

Swapping und Paging

Unter **Swapping** versteht man das Aus- bzw. Einlagern eines kompletten Prozesses.

Swapping kommt nur bei Betriebssystemen zum Einsatz, die **keine virtuelle Speicherverwaltung** unterstützen!

eine Alternative zur virtuellen Speicherverwaltung mit Hilfe der MMU, ist das **Paging**, welches flexibler als **Swapping** agiert.

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

- Eine Seite enthält einen **bestimmten Teil eines Prozesses**.
- Die *Gesamtheit aller Seiten* eines Prozesses repräsentiert somit den *kompletten Prozess*.
- Eine (virtuelle) Seite kann **einerseits** in einem (physikalischen) Seitenrahmen eingelagert sein.
 - befindet sich die Seite im RAM und der zugehörige Prozess kann auf die Befehle Daten innerhalb der Seite zugreifen.
 - Zum Einsatz kommt hier die *Adressumrechnung mit Hilfe von MMU und Seitentabelle*.
- **Andererseits** kann eine (virtuelle) Seite auf einen Hintergrundspeicher (wie beispielsweise die *Festplatte*) ausgelagert sein.
 - so steht die Seite nicht im RAM zur Verfügung, Zugriff des Prozesses auf Befehle schlägt fehl -> **Page fault**
 - ausgelagerte Seite muss vom *Hintergrundspeicher in Seitenrahmen* eingelagert werden.
 - Anschließend kann die *Adressumrechnung mit Hilfe von MMU und Seitentabelle*.

Beim Paging werden virtuelle Seiten in (physikalische) Seitenrahmen eingelagert, oder andersrum

Page Fault

Page fault tritt auf, falls die MMU bei der Umrechnung einer virtuellen in eine physische Adresse feststellt, dass die benötigte (virtuelle) Seite nicht in einem (physischen) Seitenrahmen eingelagert ist.

Seitenersetzung

Tritt **Seitenfehler** auf, lagert BS **virtuelle Seite** aus **Hintergrundspeicher** in *freien* **Seitenrahmen** des **physikalischen Speichers**

- Entscheidend ist bei der Einlagerung, ob es einen *freien Seitenrahmen* gibt
- Falls **ja**: dieser für die *Einlagerung der benötigten Seite* genutzt werden
- Falls **nein**: **Seitenersetzung** entscheidet, welche *momentan eingelagerte virtuelle Seite* in *HS* verschoben wird

Was bei der Seitenersetzung passiert

1. Die MMU stellt fest, dass **virtuelle Seite B** nicht in **Seitenrahmen** eingelagert ist und löst einen Seitenfehler aus.
2. Es wird festgestellt, dass *kein freier Seitenrahmen* verfügbar ist, das **Seitenersetzungsverfahren** wird deshalb *gestartet*.
3. Die zu **ersetzende Seite E** wird bestimmt, sie ist derzeit in **Seitenrahmen X** eingelagert.
4. Die zu **ersetzende Seite E** wird in den Hintergrundspeicher geschrieben, damit ist ihr Inhalt gesichert
5. Die **benötigte Seite B** wird eingelagert, ihr Inhalt wird in den **Seitenrahmen X** geschrieben

Schreiben der zu ersetzenden Seite E in den Hintergrundspeicher kostet viel Zeit! Das ist schlecht für das Gesamtsystem.

→ sinnvoll, eine zu *ersetzende Seite* nur **dann** in den Hintergrundspeicher zu **kopieren**, wenn dies auch **tatsächlich notwendig** ist.

Das Modifiziert-Bit

→ BS kann mit dem **M-Bit** feststellen, ob die Inhalte einer eingelagerten Seite modifiziert und zum auslagern bereit sind

Seitenrahmen-Nr.	Present-/ Absent-Bit	M-Bit
0 1111 0000 0011	1	0
0 1100 0111 0101	0	0
1 0101 0011 1110	0	0
0 1001 1111 1001	1	1
...

- Das M-Bit ist *gesetzt*, also **1**:

Der Inhalt der zugehörigen Seite wurde **modifiziert**.

- Das M-Bit ist *nicht gesetzt*, also **0**:

Der Inhalt der zugehörigen Seite wurde **nicht modifiziert**.

Seitenersetzungsverfahren → Paging

Die optimale Seitenersetzung kann nicht programmiert werden.

No Recently Used → Welche Seite wurde in der Vergangenheit am *längsten nicht genutzt*?

Für jede einzelne Seite wird mit Hilfe eines einzelnen Bits festgehalten, ob die *betreffende Seite* referenziert wurde.

Seitenrahmen-Nr.	Present-/ Absent-Bit	R-Bit	M-Bit
0 1111 0000 0011	1	1	0
0 1100 0111 0101	0	0	0
1 0101 0011 1110	0	0	0
0 1001 1111 1001	1	1	1
...

- Das R-Bit ist gesetzt, also **1**:

Inhalt der Seite wurde *referenziert*, es wird **verwendet**.

- Das R-Bit ist nicht gesetzt, also **0**:

Inhalt der Seite wurde *nicht referenziert*, es wird **nicht verwendet**.

First-In-First-Out

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

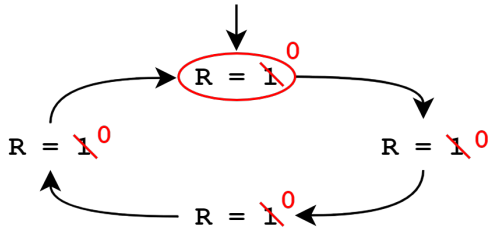
In der Praxis hat dieses Verfahren keine Bedeutung, da die ausgelagerte Seite oft benötigt wird → hohen Zahl an **Seitenfehlern**

Working Set

bei vielen nacheinander auf der CPU ausgeführten Befehlen werden nur wenig verschiedene virtuelle Seiten angesprochen

→ versucht, alle zum Working Set eines *Prozesses* gehörenden Seiten **ständig im Hauptspeicher** zu halten

Second Chance



er prüft, ob diese *Seite* in der **letzten Zeit auch**

angesprochen wurde.

- Falls **nein**: sie wird ersetzt
- Falls **ja**: wird mit der am **zweitlängsten eingelagerten Seite** fortgefahren.
- Auch hier wird zunächst geprüft, ob Seite in der letzten Zeit angesprochen wurde.
- Usw.

Revision #2

Created 13 September 2022 06:37:02 by Merith Holtmann

Updated 24 September 2022 16:08:52 by Merith Holtmann