

Musterlösungen zu den Tutoraufgaben auf  
Blatt 9  
der Übungen zur Vorlesung  
“Grundlagen Betriebssysteme und Systemsoftware

---

G.Groh, 12.01.2009

## Aufgabe 4.1, 4.2, 4.3 ,4.4 Lösungsvorschlag

- 4.1** Es sind nur Seitengrößen sinnvoll, die Teiler der Größe des vorhandenen physikalischen Speichers darstellen. Ansonsten würden nicht alle Teile abgedeckt oder es käme zu Überlagerungen. Also 1, 2, 4, 8, 16, 32, 64, 128, 256.
- 4.2** Bei 12bit-breiter virtueller Adresse werden vom System  $2^{12} = 4096$  Byte zur Verfügung gestellt. Für das Ein- und Auslagern ist zusätzlich ein Puffer in Größe einer Seite notwendig, da sonst das Austauschen der Daten nicht möglich ist. Das System muss also insgesamt mindestens über  $4096 + 32 = 4128$  Byte Speicher verfügen. Dabei werden dann keine Daten redundant gehalten, sprich Daten befinden sich dann entweder im Hintergrundspeicher oder im Hauptspeicher.
- Üblicherweise befindet sich aber der komplette virtuelle Speicher schon auf dem Hintergrundspeicher (also 4096 Byte). Es werden dann die benötigten Teile zusätzlich in den Hauptspeicher (256 Byte) geladen. Dann wird kein weiterer Puffer benötigt. Insgesamt aber dann  $4096 + 256 = 4352$  Byte Speicher.
- 4.3** Bei einer Seitengröße von 32 Byte sind für die Adressierung innerhalb der Seite (Offset) 5 Bit notwendig. Von den 12 Bit der virtuellen Adresse bleiben also 7 Bit für die Seitennummer. Es können also  $2^7 = 128$  Seiten verwaltet werden.

## Aufgabe 4.5, 4.6 Lösungsvorschlag

- 4.4** Wählt man die ersten Bits, so werden innerhalb eines Programms (meist Zugriffe auf nicht weit auseinander liegende Bereiche) die Seiten immer in die gleiche Kachel eingelagert, auch wenn andere gar nicht benutzt werden. Wählt man die letzten Bits, so werden direkt zusammenliegende Bereiche gut auf Kacheln aufgeteilt. Heutige Programme liegen aber der oft dynamisch eingebundenen Bibliotheken wegen, nicht immer direkt beieinander. Wählt man die Bits aus der Mitte, kann ein Kompromiss aus den anderen beiden Verfahren "für mittlere Entfernungen" erreicht werden.
- 4.5** Vorteil dieses Verfahrens ist, dass sich die Abbildung sehr einfach, direkt und damit schnell berechnen lässt. Es ist speziell für Hardwarelösungen geeignet, da die Abbildung per weniger, einfacher Logikschaltungen möglich ist. Der hohen Geschwindigkeit wegen ist es damit für Caches geeignet. Nachteil ist, dass bei gewissen Konstellationen immer wieder die gleichen Kacheln ersetzt werden.

## Aufgabe 5.1, 5.2 Lösungsvorschlag

### 5.1

Seitengröße	Offsetbreite	Länge der virtuellen Adresse		
		16Bit	32Bit	64Bit
4KB = 4096 Byte = $2^{12}$ Byte	12Bit	$2^4=16$	$2^{20} = 1.048.576$	$2^{52} = 4.503.599.627.370.496$
8KB = 8192 Byte = $2^{13}$ Byte	13Bit	$2^3=8$	$2^{19} = 524.288$	$2^{51} = 2.251.799.813.685.248$
16KB = 16384 Byte = $2^{14}$ Byte	14Bit	$2^2=4$	$2^{18} = 262.144$	$2^{50} = 1.125.899.906.842.624$

### 5.2

Seitengröße	Offsetbreite phys. (wp)	Offsetbreite virt. (wv)	Seitennummer (s)	Kachelnummer (k)
1KB = 1024 Byte = $2^{10}$ Byte	10Bit	10Bit	22Bit	14Bit
2KB = 2048 Byte = $2^{11}$ Byte	11Bit	11Bit	21Bit	13Bit
4KB = 4096 Byte = $2^{12}$ Byte	12Bit	12Bit	20Bit	12Bit
8KB = 8192 Byte = $2^{13}$ Byte	13Bit	13Bit	19Bit	11Bit