

# Geräteverwaltung

Unter der Geräteverwaltung fasst man alle Aufgaben und Tätigkeiten des Betriebssystems zusammen, welche einer optimierten Zusammenarbeit zwischen dem Betriebssystem und den Peripheriegeräten dienen.

- Rolle der Geräteverwaltung
- Gerätetreiber
- Geräteklassen

# Rolle der Geräteverwaltung



Wenn Prozesse mit **Peripheriegeräten** kommunizieren,

tätigen sie **Systemaufruf**

→ dadurch wird *Betriebssystem* mit der gewünschten **Kommunikation** beauftragt .

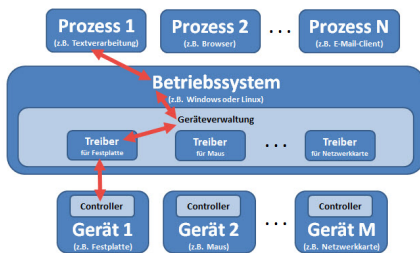
Geräteverwaltung ist Bestandteil des Betriebssystems

→ realisiert eine **Schnittstelle** zwischen den Peripheriegeräten und Betriebssystem

- muss mit unterschiedlichsten Hardware-Geräten kommunizieren können
- Die Geräteverwaltung besitzt sowohl **geräteabhängige, als auch geräteunabhängige Teile**.
  - **geräteabhängigen Teile** sind die sogenannten Gerätetreiber.
  - **geräteunabhängigen Teile** werden durch unterschiedlicher Geräteklassen realisiert

# Gerätetreiber

**Gerätetreiber** ist eine Softwarekomponente, welche Interaktionen zwischen BS und Controller des Peripheriegeräts steuert.



die Geräteverwaltung enthält verschiedene Treiber mit gleichem Ablauf:

1. Initialisierung des Geräte-Controllers
2. Gerät dem Betriebssystem bekannt machen
3. Bereitstellen einer Schnittstelle zum **Geräte-Controller**
4. Interruptbehandlung für ein Gerät
5. Bereitstellen einer Schnittstelle zur **Geräteverwaltung**

## Initialisierung des Geräte-Controllers



**Treiber** initialisiert den *Controller* seines zugehörigen **Geräts** beim **Systemstart**

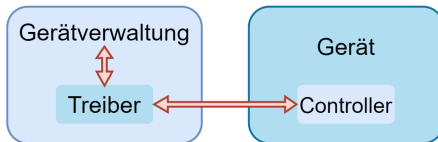
→ Damit stehen gültige Werte in Registern des Controllers

→ das Gerät selbst wird über seinen *Controller* bereit für die **Entgegennahme von Befehlen**

## Gerät dem Betriebssystem bekannt machen

Ein **Treiber** macht der **Geräteverwaltung** des Betriebssystems das **Gerät** bekannt, damit kann das Gerät vom **BS** genutzt werden

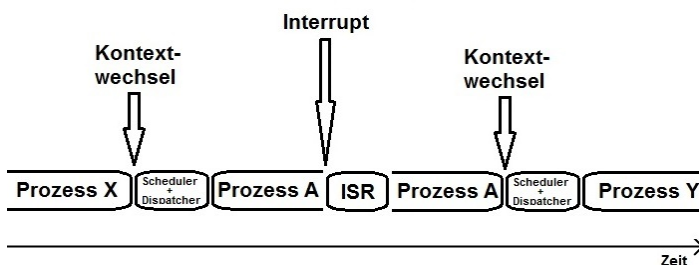
## Bereitstellen einer Schnittstelle zum Geräte-Controller



**Treiber** bildet *Schnittstelle* zwischen **Geratverwaltung** und Controller des **Gerats**  
 → Treiber muss wissen, welche Befehle der Controller zur Verfugung stellt  
 → wie diese anzusprechen und in welcher Weise die Ruckgabewerte zu interpretieren sind.

## Interruptbehandlung fur ein Gerat

**Geratetreiber** ist zustandig fur die *Behandlung von Interrupts*, die durch das zugehorige **Gerat** ausgelost werden.

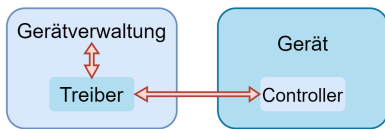


Die Interruptbehandlungsroutine ist Teil des Geratetreibers!  
 Der Treiber ist Teil der Geratverwaltung, welche wiederum ein Teil des BS ist

Auer Festplatten gibt es noch viel weitere Gerate, die von der **Geratverwaltung** des Betriebssystems unterstutzt werden mussen, und fur die ein Treiber eingebunden ist.

Lange nicht alle Gerate unterstutzen den *Einsatz eines DMA-Controllers*!  
 → die Interruptverarbeitung im Treiber kann unterschiedlich ablaufen: **mit DMA oder kein DMA**  
 → das Interruptkonzept eine einfache Reaktionsmoglichkeit des Treibers auf die ungefragt eintreffenden Daten

## Bereitstellen einer Schnittstelle zur Geratverwaltung



**Treiber** muss auch mit der **Geratverwaltung** des

Betriebssystems kommunizieren

- Er ist Mittel zwischen beiden Welten mit **Gerateklassen**, mit jeweils *eigenen Schnittstelle*.
- **Treiber** muss von **Geratverwaltung** fur **Gerateklasse** gedachte Funktionen implementieren

# Geräteklassen

Im Zuge seiner geräteunabhängigen Teile der Implementierung definiert die Geräteverwaltung des Betriebssystems unterschiedliche **Geräteklassen**. Zu den wichtigsten beiden Klassen gehören die **blockorientierten Geräte**, sowie die **zeichenorientierten Geräte**.

---

## Blockorientierte Geräte

übertragen Daten jeweils in kompletten Blöcken, beim Lesen und Schreiben von diesem Gerät (zb. Festplatte)

- Typische Blockgrößen liegen zwischen 512 und 32 768 Byte.
- Jeder Datenblock ist *direkt adressierbar*.
- BEISPIEL: Lade Datenblock Nr. 723 von Festplatte | Ändere fünfte Byte wie | Schreibe Datenblock Nr. 723 zurück auf Festplatte.
- Es können also immer nur **komplette** Datenblöcke gelesen oder geschrieben werden.

---

## Zeichenorientierte Geräte

erzeugen/empfangen Datenstrom. Zeichenfolge zum Gerät oder vom Gerät übertragen. Zeichen sind nicht adressierbar.

Beispiele für zeichenorientierte Geräte sind: Maus, Tastatur, Netzwerkkarte, ...

---

## Schnittstelle für zeichenorientierte / blockorientierte Geräte

- Geräteverwaltung definiert Standardschnittstelle, welche Treiber zeichenorientierten/blockorientierte Geräte unterstützen
- Darin vorgesehen sind beispielsweise Funktionen für: die Initialisierung des Geräts, das Lesen eines Zeichenstroms vom Gerät...
- Diese Funktionen werden vom Gerätetreiber implementiert.

---

## Zusammenarbeit von Treiber und Geräteverwaltung

- ist ein zeichenorientiertes Gerät am System angeschlossen, so wird sein Treiber in den Hauptspeicher geladen
- Der Treiber implementiert Funktionen (initDevice(), readBlock(), writeBlock(), handleInterrupt()).

Durch das Laden des Treibers in den Hauptspeicher steht ab diesem Moment fest, ab welcher Adresse im Hauptspeicher der ausführbare Code der genannten Funktionen beginnt. Dies gilt für jeden geladenen Treiber, also für jedes unterstützte Gerät.