

Rechnerorganisation

Prof. Dr. Wolfgang Karl

Vorlesung im Wintersemester 2025/2026 – Foliensatz: RO25-FS17



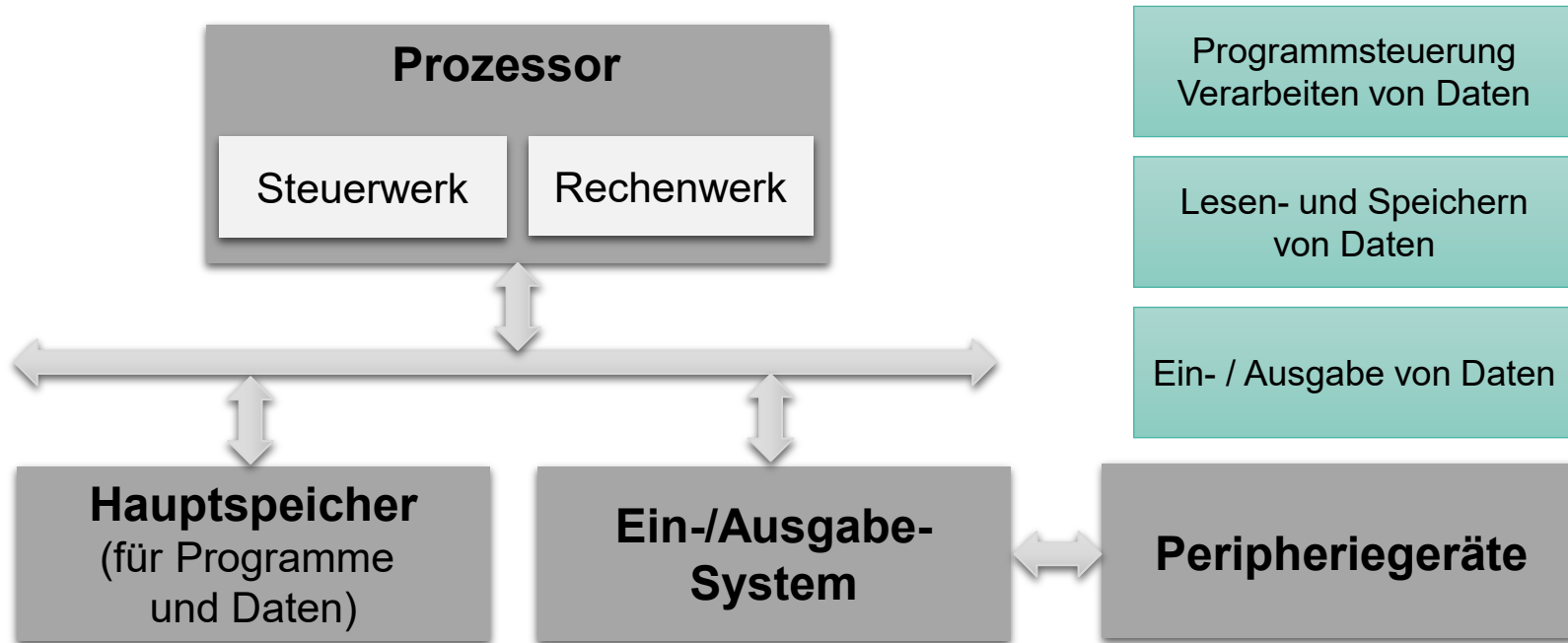
Kapitel 10

Betriebssystemunterstützung

- **Aufgaben des Betriebssystems**
- Virtuelle Speicherverwaltung
 - Fallstudie IA-32
 - Beschleunigung der Adressumsetzung
- Schutzmechanismen
- Unterbrechungsbehandlung

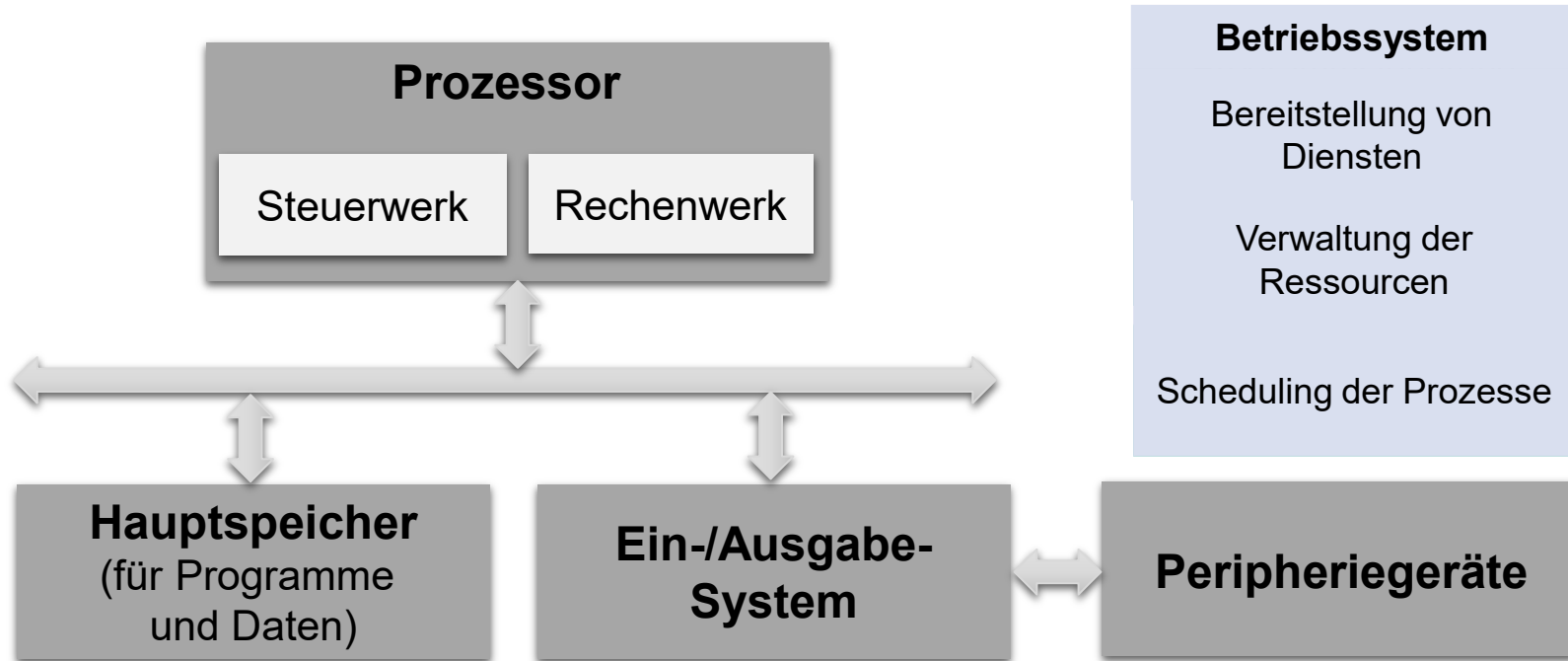
Programmierbarer Universalrechner

■ Einfaches Modell



Programmierbarer Universalrechner

■ Einfaches Modell



Betriebssystem

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

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

■ Ziele:

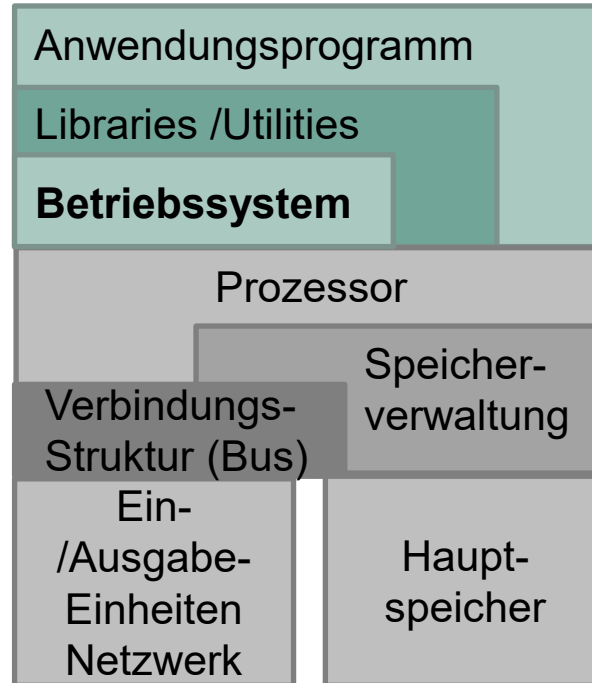
- Benutzerfreundliche Unterstützung bei den Verwendungen eines Computers;
- Effiziente Nutzung der Ressourcen eines Computers;

Betriebssystem

■ Das Betriebssystem als Schnittstelle zwischen der Computer-Hardware und dem Benutzer

Bereitstellung von **Libraries/Utilities** zur Unterstützung der Programmerstellung

Bereitstellung von Diensten für die Programmausführung, Dateiverwaltung, Zugriff auf die E/A-Einheiten, ...



Das **Betriebssystem** ist die Schnittstelle zur Nutzung des Systems und verbirgt die Details der HW vor dem Programmierer.

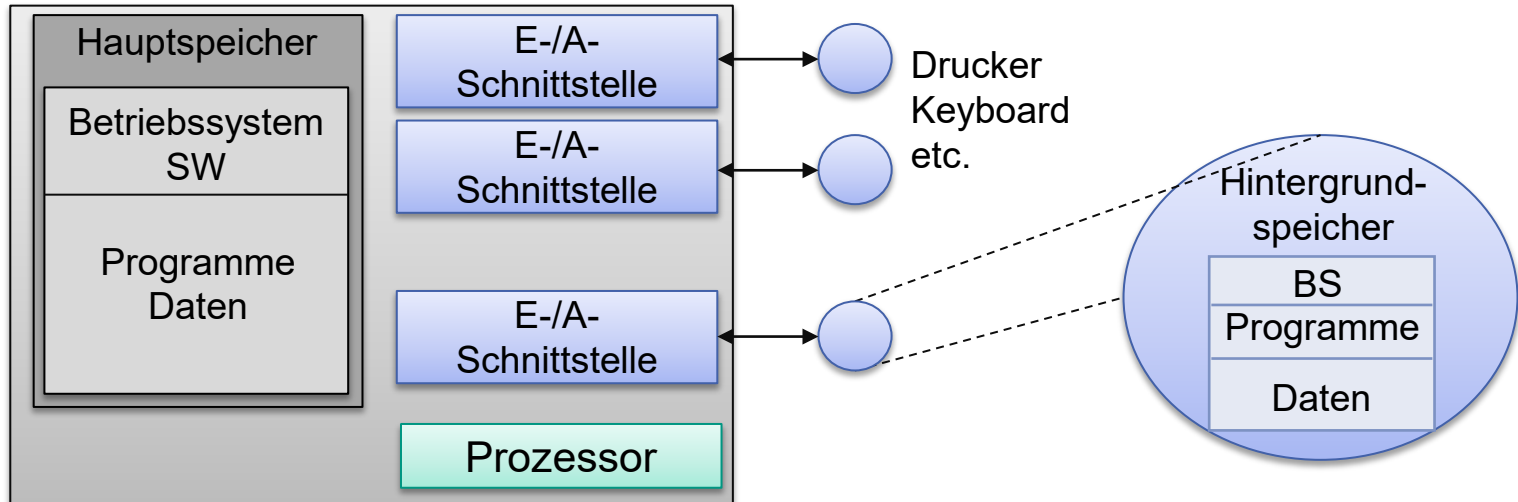
Betriebssystem

■ Weitere Aufgaben des Betriebssystems

■ Betriebsmittelverwaltung (resource management)

■ Ziel:

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

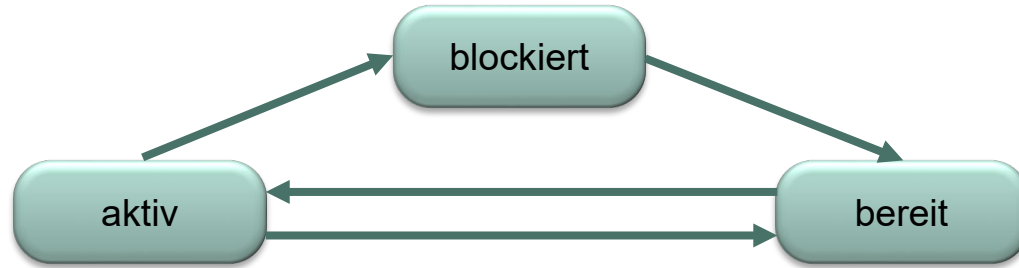


Betriebssystem

■ Weitere Aufgaben des Betriebssystems

■ Auftragsverwaltung (Prozessverwaltung)

- 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

Betriebssystem

■ Weitere Aufgaben des Betriebssystems

■ Mehrprogrammbetrieb (multi programming, multi tasking)

- Quasi-parallele Ausführung von voneinander unabhängigen Prozessen;
 - Abtrennung der Adressräume zur Absicherung;
- Aktive Programme/Daten müssen im Hauptspeicher vorhanden sein.

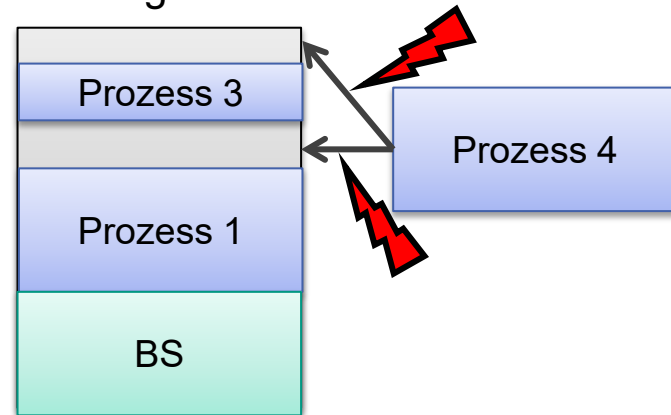
Betriebssystem

■ Weitere Aufgaben des Betriebssystems

■ Mehrprogrammbetrieb (multi programming, multi tasking)

■ Probleme:

- Hauptspeicherkapazität ist begrenzt.
- Ein Prozess kann mehr Speicherplatz benötigen, als der Hauptspeicher groß ist.
- Eine Menge von Prozessen ist aktiv, die nicht alle in den Hauptspeicher passen.
- Ein Prozess benötigt mehr Speicherplatz als die größte momentan freie Lücke groß ist (Fragmentierung).



Betriebssystem

■ Weitere Aufgaben des Betriebssystems

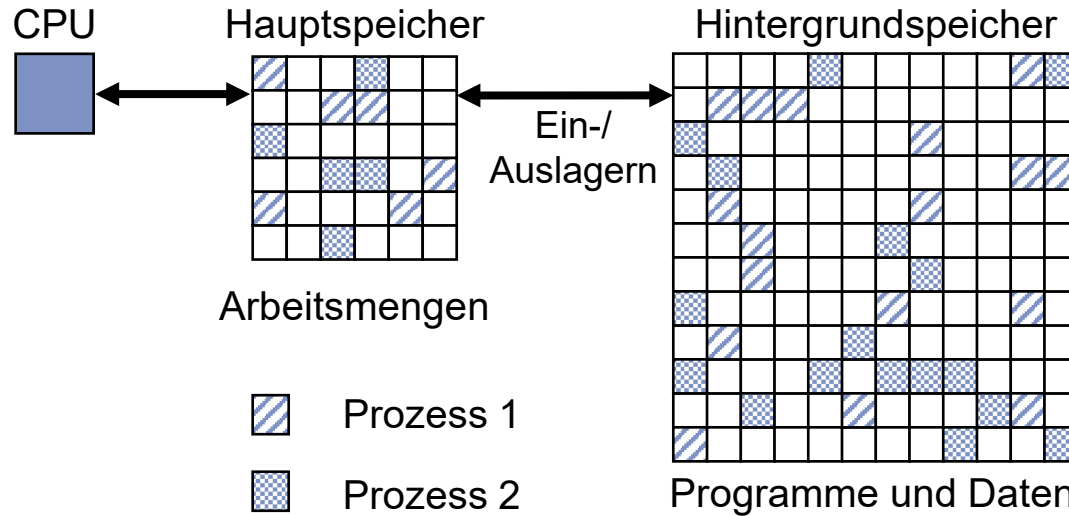
■ Mehrprogrammbetrieb (multi programming, multi tasking)

■ Lösung: Virtueller Speicher:

- Kapazitätserweiterung des Hauptspeichers durch den Hintergrundspeicher;
 - Damit wird ein Hauptspeichers mit großer Kapazität vorgespiegelt.
 - Die aktuell benötigten Programme und Daten werden vom Hintergrundspeicher in den Hauptspeicher geladen.
-
- Virtueller Speicher >> Hauptspeicher
 - Jeder Prozess hat seinen großen und einheitlichen Adressraum.

Betriebssystem

■ Virtueller Speicher



Betriebssystem

■ Virtueller Speicher

■ Speicherverwaltung (memory management)

- Laden der für die Ausführung benötigten Programme und Daten in den realen Hauptspeicher;
- Vermerken freier Speicherbereiche (Buchführung)
- Auslagern von Speicherinhalten in den Hintergrundspeicher (z. B. Festplatte);
 - Speicherinhalte sind auszulagern, wenn nicht genügend Hauptspeicherressourcen vorhanden sind und sie gegenüber dem Original auf dem Hintergrundspeicher verändert worden sind.
- Notierung der Veränderungen;

- Voraussetzungen: Lageunabhängigkeit der Programme und Daten
 - Freie Verschiebbarkeit im Hauptspeicher;
 - Keine feste Bindung an physikalische Adressen;

Betriebssystem

■ Virtueller Speicher

■ Speicherverwaltung (memory management)

- Diese Vorgänge bleiben 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)

Betriebssystem

■ Virtueller Speicher

■ Speicherverwaltung (memory management)

- Die Speicherverwaltung ist Aufgabe des Betriebssystems!
- Unterstützung durch die Hardware: **Speicherverwaltungseinheit (Memory Management Unit, MMU)**
- Die Verwaltung des virtuellen Speichers bleibt dem Benutzer verborgen.

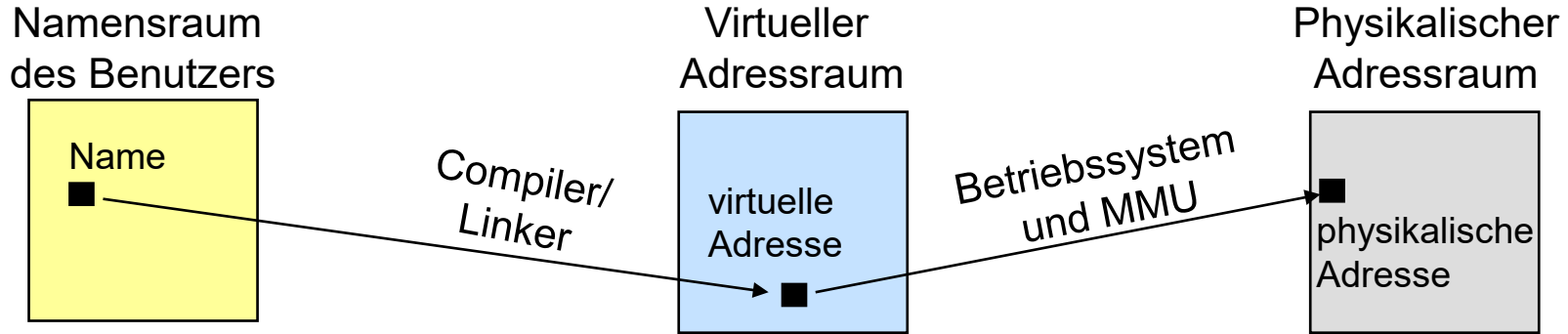
Kapitel 10

Betriebssystemunterstützung

- Aufgaben des Betriebssystems
- **Virtuelle Speicherverwaltung**
 - Fallstudie IA-32
 - Beschleunigung der Adressumsetzung
- Schutzmechanismen
- Unterbrechungsbehandlung

Virtuelle Speicherverwaltung

■ Umsetzung virtueller (logischer) Adressen in physikalische Adressen



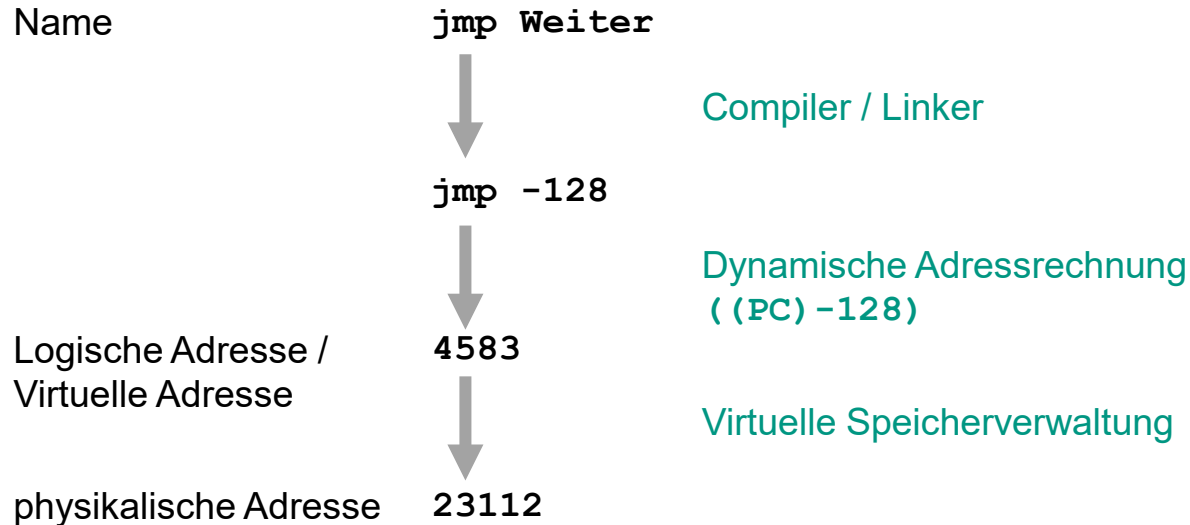
Benutzer:
kennzeichnet
seine Objekte
(Programme,
Unterprogramme
, Variablen, ...)
durch Namen

**Compiler/
Linker:**
übersetzen
diese Namen
in virtuelle
(logische)
Adressen

Virtuelle Speicherverwaltung:
wandelt diese Adressen zur
Laufzeit je nach gerade
gegebener Speicherbelegung in
die physikalische Adresse
(wirklicher Ort des Objekts im
Hauptspeicher) um

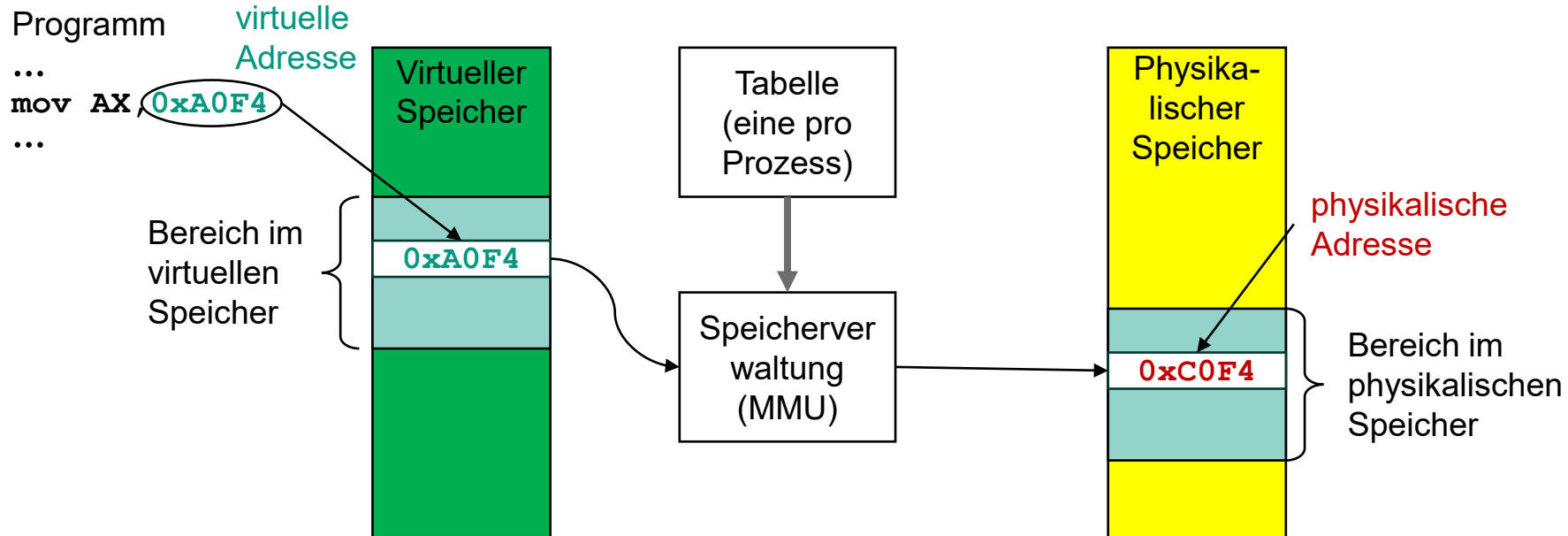
Virtuelle Speicherverwaltung

- Umsetzung virtueller (logischer) Adressen in physikalische Adressen
 - Beispiel:



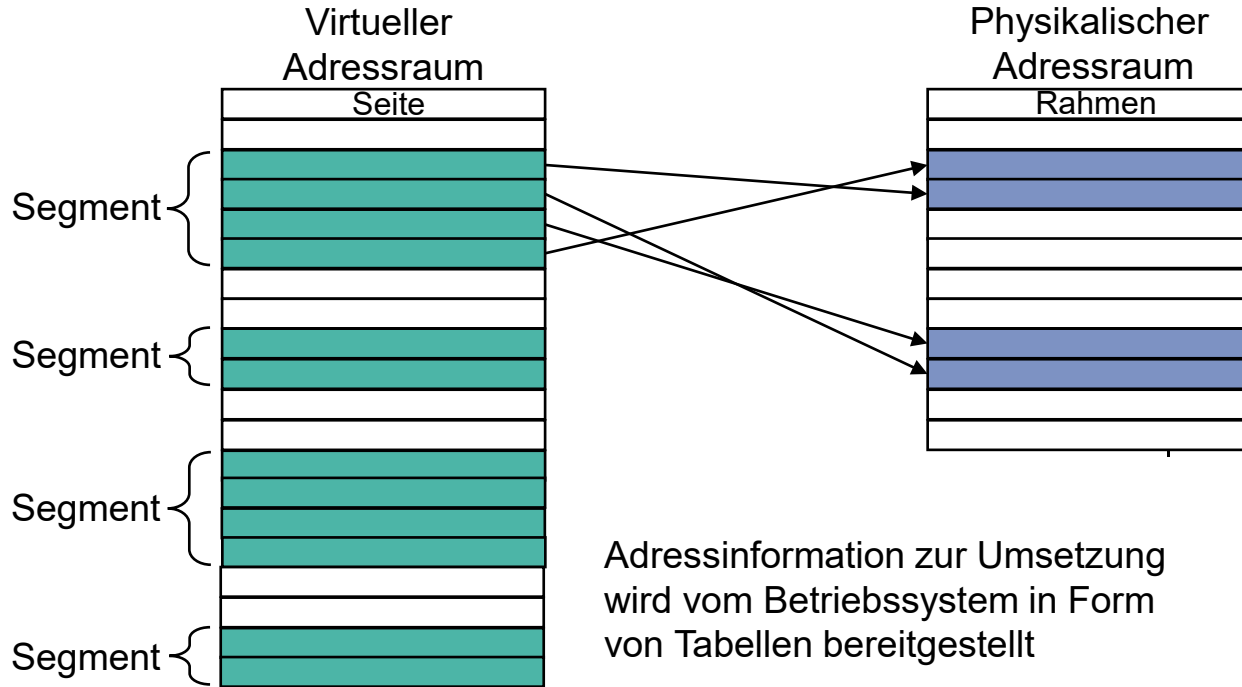
Virtuelle Speicherverwaltung

- Umsetzung virtueller (logischer) Adressen in physikalische Adressen
- Beispiel:



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*

Virtuelle Speicherverwaltung

- **Umsetzung virtueller (logischer) Adressen in physikalische Adressen**
 - **Segmentierung**
 - Hierbei wird der virtuelle Adressraum in Segmente von unterschiedlicher Länge aufgeteilt.
 - Einem Prozess sind ein oder mehrere Segmente zugeordnet.
 - Die einzelnen Segmente enthalten logisch zusammenhängende Informationen (Programm- und Datenmodule).
 - **Vorteile:**
 - Die Segmentierung spiegelt logische Programmstruktur wider.
 - **Nachteile:**
 - Segmente eines Prozesses müssen im Hauptspeicher sein. Die variabel langen Segmente erschweren das Finden freier Bereiche im Hauptspeicher.

Virtuelle Speicherverwaltung

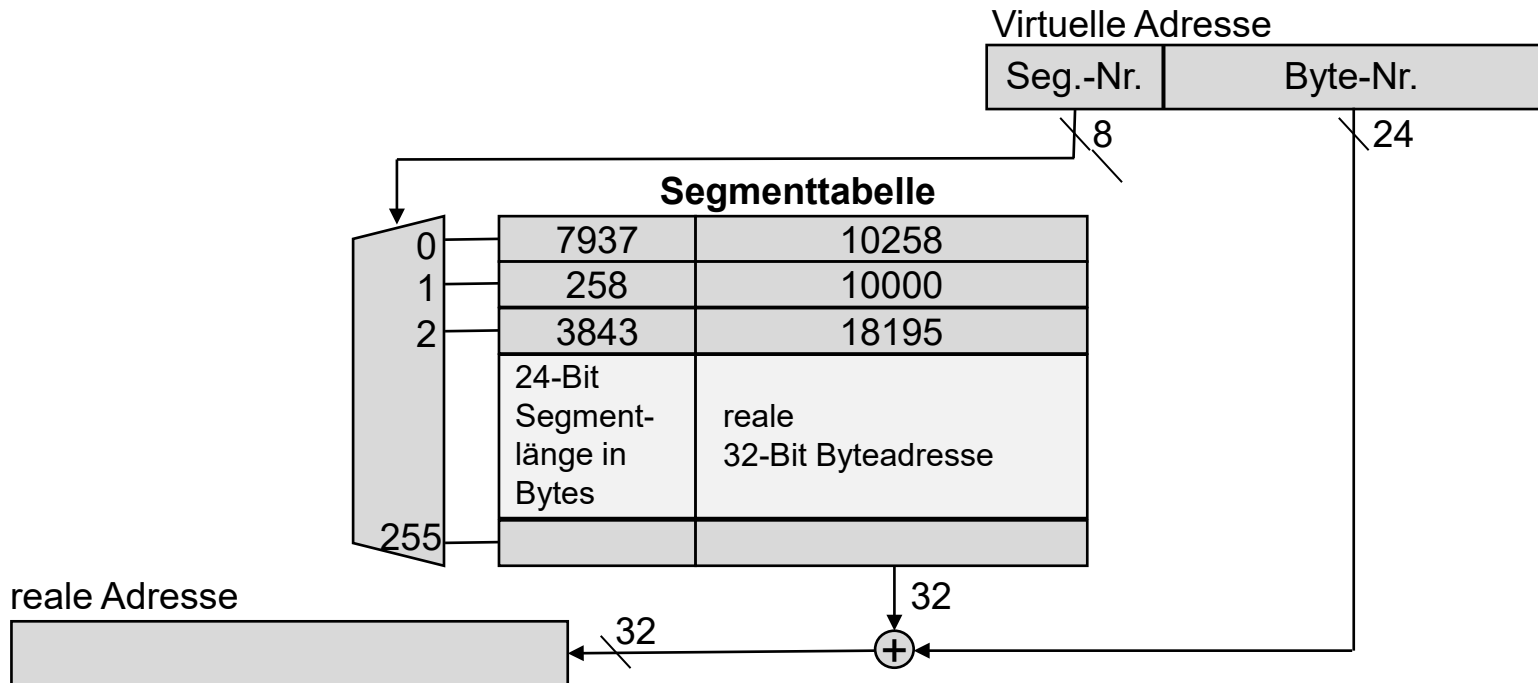
- **Umsetzung virtueller (logischer) Adressen in physikalische Adressen**
 - **Segmentierung**
 - **Unterteilung der virtuellen Adresse**
 - **Segmentnummer**
 - umfasst die n höherwertigen Bits der virtuellen Adresse als Kennung eines Segments;
 - **Byte-Nummer**
 - umfasst die niederwertigen m Bits der virtuellen Adresse als Abstand zum Segmentanfang;
 - Maximale Anzahl virtueller Segmente: 2^n
 - Maximale Größe des Segments: 2^m Bytes

Virtuelle Speicherverwaltung

- **Umsetzung virtueller (logischer) Adressen in physikalische Adressen**
 - **Segmentierung**
 - Die Adressabbildung erfolgt über eine **Segmenttabelle**.
 - Die reale Adresse ergibt sich aus der Segmentbasisadresse durch Addition der virtuellen Byte-Adresse.
 - Die Angabe der Segmentlängen ermöglicht, Zugriffe über die Grenzen eines Segments feststellen und ggf. verhindern zu können.

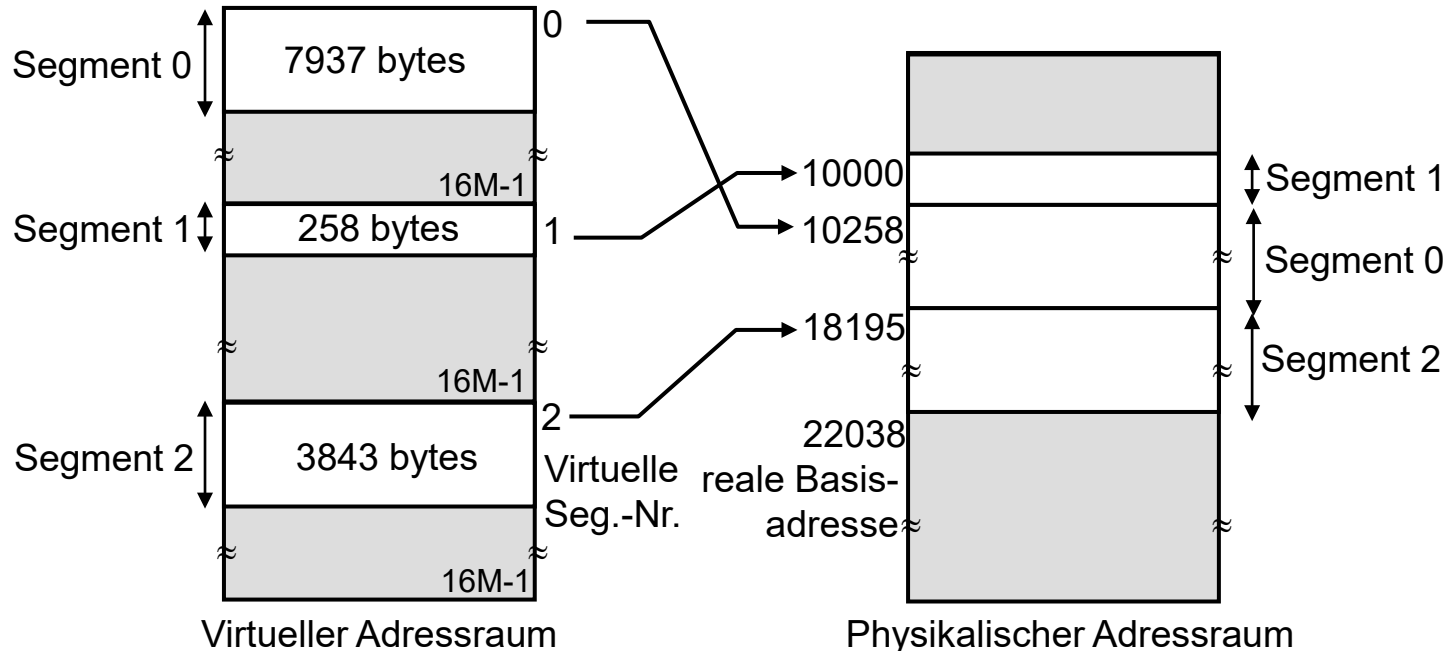
Virtuelle Speicherverwaltung

- Umsetzung virtueller (logischer) Adressen in physikalische Adressen
 - Segmentierung: Unterteilung der virtuellen Adresse



Virtuelle Speicherverwaltung

- Umsetzung virtueller (logischer) Adressen in physikalische Adressen
 - Segmentierung: Unterteilung der virtuellen Adresse



Virtuelle Speicherverwaltung

■ Umsetzung virtueller (logischer) Adressen in physikalische Adressen

■ Seitenverwaltung

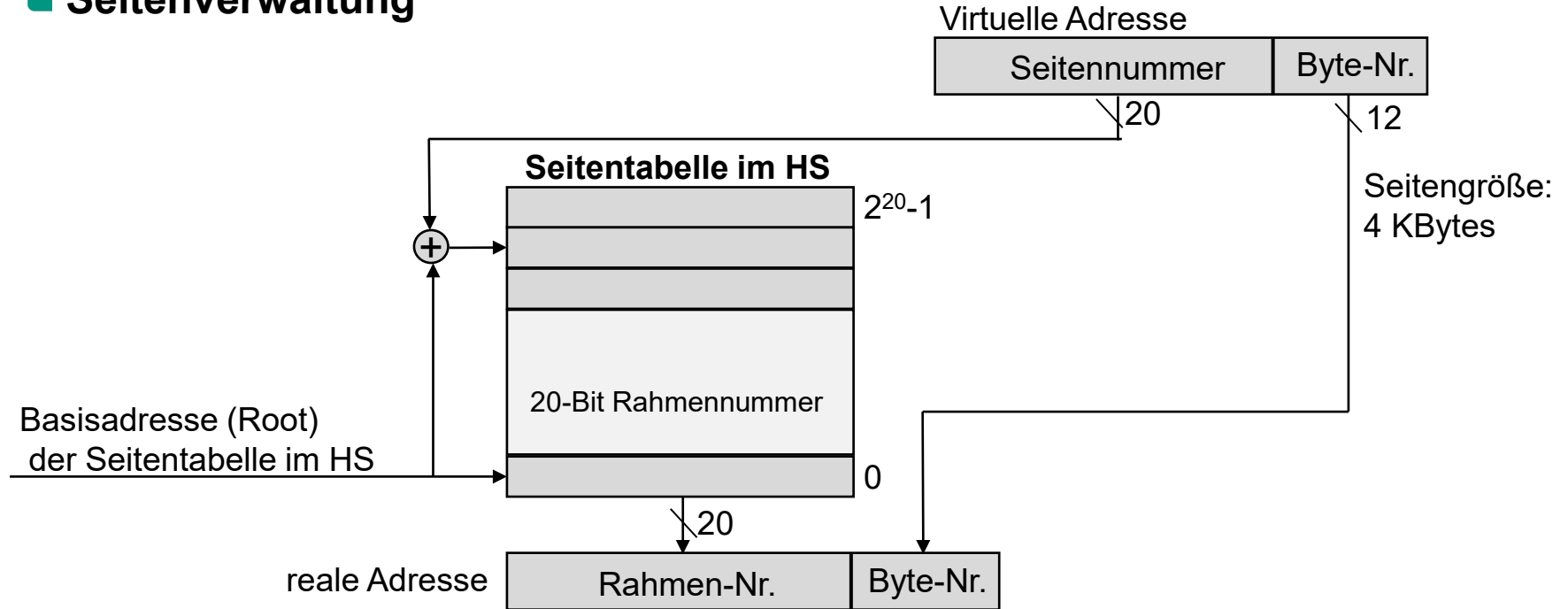
- Aufteilung des virtuellen und des physikalischen Adressraums in Seiten fester Länge;
 - Die Länge der Seite kann konfiguriert werden:
 - typischerweise 4 KBytes;
 - Es werden auch sog. Huge Pages (typisch 1 MBytes – 4 MBytes) und Large Pages (bis zu mehreren GBytes bei Servern) unterstützt
 - Seiten sind auf Seitengrenzen im Speicher ausgerichtet.
 - Ein Prozess wird auf viele dieser Seiten verteilt.

■ Vorteile

- Nur die aktuell benötigten Seiten eines Prozesses werden eingelagert.
- Keine Fragmentierung;
- Einfache Freispeicherverwaltung;

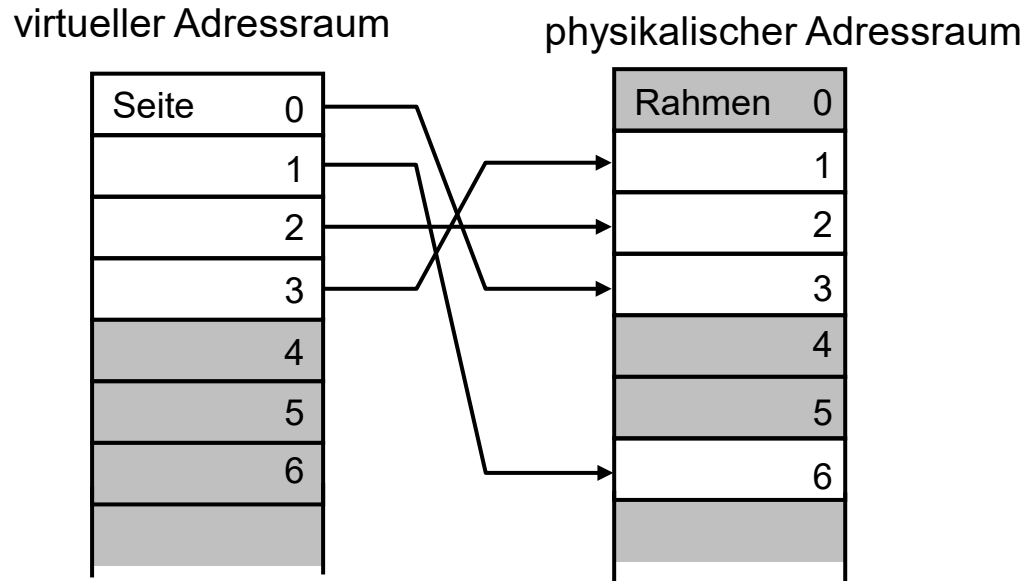
Virtuelle Speicherverwaltung

- Umsetzung virtueller (logischer) Adressen in physikalische Adressen
 - Seitenverwaltung



Virtuelle Speicherverwaltung

- **Umsetzung virtueller (logischer) Adressen in physikalische Adressen**
 - **Seitenverwaltung**
 - Beispiel: Abbildung von 4 Seiten eines Segments auf 4 Rahmen des Hauptspeichers



Virtuelle Speicherverwaltung

■ Fragen der Speicherverwaltung

- Beim Austausch von Daten zwischen Haupt- und Hintergrundspeicher ergeben sich drei Probleme:
 - **Einlagerungszeitpunkt:**
 - Wann werden Segmente oder Seiten in den Hauptspeicher eingelagert?
 - **Zuweisungsproblem:**
 - An welche Stelle des Hauptspeichers werden die Seiten oder Segmente eingelagert?
 - **Ersetzungsproblem:**
 - Welche Segmente oder Seiten müssen ausgelagert werden, um Platz für neu benötigte Daten zu schaffen?
- **Die Lösungsstrategien werden vom Betriebssystem umgesetzt.**

Virtuelle Speicherverwaltung

■ Fragen der Speicherverwaltung

■ Einlagerungszeitpunkt

- Einlagerung auf Anforderung (**Demand Paging** bei Seitenverfahren):
 - Eine Seite wird in den Hauptspeicher geholt, wenn auf Daten zugegriffen wird, aber die Seite mit den benötigten Daten nicht im Hauptspeicher ist.
 - Ein Bit in einem Tabelleneintrag zeigt an, ob die Seite oder das Segment im Hauptspeicher geladen ist.

Virtuelle Speicherverwaltung

■ Fragen der Speicherverwaltung

■ Einlagerungszeitpunkt

- Einlagerung auf Anforderung (**Demand Paging** bei Seitenverfahren):

■ Unterstützung des Prozessors:

- Das Bit im Tabelleneintrag wird beim Zugriff geprüft, ob es gesetzt ist oder nicht.
- Beim Zugriff auf ein nicht im Hauptspeicher vorhandenes Segment oder Seite wird ein Alarm ausgelöst:
 - Segment-fehlt-Alarm, segment fault oder
 - Seiten-fehlt-Alarm page fault)
- Unterbrechungsbehandlung

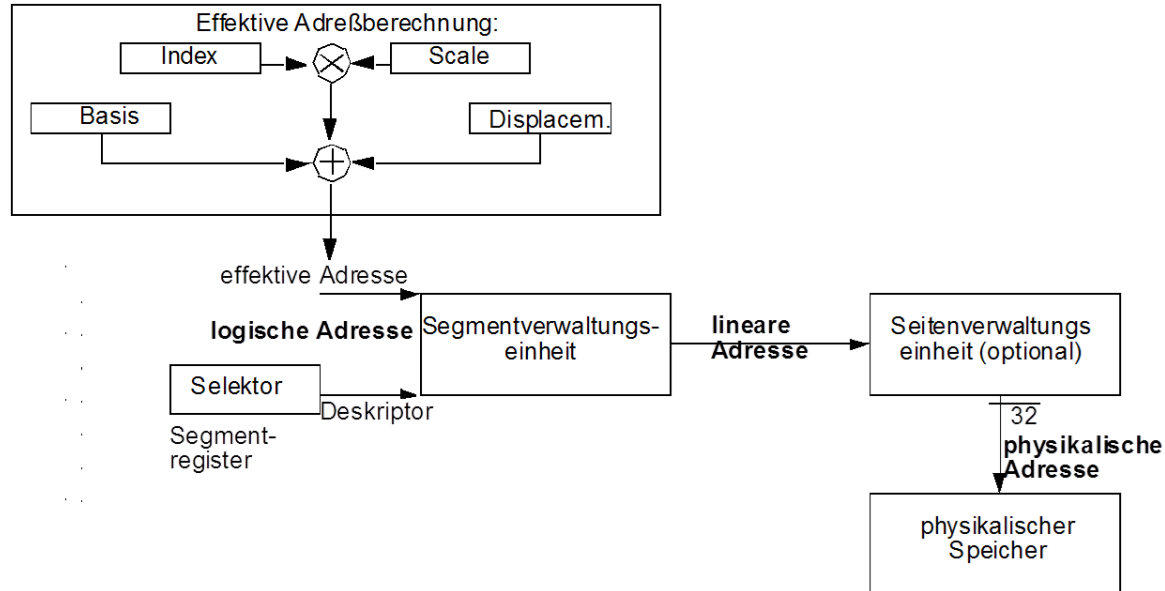
Kapitel 10

Betriebssystemunterstützung

- Aufgaben des Betriebssystems
- **Virtuelle Speicherverwaltung**
 - **Fallstudie IA-32**
 - Beschleunigung der Adressumsetzung
- Schutzmechanismen
- Unterbrechungsbehandlung

Virtuelle Speicherverwaltung

- Fallstudie: Intel x86 Speicherverwaltung
 - Umsetzung von virtuellen in physikalische Adressen
 - Segmentierung und Seitenverwaltung

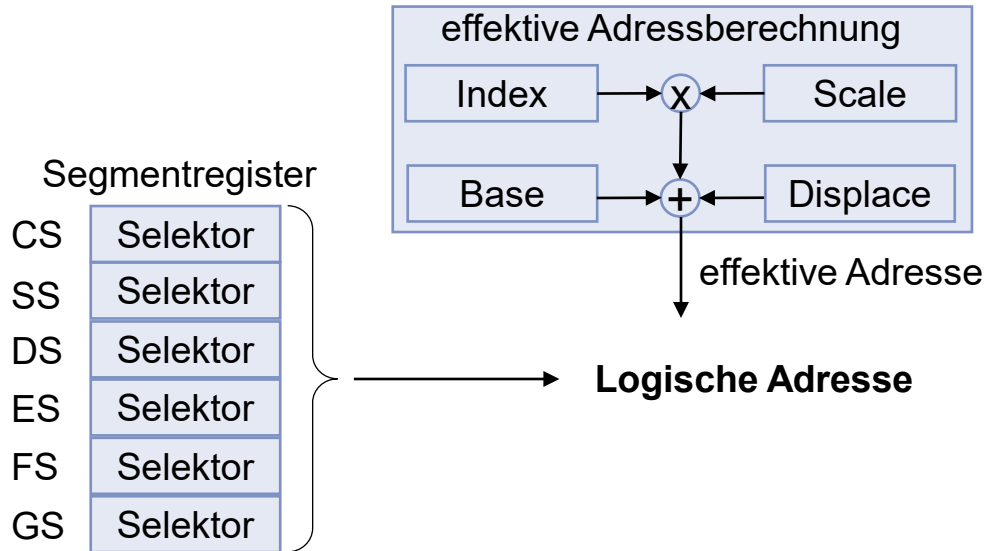


Intel x86 Speicherverwaltung

- **Segmentierung mittels Erweiterung der virtuellen Adresse:**
 - 64 TBytes (2^{46}) logischer Adressraum
 - Pro Task (Prozess) maximal 16K (2^{14}) Selektoren und damit Segmente
 - Pro Segment max. 4GBytes

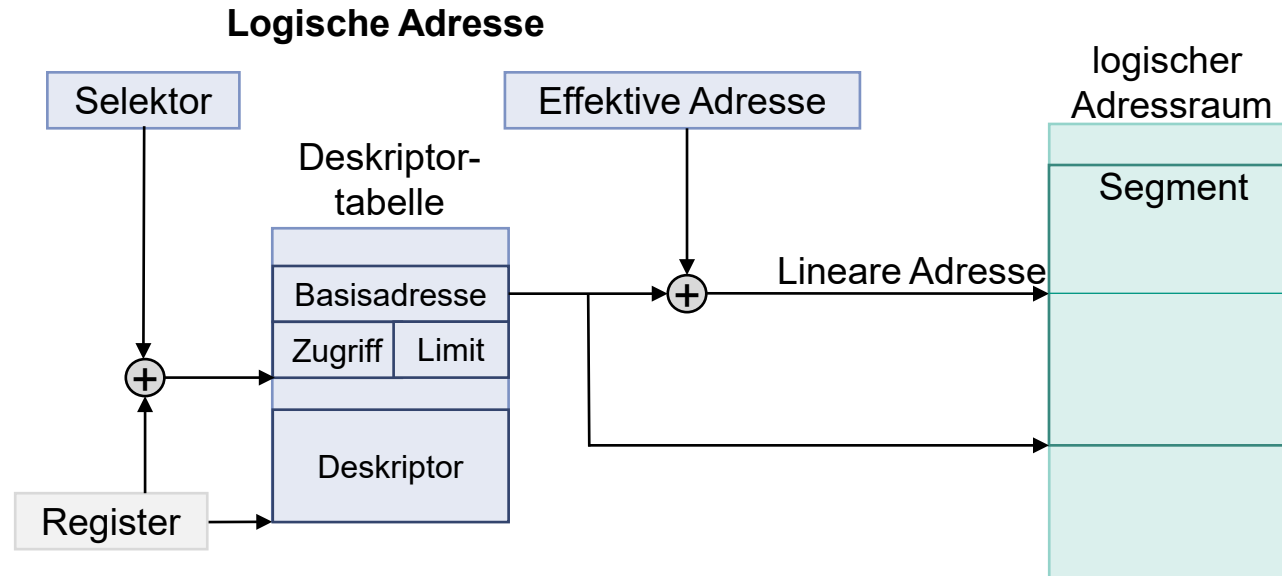
Intel x86 Speicherverwaltung

- **Segmentierung mittels Erweiterung der virtuellen Adresse:**
 - Die logische Adresse setzt sich aus effektiver Adresse (32 Bit) und Selektor (13 Bit) zusammen



Intel x86 Speicherverwaltung

■ Segmentverwaltung



Intel x86 Speicherverwaltung

■ Segmentverwaltung

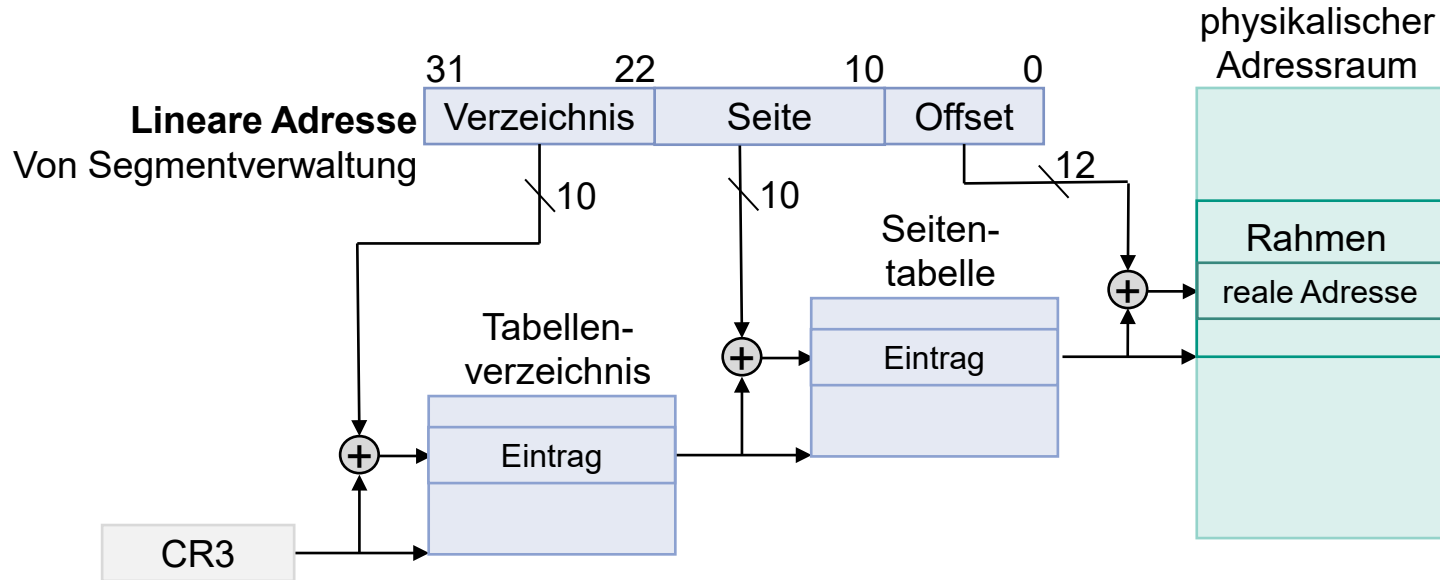
- Die Adressrechnung im Prozessor liefert die **effektive Adresse**.
- Ein **Selektor** steht in einem Segmentregister und bildet einen Index in eine Deskriptortabelle.
- **Deskriptortabelle:**
 - Eine Deskriptortabelle enthält die Deskriptoren.
 - Die Anfangsadresse einer Deskriptortabelle steht in einem Spezialregister.
 - Unterscheidung zwischen der globalen Deskriptortabelle und den lokale Deskriptortabellen;
- Ein **Deskriptor** beschreibt das Segment und umfasst folgende Informationen:
 - Basisadresse des Segments im linearen Adressraum
 - Limit: Größe des Segments
 - Zugriff: Zugriffsrechte auf Segment, Statusinformationen

Intel x86 Speicherverwaltung

■ Seitenverwaltung

■ Zweistufige Adressumsetzung mit 32-Bit physikalischer Adressierung

- 4 KBytes Seiten



Intel x86 Speicherverwaltung

■ Seitenverwaltung

■ Zweistufige Adressumsetzung mit 32-Bit physikalischer Adressierung

- Die lineare Adresse ist das Ergebnis der Segmentverwaltung.
- Aufteilung der linearen Adresse in drei Felder:
- **Verzeichnis (Bits 31 – 22):**
 - Index in das **Seitentabellenverzeichnis (Verzeichnistabelle page directory)**, dessen Basisadresse in einem speziellen Systemregister (CR3) abgelegt ist
 - Seitentabellenverzeichnis enthält 1024 Einträge;
 - Ein Eintrag verweist auf eine Seitentabelle der 2. Stufe;

■ Seite (Bits 21 – 12):

- Index in die ausgewählte **Seitentabelle (page table) der 2. Stufe**
- Eine Seitentabelle der 2. Stufe enthält 1024 Einträge.
- Ein Eintrag verweist auf die Anfangsadresse eines Seitenrahmens im physikalischen Speicher.

Intel x86 Speicherverwaltung

■ Seitenverwaltung

■ Zweistufige Adressumsetzung mit 32-Bit physikalischer Adressierung

■ Offset (Bits 11 – 0)

- Wird mit der Anfangsadresse des Seitenrahmens konkateniert, um die endgültige physikalische Adresse zu erhalten;

Kapitel 10

Betriebssystemunterstützung

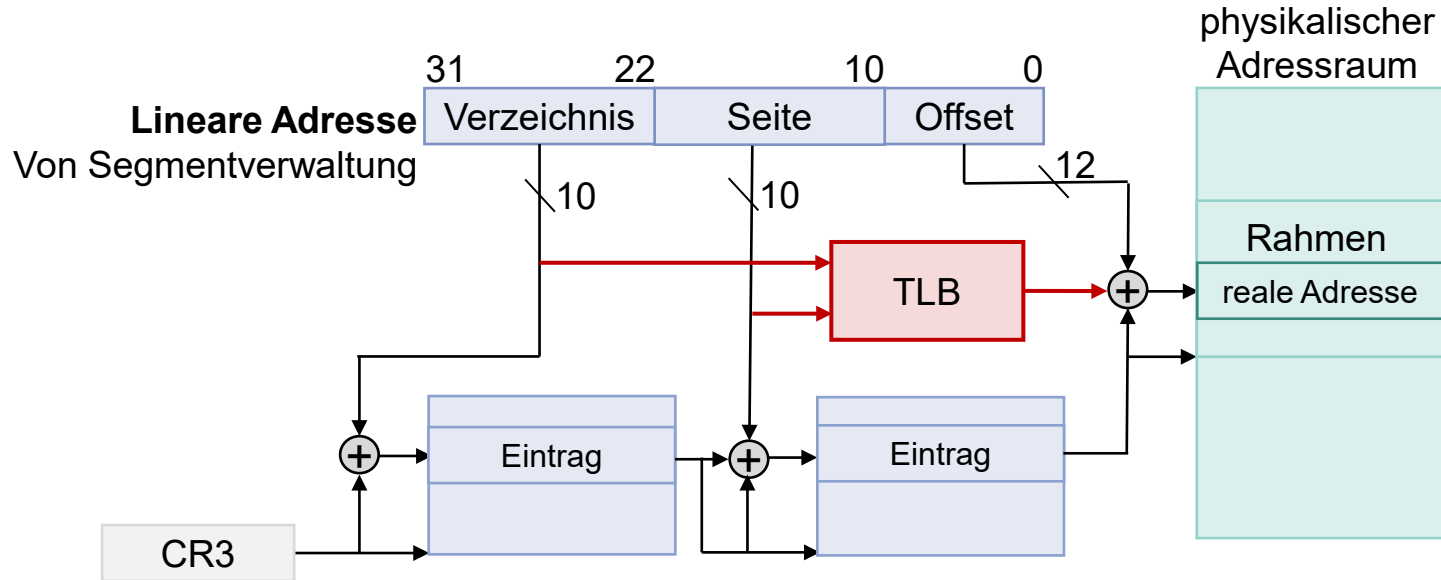
- Aufgaben des Betriebssystems
- **Virtuelle Speicherverwaltung**
 - Fallstudie IA-32
 - **Beschleunigung der Adressumsetzung**
- Schutzmechanismen
- Unterbrechungsbehandlung

Virtuelle Speicherverwaltung

■ Seitenverwaltung

■ Beschleunigung der Adressumsetzung: Translation Lookaside Buffer (TLB)

■ Beispiel: Intel x86 Speicherverwaltung



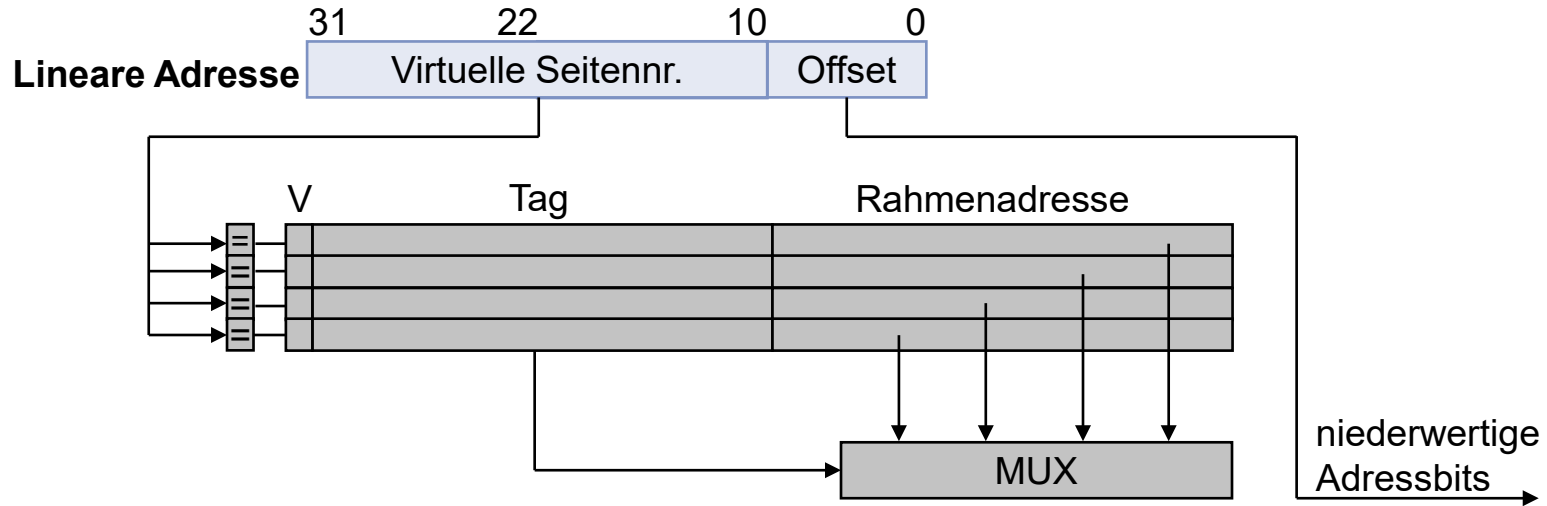
Virtuelle Speicherverwaltung

■ Translation Lookaside Buffer (TLB)

- Unterstützung der Seitenverwaltung bei der Umsetzung der virtuellen in die entsprechende physikalische Adresse
- **Aufbau:**
 - Cache-Speicher, der die Einträge der Seitentabellen enthält, die zuletzt verwendet worden sind.
 - Ausnützen des Lokalitätsprinzips:
 - Aufeinanderfolgende Zugriffe auf eine Seite, auf die einmal zugegriffen worden ist;
 - Getrennte TLBs für Befehle und für Daten
- **Prinzip:**
 - Bei einem Zugriff wird geprüft, ob ein Eintrag in dem TLB vorhanden ist:
 - Treffer: dann wird mit dem TLB-Eintrag auf den physikalischen Speicher zugegriffen
 - Fehlzugriff: Adressumsetzung mit Seitenverwaltung (über die Seitentabellen im Hauptspeicher)

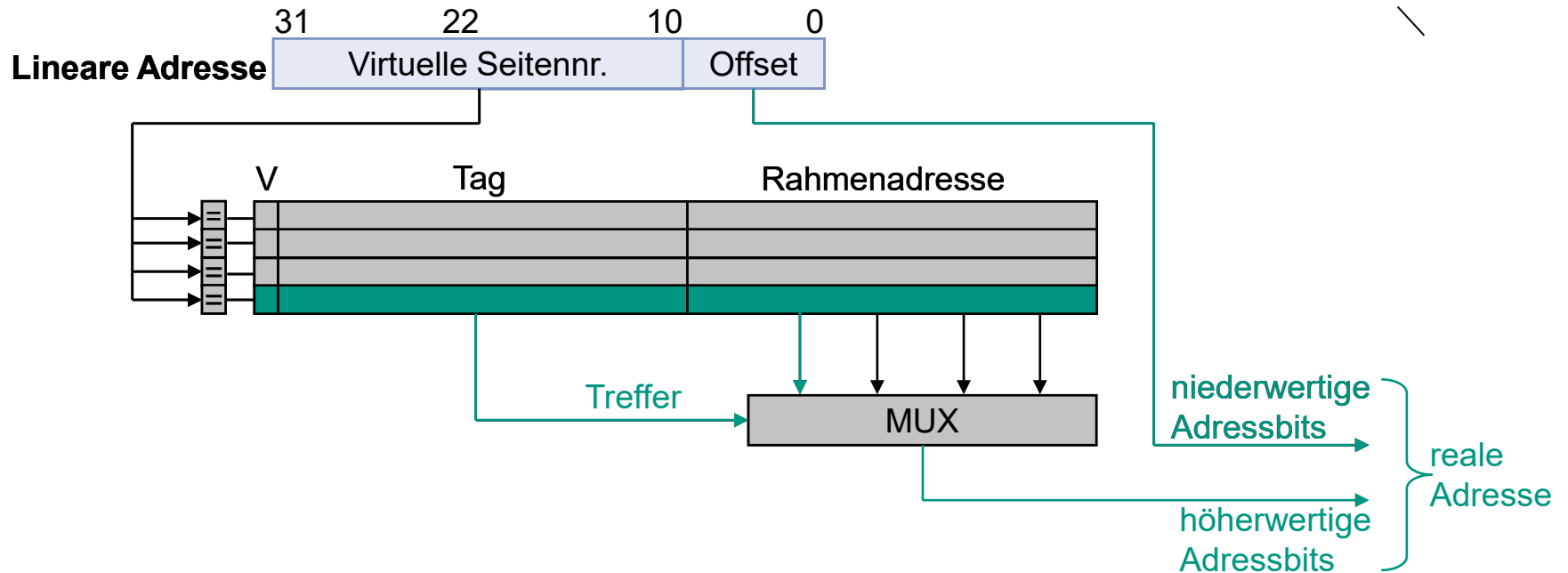
Virtuelle Speicherverwaltung

■ Translation Lookaside Buffer (TLB)



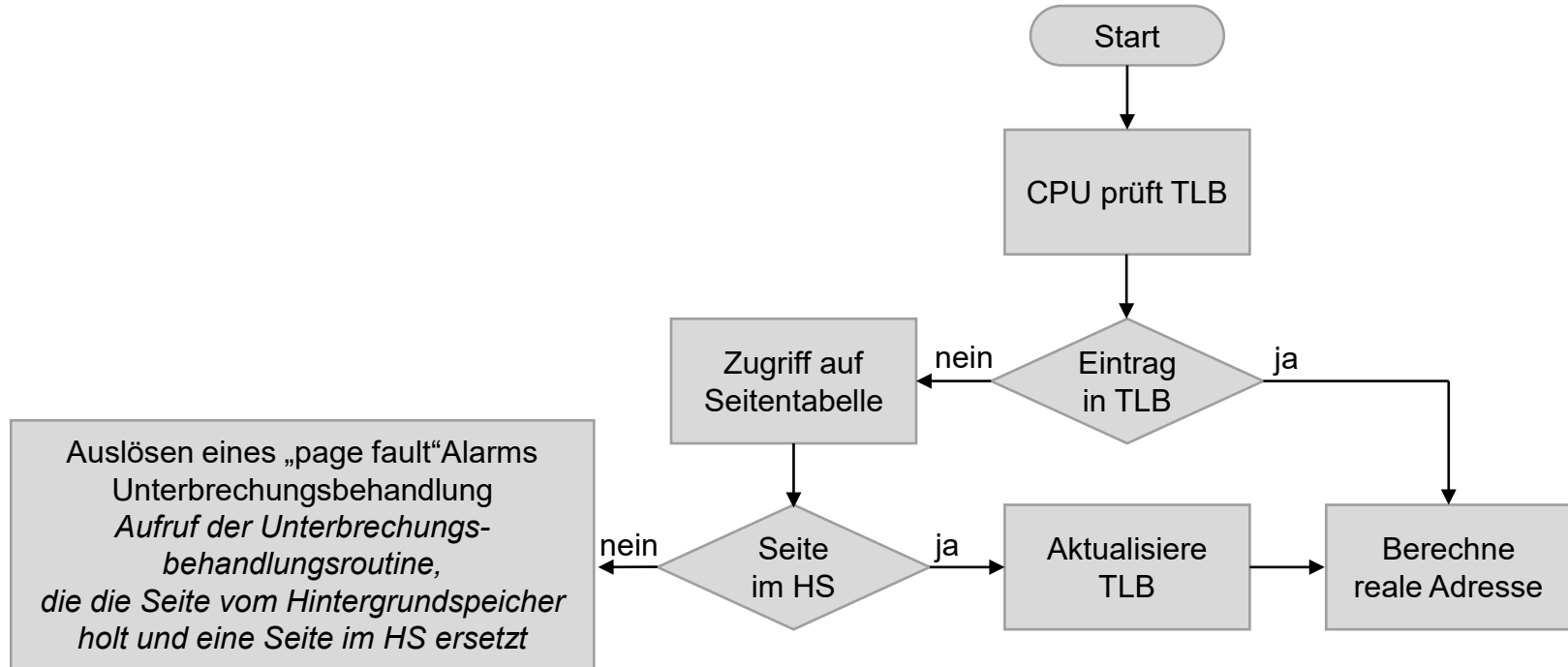
Virtuelle Speicherverwaltung

■ Translation Lookaside Buffer (TLB)



Virtuelle Speicherverwaltung

■ Translation Lookaside Buffer (TLB)



Kapitel 10

Betriebssystemunterstützung

- Aufgaben des Betriebssystems
- Virtuelle Speicherverwaltung
 - Fallstudie IA-32
 - Beschleunigung der Adressumsetzung
- **Schutzmechanismen**
- Unterbrechungsbehandlung

Schutzmechanismen

■ Schutzmechanismen (protection)

- Menge von Mechanismen, die gewährleisten, dass
 - Prozesse, die Ressourcen des Rechensystems (Prozessor, Speicher, Ein-/Ausgabe-Geräte) gemeinsam nutzen, sich weder beabsichtigt noch unbeabsichtigt gegenseitig beeinflussen können.
 - Aufgabe des Betriebssystems, die von Hardware unterstützt wird:
 - Beim Erkennen eines Fehlers oder einer Zugriffsverletzung wird ein Alarm ausgelöst.

Schutzmechanismen

■ Unterscheidung von mindestens zwei Betriebsmodi

- Trennung der Systemsoftware, z.B. des Betriebssystems, insbesondere des Ein-/Ausgabe-Subsystem (BIOS, Basic I/O-System), von den Anwendungsprozessen;
- **Supervisor-Modus (supervisor mode, Kernel-Modus):**
 - Zeigt an, dass ein ausführender Prozess ein Betriebssystemprozess ist;
 - Erlaubt diesem Prozess den Zugriff auf alle Ressourcen des Rechnersystems:
 - z. B. Spezialregister, TLB, ...;
 - Systemdaten (z. B. Seitentabellen, Vektortabelle, ...);
 - Ausführung von Spezialbefehlen;

Schutzmechanismen

■ Unterscheidung von mindestens zwei Betriebsmodi

- Trennung der Systemsoftware, z.B. des Betriebssystems, insbesondere des Ein-/Ausgabe-Subsystem (BIOS, Basic I/O-System), von den Anwendungsprozessen;

■ Benutzermodus

- Zeigt an, dass ein ausführender Prozess ein Benutzerprozess ist;
- Eingeschränkter Zugriff auf die Ressourcen des Rechnersystems;
- Alarm bei Zugriffsverletzung;

Schutzmechanismen

■ Schutzmechanismen der Speicherverwaltung

- Trennung der Anwendungsprozessen voneinander;
 - Ist dies nicht gewährleistet, könnte ein fehlerhaftes Anwenderprogramm andere, fehlerfreie Programme beeinflussen (Schutzebenen und Zugriffsrechte).
- **Virtueller Speicher**
 - Unterstützt die gemeinsame Nutzung des Hauptspeichers durch mehrere gleichzeitig aktive Prozesse.
 - Erfordert den Schutz des Adressraums eines Prozesses vor den anderen Prozessen;

Schutzmechanismen

■ Schutzmechanismen der Speicherverwaltung

■ Segmentdeskriptoren

■ Zugriffsattribute für Segmente:

- Garantieren, dass nur unter den vereinbarten Bedingungen auf ein Segment zugegriffen werden darf.

- Lese- und Schreibzugriff erlaubt oder

- Nur Lesezugriff erlaubt;

- Segmentgröße

- Schutz vor einer Überschreitung der Segmentgrenze

- Überprüfung der Zugriffsattribute durch den Prozessor

- Alarm bei Verletzung der Zugriffsattribute

Schutzmechanismen

■ Schutzmechanismen der Speicherverwaltung

■ Einträge in Seitentabellen

■ Zugriffsattribute für Seite

- Garantieren, dass nur unter den vereinbarten Bedingungen auf ein Segment zugegriffen werden darf.

- Lese- und Schreibzugriff erlaubt oder

- Nur Lesezugriff erlaubt;

- Supervisor- / oder Benutzer-Modus

■ Weitere Informationen: Statusinformationen

- Auf eine Seite wurde schreibend zugegriffen.

- Angabe eine Seite in den Cache geladen werden darf (cacheable / noncachable).

- Aktualisierungsstrategie für Cache (write-through, copy-back).

Kapitel 10

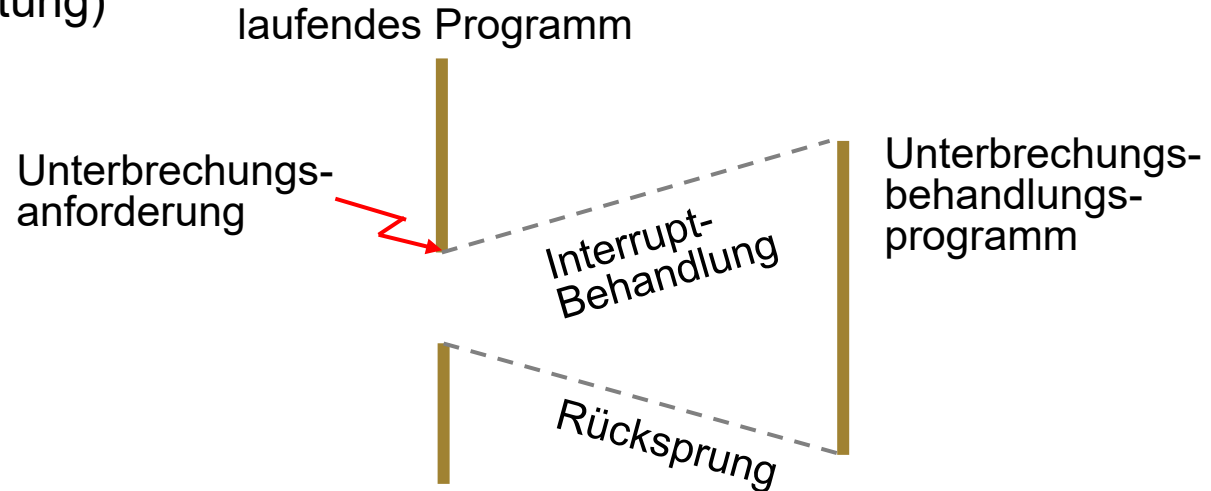
Betriebssystemunterstützung

- Aufgaben des Betriebssystems
- Virtuelle Speicherverwaltung
 - Fallstudie IA-32
 - Beschleunigung der Adressumsetzung
- Schutzmechanismen
- **Unterbrechungsbehandlung**

Unterbrechungsbehandlung

■ Unterbrechungsbehandlung (Exception Processing)

- Bei der Abarbeitung eines Programms können Situationen eintreten, die eine Unterbrechung dieser Abarbeitung erforderlich machen, um auf die eingetretene Situation entsprechend reagieren zu können
- Reaktion eines Prozessors auf eine Unterbrechungsanforderung (Ausnahmebearbeitung)



Unterbrechungsbehandlung

■ Unterbrechungsbehandlungsroutine vs. Unterprogramm

■ Aktivierung:

- `call subroutine` bei Unterprogramm
- Hardware-Aktivierung durch Alarm bei Ausnahmeroutine;

■ Beendigung:

- `ret` - Befehl bei Unterprogrammen (return from subroutine)
- `reti` - Befehl bei Ausnahmebehandlung (return from interrupt)

■ Einsprungsadresse

- ins Unterprogramm: direkt im Programm;
- bei Ausnahmebehandlung über Vektortabelle (Interrupttabelle)

Unterbrechungsbehandlung

■ Unterbrechungsbehandlungsroutine vs. Unterprogramm

■ Status retten:

- Unterprogrammaufruf sichert meist nur die Inhalte von PC, Statusregister und Register auf den Laufzeitkeller.
- Bei Ausnahmebehandlungs-Aufruf meist auch das Prozessorstatuswort (PSW)

■ Aufruf:

- Unterprogrammaufrufe werden immer durchgeführt
- die meisten Ausnahmebehandlungen werden nur aktiviert, falls das Interrupt-Enable-Bit im Steuerregister (PSW) gesetzt ist;
 - Privilegierte Befehle zum Setzen und Zurücksetzen des IF-Flags

Unterbrechungsbehandlung

■ Unterbrechungsbehandlung

■ Unterbrechungen aktivieren

- Mit einem Bit im Statusregister können Unterbrechungen aktiviert oder deaktiviert werden.
- Privilegierte Befehle zum Setzen und Zurücksetzen des IF-Flags

■ Vektortabelle

- Jeder Unterbrechungsart ist eine Unterbrechungsbehandlungsroutine zugeordnet, die die eigentliche Ausnahmebehandlung durchführt;
- Enthält als Einträge die Anfangsadresse der Unterbrechungsbehandlungsroutinen;
- Die Basisadresse der Vektortabelle steht in einem Spezialregister.
- Vektornummer ist ein Index in die Tabelle.

Unterbrechungsbehandlung

■ Arten von Unterbrechungen

- Es gibt prozessorinterne und prozessorexterne Ursachen für die Situationen, die eine Programmunterbrechung bewirken:
 - Interrupts
 - Traps
 - Exceptions

Unterbrechungsbehandlung

■ Arten von Unterbrechungen: Beispiel Intel IA-32

Vektornummer	Unterbrechungs- bedingung	Quelle	Art	
0	Division durch 0	DIV, IDIV	Fault	Prozessor entdeckt bei Ausführung eines Divisionsbefehls einen Divisor mit dem Wert 0
1	Debug Exception	jeder Befehl	Traps/ Faults	
2	NMI	extern	NMI	
...
10	Invalid TSS	JMP,CALL, IRET, INT (intern)	Fault	

Unterbrechungsbehandlung

■ Arten von Unterbrechungen: Beispiel Intel IA-32

Vektornummer	Unterbrechungs- bedingung	Quelle	Art	
11	Segment not Present	Segment- Registerbef.,	Fault	Present-Bit eines Deskriptors ist zurückgesetzt
12	Stack Fault	Zugriff auf Stack	Fault	
13	General Protection	Speicherzugriff (int.)	Fault	Verletzung von Zugriffsrechten
14	Page Fault	Speicherzu- zugriff (int.)	Fault	Seite nicht im Speicher
16	FP Error	ESC, FP-Bef.	Fault	Fehler bei der Ausführung eines Gleitkommabefehls

Unterbrechungsbehandlung

■ Arten von Unterbrechungen: Beispiel Intel IA-32

Vektornummer	Unterbrechungs- bedingung	Quelle	Art	
0-255	Software Interupts	INTn	Trap	
32-255	Maskierbare Interrupts	Extern	Inter.	