

# Speicherverwaltung

Die Speicherverwaltung ist Teil des Betriebssystems und erledigt alle erforderlichen Arbeiten zur Verwaltung des physischen und des virtuellen Speichers eines Computers.

## Grundlagen

### Virtuelle Speicherverwaltung

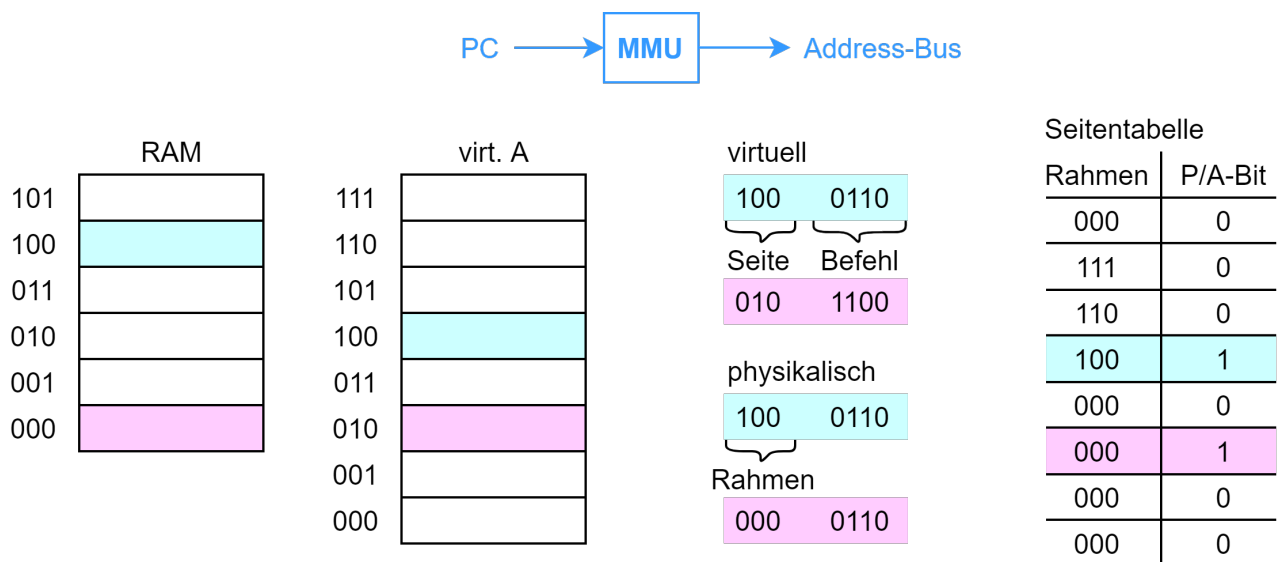
Prozesse werden in einem virtuellen Speicher gehalten, um im RAM nur tatsächlich benötigte Programmteile zu speichern. Die MMU (Memory Management Unit) rechnet die virtuellen Speicheradressen in die physischen Adressen um.

Ein **Seitenrahmen** ist ein zusammenhängender Block von Speicherzellen des virtuellen Speichers (RAM).

Eine **Seite** ist ein zusammenhängender Block von Speicherzellen des physischen Speichers (RAM). Die Größe einer Seite entspricht der Größe eines Seitenrahmens.

### Seitentabellen

Zur Umrechnung der virtuellen Speicheradresse in die physische werden Seitentabellen benötigt.



Die **MMU** prüft in der Seitentabelle des aktuellen Prozesses den Eintrag an der entsprechenden Stelle aus der virtuellen Adresse.

Ist das **P/A-Bit = true (1)**, kann die Rahmennummer aus der Tabellenzeile entnommen werden.

Ist das **P/A-Bit = false (0)**, kommt es zu einem **Page-Fault (Seitenfehler)**. Die Seite muss zunächst vom DMA-Controller in den RAM-Speicher geladen werden. Ist der RAM bereits voll, kommen **Seitenersetzungsverfahren** ins Spiel.

### Swapping

Kompletter Prozess wird ausgetauscht (Basis- & Limitregister, ...)

Wenn der RAM voll ist wird ein gesamter Prozess in den HD Speicher verschoben, sodass Platz im RAM für einen neuen Prozess geschaffen wird.

Ein kompletter Prozess ist entweder ganz eingelagert (RAM) oder ganz ausgelagert (HD) (Altes Verfahren)

## Seitenersetzungsverfahren → Paging

Unter Paging versteht man das Ein- bzw. Auslagern von Teilen eines Prozesses.

### Optimale Seitenersetzung

→ Welche Seite wird in der Zukunft am längsten nicht benötigt?

Die optimale Seitenersetzung kann nicht programmiert werden.

### No Recently Used

→ Welche Seite wurde in der Vergangenheit am längsten nicht genutzt?

Es wird davon ausgegangen, dass diese Seite auch in Zukunft lange nicht benötigt wird.

Es wird ein Referenziert-Bit und ein Modifiziert-Bit eingeführt.

Das **R-Bit** wird **true (1)** gesetzt, wenn die Seite verwendet wurde. Es wird in regelmäßigen Abschnitten auf **false (0)** gesetzt.

Das **M-Bit** wird **true (1)** gesetzt, wenn die Seite verändert wurde.

In folgender Reihenfolge werden Seiten gesucht und ersetzt. Wenn:

1. **R = 0 && M = 0** (Seite nicht verwendet)
2. **R = 0 && M = 1** (Seite modifiziert, lange nicht verwendet)
3. **R = 1 && M = 0** (Seite verwendet, nicht modifiziert)
4. **R = 1 && M = 1** (Seite verwendet und modifiziert)

### First-In-First-Out

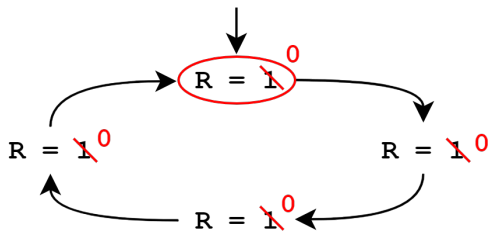
→ Die Seiten, die zuerst im RAM gespeichert wurden, werden als erstes überschrieben.

## Working Set

→ Die wichtigsten Seiten, das "Working-Set" eines Prozesses werden eingelagert.

## Second Chance

→ In einem kreisförmigen Durchlauf wird die Schlange der zuerst eingelagerten Seiten durchlaufen. Ist das `R-Bit = true (1)` wird es auf `false (0)` gesetzt und die Seite bekommt eine zweite Chance.



Sind alle `R-Bit = 1` wird der erste Prozess ersetzt (die

Seite).

---

## Shared Memory

Zur Interkommunikation können gemeinsame Speicherbereiche sinnvoll sein. Dazu wird der Seitenrahmen des RAMs in allen Seitentabellen aller beteiligten Prozesse hinterlegt.

---

Revision #13

Created 12 August 2022 08:27:41 by Martin Tienken

Updated 27 September 2022 07:48:13 by Lukas Berding