

# Echtzeitbetriebssysteme

## **Speicherverwaltung (*Memory Management*)**

**Aufgaben der Memory-Management-Unit ist**

- **der Speicherschutz und**
- **die Adressumsetzung**

**Wird durch Hardware unterstützt**

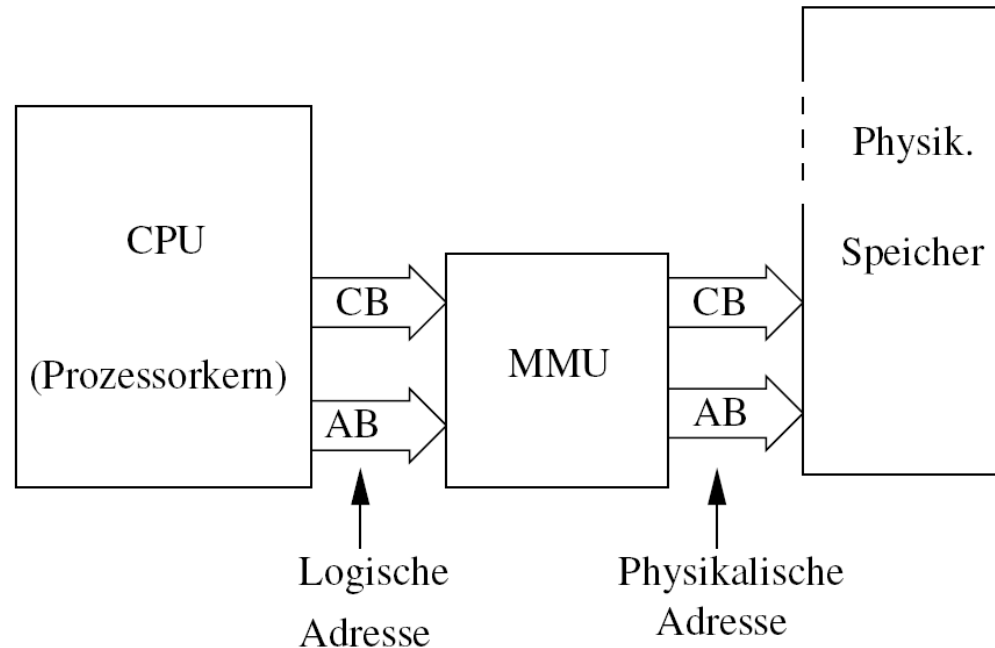
- **Memory Management Unit (MMU)**
- **MMU wird vom Betriebssystem konfiguriert  
(Speicherabbildungstabellen)**



# Echtzeitbetriebssysteme

## Speicherverwaltung (*Memory Management*)

### Struktur einer MMU



MMU = Memory Management Unit

AB=Adreßbus

CB=Controlbus



## Speicherverwaltung: Speicherschutz

- Jeder Prozess (nicht Threads) hat eigenen Adressraum (Prozessadressraum)
- Zugriff nur auf eigene Daten-, Stack- und Codesegmente
- Zugriff auf nicht abgebildete Adresse: MMU erzeugt Interrupt (*Segmentation Fault*)



# Echtzeitbetriebssysteme

## Speicherverwaltung : Keine Adressumsetzung

Alle Programme zur Link-Zeit bekannt: Linker kann unterschiedliche Adressbereiche pro Programm zuordnen und Sprungadressen (Funktionsaufrufe) auflösen, ein Executable (*Image*)

Mehrere Programme dynamisch zur Laufzeit im Hauptspeicher -> unterschiedliche, zur Ladezeit festzulegende Programmadressen (Sprünge bei Funktionsaufrufen) erforderlich

Laden: (Programm in Hauptspeicher bringen; zu Prozess machen)

- Lader (*loader*): Ersetzt Adressen (Sprünge absolut, Variablenzugriffsadressen absolut) zur Lade-Zeit eines Programms -> aufwändig
- Sonderfall: Speicherposition unabhängiger Code → Compiler erzeugt nur Sprungbefehle relativ zum PC und Variablenzugriffe relativ zu einem einmal geladenen danach „readonly“  
Indexregisterinhalt → Lader: Indexregister setzen beim Start



# Echtzeitbetriebssysteme

## Speicherverwaltung : Adressumsetzung

### Einheitlicher (virtuellen) Adressraum für Programme

- **Virtueller Adressraum beginnt bei 0, umfasst kompletten adressierbaren (Adressbusbreite) Adressbereich *für jedes Programm***
- **Linker legt virtuelle Adressen in Executable fest, Adressen werden nicht mehr verändert, nur durch MMU auf reale Adressen abgebildet**  
→ **schnelles Laden, da keine Veränderung des Executables erforderlich (lediglich Konfiguration der MMU, Speichertabellen)**
- **Ermöglicht leichte Verwendung von Shared Libraries: Mehrere Tasks teilen sich ein (Code-) Segment (beliebige virtuelle Adressen durch Linker vergeben, Abbildung auf bereits geladene Shared Library durch MMU)**  
→ **reduziert Hauptspeicherbedarf**



# Echtzeitbetriebssysteme

## Speicherverwaltung Adressumsetzung

### Einheitlicher (virtueller) Adressraum für Programme

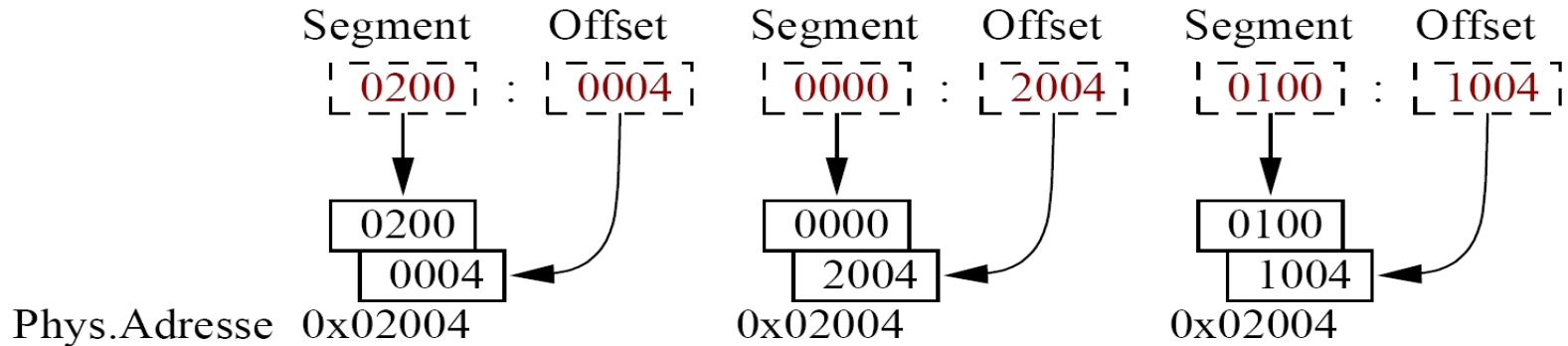
- **Physikalischer Adressraum kann kleiner sein als Virtueller Adressraum**
- **Abbildung durch Swappen und Paging**
- **Swappen: Auslagern des zugeordneten physikalischen Speichers eines gesamten Prozesses auf den Hintergrundspeicher**
- **Paging: Auslagern selten benutzter Speicherseiten (z.B. 4kByte) auf Hintergrundspeicher**
- **Kaum Bedeutung bei Echtzeitsystemen: führt zu (zeitlichem) Nichtdeterminismus**



# Echtzeitbetriebssysteme

## Speicherverwaltung (*Memory Management*)

### Segmentierung (x86)



Normalisieren:

```
SegmentRegister = SegmentRegister + (Offset >> 4);  
Offset = Offset & 0x000F;
```

```
Phys.Adresse = SegmentRegister * 16 + Offset ;
```

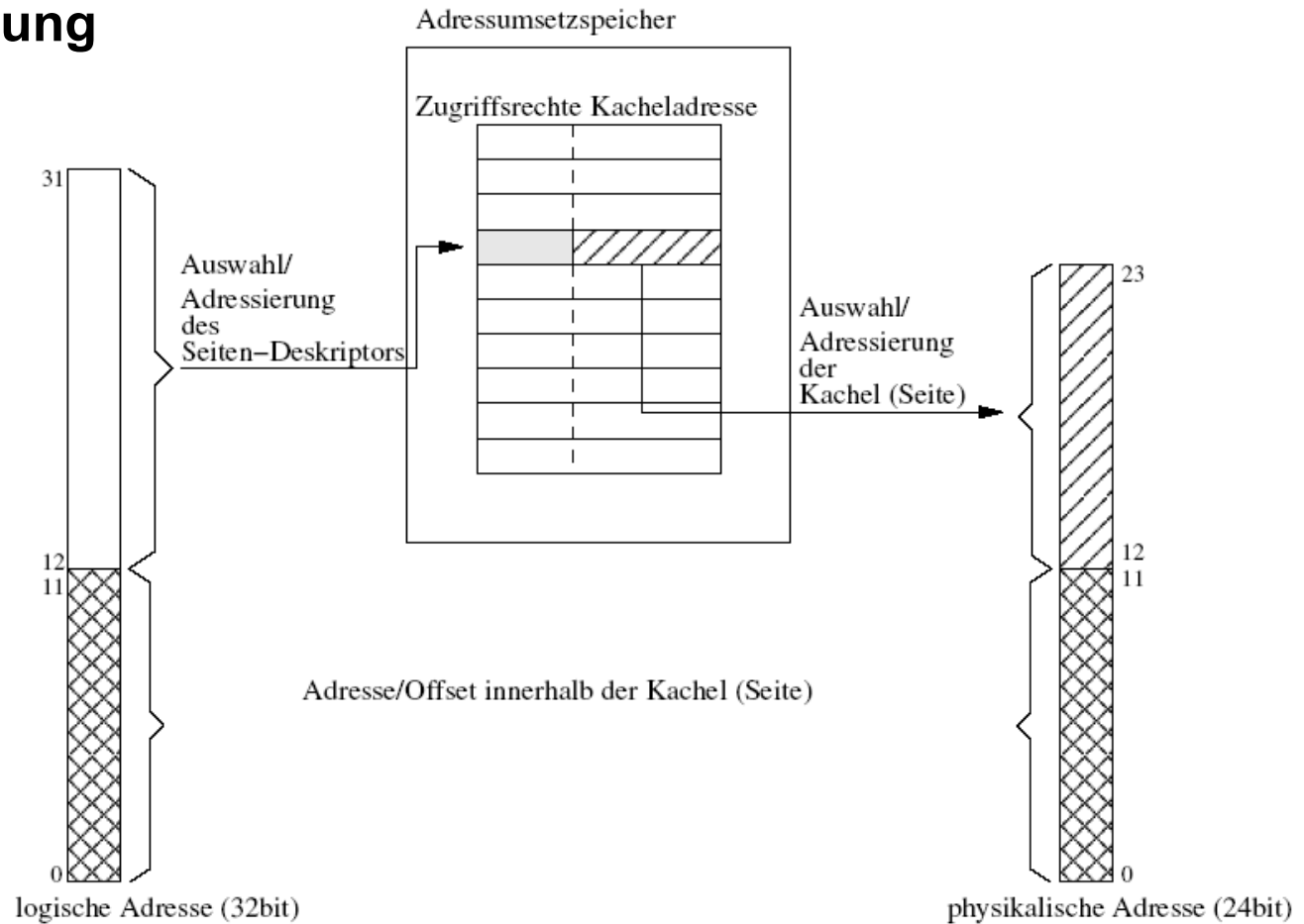
**Pointervergleich: Normalisieren (Umrechnen in „lange“ Adresse); ist aufwändig**



# Echtzeitbetriebssysteme

## Speicherverwaltung, Seitenorganisation

### Adressabbildung

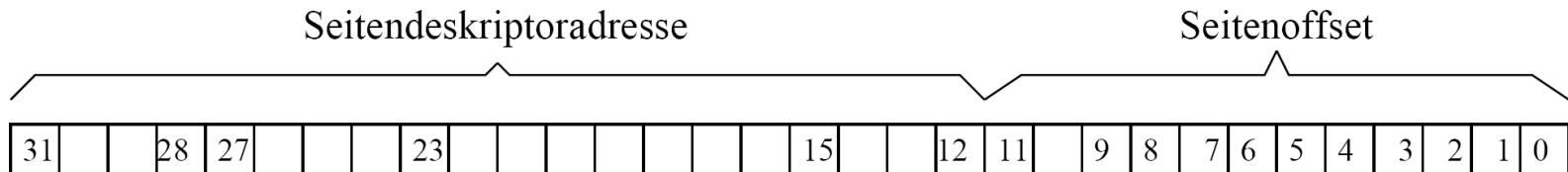
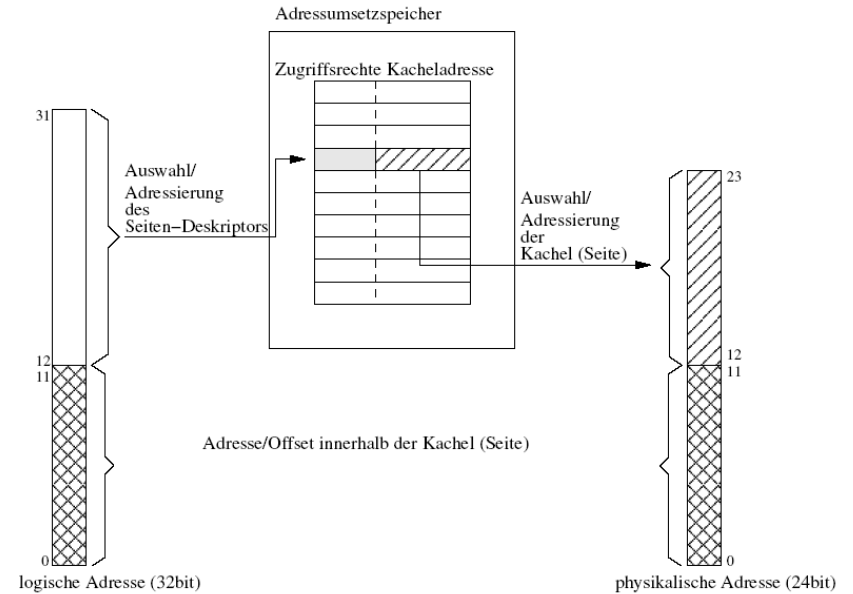




# Echtzeitbetriebssysteme

## Speicherverwaltung, Seitenorganisation

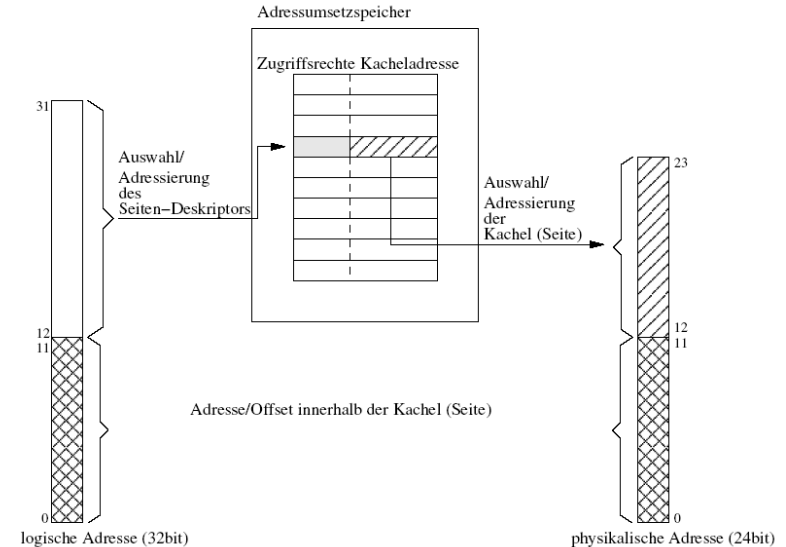
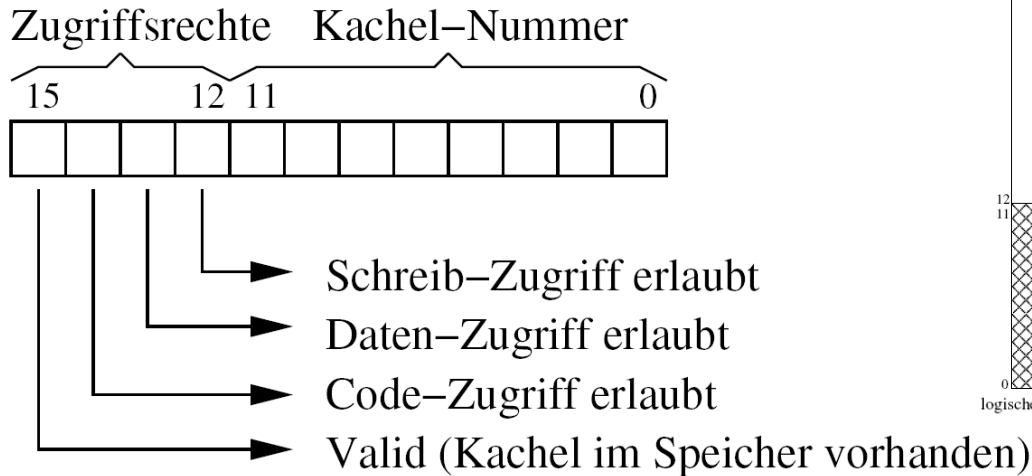
### Virtuelle Adresse



# Echtzeitbetriebssysteme

## Speicherverwaltung, Seitenorganisation

### Seitendeskriptor



### Schreibflag

**1: auf die Seite darf schreibend zugegriffen werden**

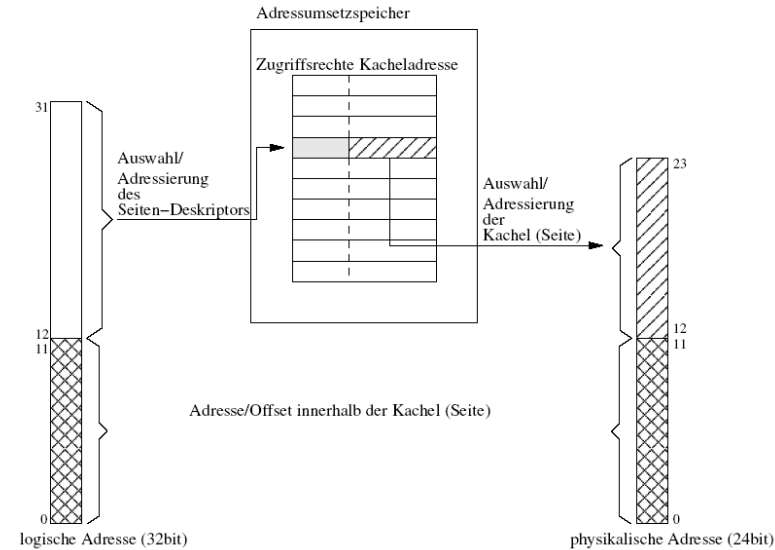
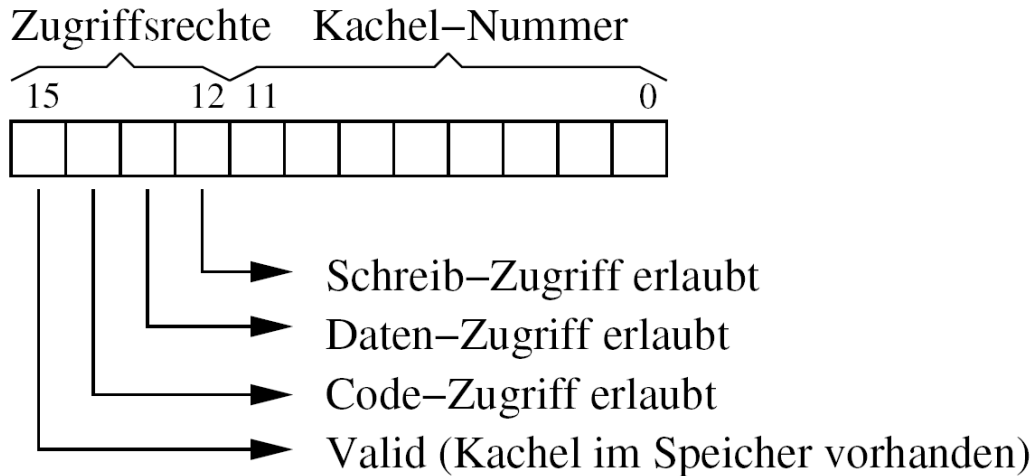
**0: Versuchter Schreibzugriff führt zu Bus-Error**



# Echtzeitbetriebssysteme

## Speicherverwaltung (*Memory Management*)

### Seitenorganisation



### Daten-Zugriff erlaubt

1: Daten dürfen aus der Kachel gelesen werden

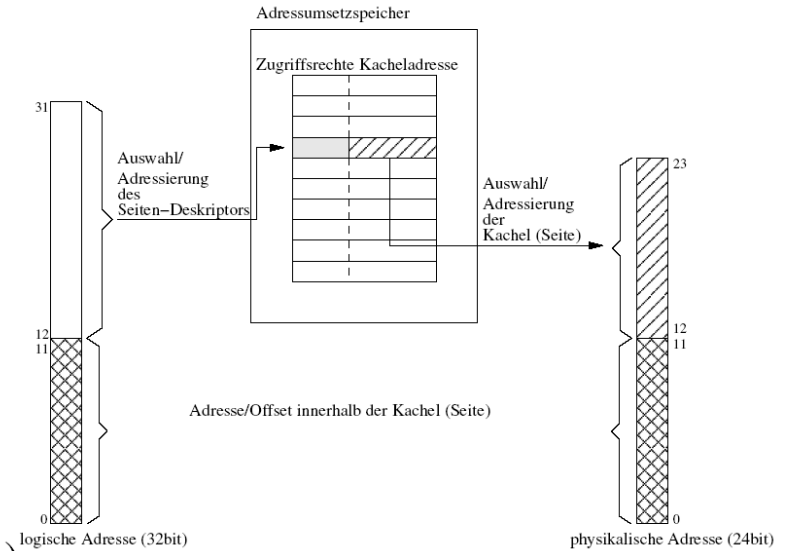
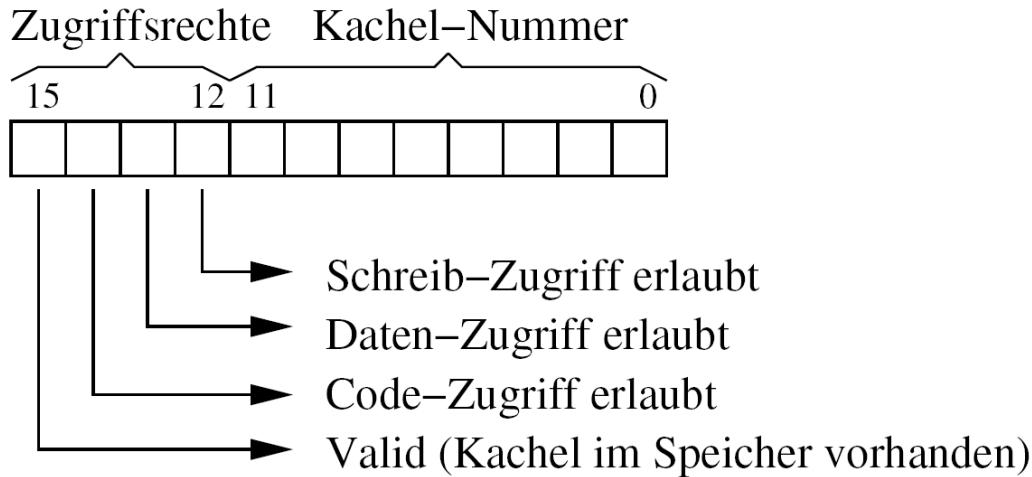
0: Datenzugriff führt zu Bus Error



# Echtzeitbetriebssysteme

## Speicherverwaltung (*Memory Management*)

### Seitenorganisation



### Code-Zugriff

1: Prozessor darf Inhalt als Befehl ausführen

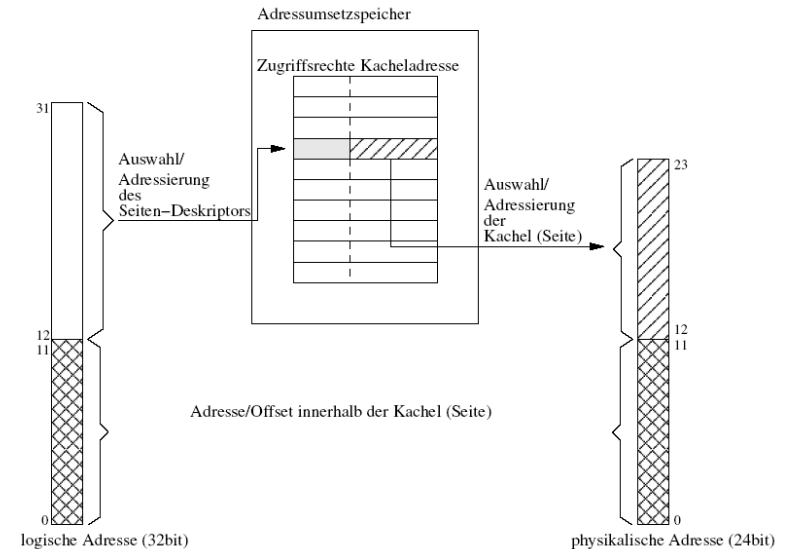
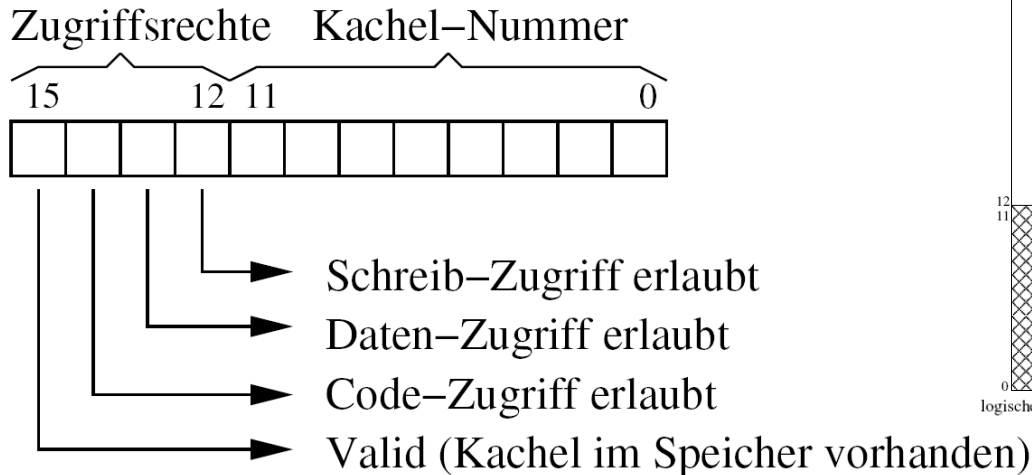
0: Versuch, Inhalt als Code auszuführen führt zu Bus Error



# Echtzeitbetriebssysteme

## Speicherverwaltung (*Memory Management*)

### Seitenorganisation



### Kachel im Speicher

**1: Seite befindet sich im Speicher**

**0: Seite ist ausgelagert; „Seite-Fehlt-Alarm“ (Hardware-Interrupt) wird ausgelöst: Kernel muss Seite laden**



# Echtzeitbetriebssysteme

**Fertig mit der Speicherverwaltung  
Weiter geht's mit dem IO-System**

