

□ Anmeldung zur TI-Klausur:

- Einwurf der Zulassungsbescheinigung in den Briefkasten im Untergeschoss des Informatikgebäudes am Fasanengarten **bis spätestens 27. August**. Es handelt sich um den gleichen Briefkasten, in dem die Übungsblätter eingeworfen werden **UND**
- **Online-Anmeldung auf der TI-Homepage**

□ Hilfsmittel sind nicht erlaubt

□ Dauer der Klausur:

- **Informatik: 120 Minuten** (9.00-11.00 Uhr)
- **Informationswirtschaft: 60 Minuten** (9.00-10.00 Uhr)

□ Studentenausweise unbedingt in die Klausur mitbringen

□ Hörsaal-Verteilung wird rechtzeitig bekannt gegeben (TI-Homepage)



Segmentierung vs. Seitenwechsel

□ Segmentierung (Segmente variabler Größe)

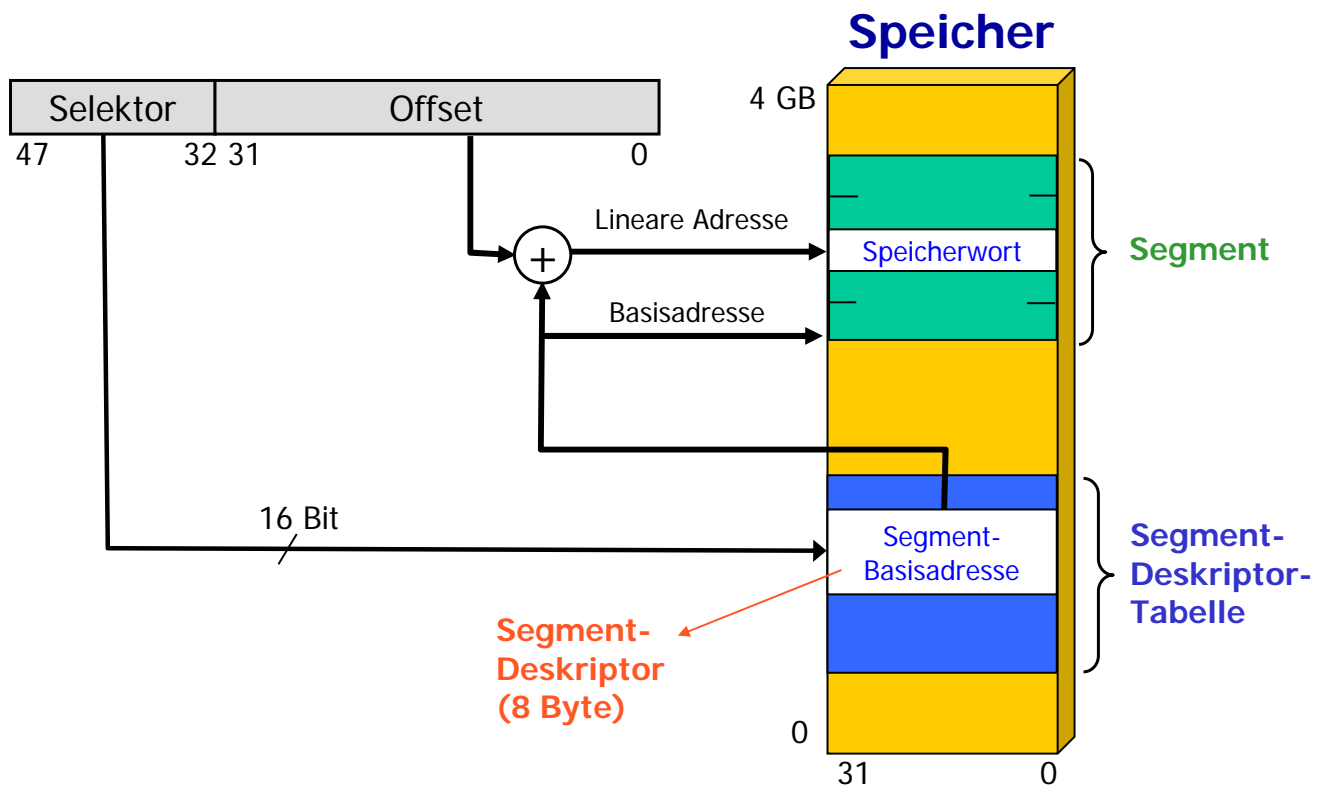
- logische Abbildung einer Programmstruktur
- geringer Datentransfer
- umfangreicher Datentransfer beim Ein-/Auslagern
- externe Fragmentierung

□ Seitenwechsel-Verfahren (Seiten fester Größe)

- geringerer Verwaltungsaufwand
- bessere Hauptspeicherauslastung
- häufiger Datentransfer
- interne Fragmentierung

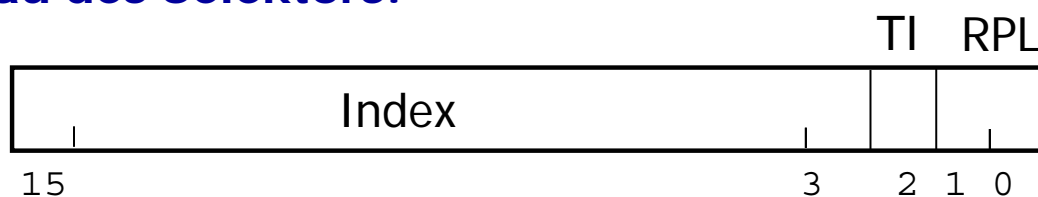


Segmentorientierte Speicherverwaltung (x86-Prozessoren)



Segmentorientierte Speicherverwaltung (x86-Prozessoren)

Aufbau des Selektors:



- Indexfeld und TI (Tabellenindikator) selektieren einen Eintrag in der Segment-Deskriptor-Tabelle
- RPL-Bits (Requested Privilege Level) geben eine Privileg-Ebene an, die ein Befehl besitzen muss, um auf das gewünschte Segment zugreifen zu dürfen.



Segment-Deskriptoren: Jeder Segment-Deskriptor beschreibt das zugehörige Segment durch folgende Attribute:

- Segment-Basisadresse (*base address*)
- Segment-Größe in Bytes (*limit*)
- Zugriffsrechte auf das Segment (*access rights*) zur Realisierung von Schutzmechanismen.

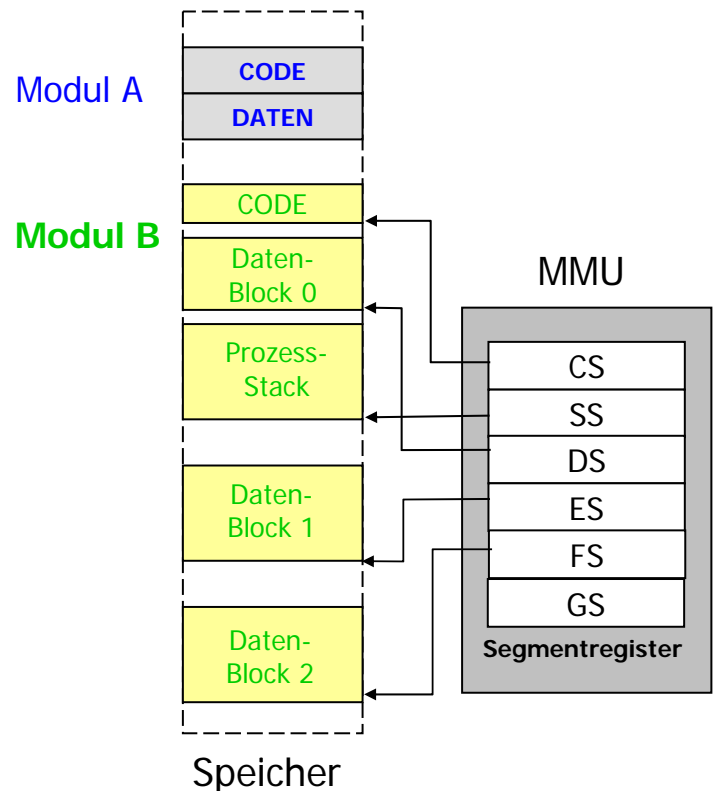


- ❑ Jedes Speicherwort wird durch einen **Segment-Selektor**, der den Segmentanfang kennzeichnet, und einen **Offset** innerhalb des Segments adressiert
- ❑ Durch Lokalitätseigenschaften in Programmen wird nicht bei jedem Zugriff auf den Hauptspeicher ein neues Segment benutzt → Segment-Selektoren wechseln selten.
- ❑ Aus Geschwindigkeitsgründen werden die Segment-Selektoren in speziellen Segment-Registern gespeichert. Hierzu existieren verschiedene Register für verschieden Segment-Typen: Code-Segment (CS), Stack-Segment (SS), Daten-Segment (DS), Extra-Segment (ES, FS, GS, ...) → Adressierung eines Speicherworts durch ein **Segment-Register** und ein **Offset**
- ❑ Wichtige Informationen über das ausgewählte Segment werden in einem **Segment-Deskriptor-Cache** abgelegt → keine Speicherzugriffe beim Lesen der Segmenteigenschaften



Segmentorientierte Speicherverwaltung (x86-Prozessoren)

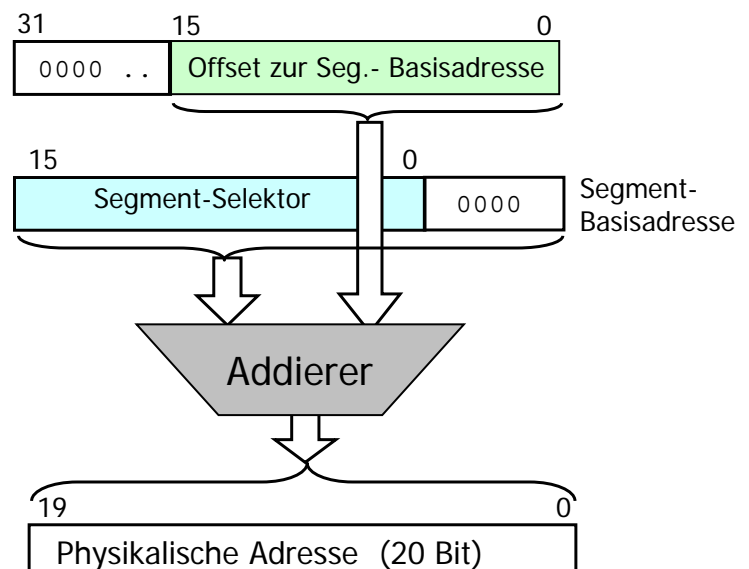
- ❑ Jedes Segment, dessen Anfangsadresse in einem Segmentregister abgelegt ist, befindet sich physikalisch im Hauptspeicher.
- ❑ Die durch die Segmentregister spezifizierten Segmente bilden die Arbeitsmenge (*working set*) eines Prozesses



Adressierungs-Modi (x86-Prozessoren)

Um Kompatibilität zu den älteren Mikroprozessoren innerhalb der x86-Prozessorfamilie zu gewährleisten, können die Prozessoren in verschiedenen Adressierungsmodi arbeiten:

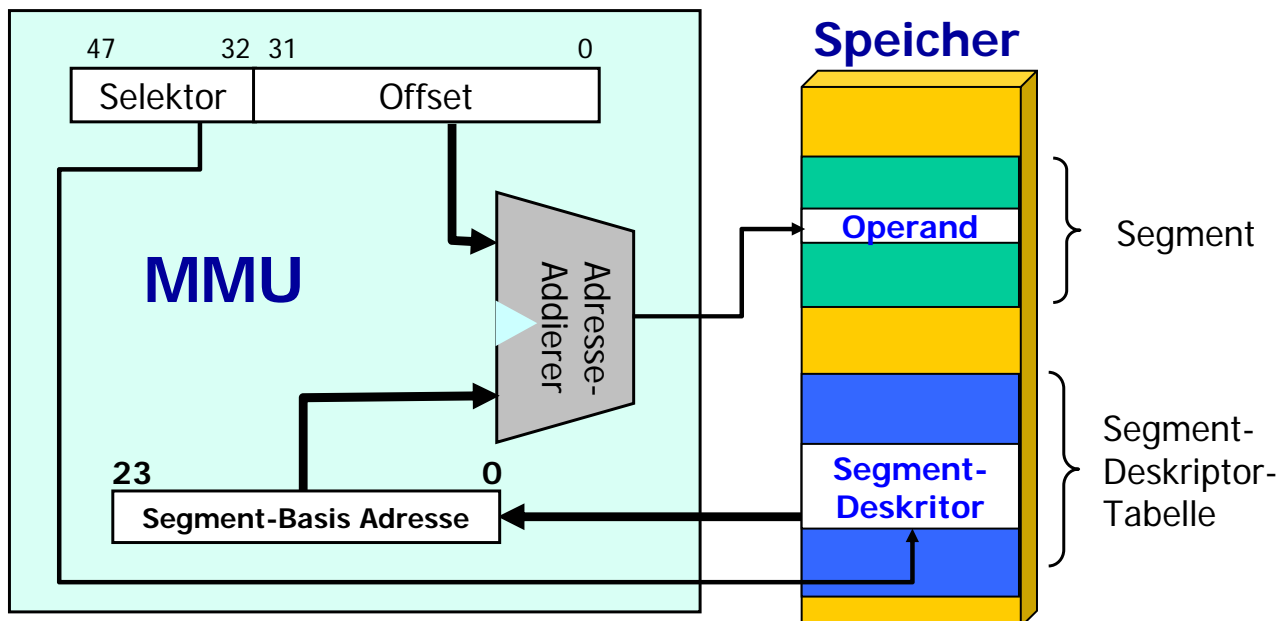
- ❑ **Real (Address) Mode**
(Kompatibilitätsmode zu 8086)
 - 20 Bit lange physikalische Adressen
 - 1 MByte max. adressierbare Hauptspeicherkapazität
 - max. Segmentlänge: 64 KByte



Adressierungs-Modi (x86-Prozessoren)

❑ Protected (Virtual Address) Mode

Modus mit vergrößertem Adressraum und erweiterten Fähigkeiten zu Speicherschutz, Multitasking



Adressierungs-Modi (x86-Prozessoren)

- ❑ Abbildung eines virtuellen Adressraums von mehreren Tera-Bytes auf einen 4 GByte großen physikalischen Adressraum
- ❑ Der Segment-Selektor spezifiziert hier nicht die Basisadresse des Segments selbst, sondern verweist auf den **Segment-Deskriptor** in der **Segment-Deskriptor-Tabelle** im Hauptspeicher
- ❑ Jeder Segment-Deskriptor beschreibt das zugehörige Segment durch folgende Attribute:
 - die Segment-Basisadresse (*base address*)
 - die Segment-Größe in Bytes (*limit*)
 - die Zugriffsrechte auf das Segment (*access rights*)



Seitenorientierte Speicherverwaltung

Der Speicher wird in viele kleine Seiten gleicher Länge unterteilt

Seitenlängen von 256 Byte bis 8 kByte

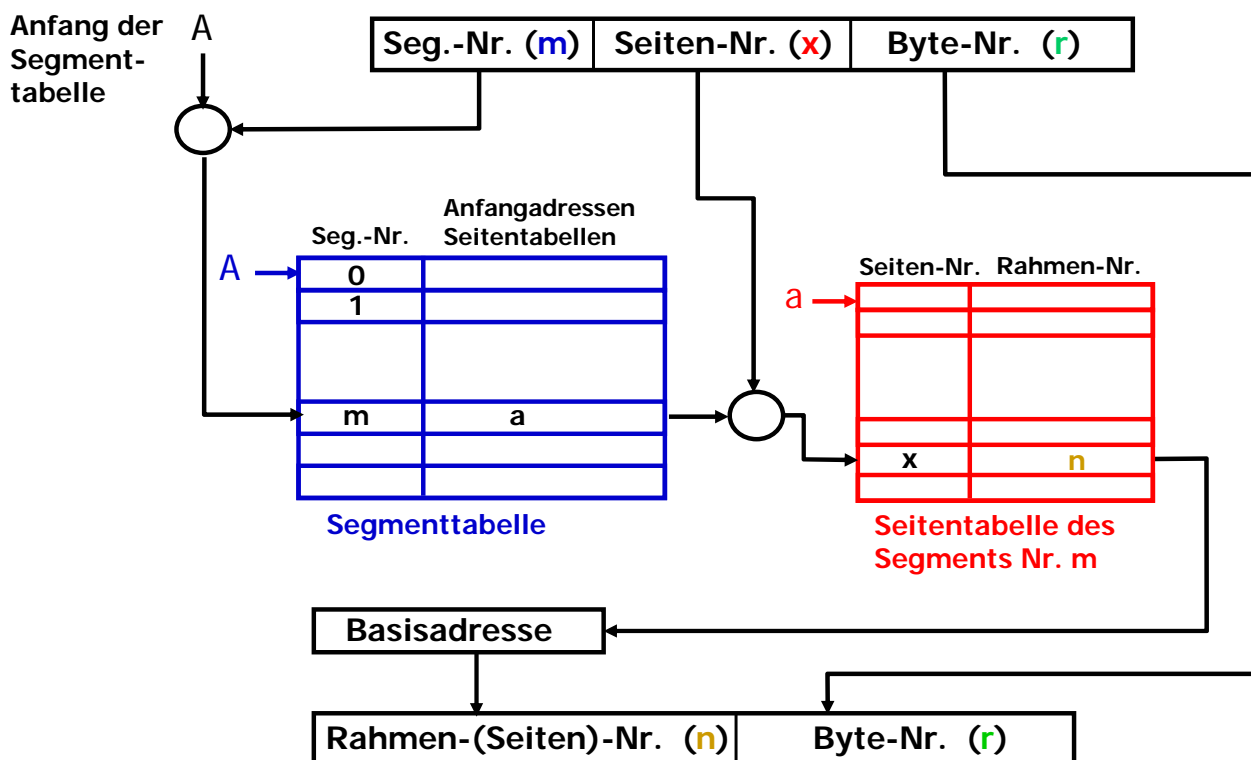
Im Unterschied zu Segmenten können Seiten nicht an beliebiger Stelle im Speicher beginnen, sondern nur in einem festen an der Seitengröße orientierten Raster

Fallbeispiel x86-Prozessoren (ab 80386):

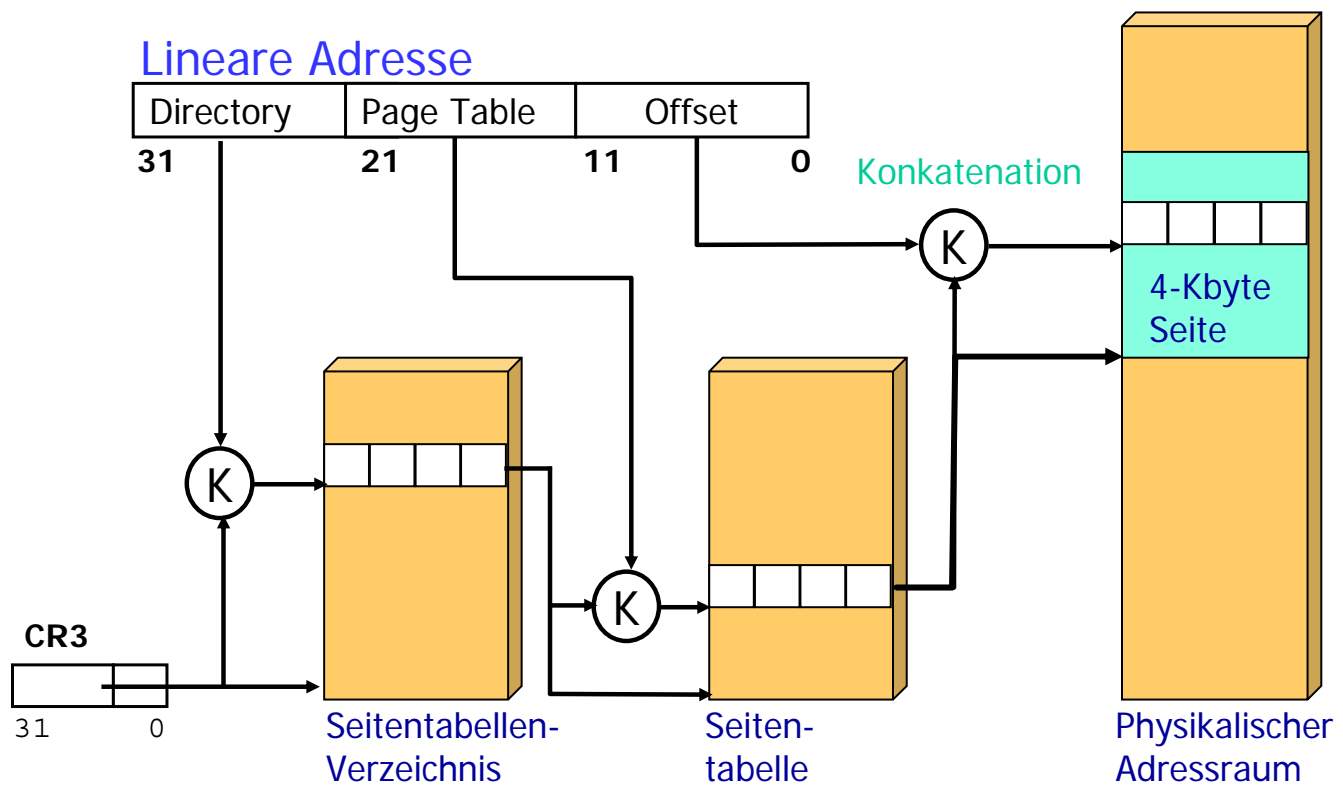
Aus Kompatibilitätsgründen zum 80286 unterstützen die x86-Prozessoren ab dem 80386 sowohl eine segment- als auch eine seitenorientierte Speicherverwaltung



Segmentierung mit Seitenwechsel



Zweistufiges Seitenwechsel-Verfahren



Zweistufiges Seitenwechsel-Verfahren

- ❑ Seitentabellenverzeichnis → Seitentabellen → Seiten
- ❑ Seitentabellenverzeichnis, Seitentabellen und Seiten sind jeweils 4 KByte groß und jeder Tabellen-Eintrag umfasst 4 Byte
- ❑ In den Seitentabellen und -verzeichnis sind jeweils 1024 Einträge enthalten.
- ❑ Lineare Adresse wird in drei Teilen zerlegt. Die höchstwertigen 10 Bits selektieren einen Eintrag im Seitentabellen-Verzeichnis, dessen Basisadresse in einem speziellen Systemregister (CR3) abgelegt ist.
- ❑ Die nächsten 10 Bits der linearen Adresse selektieren einen der 1024 Einträge aus der Seitentabelle. In diesem Eintrag steht die Basisadresse einer Seite.
- ❑ Die niedrigstwertigen 12 Bits werden als Offset zur Seitenadresse addiert, um die endgültige, physikalische Adresse zu erhalten.



Anmerkungen

Sowohl bei segmentorientierter wie bei seitenorientierter Speicherverwaltung gilt:

Befindet sich eine Seite oder ein Segment nicht im Hauptspeicher, so löst der Prozessor eine Unterbrechung aus, um die Seite oder das Segment durch das Betriebssystem zu laden (Seiten- oder Segmentfehler).



Anmerkungen

Erkennung eines Segmentfehlers:

Bit im Segment-Deskriptor zeigt an, ob das Segment im Hauptspeicher ist oder nicht.

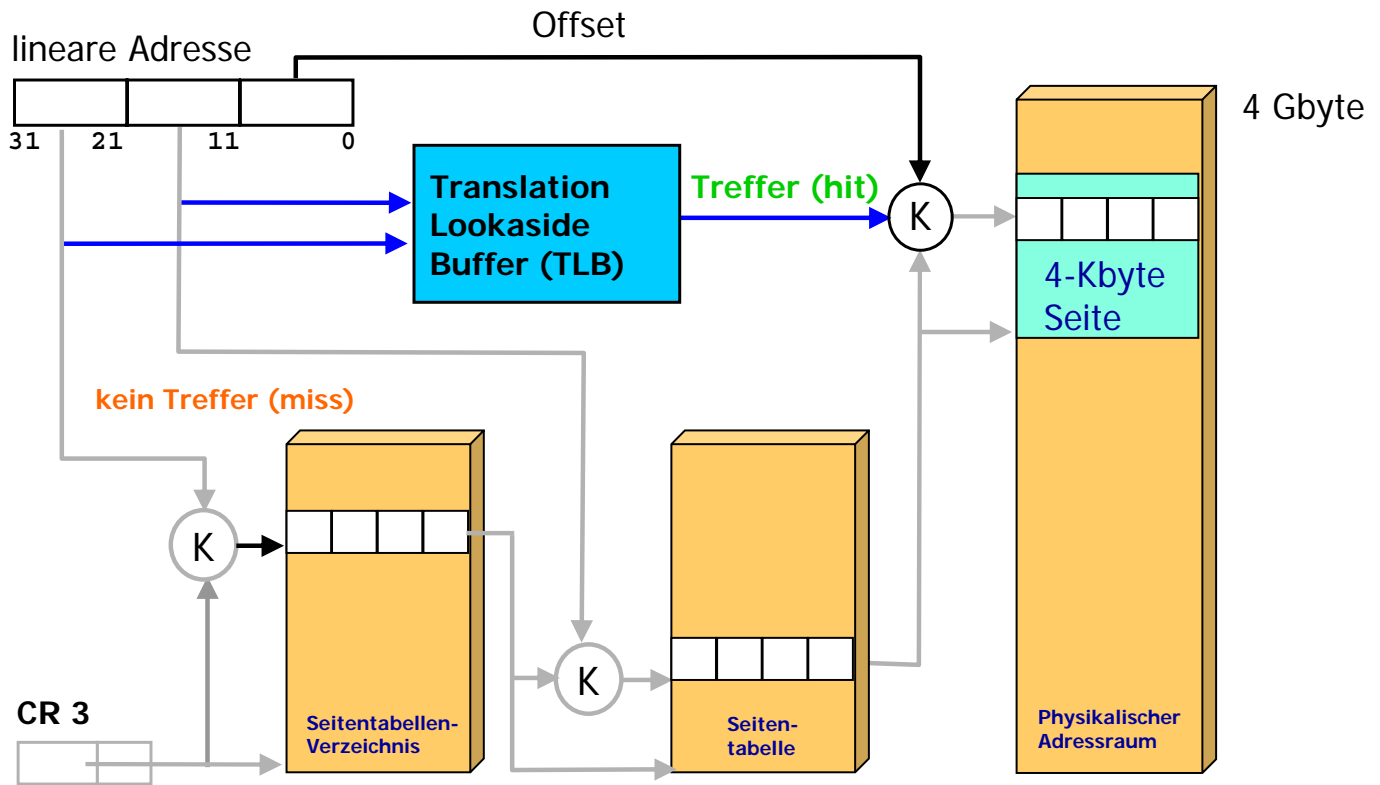
Erkennung eines Seitenfehlers:

Seite oder Seitentabelle befindet sich nicht im Hauptspeicher (Seitenfehler).

Spezielles Kennungbit im Seitentabellen-Verzeichnis (Seitentabellenfehler)



Beschleunigung der Adressberechnung durch einen Cache



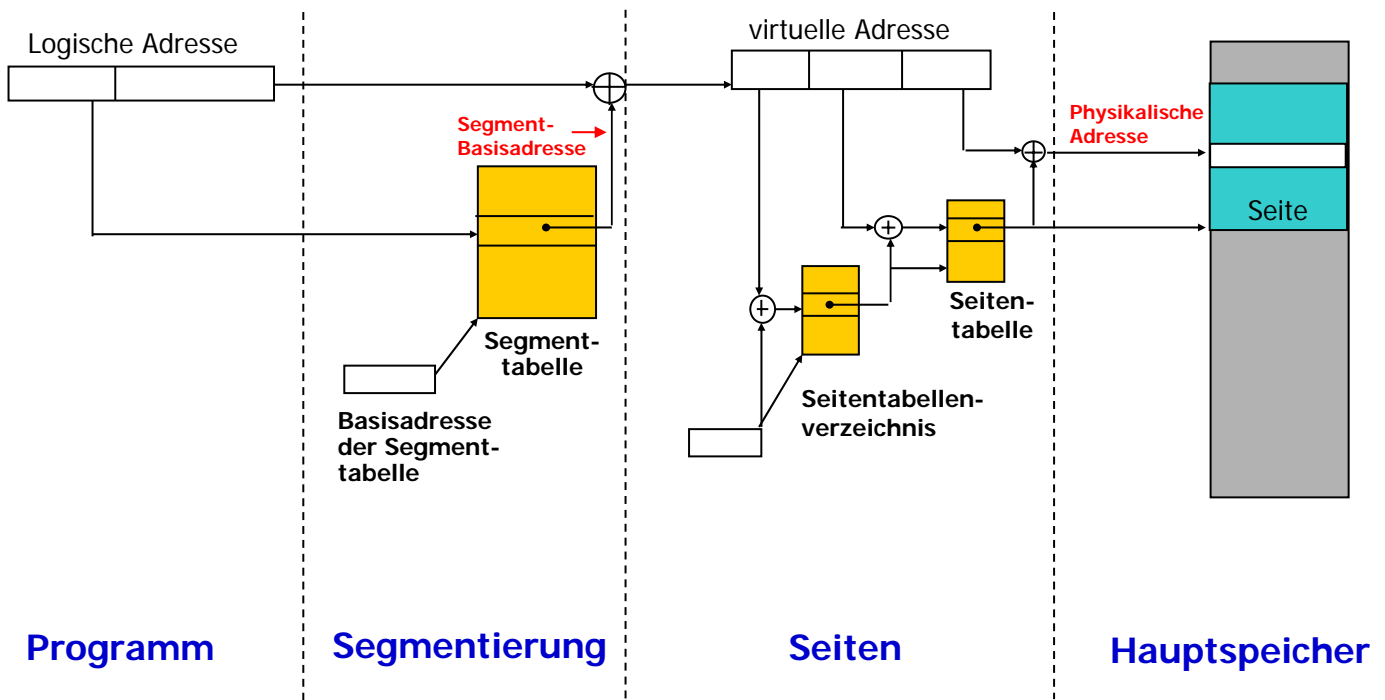
Beschleunigung der Adressberechnung durch einen Cache

Ein schneller voll-assoziativer Cache (*Translation Lookaside Buffer*, TBL) speichert automatisch die zuletzt benutzen Einträge aus dem Seitentabellenverzeichnis und der Seitentabelle

- ➔ Im Trefferfall (Trefferquote ca. 90 %) muss nicht auf die im Hauptspeicher liegenden Seitentabellen zugegriffen werden



Automatische Adressübersetzung bei 80486 und Pentium-Prozessoren



Schutzmechanismen

Moderne Mikroprozessoren bieten Schutzmechanismen an, um während der Laufzeit von Programmen unerlaubte Speicherzugriffe zu verhindern. Dies geschieht im wesentlichen durch:

- Trennung der Systemsoftware, z. B. des Betriebssystems, insbesondere des Ein-/Ausgabe-Subsystem (BIOS, basic I/O system), von den Anwendungsprozessen.
- Trennung der Anwendungsprozessen voneinander. Ist dies nicht gewährleistet, könnte ein fehlerhaftes Anwenderprogramm andere, fehlerfreie Programme beeinflussen (Schutzebenen und Zugriffsrechte)



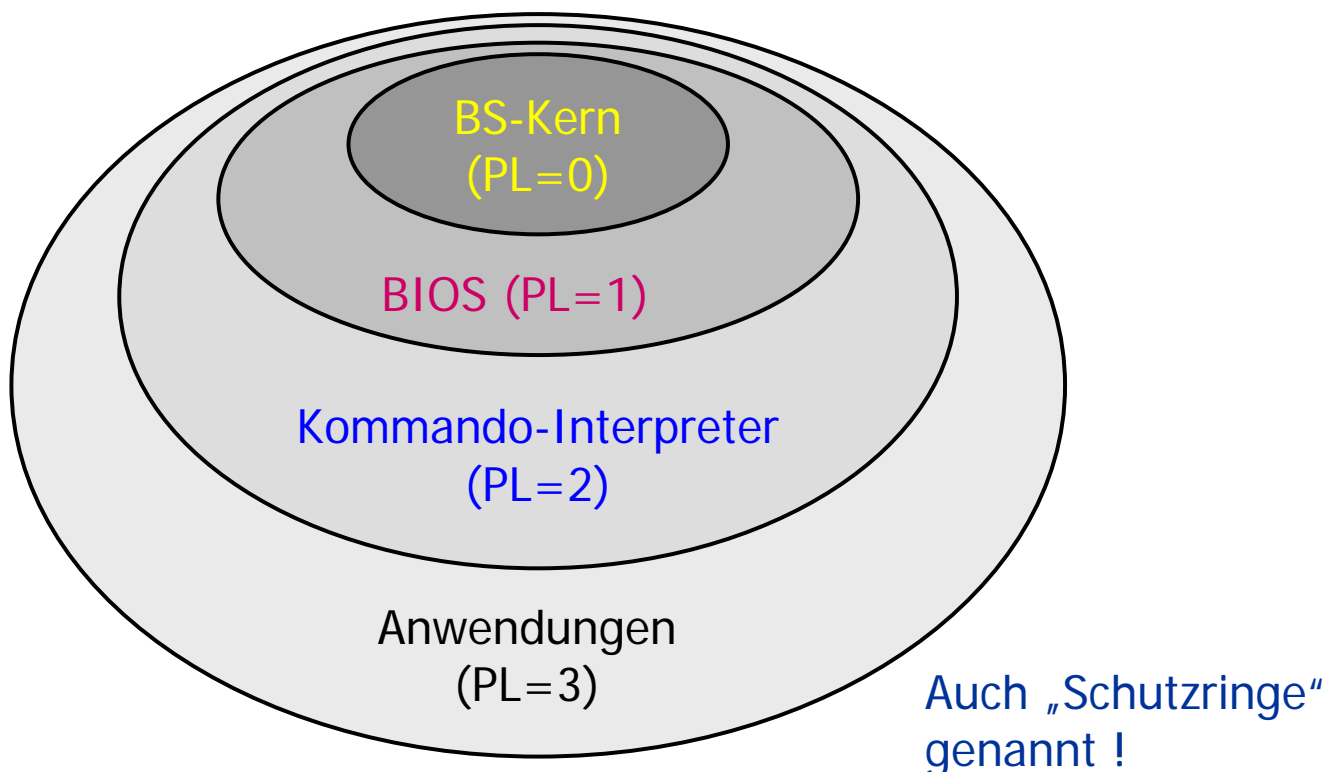
Schutzmechanismen

Schutzebenen (PL: Privilege Levels): Wichtigstes Mittel zur Realisierung von Schutzmechanismen

- Zweischatzebenen (bei Seitenverwaltung der x86-Prozessoren):
 - Betriebs-Systemmodus (supervisor mode)
 - Benutzermodus (user mode)
 - Ein Auftrag im Benutzermodus darf keine Daten und Programme des höherprivilegierten Betriebssystemmodus benutzen.
- Vierstufige Hierarchie bei Segmentverwaltung:
 - Privileg-Ebene PL = 0 entspricht der vertrauenswürdigsten Ebene (most trust level). Privileg-Ebene PL = 3 entspricht der am wenigsten vertrauenswürdigsten Ebene (least trust level)



Beispiel eines Systems mit vier Schutzebenen



Regeln für den Zugriffsschutz (*protection rules*)

- ❑ Ein Prozess darf nur auf Daten zugreifen, die höchstens genauso vertrauenswürdig (*trusted*) sind wie er selbst
- ❑ Ein Prozess darf nur Code benutzen, der mindestens genauso vertrauenswürdig ist wie er selbst
- ❑ Zugriffsrechte (*access rights*) garantieren, dass nur unter bestimmten Voraussetzungen auf die im Speicher abgelegten Informationen zugegriffen werden darf.

Sowohl die Schutzebenen als auch die Zugriffsrechte werden hardwaremäßig bei der Speicherverwaltung durch die Vergabe von Privileg-Ebenen und Rechten an Speichersegmente und Speicherseiten unterstützt.

