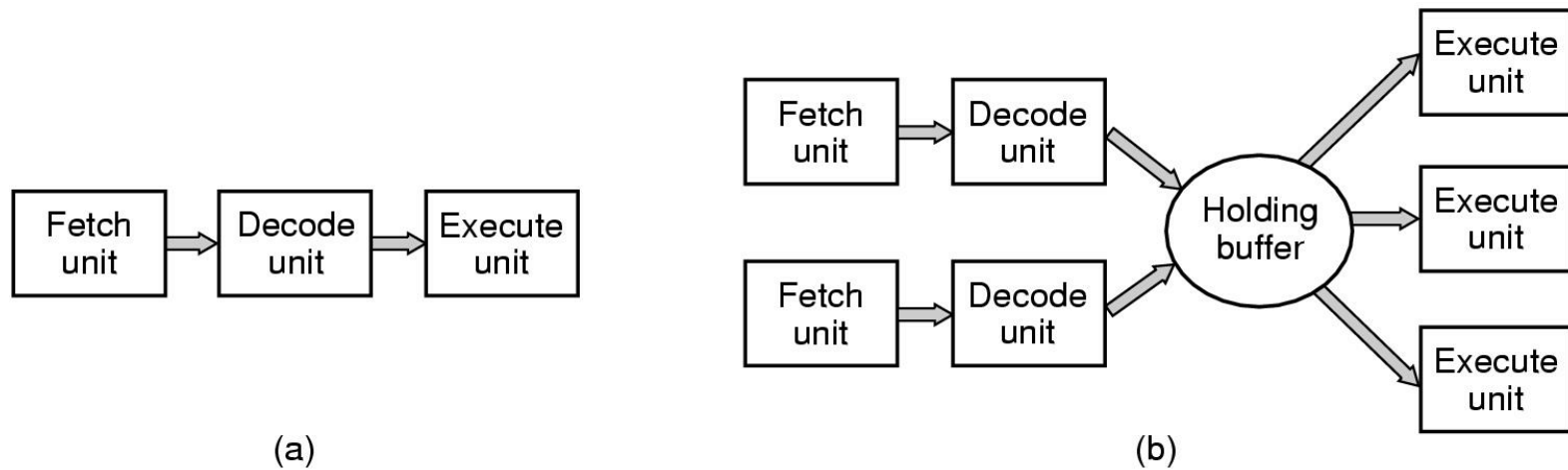


# Systemadministration Teil 3

Prof. Dr.-Ing. Jörn Schneider

# WIEDERHOLUNG

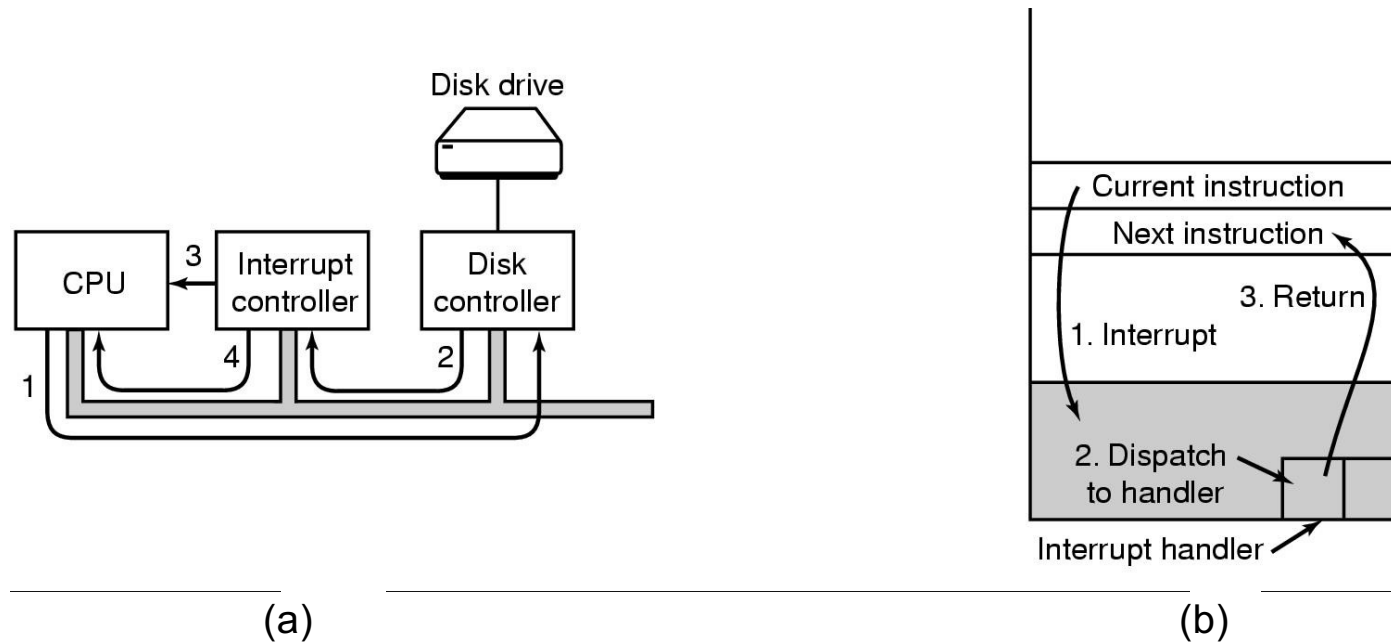
# Computer Hardware: Pipelines



(a) A three-stage pipeline

(b) A superscalar CPU

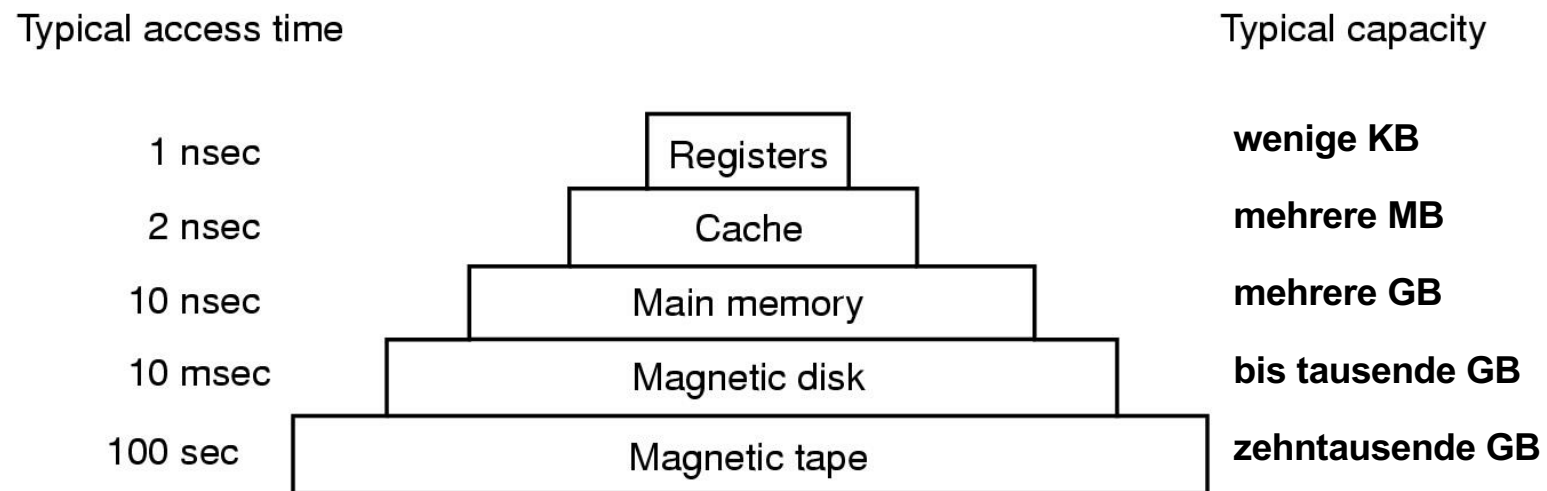
# Interrupt getriebene I/O-Kommunikation



(a) Steps in starting an I/O device and getting interrupt

(b) How the CPU is interrupted

# Computer Hardware: Memory Hierarchy



- Typical memory hierarchy
  - numbers shown are rough approximations

# ENDE WIEDERHOLUNG

## Teil 3

- Was ist ein Rechnersystem?
- Was ist ein Betriebssystem?
- Aufgaben eines Systemadministrators
- Rechneraufbau
- **Betriebssystemkonzepte**

# Betriebssystemkonzepte

- Prozesse
- Adressraum
- Files
- Protection

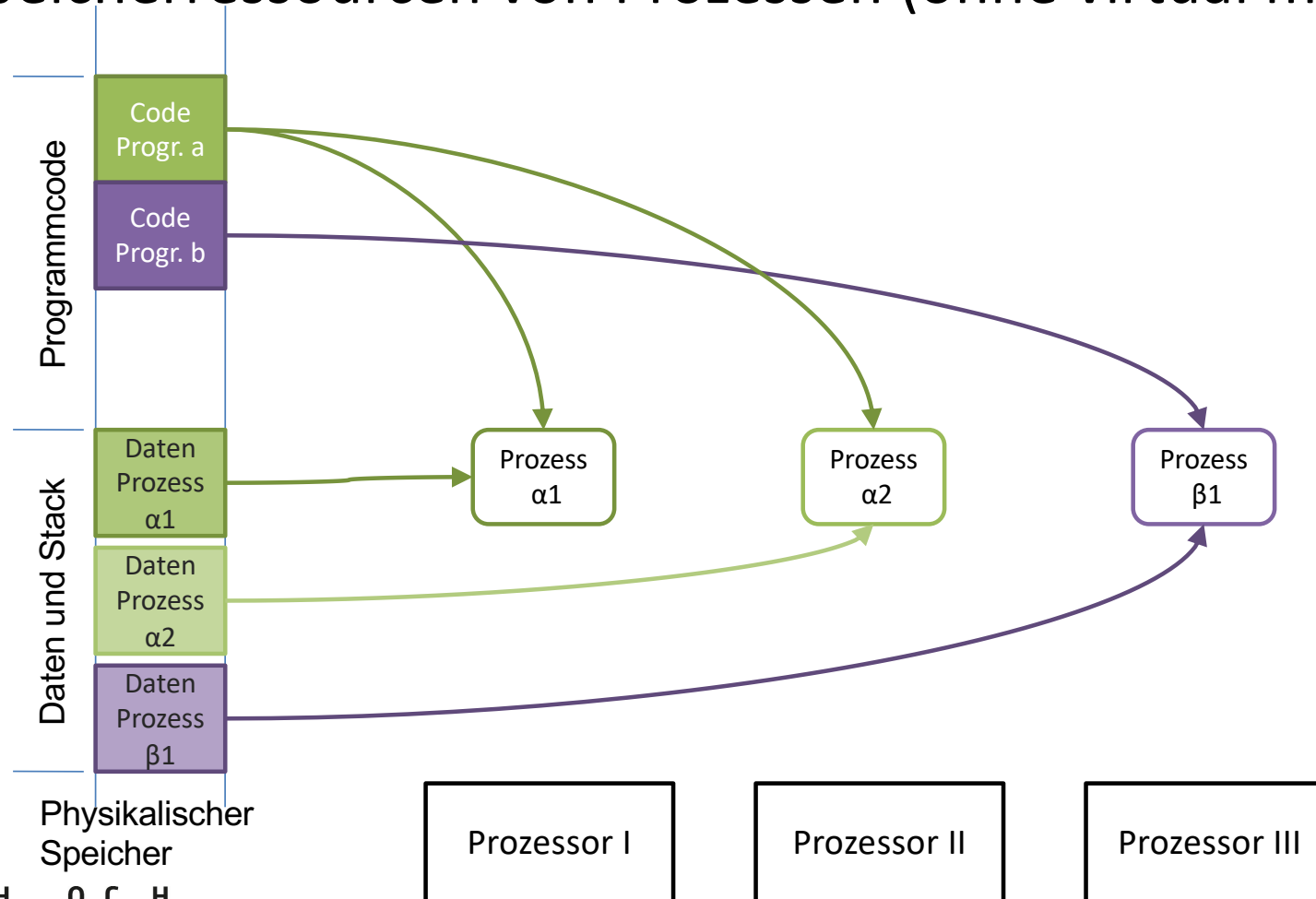


# PROZESSE

# Prozess

- Instanz eines Programmes mit eigenem Ausführungskontext
- Zum Ausführungskontext gehören Ressourcen wie:
  - Befehlszähler
  - Inhalt der General Purpose Register (= normale Register)
  - Inhalt von Statusregistern
  - Stack
  - Sonstige Daten im Hauptspeicher
  - Informationen über offene Dateien, etc.

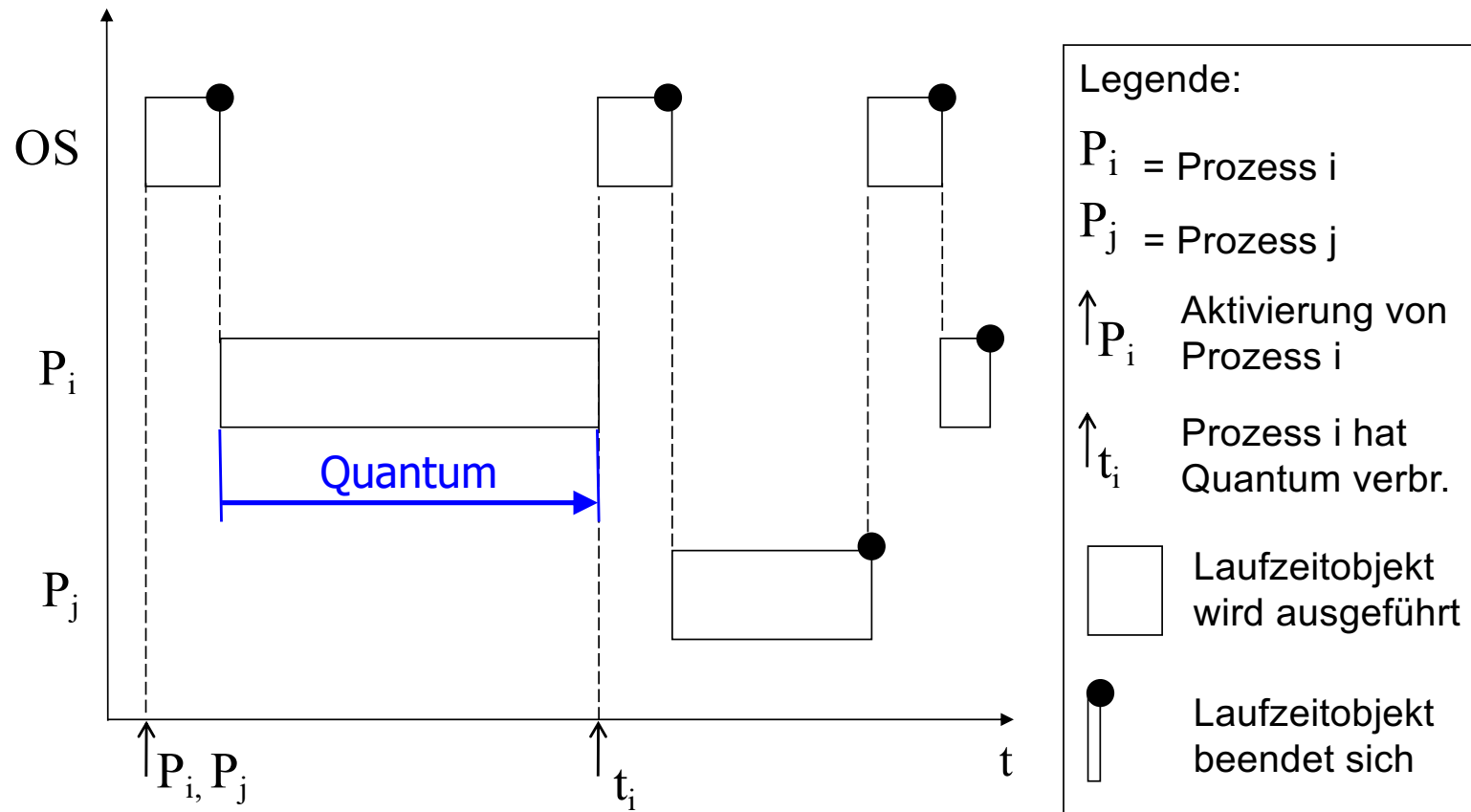
# Speicherressourcen von Prozessen (ohne virtual memory)



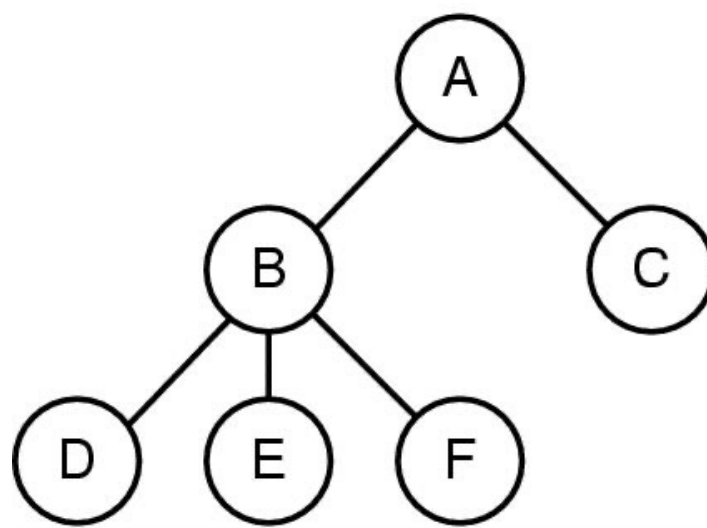
# Multiprocessing

- Es können mehrere Prozesse gleichzeitig aktiv sein, pro Prozessorkern kann aber zu einem gegebenen Zeitpunkt nur ein Prozess ausgeführt werden
- Das Betriebssystem entscheidet welche Prozesse auf welchen Prozessoren ausgeführt werden, das nennt man Prozess *Scheduling*
- Je nach Schedulingverfahren werden Prozesse von anderen Prozessen unterbrochen
  - Beispiele:
    - Videocodierung, Web surfen und Maileingang kontrollieren
    - Shell, „ls /etc | more“

# Beispiel für Ablauf im Rechner (Single Core mit Round-Robin Scheduling)



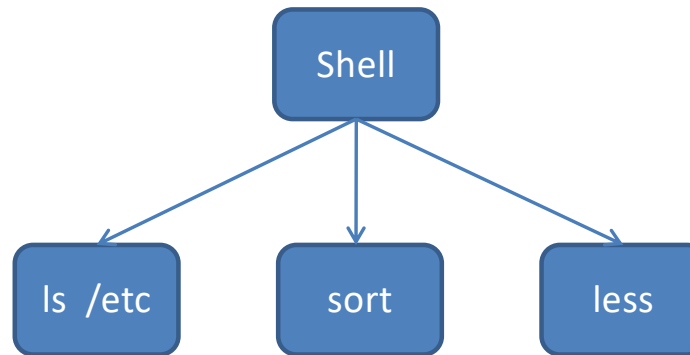
# Prozesshierarchie



- A process tree
  - A created two child processes, B and C
  - B created three child processes, D, E, and F

# Beispiel Prozesshierarchie

- „ls /etc | sort | less“

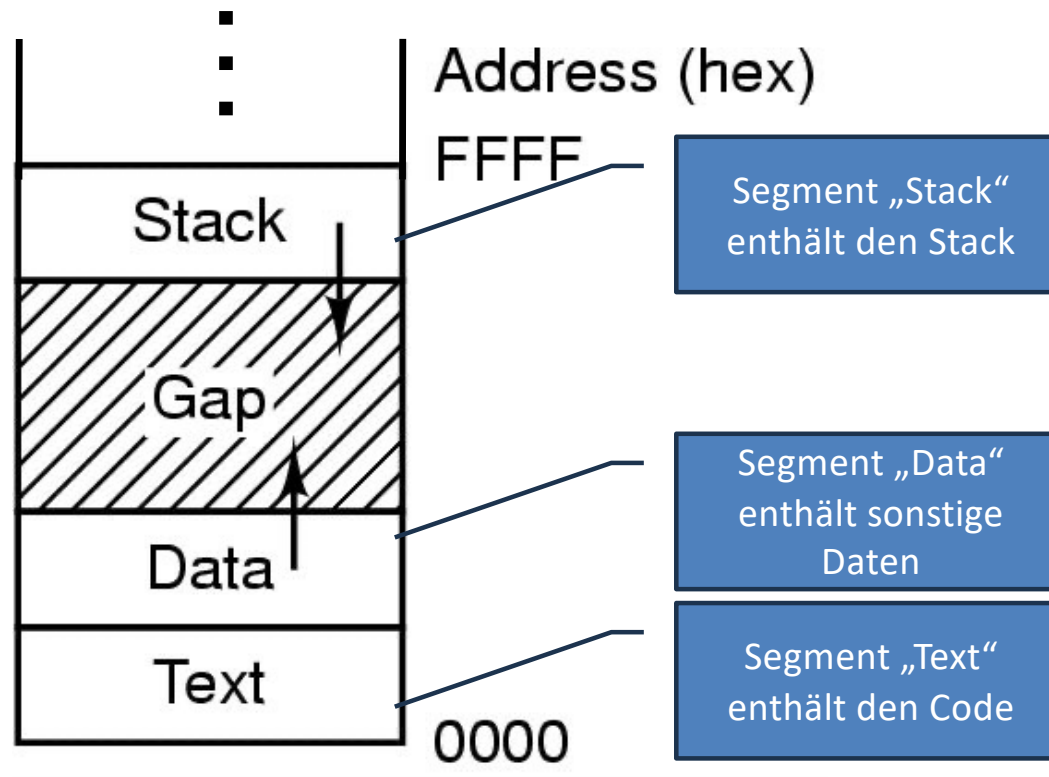


# ADRESSRAUM



# Speichersegmente

Prozesse organisieren ihren  
Adressraum in drei  
Segmenten: Text, Data, Stack

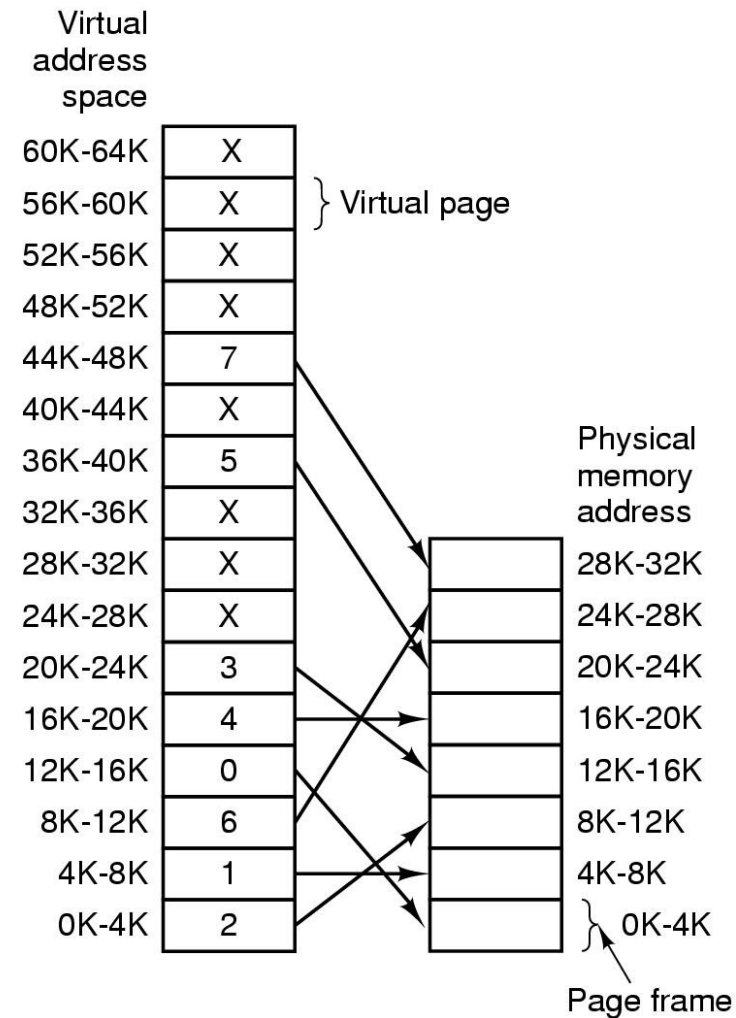


# Adressraum

- Jeder Prozess hat seinen eigenen Adressraum (Virtual Memory)
- Bsp. 32-Bit Rechner:  $2^{32}$  Speicherstellen adressierbar
- Spezielle Hardware (MMU – Memory Management Unit) übersetzt virtuelle Adressen in physikalische Adressen
- Prozess benötigt mehr Speicher als verfügbar  
➔ Nachladen aus Hintergrundspeicher (Festplatte)  
diesen Vorgang nennt man *Paging*

# Paging

The relation between virtual addresses and physical memory addresses given by page table



# Paging: Ablaufschritte

1. Bei Page Fault wird Trap (Interrupt) ausgelöst
2. OS sichert wenig benutzte Page auf Festplatte
3. OS lädt angefragte Page in Speicher
4. OS ändert MMU Mapping, d.h. die Seitentabelle des betroffenen Prozesses
5. OS kehrt zum Befehl zurück, der Trap auslöste\*

\* Nachdem der unterbrochene Prozess wieder vom Scheduler zur Ausführung gebracht wird.

# Paging: Least Recently Used (LRU) Verfahren

- Algorithmus zur Auswahl zu ersetzender Pages im physikalischen Speicher. Der Algorithmus arbeitet wie folgt:
- Zugriff auf im physikalischen Speicher vorhandene Page:
  - Zugriffene Page wird jüngste Page
  - alle, die zuvor jünger waren altern um eins
  - das Alter der sonstigen Pages bleibt unverändert
- Zugriff auf nicht vorhandene Page:
  - Page wird in physikalischen Speicher aufgenommen
  - Neue Page wird jüngste Page
  - die anderen Pages altern um eins
  - Falls Verdrängung notwendig:
    - neue Page nimmt Speicherstelle der ältesten Page ein (diese wird verdrängt)

# Aufgabe: Paging mit LRU

- Gegeben folgende Page Table (Zuordnung von virtuellem Speicher (links) zu physikalischem Speicher (rechts)):

28K-32K	0
24K-28K	X
20K-24K	X
16K-20K	3
12K-16K	X
8K-12K	X
4K-8K	1
0K-4K	2

Zahl ist Nummer des  
Seitenrahmens, in dem sich  
die jeweilige Seite befindet

Die Zugriffreihenfolge  
bestimmt das Alter der  
Seiten

- Bisherigen Zugriffe gingen auf die Adressen: 3K, 7K, 31K und 19K.
- Neue Zugriffe in folgender Reihenfolge: 5K, 30K, 13K und 21K
- **Welche Pages sind im phys. Speicher?**
- **Zeichne neue Zuordnung!**