

Kapitel 9

Cache-Speicher

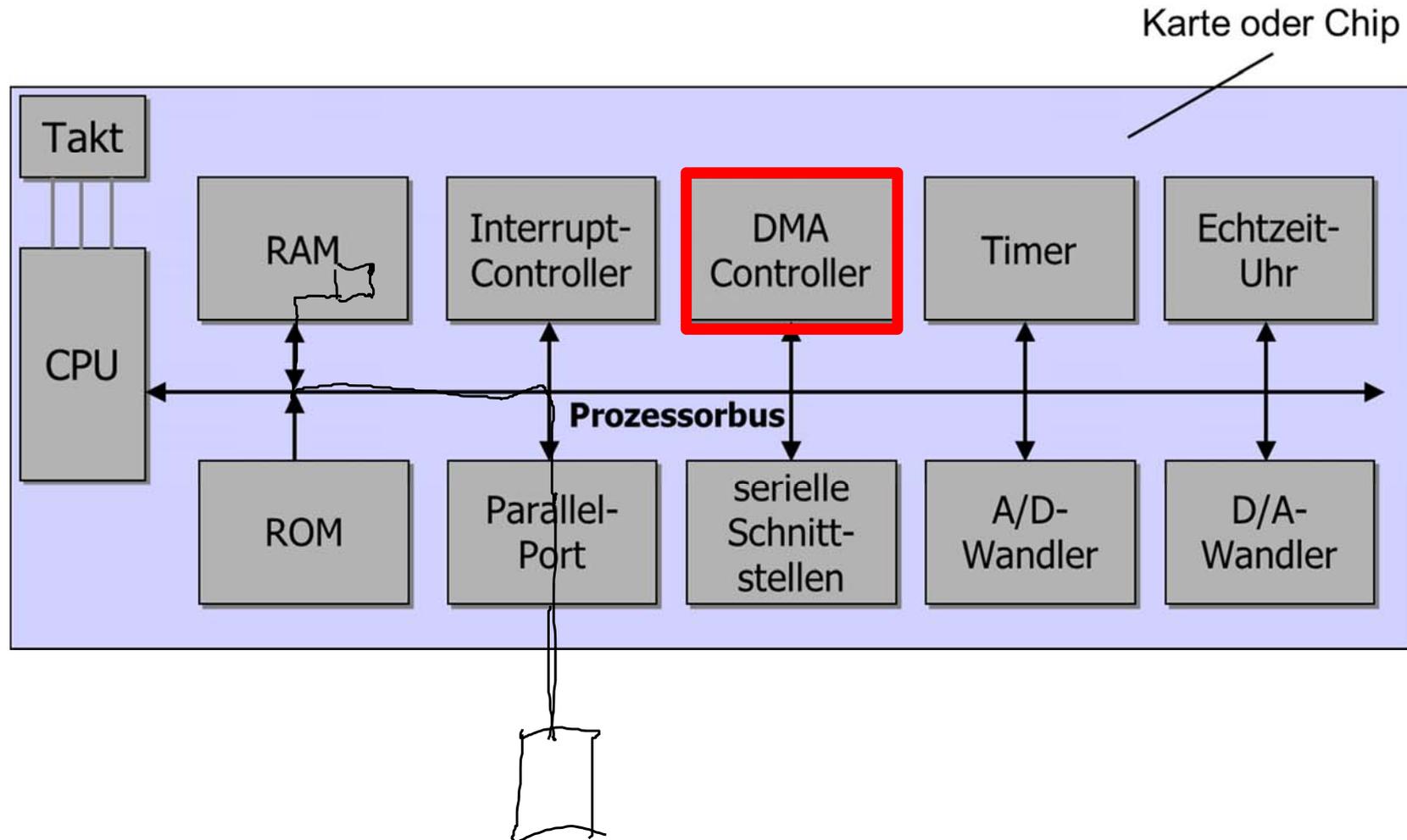
9.2 Cache-Speicher

■ Gültigkeitsproblem

- wenn mehrere Verarbeitungskomponenten jeweils unabhängig voneinander auf Speicherwörter des Hauptspeichers zugreifen können.
- Mehrere Kopien des gleichen Speicherwortes müssen miteinander in Einklang gebracht werden.
- Eine Cache-Speicherverwaltung heißt **cache-kohärent**, wenn ein Lesezugriff immer den Wert des zeitlich letzten Schreibzugriffs auf das entsprechende Speicherwort liefert.

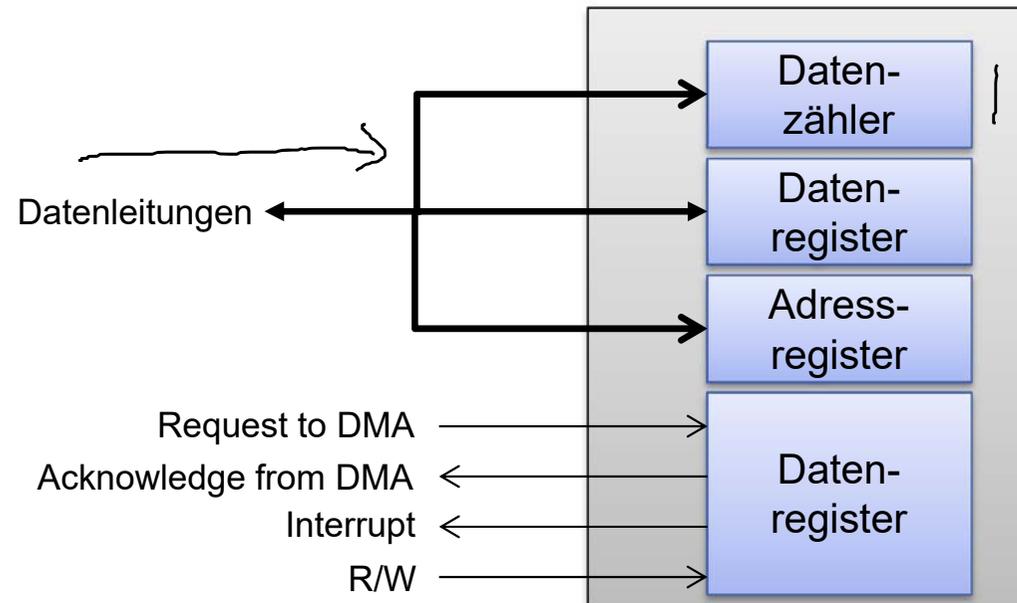
9.2 Cache-Speicher

■ Mikroprozessorsystem mit DMA-Controller



9.2 Cache-Speicher

- **Direkter Speicherzugriff (Direct Memory Access, DMA)**
 - DMA Controller (DMA Steuerbaustein, DMA Einheit):
 - Führt Datenübertragung unabhängig vom Prozessor durch



9.2 Cache-Speicher

■ Direkter Speicherzugriff (Direct Memory Access, DMA)

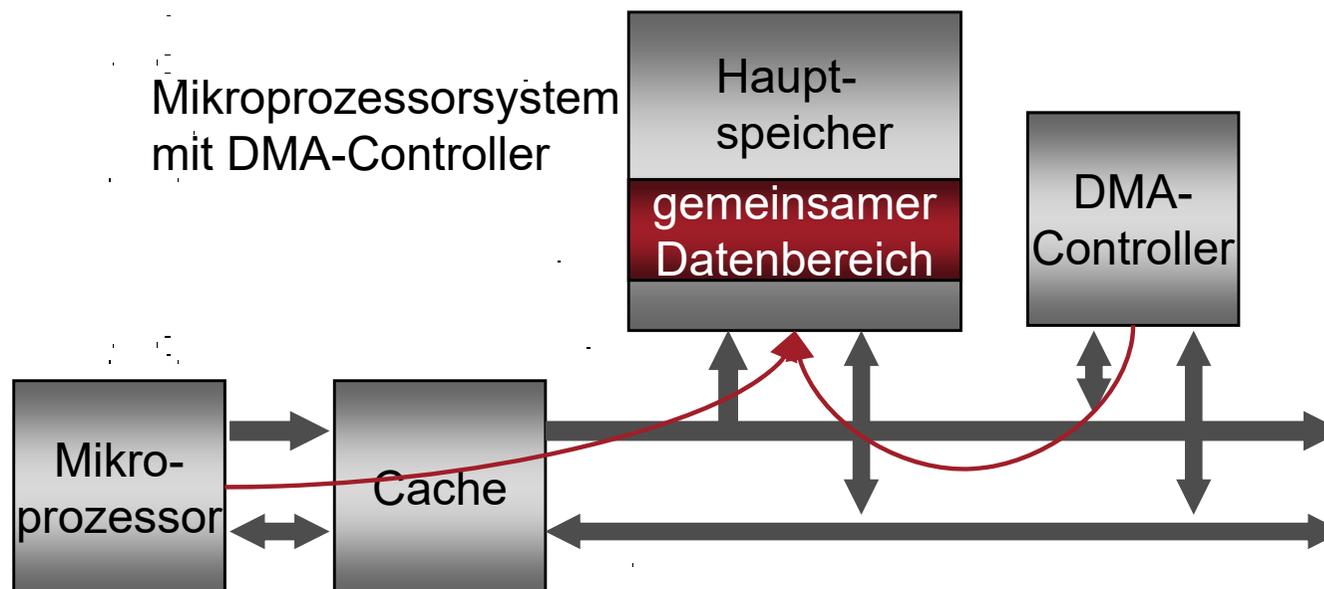
■ DMA Controller (DMA Steuerbaustein, DMA Einheit):

- Wenn der Prozessor einen Block von Daten lesen oder schreiben möchte, initiiert er ein entsprechendes Kommando an die DMA-Einheit mit folgenden Informationen:
 - Lese-/Schreib-Steuerleitung
 - Adresse der Ein-/Ausgabeschnittstelle, die betroffen ist (über die Datenleitungen)
 - Startadresse im Speicher, ab der gelesen oder geschrieben werden soll
 - Wird über die Datenleitungen übertragen und im Adressregister der DMA-Einheit gespeichert
 - Anzahl der Wörter
 - Wird ebenfalls über die Datenleitungen geliefert und im Zählregister der DMA-Einheit abgelegt
- Prozessor hat die Datenübertragung an DMA-Einheit delegiert und fährt unabhängig mit seiner Programmausführung fort
- DMA-Einheit überträgt den Datenblock direkt vom bzw. zum Speicher durch
- DMA-Einheit informiert den Prozessor nach der Datenübertragung

9.2 Cache-Speicher

Cache-Kohärenzproblem:

- Beispiel: System mit einem Mikroprozessoren und weiteren Komponenten mit Master-Funktion (ohne Cache)
 - Zusätzlicher Master kann Kontrolle über Bus übernehmen
 - Kann damit unabhängig vom Prozessor auf Hauptspeicher zugreifen
 - Mikroprozessor und Master teilen sich gemeinsamen Datenbereich



9.2 Cache-Speicher

■ Cache-Kohärenzproblem:

- Mikroprozessorsystem mit DMA-Controller: Zugriff auf veraltete Daten (stale data)
 - Problem beim Write-Through-Verfahren
 - DMA-Controller beschreibt eine Speicherzelle, deren Inhalt im Cache als gültig eingetragen war, der Prozessor führt danach einen Lesezugriff mit der Adresse dieser Speicherzelle durch:
 - Prozessor liest veraltetes Datum
 - Problem beim Copy-Back-Verfahren:
 - der Prozessor führt Schreibzugriff mit der Adresse aus dem gemeinsamen Bereich aus und aktualisiert nur Cache; der DMA-Controller liest anschließend die Speicherzelle mit dieser Adresse:
 - der DMA-Controller liest veraltetes Datum (im Hauptspeicher);

9.2 Cache-Speicher

■ Cache-Kohärenzproblem:

- Mikroprozessorsystem mit DMA-Controller: Zugriff auf veraltete Daten (stale data)
- Lösung des Kohärenzproblems (1):
 - Non-Cachable Data
 - der vom Prozessor und dem zusätzlichen Master gemeinsam benutzte Speicherbereich wird von der Speicherung im Cache ausgeschlossen;
 - Aufgabe der Speicherverwaltung:
 - Der Adressbereich wird in seinem für die Speicherverwaltungseinheit bereitgestelltem Deskriptor als „non-cacheable“ gekennzeichnet.
 - Die Cache-Steuerung wird bei Zugriffen auf den so gekennzeichneten Bereich nicht aktiv.
 - Es werden auch die für Schnittstellen und Controller reservierten Adressbereiche als „non-cacheable“ gekennzeichnet, um den direkten Zugriff auf deren Daten-, Steuer- und Statusregister zu gewährleisten.

9.2 Cache-Speicher

■ Cache-Kohärenzproblem:

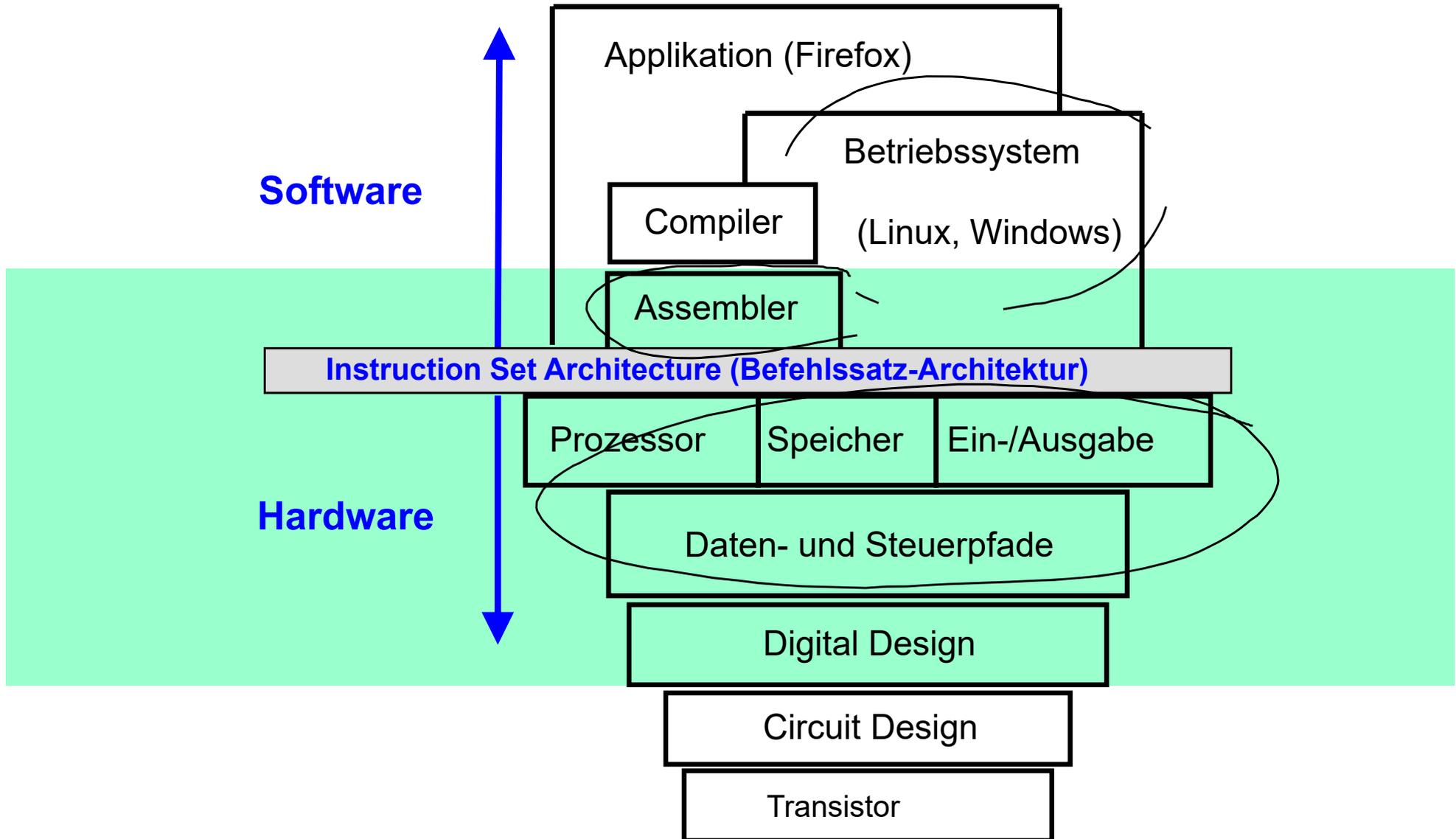
- Mikroprozessorsystem mit DMA-Controller: Zugriff auf veraltete Daten (stale data)
- Lösung des Kohärenzproblems (2)
 - Cache-Clear, Cache-Flush
 - Die Zugriffe von Prozessor und DMA-Controller auf den gemeinsamen Datenbereich werden von zwei unterschiedlichen Tasks ausgeführt;
 - In diesem Fall kann die Task, die den DMA-Vorgang auslöst, dafür sorgen, dass der Cache gelöscht wird, d.h nachfolgende Prozessorzugriffe führen zu einem Neuladen des Cache;
 - Write-Through: Cache-Clear:
 - Die Cache-Einträge werden auf ungültig gesetzt.
 - Copy-Back: Cache-Flush:
 - Alle mit „dirty“ gekennzeichneten Einträge im Cache werden in den Hauptspeicher zurückgeschrieben, danach werden Cache-Einträge auf ungültig gesetzt.

Kapitel 10

Betriebssystemunterstützung

Virtuelle Speicherverwaltung
Schutzmechanismen
Unterbrechungsbehandlung

10.1 Aufgaben des Betriebssystems



10.1 Aufgaben des Betriebssystems

■ Betriebssystem

■ Definition (Deutsches Institut für Normung, DIN 44300):

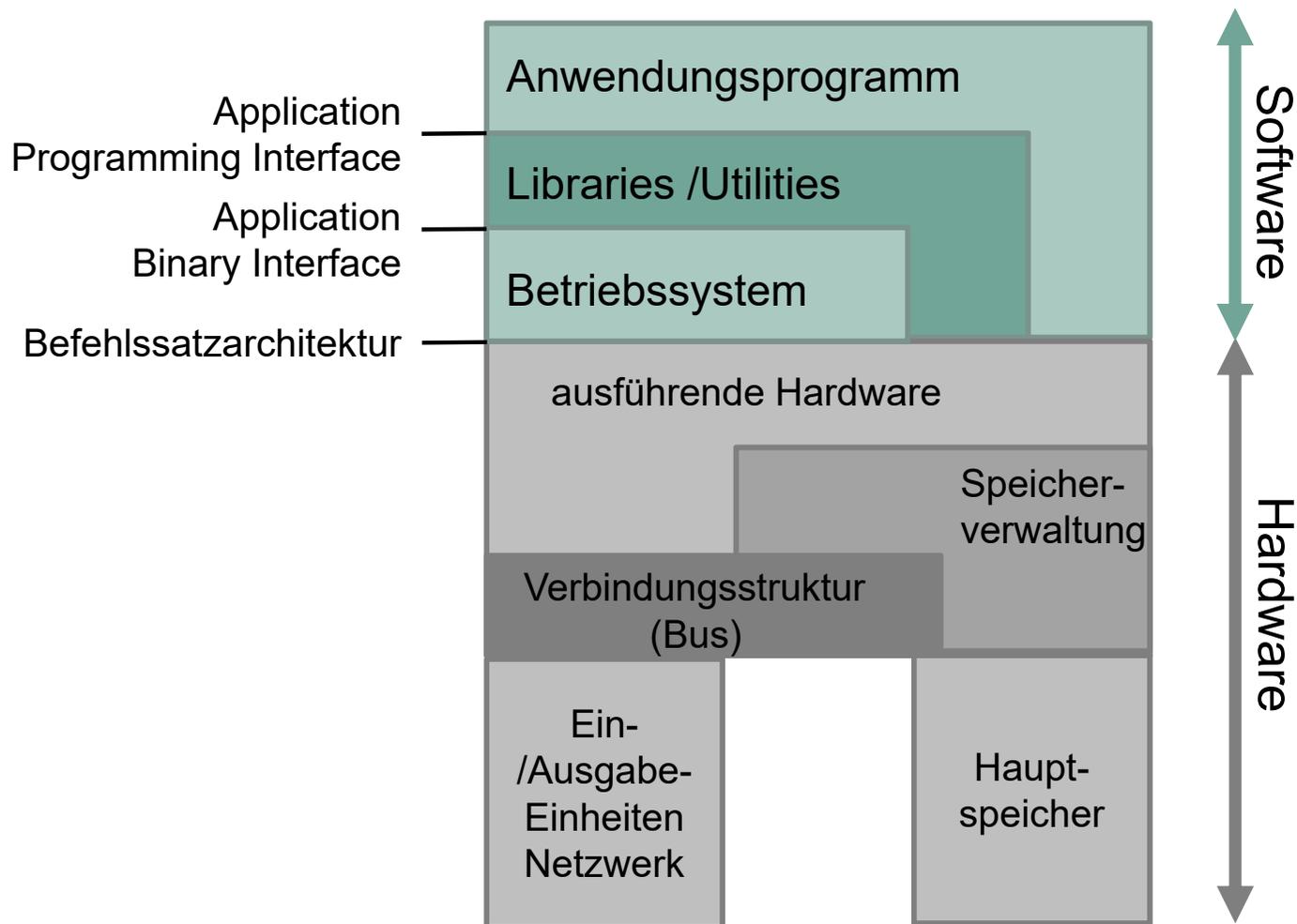
- Ein Betriebssystem umfasst die Programme eines digitalen Rechensystems, die zusammen mit den Eigenschaften der Rechanlage die Grundlage der möglichen Betriebsarten des digitalen Rechensystems bilden und insbesondere die Abwicklung von Programmen steuern und überwachen.

■ Ziele:

- **Einfachheit** in der Benutzung eines Computers
- **Effizienz:** BS soll ermöglichen, dass die Ressourcen eines Computersystems in effizienter Weise benutzt werden

10.1 Aufgaben des Betriebssystems

- Betriebssystem als Schnittstelle zwischen Computer Hardware und Benutzer



10.1 Aufgaben des Betriebssystems

■ Betriebssystem

- Bietet Dienste für folgende Bereiche an:

■ Programmerstellung

- Bietet eine Reihe von Diensten und Programmen (bsw. Editoren, Debugger) an, die den Programmierer bei der Programmerstellung unterstützen
- Dienstprogramme (utility programs), sind nicht Teil des BS, aber werden durch zugreifbar

■ Programmausführung

- Für die Ausführung eines Programms sind eine Reihe von Schritten notwendig:
 - Befehle und Daten müssen in den Hauptspeicher geladen werden, Ein-/Ausgabe-Einheiten initialisiert, ...

10.1 Aufgaben des Betriebssystems

■ Betriebssystem

- Bietet Dienste für folgende Bereiche an:

■ Systemzugriff

- BS kontrolliert den Zugriff auf das System als Ganzes und auf spezielle Systemressourcen
- Schutz von nicht autorisiertem Zugriff
- Bei Blockierung von Ressourcen: Auflösung der Konflikte

■ Fehlererkennung und Antwort

- Bei Fehlersituationen antwortet das System mit der geringsten Auswirkung auf das Programm

■ Monitoring (Accounting)

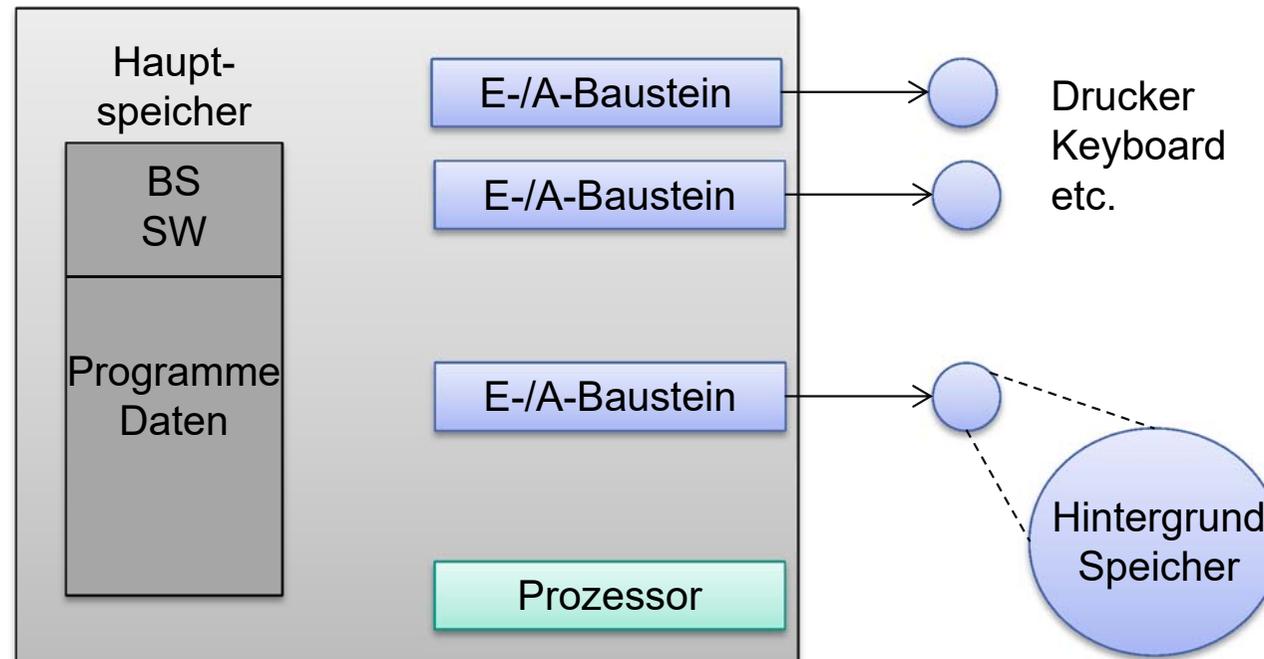
10.1 Aufgaben des Betriebssystems

■ Grundlegende Aufgaben

■ Betriebsmittelverwaltung (resource management)

- Verwaltet alle peripheren Betriebsmittel des Rechnersystems, wie z.B. Festplatte, Floppy Disk, CDROM, Drucker, ...

- Ziel: einfache, effektive und konfliktfreie Nutzung der Betriebsmittel durch die verschiedenen Prozesse

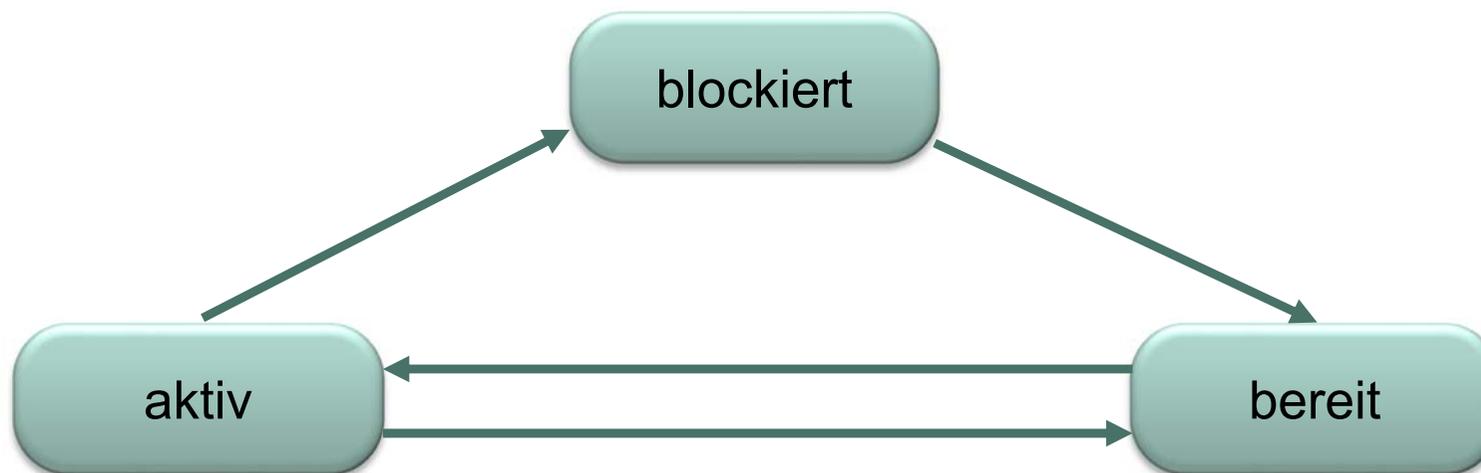


10.1 Aufgaben des Betriebssystems

■ Grundlegende Aufgaben

■ Auftragsverwaltung (Prozessverwaltung)

- Verwaltung der Prozesse (tasks)
- Ein Prozess kann im Laufe seiner Bearbeitung im wesentlichen folgende unterscheidbaren Zustände annehmen



- Verwaltung dieser Zustände und Übergänge sind wesentliche Aufgaben der Auftragsverwaltung
- Ziel: möglichst effiziente und scheinbar parallele Bearbeitung mehrerer Aufgaben

10.1 Aufgaben des Betriebssystems

■ Grundlegende Aufgaben

■ Speicherverwaltung (memory management)

■ Mehrprogrammbetrieb

■ Quasi-parallele Ausführung von unabhängigen Tasks

■ Abtrennung der Adressräume zur Absicherung

■ Virtuelle Speicherorganisation

■ Jeder Task sieht eigenen Adressraum

■ Keine Interferenz

■ Zugriffsschutz

■ Voraussetzung: Speicherpräsenz

■ Aktive Programme/Daten müssen im Hauptspeicher vorhanden sein

■ Aufgabe der Virtuellen Speicherverwaltung (OS)

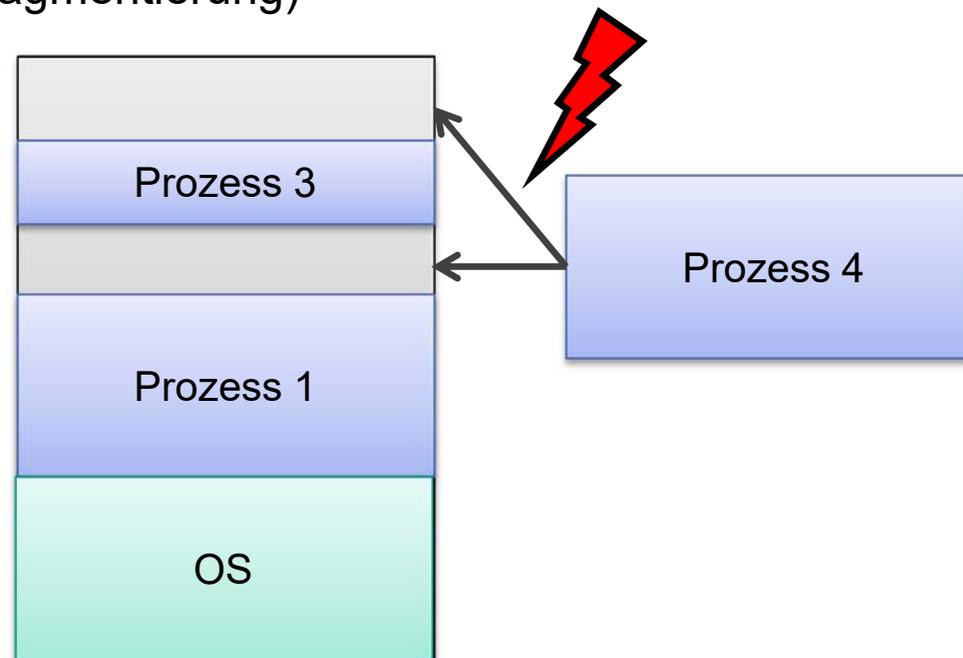
10.1 Aufgaben des Betriebssystems

■ Grundlegende Aufgaben

■ Speicherverwaltung (memory management)

■ Mehrprogrammbetrieb

- Hauptspeicherkapazität ist begrenzt
- Prozess benötigt mehr Speicher als der Hauptspeicher groß ist
- Mehr aktive Prozesse als in den Hauptspeicher passen
- Prozess benötigt mehr Speicher als die größte momentan freie Lücke groß ist (Fragmentierung)



10.1 Aufgaben des Betriebssystems

■ Grundlegende Aufgaben

■ Virtuelle Speicherverwaltung (memory management)

- Teil des OS
- Ggf. Hardwareunterstützung (MMU)

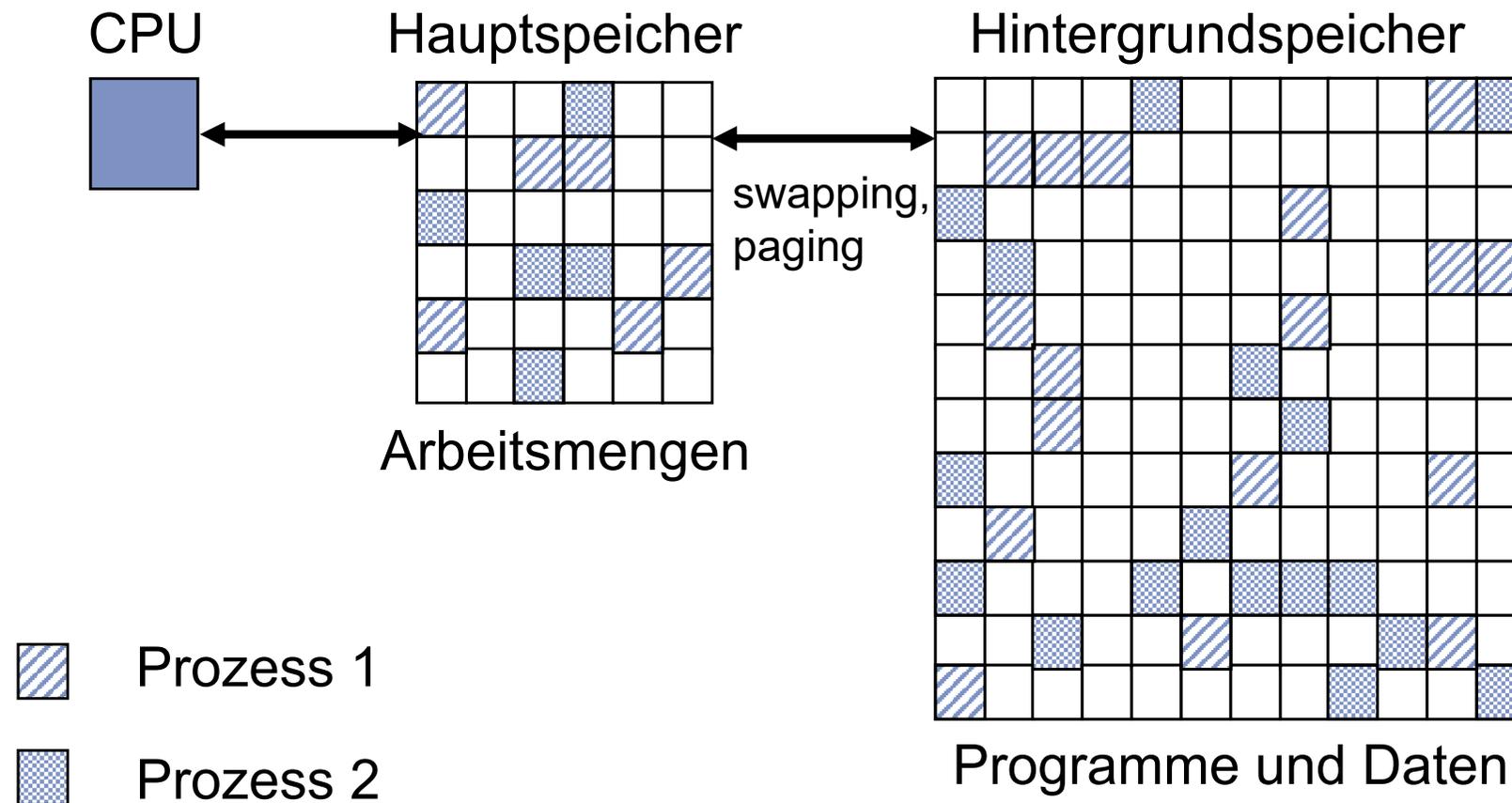
■ Aufgaben:

- Einladen von Programmen und Daten in den realen Hauptspeicher
- Vermerken freier Speicherbereiche (Buchführung)
- Auslagern von Speicherinhalten in Hintergrundspeicher bei ungenügenden Hauptspeicherressourcen
- Notierung von Veränderungen

10.1 Aufgaben des Betriebssystems

- Grundlegende Aufgaben

- Virtuelle Speicherverwaltung (memory management)



10.1 Aufgaben des Betriebssystems

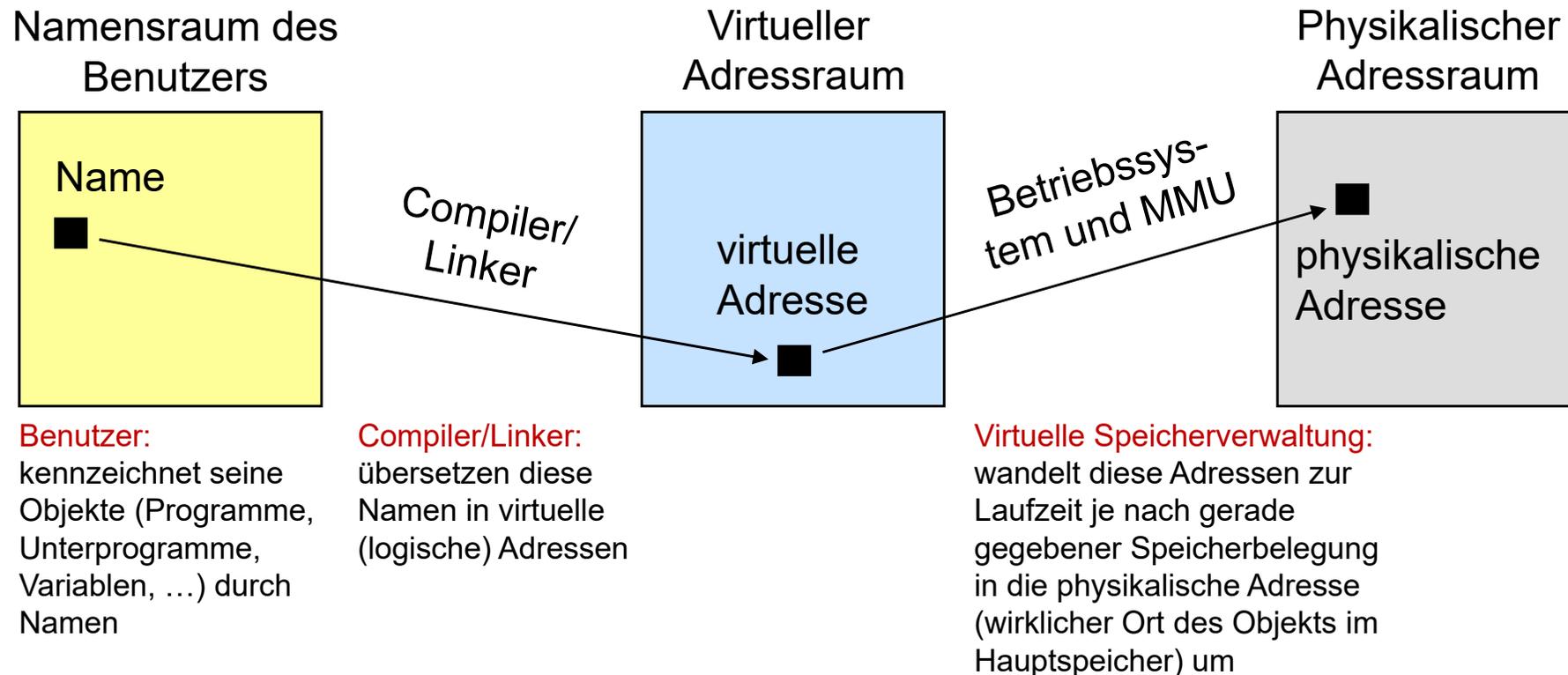
■ Grundlegende Aufgaben

■ Virtuelle Speicherverwaltung

- Dieser Vorgang bleibt dem Anwender verborgen, d.h. der Arbeitsspeicher erscheint dem Anwender wesentlich größer, als er in Wahrheit ist
- Transparent; aber durchaus sichtbar bzgl. nicht-funktionaler Eigenschaften (z.B. Speicherzugriffszeit)
- Ein nach diesem Konzept verwalteter Speicher heißt deshalb virtueller Speicher
- Von modernen Prozessoren wird die Verwaltung dieses virtuellen Speichers hardwaremäßig durch eine MMU (Memory Management Unit) unterstützt

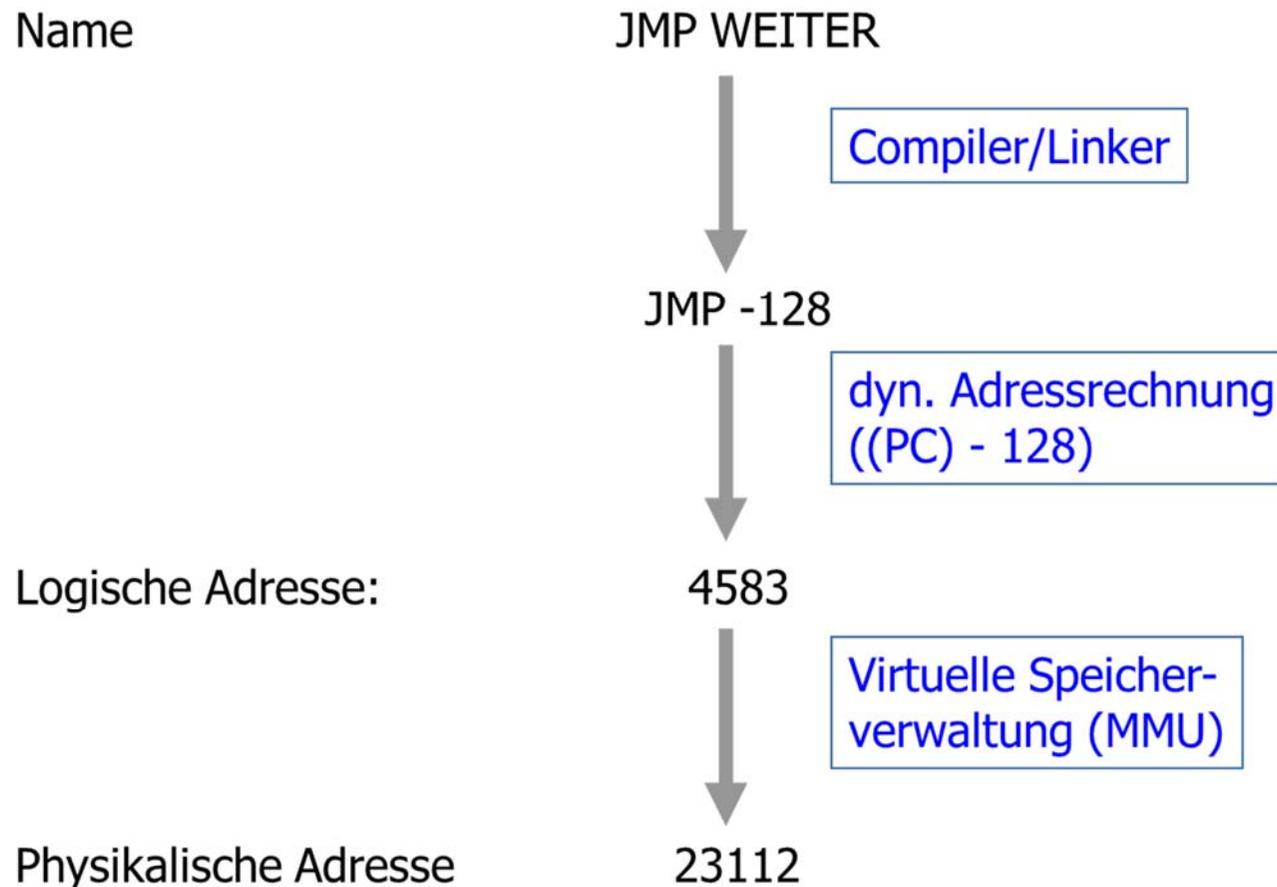
10.2 Virtuelle Speicherverwaltung

- Hauptaufgabe dieser virtuellen Speicherverwaltung: Umsetzung virtueller (logischer) Adressen in physikalische Adressen



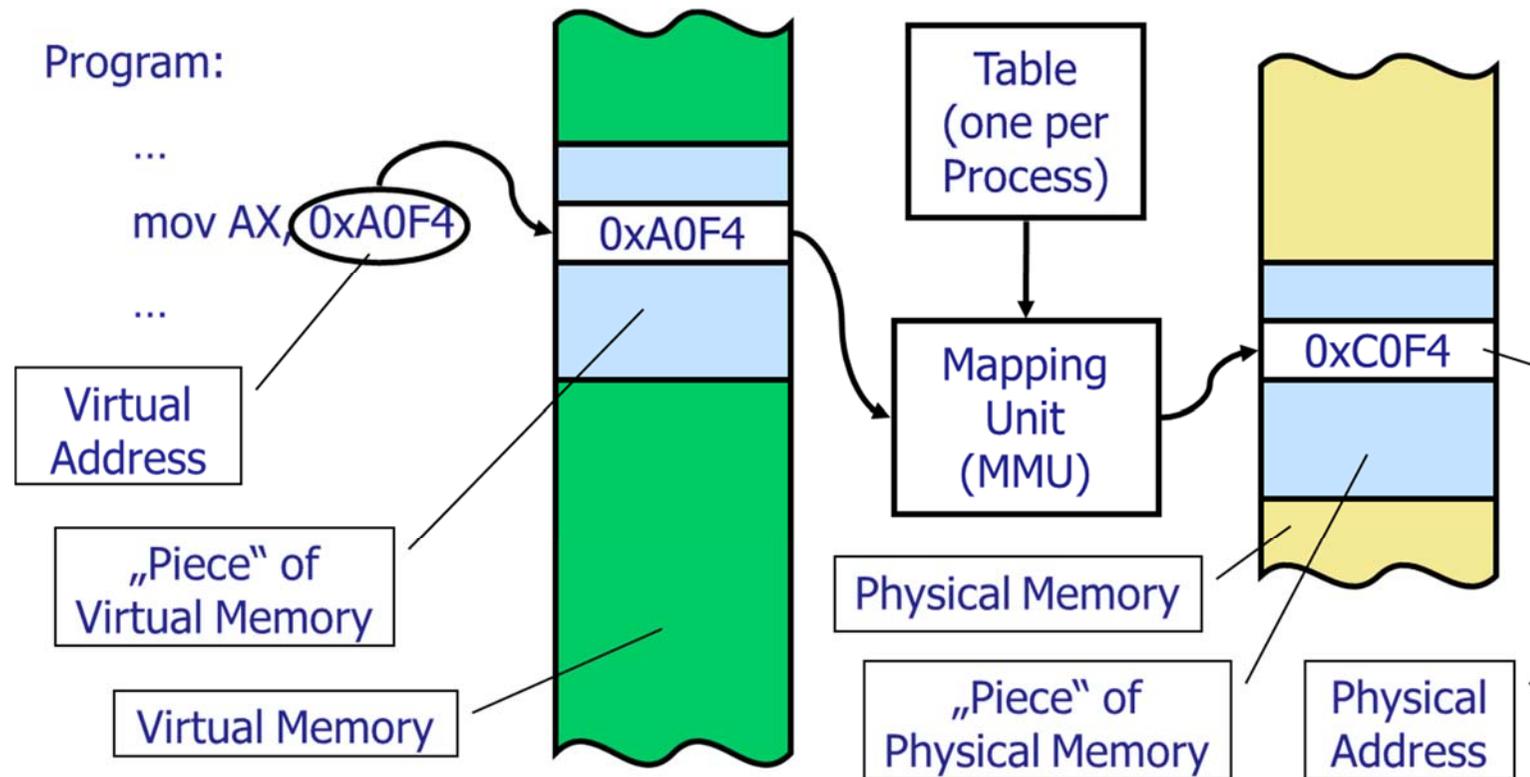
10.2 Virtuelle Speicherverwaltung

- Hauptaufgabe dieser virtuellen Speicherverwaltung: Umsetzung virtueller (logischer) Adressen in physikalische Adressen



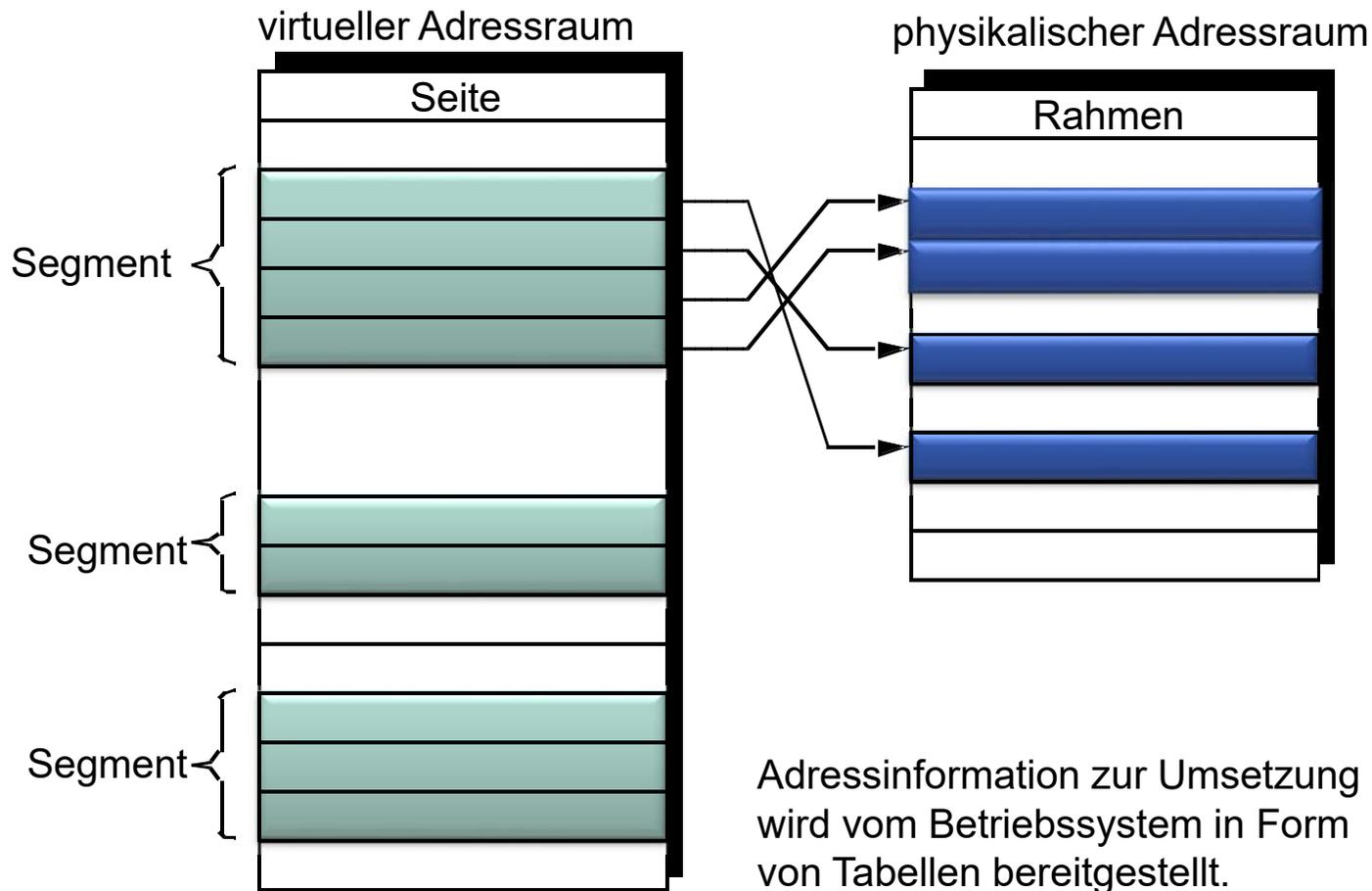
10.2 Virtuelle Speicherverwaltung

- Hauptaufgabe dieser virtuellen Speicherverwaltung: Umsetzung virtueller (logischer) Adressen in physikalische Adressen



10.2 Virtuelle Speicherverwaltung

- Umsetzung virtueller (logischer) Adressen in physikalische Adressen



Segmentierung:

Aufteilen des virtuellen Adressraums in *Segmente variabler* Länge. Segmente umfassen logische Einheiten wie Programmcode, Daten oder Stack.

Seitenaufteilung:

Aufteilen einer logischen Einheit in *Seiten fester* Länge.

10.2 Virtuelle Speicherverwaltung

■ Segmentierung

- Hierbei wird der virtuelle Adressraum in Segmente von verschiedener Länge zerlegt
- Pro Prozess sind ein oder mehrere Segmente zugeordnet (z.B. für Programmcode und Daten)
- Die einzelnen Segmente enthalten logisch zusammenhängende Informationen (Programm- und Datenmodule) und können relativ groß sein

■ Vorteile:

- Segmentierung spiegelt logische Programmstruktur wider
- Durch große Segmente relativ seltener Datentransfer

■ Nachteile:

- Wenn Datentransfer, dann jedoch umfangreich
- besteht ein Programm nur aus einem Code- und Daten-Segment (wird vom Compiler oder Benutzer festgelegt), so muss es vollständig eingelagert werden

10.2 Virtuelle Speicherverwaltung

■ Seiten:

- Hierbei wird der logische und der physikalische Adressraum in "Teile fester Länge", die sogenannten Seiten (Pages) unterteilt
- Größe: 4 KBytes, auch andere Größen einstellbar
 - Manchmal zusätzlich sog. Huge Pages (typisch 1 MByte – 4 MByte) und Large Pages (bis zu mehreren GByte bei Servern)
- Im Unterschied zu Segmenten können Seiten nicht an beliebiger Stelle im Speicher beginnen, sondern nur in einem festen an der Seitengröße orientierten Raster
- Ein Prozess wird auf viele dieser Seiten verteilt
- Keine logischen Zusammenhänge wie bei der Segmentierung

■ Vorteile:

- Einlagerung der benötigten Teile des Programms
- geringerer Verwaltungsaufwand als Segmentierung

■ Nachteil:

- häufigerer Datentransfer

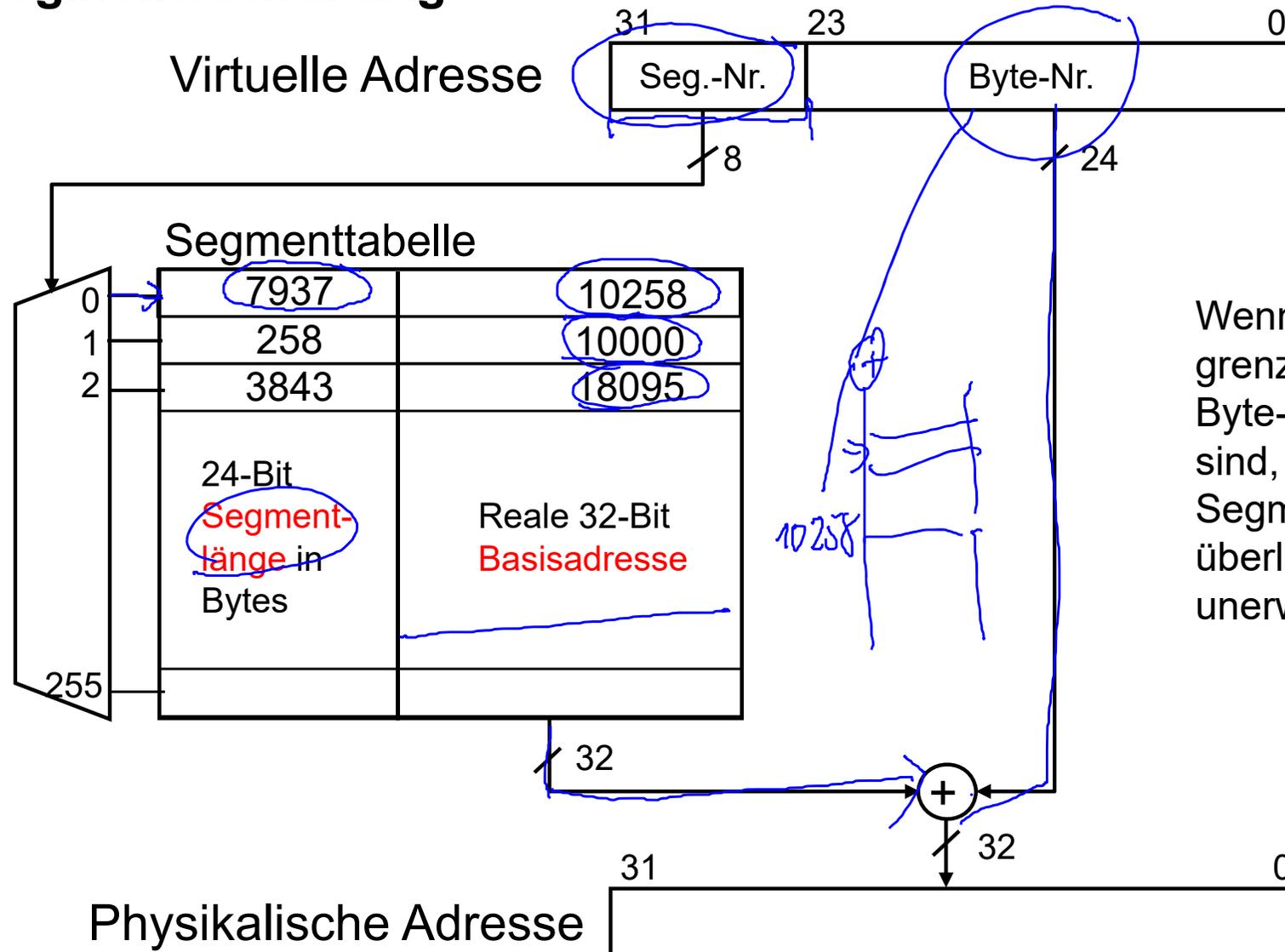
10.2 Virtuelle Speicherverwaltung

■ Segmentverwaltung

- Virtuelle Adresse wird in eine Segmentnummer (n höherwertige Bits der virtuellen Adresse) als Kennung eines Segments und in eine Bytenummer (verbleibenden m Bits der Adresse) als Abstand zum Segmentanfang unterteilt
 - Maximale virtuelle Segmentanzahl = 2^n
 - Maximale Segmentgröße = 2^m
- Die Adressabbildung erfolgt über eine Segmenttabelle (beispielsweise im Registerspeicher der MMU)
- Reale Adresse ergibt sich aus der Segmentbasisadresse, zu der die virtuelle Byte-Nummer addiert wird
- Segmentlängenangaben: um Segment-überschreitende Zugriffe feststellen und ggf. verhindern zu können
- Bei Segmenten mit einer geringeren Größe als 2^m gilt der ungenutzte Raum als Verschnitt
- Gute Ausnutzung des Hauptspeichers, wenn man die Segmentgrenzen an jeder Byteadresse zulässt

10.2 Virtuelle Speicherverwaltung

■ Segmentverwaltung



Wenn Segmentgrenzen beliebige Byte-Grenzen sind, können sich Segmente überlappen (i.A. unerwünscht)