

Nr. 6.1

Aufgabe 1

Es fällt auf: nach Memorymapping des Files wird die virtuelle Größe auf 1GB gesetzt, resident size aber erstmal nur 800MB und Dirty Size 0KB. Nachdem alles geschrieben ist, sind alle 3 Größen auf 1GB. Head und Tail: keine Auffälligkeiten, da wir nur komplette Zeilen schreiben.

Aufgabe 2

1. 51 Bit
2. *Caching* | *R-Bit* (setzt MMU bei jedem Zugriff) | *M-Bit* (setzt MMU bei Schreibzugriff) | *Zugriffsrechte* | *Present-Bit* (überhaupt physikalisch gelagert?) | *Seiten- / Kachelnummer*
3. $2^{51-12} = 2^{39}$ Seiten
4. Adressumsetzung:
 - a. Programm-Adr besteht aus Seitennummer + relative Adresse (Offset) innerhalb Seite
 - b. Die Seitennr (=Index in Seitentabelle) wird in der Seitentabelle nachgeschaut und zur Rahmennummer, wobei die Größen dieser Adressen auch unterschiedlich sein können
 - c. Der Offset bleibt Offset
 - d. Falls Seite nicht im Speicher liegt, ist Present-Bit gelöscht
→ Page Fault
5. 3 Stufig → 1. Stufe: Hauptseitentabelle mit 8192 Einträge
2. Stufe: ≤ 8192 Seitentabellen mit je 8192 Einträgen
3. Stufe: ≤ 8192 Seitentabellen mit je 8192 Einträgen
6. Seitentabellen haben je 2^{13} Einträge (= 8192)

Aufgabe 3

1. 20 Bit
2. So groß, wie physikalisch Speicher vorhanden ist
3. 20 Bit virtuelle Adressen
4 kByte Seitengröße (== 12 Bit = Offset)
4 Byte je Tabelleneintrag
→ $2^{20-12} = 2^8$ Einträge / Seiten
→ $2^8 * 4 \text{ Byte} = 1024 \text{ Byte} = 1 \text{ KB}$ Tabellengröße
4. %X 2B3
5. Schreibzugriff auf virtuelle Adr:
 - a. Page fault, weil sie nicht drin ist
 - b. Page fault, weil sie nicht drin ist
6. Dann wird sie rausgeschmissen um Platz zu schaffen. Present und Modify Bit werden auf 0 gesetzt.