

# Virtuelle Speicherverwaltung

- Ein **Prozess** sollte auch dann noch *ablaufen* können, wenn er nur *teilweise im Hauptspeicher* ist.
- Wichtig ist, dass die Teile des Prozesses (Daten und Code) im **physikalischen Speicher** sind, die gerade *benötigt* werden.
- Der **Speicherbedarf** eines Prozesses sollte *größer* als der *physikalisch vorhandene Hauptspeicher* sein können.

Ein **Seitenrahmen** versteht man einen zusammenhängenden Block von Speicherzellen des *physikalischen* Speichers.

Eine **Seite** ist ein Block von Speicherzellen des *virtuellen* Speichers. Blockgröße einer Seite ist Größe eines Seitenrahmens.

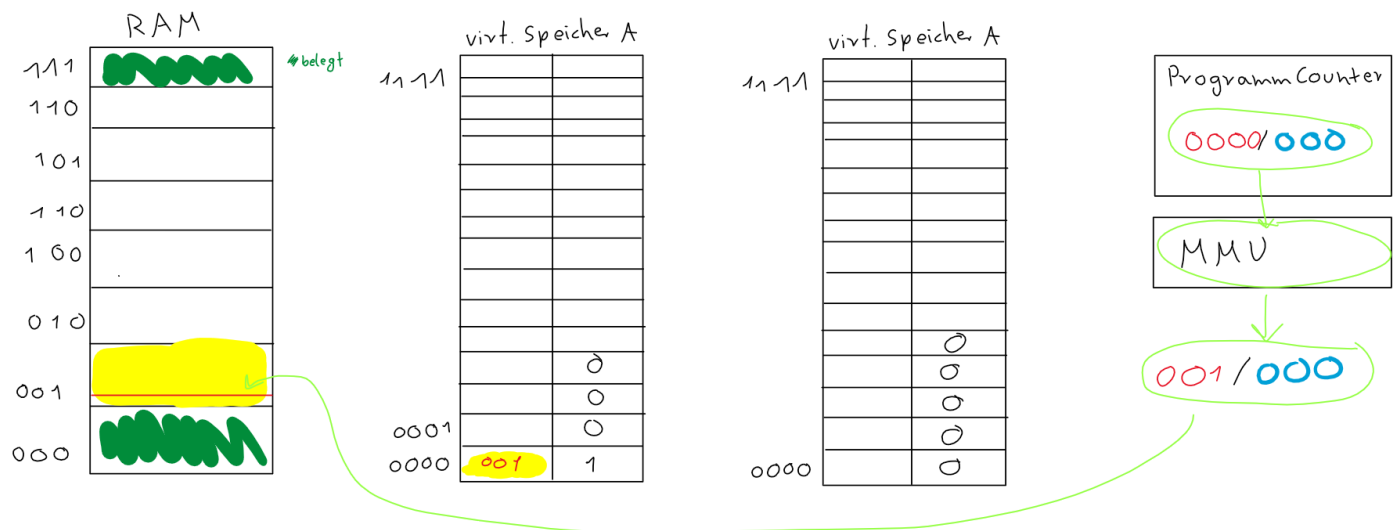
- **alle Seiten eines Systems haben stets die gleiche Größe**

die Aufgabe der MMU besteht in der **Umrechnung von virtuellen in physikalische Adressen**

---

## Seitentabellen

- werden benötigt, damit **MMU die Umrechnung einer virtuellen Adresse** in eine physikalische Adresse vornehmen kann
- Für jeden vom Betriebssystem zu verwaltenden **virtuellen Adressraum** gibt es jeweils **eine zugehörige Seitentabelle**
- **Jeder Prozess besitzt seine eigene Seitentabelle!**



1. **MMU** prüft in der Seitentabelle des aktuellen Prozesses den Eintrag an entsprechenden Stelle aus der virtuellen Adresse.
  2. Ist **P/A-Bit = false (0)**, kommt es zu **Page-Fault (Seitenfehler)**
    - Liegt daran, dass noch keine physische Seite eingelagert wurde
    - Die Seite muss vom **DMA-Controller** in den RAM-Speicher geladen werden
    - Ist der RAM bereits voll, kommen **Seitenersetzungsverfahren** ins Spiel.
  3. Ist das dann **P/A-Bit = true (1)**, kann die Rahmennummer aus der Tabellenzeile entnommen werden.
  4. **MMU** guckt sich bei 0000 das *present absent bit* an, weil es eine 1 ist, kann es umgerechnet werden
  5. wird im **RAM** bei 001 abgespeichert
  6. wenn die fetch Phase beendet ist, wird der **PC** um 1 hochgesetzt und auf 0000001 gesetzt
- so viel wie in Seitenrahmen reinpasst muss der DMA-Controller kopieren
  - bei Pagefault wird der Prozess auf **blocket** gesetzt, da er so nicht weiterarbeiten kann
  - wenn alle Daten im Ram sind wird ein **Interrupt** eingestellt und es geht auf **Ready**
  - Prozess kann mit den Daten im Ram arbeiten und wenn Scheduler den Prozess wieder nimmt geht es wieder in **Running**

## KONTEXTWECHSEL

1. Wert des PC wird überschrieben mit Wert vom virtuellen Speicher B
  2. In Seitentabelle wird gesehen das absent 0 ist -> **Seitenfehler** -> muss in Ram eingelagert werden (egal welche Stelle)
  3. Present wird zu Absend
- in 111 wird der Wert des PC gespeichert

DLL Dateien sorgen dafür, dass weniger Speicher gebraucht wird

---

Revision #7

Created 6 September 2022 11:39:06 by Merith Holtmann

Updated 25 September 2022 07:09:56 by Merith Holtmann