Aufgabe 3 – Erweiterung um MMU mit Segmentierung und Paging

Rainer Müller

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

WS 2014/2015

Fragestunde

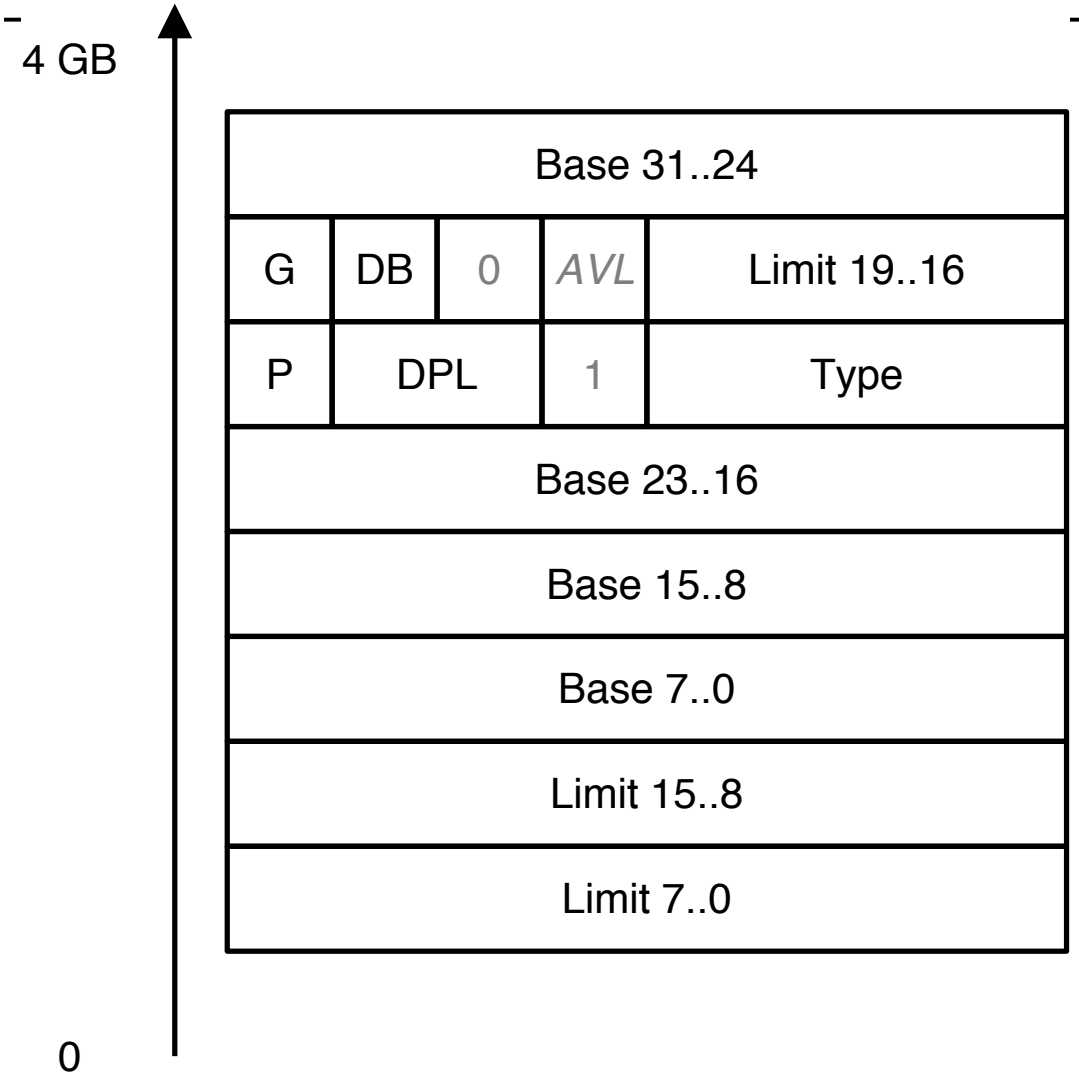
Fragen zu Aufgabe 2?

Segmentierung - Überblick

- 6 Segmentregister
  - öffentlicher Anteil: Offset in GDT
  - versteckter Anteil: Eintrag aus GDT
  - Laden des öffentlichen Anteils aktualisiert versteckten Anteil
  - Laden des öffentlichen Anteils aktualisiert `Accessed`-Bit in GDT
- automatisch aktiv im *Protected Mode*
- Null-Deskriptor ist ungültg und erzeugt `GP`-Fault beim Laden
- Zugriffsverletzungen bei Befehlen erzeugen `GP`-Fault, und verändern den Prozessorzustand **nicht**[1]

[1] Ausnahmen bestätigen die Regel: `popad`

Segmentierung - GDT



P: Present
DPL: Privilege Level
Type: Read/Write, ...
G: 0 $\rightarrow$ limit=bytes
DB: 1 $\rightarrow$ 32-Bit

Siehe: „Intel 64 and IA-32 Architectures Software Developer's Manual“, Volume 3: System Programming Guide, Kapitel 3.4.5.

Segmentierung: Bootstrapping im realen System

- startet im *Real Mode*
- Segmentregister werden daher als Bit 19..4 zur Adresse addiert
- privater Anteil: Basis=0, Limit=64 KiB
- Ausnahme `CS`: Basis = `0xffff0000`, Limit=64 KiB

Segmentierung: Bootstrapping im realen System

1. Interrupts ausschalten
2. GDT laden (`lgdt`)
3. in *Protected Mode* schalten (`CR0`, Bit 0 $\rightarrow$ 1)
4. **alle** Segmentregister initialisieren
5. ggf. Interrupts wieder anschalten

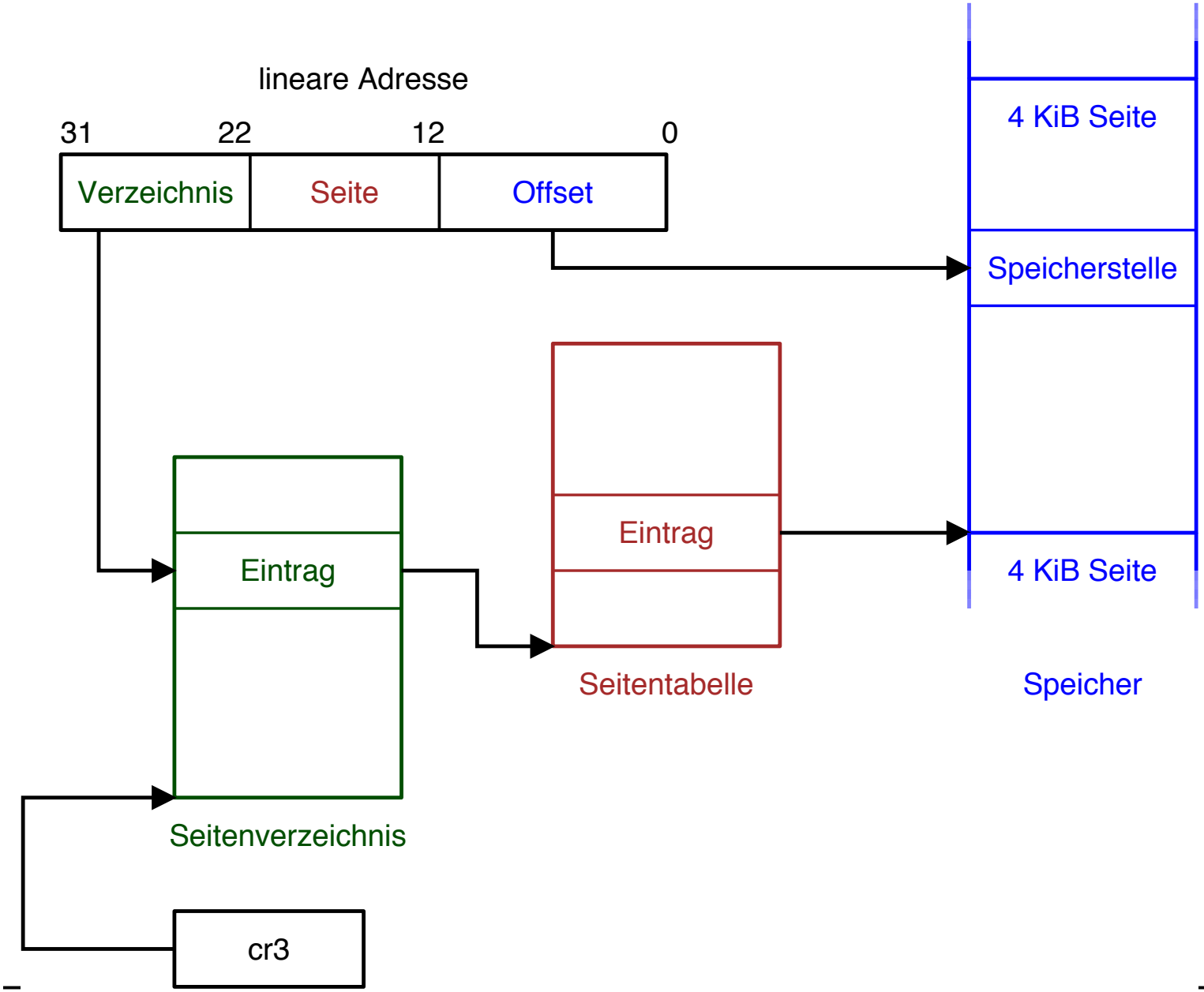
Bootstrapping: Fragen

- Warum sollten die Interrupts vor Laden der GDT ausgeschaltet werden?
- Wie kann das `CS`-Register initialisiert werden?

Segmentierung - Exceptions

- Ausführung analog zu Interrupts
- als letztes wird Fehlercode auf Stack gelegt
- Fehlercode entspricht Inhalt des Segment Registers, welches den Fehler erzeugt hat
- `iret` entfernt den Fehlercode nicht!
- Exception 11 (`NP`): „Segment nicht vorhanden“, bei Laden des Segment Registers
- Exception 12 (`SS`): „Stack Segment Fehler“, bei Zugriff auf das Stack-Segment
- Exception 13 (`GP`): „General Protection Fault“, bei Zugriff auf andere Segmente

Paging - Überblick

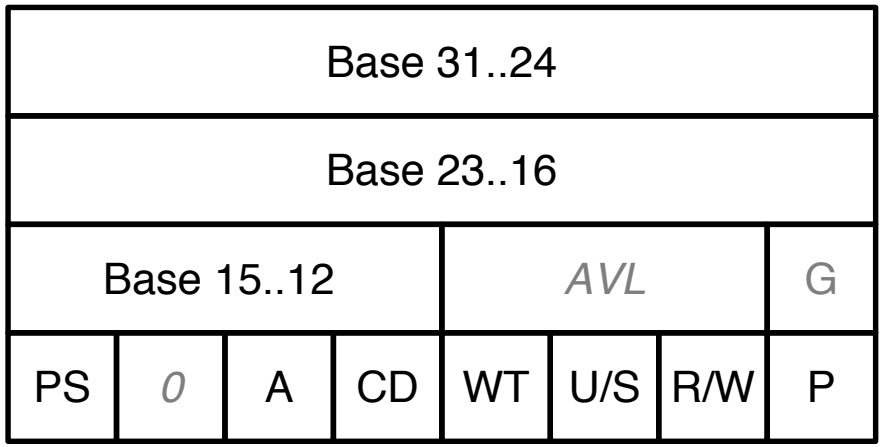


Paging - Überblick

Kontrollregister:

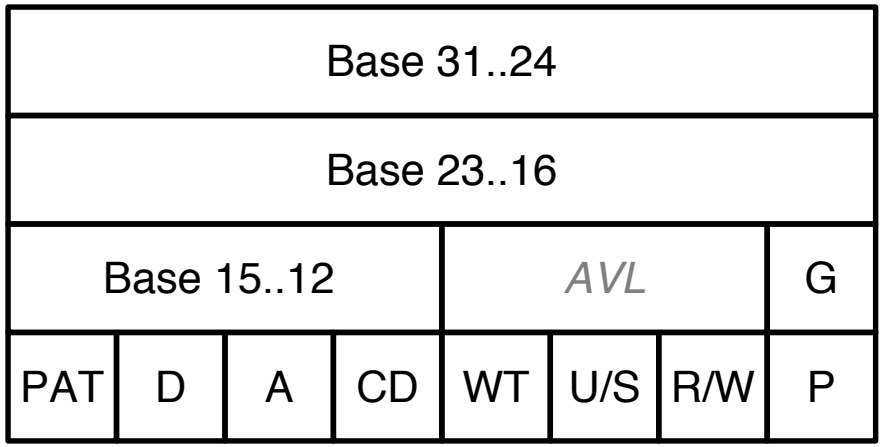
- `CR0`, Bit 31: Paging
- `CR3`, Bits 31..12: Basisadresse der Page Directory
- `CR2`: Lineare Adresse des Page Faults

Paging - Tabellenaufbau: Seitentabellenverzeichnis



P	Present
R/W	Schreiben erlaubt (1)
U/S	User-Seite (1)
WT	Write-Through forcieren (1)
CD	nicht cachebar (1)
A	Accessed
PS	Page Size, 0= 4 KiB
G	Global Page

Paging - Tabellenaufbau: Seitentabellenverzeichnis



P	Present
R/W	Schreiben erlaubt (1)
U/S	User-Seite (1)
WT	Write-Through forcieren (1)
CD	nicht cachebar (1)
A	Accessed
D	Dirty
PAT	Page Attribute Table Index
G	Global Page

Caching - Fragen

- Wozu kann das Caching einzelner Seiten komplett verhindert werden?
- Wozu kann das Caching einzelner Seiten auf `write-through` gestellt werden?

Ausblick (nicht mehr Teil der Aufgabe) . . .

- Wie funktionieren Privilegierungsstufen?
- Wie kann ein Betriebssystem seinen Stack schützen?