

MMU

Rainer Müller

Department Informatik 4
Verteilte Systeme und Betriebssysteme
Friedrich-Alexander-Universität Erlangen-Nürnberg

WS 2014/2015



Fragen zu Aufgabe 2?



Motivation

- Befehle können auf Speicher zugreifen
- Befehle können dadurch
 - andere Prozesse stören
 - wichtige Daten anderer Prozesse lesen

⇒ Einschränken des Zugriffs auf Speicher für Mehrbenutzer-, Mehrprozesssysteme notwendig!



Definitionen

- logische Adresse: Adresse im Operanden (und welches Basisregister benötigt)
- lineare Adresse: logische Adresse + Basis
- physikalische Adresse: Adresse im Speicher



Speichermodelle

- *Real Mode:*
Basis $\ll 4$ + logische Adresse = physikalische Adresse
- *Flach:*
logische Adresse = physikalische Adresse
- *Segmentierung:*
Basis + logische Adresse = physikalische/lineare Adresse
- *Paging:*
Aufteilen des Speichers in Seiten und Abbilden der linearen auf eine physikalische Adresse



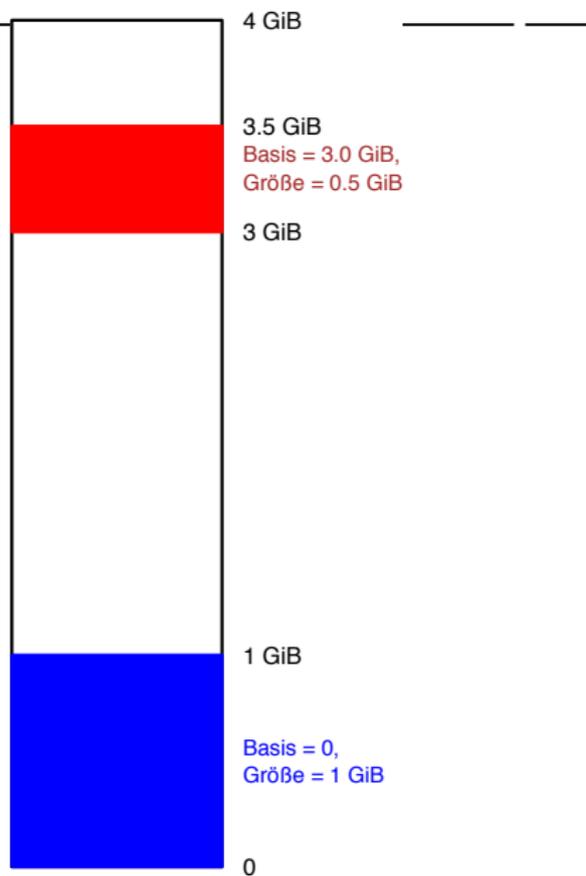
Segmentregister

- 6 Segmentregister: CS, DS, SS, ES, FS, GS
- Register für Codesegment CS
 - relative Sprünge
 - Laden von Opcodes
 - kann nur implizit verändert werden (`call`, `ret`, `ljmp`, ...)
- Register für Stack-Segment SS
 - Adressierung relativ zu `%esp`
 - Adressierung relativ zu `%ebp`
 - Befehle mit Stackoperationen `push`, `pop`, ...
- Datensegmente DS, ES, FS, GS
- Standard: DS
- Bestimmte Opcodes: Anderes Segmentregister
- Prefixe zum Auswählen anderer Segmentregister



Probleme Segmentierung x86

- GDT enthält Basis und Limit für alle Segmente
- Segmentregister: Offset in GDT
- Adresse: Operand ist Offset im Segment



Wie viele Speicherzugriffe benötigt folgender Befehl bei Segmentierung?

```
addl $3, (%ebx)
```



Lösung

- Segmentregister enthalten nicht öffentlichen Anteil
- Eintrag in GDT
- Laden **nur** bei Aktualisieren des öffentlichen Anteils

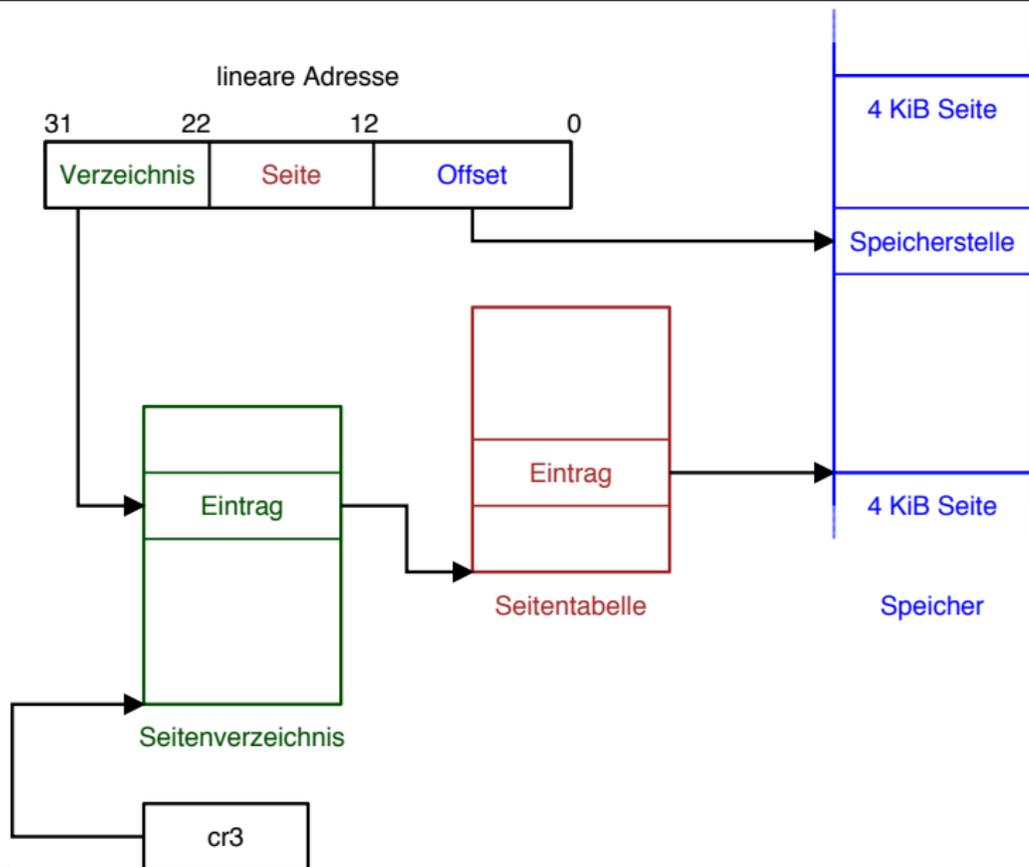


Paging

- Baut auf Segmentierung auf
- Aufteilen des Speichers in Pages (4 KiB)
- Übersetzen der linearen Adresse mittels Seitentabelle auf physikalische Adresse
- Problem 1:1 Übersetzung: 4 GiB / 4 KiB wären 1048576 Einträge
- Daher Indirektionsstufe über Seitenverzeichnis
- Register `cr3` zeigt auf Anfang der Seitenverzeichnis



Paging x86



1. Wie viele Speicherzugriffe benötigt folgender Befehl mit Segmentierung und Paging?
`addl $3, (%ebx)`
2. Was muss ein Betriebssystem beim Prozesswechsel tun, um unterschiedliche Adressräume zu garantieren?



1. Abhilfe sorgt ein Cache für die Einträge: Translation Lookaside Buffer (TLB).
2. Das Register `cr3` umsetzen.

