

## Beispiellösung - Übungsblatt 8

Erstellt von Daniel Bub

### Aufgabe 1.

Ist das Konzept der Adressräume mit der Interprozesskommunikation (IPC) vereinbar? Begründen Sie Ihre Antwort.

### Lösung 1.

Das Konzept der Adressräume ist mit der Interprozesskommunikation **nicht** vereinbar. Jeder Prozess besitzt seinen eigenen Adressraum. Ein Prozess A darf nicht in den Adressraum von Prozess B schreiben oder von diesem lesen. Eine Kommunikation zwischen Prozessen kann somit nur mit Hilfe von Pipe's, FIFO's, Message-Queues, etc. realisiert werden.

### Aufgabe 2.

Ein Computer verfügt über einen 64-Bit-CPU und 64 GByte RAM. Wie groß ist der virtuelle und wie groß der physikalisch adressierbare Speicher?

### Lösung 2.

- virtuell:  $2^{64}$  Byte =  $1.8 \cdot 10^{19}$  Byte, d.h. jedes Byte wird durch einen eigenen Bit-Wert abgebildet.
- physikalisch: genau die zur Verfügung stehenden 64 GByte (=  $2^{36}$  Byte)

### Aufgabe 3.

Folgende Pages wurden im System geladen. Spielen Sie das Szenario der Seitenersetzung jeweils mit den folgenden Algorithmen durch: NRU, FIFO, Second-Chance, Clock, LRU. Ersetzen Sie jeweils drei Seiten.

Page-Number $\hat{=}$ Reihenfolge	R, M	Zeit seit letztem Zugriff (ms)
1	1, 0	10
2	0, 1	15
3	0, 0	20
4	1, 1	10
5	1, 1	5
6	1, 1	10

### Lösung 3.

- **NRU:** Page 3  $\Rightarrow$  Page 2  $\Rightarrow$  Page 1
- **FIFO:** Page 1  $\Rightarrow$  Page 2  $\Rightarrow$  Page 3
- **Second-Chance:** Page 2  $\Rightarrow$  Page 3  $\Rightarrow$  Page 1
- **Clock:** Page 2  $\Rightarrow$  Page 3  $\Rightarrow$  Page 1
- **LRU:** Page 3  $\Rightarrow$  Page 2  $\Rightarrow$  (Page 1 oder Page 4 oder Page 6)

**Aufgabe 4.**

Beschreiben Sie die Funktionsweise des WSClock-Algorithmus.

**Lösung 4.**

Der WSClock-Algorithmus (WS steht für WorkingSpace) speichert alle Informationen über Pages, die von einem Prozess genutzt werden. Findet ein Kontextwechsel statt, und der Prozess wird erneut zur Ausführung aufgerufen, werden dementsprechend die gespeicherten Pages gleich mit geladen. Dies hat den Vorteil, dass nicht jede Page nach einem expliziten Page-Fault geladen werden muss. Anschließend wird mit dem ‘normalen’ Clock-Algorithmus weiter verfahren.

**Aufgabe 5.**

Beim Clock-Algorithmus (und auch bei anderen Algorithmen, etwa Second Chance) kann es zu einem unendlichen Loop während der Suche nach einer zu ersetzenden Seiten kommen – weshalb?

Wie ließe sich ein solcher Loop im Falle des Clock-Algorithmus verhindern?

**Lösung 5.**

Stellen Sie sich folgendes Szenario vor:

Der Clock-Algorithmus wählt eine Page zum “Dereferenzieren” aus. Kurz darauf wird genau diese Page wieder von einem Prozess benötigt und referenziert. Dieser Ablauf wiederholt sich nun zyklisch, so dass  $\neg R$  immer wieder auf  $R$  gesetzt wird.

Eine möglich Lösung für dieses Problem: Nach einem kompletten Umlauf ohne Ersetzung, wählt man die erste Page in diesem Zyklus als “Victim”-Page aus und ersetzt diese.