

PUNKTEVERTEILUNG:

1	2	3	Σ

Aufgabe (1)

- (a) $32 - (9 + 11) = 12$ also gibt es 12 Bit für den Offset, daraus folgt dass eine Seite 2^{12} Bytes groß ist, was dann 4 Kilybyte entspricht.
- (b) $2^9 \cdot 2^{11} = 2^{9+11} = 2^{20} = 1048576$
- (c) $4 \cdot 2^{20} = 2^2 \cdot 2^{20} = 2^{2+20} = 2^{22} = 4194304$ Byte, also 4096 Kilobyte, also 4 Megabyte.

Aufgabe (2)

- (a) • first fit

Sucht vom Anfang der Liste nach der ersten Partition die groß genug ist um die Anfrage zu erfüllen. Einfache Speicherverwaltung, da Löcher bei der alloka-tion nur geteilt werden müssen. Suche ist simpel, Fragmentierung tritt aber dennoch auf.

100kB	400kB	250kB	200kB	50kB
70kB 10kB	280kB 260kB 160kB	0kB		

- next fit

Variation von first fit bei der noch vom Anfang gesucht wird sondern von der letzten gefundenen Lücke. Allerdings in Simulationen eine schlechtere Leistung als first fit.

100kB	400kB	250kB	200kB	50kB
70kB 10kB	280kB 260kB 160kB	0kB		

- best fit

Sucht die kleinste passende Lücke. Suche muss alle Lücken durchsuchen um die kleinste passende Lücke zu finden. Fragmentiert erstaunlicherweise stärker als first fit, da die dabei entstehenden Lücken klein aber zahlreich sind und kaum ein Prozess diese mehr ausnutzen will.

100kB	400kB	250kB	200kB	50kB
40kB	150kB	150kB	80kB	20kB 0kB

- worst fit

Inverses von best fit. Sucht die größte Lücke, muss also diese auch erstmal finden. Verbraucht viel Speicher.

100kB	400kB	250kB	200kB	50kB
	370kB 310kB 190kB	230kB 130kB		

Allokation von 250kB klappt nicht, nicht genügend zusammenhängender Speicher.

Aufgabe (3)

$1\mu s$ für 1 Instruktion mit 2 Speicherzugriffen, also $10\mu s$ für 10 Instruktionen mit 20 Speicherzugriffen, davon 2% Seitentabellenzugriffe/Page Faults. Im (optimistischen) Falle von Seitentabellen also $1\mu s$ Overhead. Daraus folgt dass 10 Instruktionen mindestens $11\mu s$ dauern. Wenn das alles wäre, könnte man davon ausgehen dass eine Instruktion im Mittel $1.1\mu s$ dauert.

Nun zu den Pagefaults: ein Pagefault ist $200\mu s$ Overhead, also dürften 1000 Instruktionen $1200\mu s$ dauern. Davon $200\mu s$ für den Pagefault, die restlichen $1000\mu s$ können dann auf Speicherzugriffe mit TLB oder Seitentabellen entfallen. Da diese $1.1\mu s$ dauern können davon $1000\mu s \div 1.1\mu s = 909$ passieren, also jede 910-te Instruktion dürfte zu einem Page Fault führen. Da jede Instruktion 2 Speicherzugriffe macht, wäre somit die "Anzahl Page-Faults pro Anzahl Gesamtzugriffe auf Speicher" 1 Page Fault pro 1820 Speicherzugriffe um $1.2\mu s$ mittlere Ausführungszeit zu erreichen.