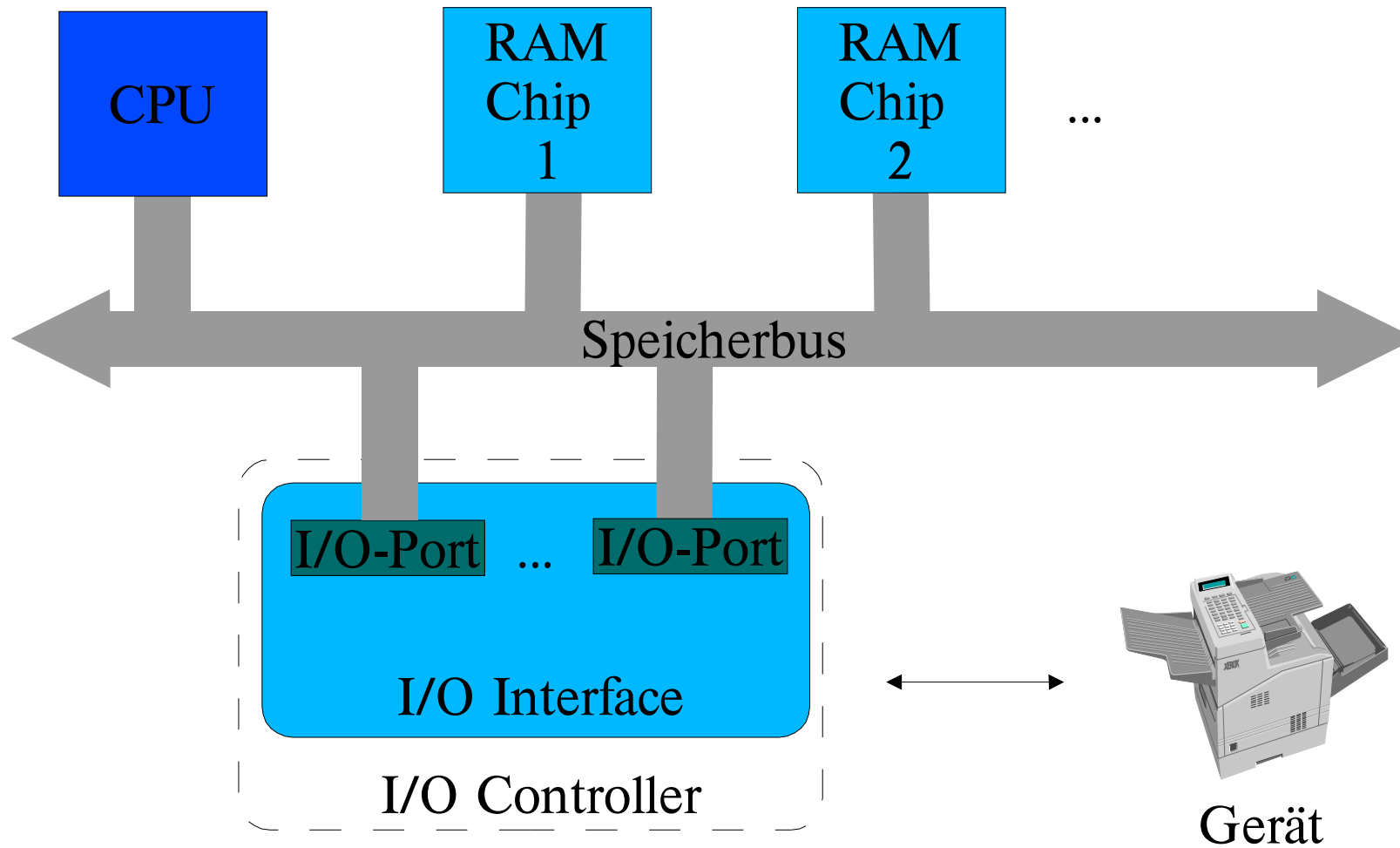
Zugriff auf Control-/Status-Register



Zugriff auf Control-/Status-Register



Zugriff auf Control-/Status-Register (2)

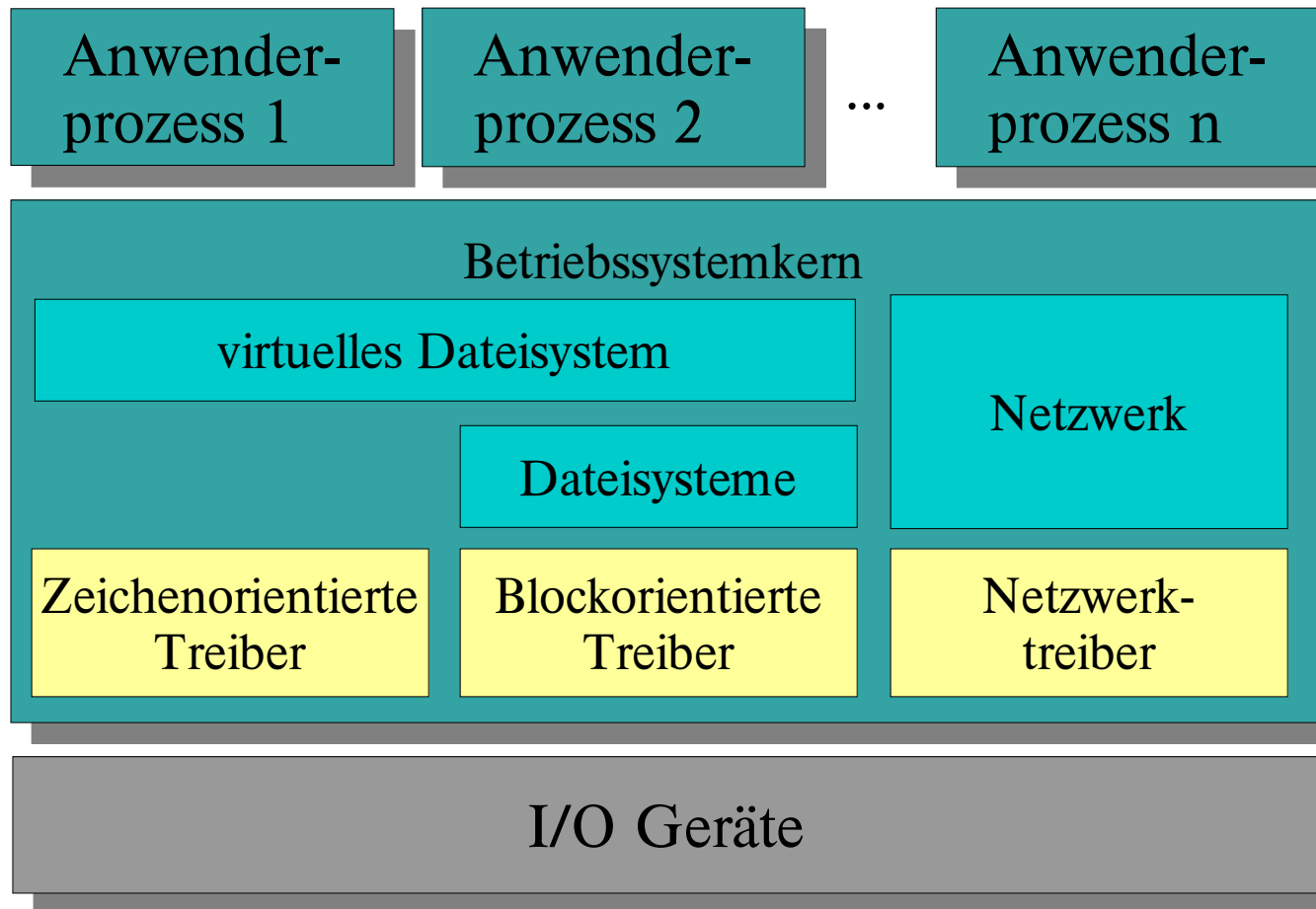




Schnittstelle zum Betriebssystem

- Hängt von Betriebssystem und Gerätetyp ab
- Beispiel: Linux, Kernel 2.4
- Unter Linux sind Treiber im Allgemeinen Bestandteile des Kernels, entweder
 - Fest integriert, oder
 - Als Kernelmodul realisiert

Treiber im Linux-Kernel



Zeichenorientierte Gerätetreiber

- Übertragung von Datenströmen vom/zum Gerät
- z.B. Tastatur, serielle und parallele Schnittstelle
- Verwendung der Schnittstelle für normale Dateien
 - Auf Benutzerebene
 - Auf Kernelebene
- Identifikation innerhalb des Systems durch „major device number“
- Ähnlich der bereits vorgestellten Schnittstelle bei Dateisystemen

Registrierung zeichenorientierter Gerätetreiber

```
int register_chrdev(
    unsigned int major,
    const char* name,
    struct file_operations* fops)
```

- Ausführung während Systemstart oder Laden des Treibermoduls
- Alle registrierten Geräte werden in `/proc/devices` aufgeführt

struct file_operations

- deklariert in `<linux/fs.h>`
- enthält Funktionspointer auf die Implementierung des Treibers
- wesentliche Funktionen:
 - `read`, `write` zum Transfer von Daten
 - `open`, `release` zur Initialisierung/Freigabe
 - `ioctl` zur Realisierung gerätespezifischer Funktionen

Gerätedatei erstellen

➤ Benutzerprogramme erhalten „direkten“ Zugriff auf zeichenorientierte Geräte über Gerätedateien

➤ Erstellung durch

```
mknod <Dateiname> c <major> <minor>
```

➤ „minor device number“ dient der Unterscheidung mehrerer Geräte innerhalb eines Treibers

Blockorientierte Gerätetreiber

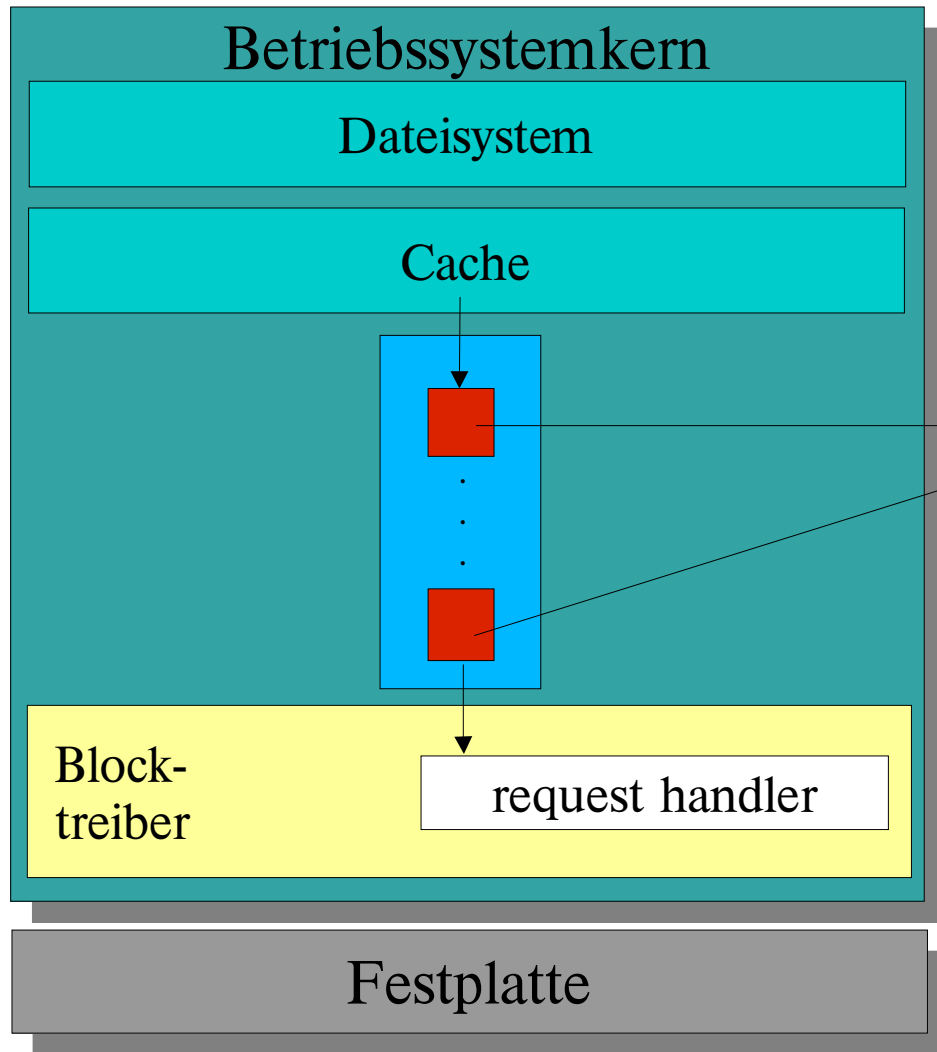
- Datentransfer in Blöcken fester Größe
- Treiber für Festplatten und ähnliche Massenspeicher
- hohe Performanceanforderungen
- Zugriff auf blockorientierte Gerätetreiber erfolgt i.A. implizit in Dateisystemzugriffen
- „major/minor number“ und Erstellung von Gerätedateien analog zu zeichenorientierten Treibern

Registrierung blockorientierter Treiber

```
int register_blkdev(
    unsigned int major,
    const char* name,
    struct block_device_operations* bdops)
```

- analog zu zeichenorientierten Treibern
- struct block_device_operations:
 - Funktionszeiger für open, release und ioctl
 - keine read/write Funktionen

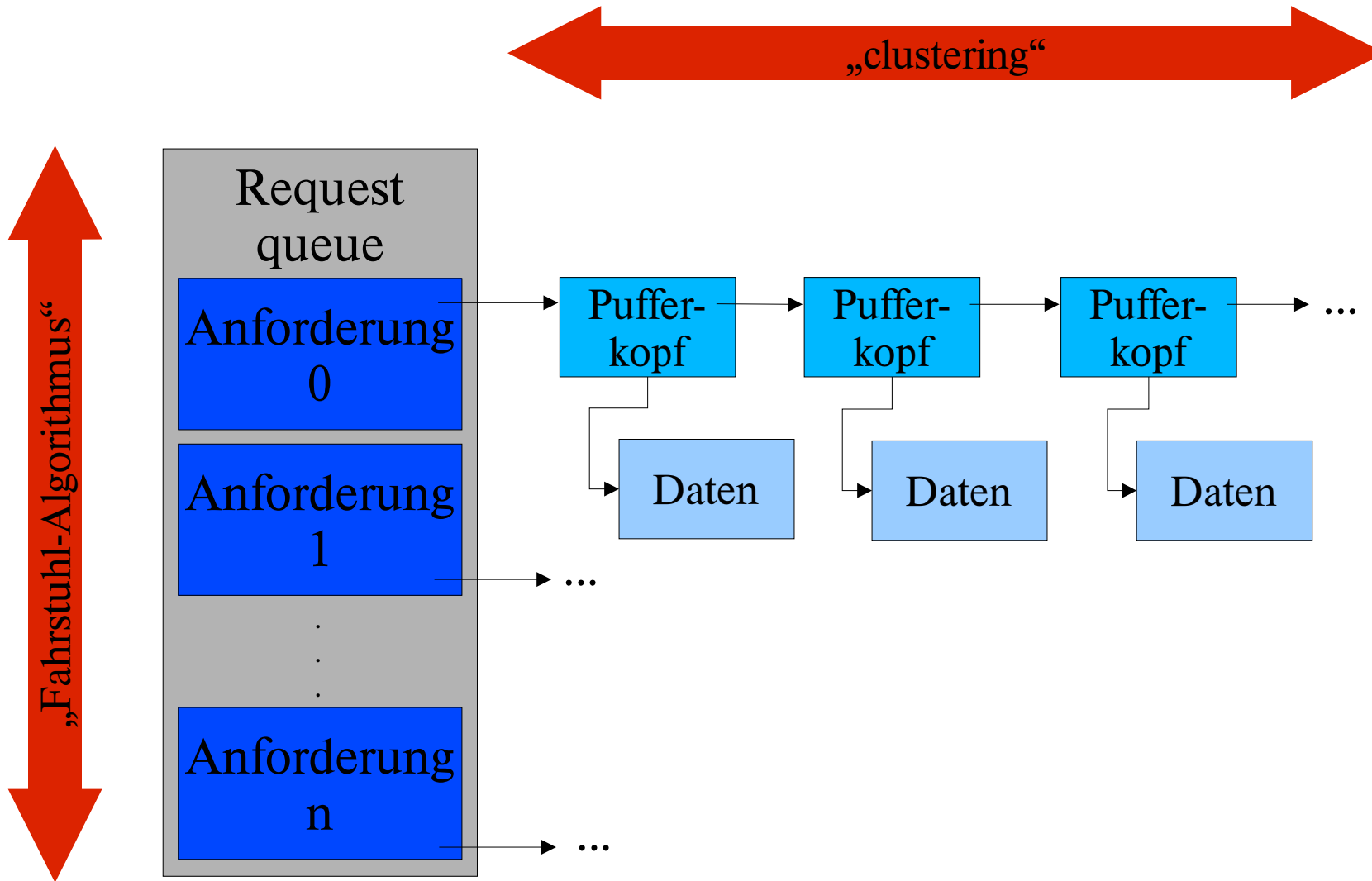
Request queues



➤ Auftrag zum Lesen/Schreiben von Datenblöcken

➤ Sortierung der Aufträge durch den Kernel nach Performanceaspekten

Request queues





Netzwerktreiber

- unterhalb des Netzwerksystems im Kernel
- Benutzerprogramme erhalten keinen direkten Zugriff
- eigene Schnittstelle
- protokollunabhängig



Zusammenfassung

- Gerätetreiber implementieren den Zugriff auf I/O Geräte
- kommunizieren über definierte Schnittstellen mit der Hardware und dem Betriebssystem
- Linux:
 - allgemeine Schnittstelle (zeichenorient. Treiber)
 - spezielle Schnittstellen für Laufwerke und Netzwerkkarten