

# Treiberentwicklung, Echtzeit- und Betriebssysteme

Prof. Dr. rer. nat. Peter Gerwinski

26. Mai 2025

# Treiberentwicklung, Echtzeit- und Betriebssysteme

**1 Einführung**

**2 Unix**

**3 Treiberentwicklung**

**4 Speicherverwaltung**

4.1 Mikrocontroller

4.2 Speichersegmentierung

4.3 Speicherschutz (*Protected Mode*)

4.4 Virtueller Speicher

**5 Dateisysteme**

...

## 4 Speicherverwaltung

### 4.1 Microcontroller

Speicherverwaltung „einfach so“

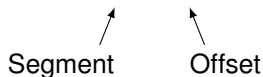
- PC erzeugt Speicher-Layout (z. B. Intel-[.hex](#)-Datei)
- Darin enthalten: Programm, Interrupt-Vektoren, ...
- Aufspielen in den Flash-Speicher

# 4 Speicherverwaltung

## 4.2 Speichersegmentierung

Intel 8086: Segment- und Offset-Adresse

- Beispiel: Speicherzelle **28F5:0100**: für das aktuelle Programm vorgesehen

  
Segment                  Offset

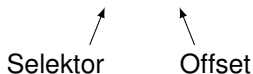
- Speicherzelle **B800:0000**: Bildschirmspeicher (Textmodus)
- physikalische Adresse =  $0x10 \cdot \text{Segment} + \text{Offset}$ , hier also: **B8000**
- Speicher insgesamt: 5 Hex-Ziffern = 20 Bit für Speicher-Adressierung  
→ maximal 1 MiB adressierbar
- „Speicherschutz“: Das Betriebssystem weist jedem Programm ein Segment zu (hier z. B.: **28F5**). Außerhalb davon **sollte** man nicht schreiben.  
(→ effiziente Bildschirmausgabe: direkt in Segment **B800** schreiben)

## 4 Speicherverwaltung

### 4.3 Speicherschutz (*Protected Mode*)

Intel 80286: Selektoren

- Beispiel: Speicherzelle **0007:0100**: für das aktuelle Programm vorgesehen

  
Selektor                      Offset

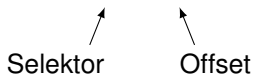
- Selektor = Index für ein Array innerhalb des Prozessors: *Deskriptorentabelle*  
Segment-Deskriptor:
  - physikalische Adresse (z. B.: **28F5**) des Segments
  - Länge des Segments (maximal 64 kiB)
  - Zugriffsrechte: Lesen, Schreiben, Ausführen→ Programme gegeneinander absichern (mit Hardware-Unterstützung)
- In der Praxis (MS-DOS): *kein* Speicherschutz, sondern:  
Jedes Programm verwaltet die Deskriptorentabelle selbst.

## 4 Speicherverwaltung

### 4.3 Speicherschutz (*Protected Mode*)

Intel 80386: 32-Bit-Offsets

- Beispiel: Speicherzelle 0007:00000100



- Selektor = Index für ein Array innerhalb des Prozessors: *Deskriptorentabelle*  
Segment-Deskriptor:
  - physikalische Adresse (z. B.: 28F5) des Segments
  - Länge des Segments (maximal 4 GiB)
  - Zugriffsrechte: Lesen, Schreiben, Ausführen→ Programme gegeneinander absichern (mit Hardware-Unterstützung)
- MS-DOS und MS-Windows bis Version ME: *kein* Speicherschutz, sondern: Jedes Programm verwaltet die Deskriptorentabelle selbst.
- MS-Windows ab Version NT, OS/2, MacOS, Linux und andere Unixe: Speicherschutz mit Hardware-Unterstützung

## 4 Speicherverwaltung

### 4.3 Speicherschutz (*Protected Mode*)

#### Speicherschutz mit Hardware-Unterstützung

- spezielle Variable („Register“) im Prozessor:  
Berechtigungsstufe des aktuell laufenden Programms
- Benutzerprogramme dürfen Befehle zur Manipulation der Deskriptorentabelle nicht ausführen. Dies darf nur der Betriebssystemkern.
- Speicherzugriffe außerhalb des zugewiesenen Segments lösen eine *Exception* aus (ähnlich Interrupt). Das Betriebssystem kann daraufhin das Programm kontrolliert beenden („Speicherzugriffsfehler“).
- Das Betriebssystem kann Code als „nicht ausführbar“ markieren. Versuche, ihn auszuführen, lösen eine Exception aus.

## 4 Speicherverwaltung

### 4.3 Speicherschutz (*Protected Mode*)

Speicherschutz mit Hardware-Unterstützung

Beispiel: Puffer-Überlauf

- Die Rücksprung-Adresse (auf dem Stack) wird durch eine Variable (auf dem Stack) überschrieben
- Gezielter Sprung in die Variable hinein ist möglich.  
offene Sicherheitslücke in alten Versionen von MS-Windows
- Lösung: Stack als „nicht ausführbar“ markieren



## 4 Speicherverwaltung

### 4.3 Speicherschutz (*Protected Mode*)

Speicherschutz mit Hardware-Unterstützung

Beispiel: Puffer-Überlauf

- Die Rücksprung-Adresse (auf dem Stack) wird durch eine Variable (auf dem Stack) überschrieben
- Gezielter Sprung in die Variable hinein ist möglich.  
offene Sicherheitslücke in alten Versionen von MS-Windows
- Lösung: Stack als „nicht ausführbar“ markieren
- Weiterhin möglich: *return-oriented programming*

## 4 Speicherverwaltung

### 4.4 Virtueller Speicher

Hardware-Untersützung durch *Memory Management Unit (MMU)*

- Unterteilung des Speichers in *Seiten* (typischerweise: 4 kiB)
- Zeiger: Zugriff auf *virtuellen Speicher*
- Tabelle innerhalb der MMU: Zuordnung der Seiten zu physikalischem oder nicht zugeordnetem Speicher
- Anwendung: ausgelagerter Speicher, bei Zugriff: Exception  
Betriebssystem kann ausgelagerte Seite bereitstellen

# 4 Speicherverwaltung

## 4.4 Virtueller Speicher

Hardware-Unterstützung durch *Memory Management Unit (MMU)*

- Unterteilung des Speichers in *Seiten* (typischerweise: 4 kiB)
- Zeiger: Zugriff auf *virtuellen Speicher*
- Tabelle innerhalb der MMU: Zuordnung der Seiten zu physikalischem oder nicht zugeordnetem Speicher
- Anwendung: ausgelagerter Speicher, bei Zugriff: Exception  
Betriebssystem kann ausgelagerte Seite bereitstellen
- Weitere Anwendungen:
  - *Shared Memory*
  - Datei in Speicher abbilden
  - kontrollierter Direktzugriff auf Speicher

# 4 Speicherverwaltung

## 4.4 Virtueller Speicher

Hardware-Untersützung durch *Memory Management Unit (MMU)*

- Unterteilung des Speichers in *Seiten* (typischerweise: 4 kiB)
- Zeiger: Zugriff auf *virtuellen Speicher*
- Tabelle innerhalb der MMU: Zuordnung der Seiten zu physikalischem oder nicht zugeordnetem Speicher
- Anwendung: ausgelagerter Speicher, bei Zugriff: Exception  
Betriebssystem kann ausgelagerte Seite bereitstellen
- Weitere Anwendungen:
  - *Shared Memory*
  - Datei in Speicher abbilden
  - kontrollierter Direktzugriff auf Speicher
- Hardware-Fehler können zu Sicherheitslücken führen,  
z. B. *Meltdown* (2017)

# 4 Speicherverwaltung

## 4.4 Virtueller Speicher

Hardware-Unterstützung durch *Memory Management Unit (MMU)*

- Unterteilung des Speichers in *Seiten* (typischerweise: 4 kiB)
- Zeiger: Zugriff auf *virtuellen Speicher*
- Tabelle innerhalb der MMU: Zuordnung der Seiten zu physikalischem oder nicht zugeordnetem Speicher
- Anwendung: ausgelagerter Speicher, bei Zugriff: Exception  
Betriebssystem kann ausgelagerte Seite bereitstellen
- Weitere Anwendungen:
  - *Shared Memory*
  - Datei in Speicher abbilden
  - kontrollierter Direktzugriff auf Speicher
- Hardware-Fehler können zu Sicherheitslücken führen,  
z. B. *Meltdown* (2017)