

Aufgabe 5.9 Virtueller Speicher

a) Mit 30-Bit lassen sich $2^{30} = 1073741824 \text{Byte} = 1 \text{GiByte}$ adressieren.

b) Für Jede Seite wird ein Eintrag von 4 Byte in der Seitentabelle benötigt. Eine Seite hat eine Größe von 4 KiByte. Die Seitentabellengröße S ergibt sich also wie folgt:

$$S = 4 \text{Byte} * (1 \text{GiByte} / (4 \text{KiByte})) = 1 \text{MiByte}$$

c) Um zu ermitteln, wie viele Prozesse gleichzeitig im Speicher gehalten und somit unter den genannten Bedingungen ausgeführt werden können teilt man den zu Verfügung stehenden Speicherplatz durch den Speicherbedarf eines Prozesses.

$$(2^{36} \text{Byte}) / (2^{30} \text{Byte}) = 64$$

Es können also 64 Prozesse unter den genannten Bedingungen gleichzeitig ausgeführt werden.

d) Im jetzt zu betrachtenden Fall erhöht sich der Speicherbedarf eines Prozesses um den für die Seitentabelle benötigten Speicher. Es ergibt sich somit ein Speicherplatzbedarf pro Prozess von $(2^{30} \text{Byte} + 2^{20} \text{Byte})$. Zusätzlich steht aufgrund des Platzbedarfs des Betriebssystems 1 MiByte weniger Speicher für die Prozesse zur Verfügung.

$$((2^{36} \text{Byte} - 2^{20} \text{Byte}) / (2^{30} \text{Byte} + 2^{20} \text{Byte})) \approx 63,9366$$

Es können also maximal 63 Prozesse mit vollem Speicherbedarf gleichzeitig ausgeführt werden.

e) Wird die Kachelgröße auf 1 KiByte reduziert, so erhöht sich der Platzbedarf der Seitentabelle um den Faktor 4. Es werden jetzt viermal so viele Einträge pro Prozess benötigt, da sich die Zahl der Kacheln (und somit auch der Seiten pro Prozess) vervierfacht hat. Durch die geringere Kachelgröße nimmt die innere Fragmentierung ab. Als Nachteil treten mehr Seitenfehler auf, da bei einem Seitenfehler weniger Daten in den Hauptspeicher geladen werden. Dieser Effekt relativiert sich etwas durch die Tatsache, dass die bei einem Seitenfehler zu bewegendende Datenmenge geringer ist, jedoch ist der Overhead insgesamt größer.

f) Die Zahl der zu verwaltenen Kacheln ergibt sich durch $\text{Speicher} / \text{Kachelgröße}$.
Für 4 KiByte große Kacheln ergeben sich also:

$$(2^{36} \text{Byte} / 4096 \text{Byte}) \text{Kacheln} = 16777216 \text{Kacheln} = 2^{24} \text{Kacheln} (= 16 \text{MiKacheln})$$

Für 1 KiByte große Kacheln ergeben sich:

$$(2^{36} \text{Byte} / 1024 \text{Byte}) \text{Kacheln} = 2^{26} \text{Kacheln} (= 64 \text{MiKacheln})$$

Aufgabe 5.10 Seitentausch

- a) In jeder Seite sind 16 Zeilen der Matrix abgelegt. Die Matrix benötigt also 4 Seiten. Während der Laufzeit werden zunächst beim jeweils ersten Zugriff die ersten beiden Seiten in die zwei freien Kacheln eingelagert. Beim Zugriffsversuch auf die Daten der dritten Seite wird die Kachel mit dem Programmcode ausgelagert und durch die dritte Seite mit Daten der Matrix ersetzt. Die Seite mit dem Code muss natürlich direkt wieder in den Hauptspeicher geladen werden. Die letzte Seite verursacht keine Probleme, da nur eine nicht mehr benötigte Seite ausgelagert wird. Insgesamt ergeben sich 5 Seitenfehler.
- b) Da nur drei Kacheln zu Verfügung stehen ist der Ausgangspunkt bei Durchlaufen einer Spalte immer isomorph. Es sind jeweils der Code und zwei nicht benötigte Seite in den Kacheln eingelagert. Bei jeder Spalte treten wie bei a) 5 Seitenfehler auf. Insgesamt treten also $64 * 5 = 320$ Seitenfehler auf.

c) „FIFO“

A	B	Kachel 1	Kachel 2	Kachel 3	Bemerkungen
running	ready	Code	-	-	Initialzustand
PF[0..15]	⇒ running	Code	A[0..15]	-	Page-Fault bei A.m[0..15]
⇒ running	PF[0..15]	Code	A[0..15]	B[0..15]	Page-Fault bei B.m[0..15]
PF[16..31]	⇒ running	A[16..31]	A[0..15]	B[0..15]	Page-Fault bei A.m[16..31]
⇒ running	PF[Code]	A[16..31]	Code	B[0..15]	Page-Fault bei B.m[Code]
PF[32..47]	⇒ running	A[16..31]	Code	A[32..47]	Page-Fault bei A.m[32..47]
⇒ running	PF[16..31]	B[16..31]	Code	A[32..47]	Page-Fault bei B.m[16..31]
PF[48..61]	⇒ running	B[16..31]	A[48..63]	A[32..47]	Page-Fault bei A.m[48..61]
⇒ terminated	PF[Code]	B[16..31]	A[48..63]	Code	Page-Fault bei B.m[Code]
terminated	⇒ running	B[16..31]	A[48..63]	Code	
terminated	PF[32..47]	B[32..47]	A[48..63]	Code	Page-Fault bei B.m[32..47]
terminated	⇒ running	B[32..47]	A[48..63]	Code	
terminated	PF[48..63]	B[32..47]	B[48..63]	Code	Page-Fault bei B.m[48..63]
terminated	⇒ terminated	B[32..47]	B[48..63]	Code	

d) „Optimal Replacement“

A	B	Kachel 1	Kachel 2	Kachel 3	Bemerkungen
running	ready	Code	-	-	Initialzustand
PF[0..15]	⇒ running	Code	A[0..15]	-	Page-Fault bei A.m[0..15]
⇒ running	PF[0..15]	Code	A[0..15]	B[0..15]	Page-Fault bei B.m[0..15]
PF[16..31]	⇒ running	Code	A[16..31]	B[0..15]	Page-Fault bei A.m[16..31]
⇒ running	PF[16..31]	Code	A[16..31]	B[16..31]	Page-Fault bei B.m[16..31]
PF[32..47]	⇒ running	Code	A[32..47]	B[16..31]	Page-Fault bei A.m[32..47]
⇒ running	PF[32..47]	Code	A[32..47]	B[32..47]	Page-Fault bei B.m[32..47]
PF[48..63]	⇒ running	Code	A[48..63]	B[32..47]	Page-Fault bei A.m[48..63]
⇒ terminated	PF[48..63]	Code	A[48..63]	B[48..63]	Page-Fault bei B.m[48..63]
terminated	⇒ terminated	Code	A[48..63]	B[48..63]	