

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

G.Groh, 11.01.2009

Aufgabe 1.1 Lösungsvorschlag

Frage: Erklären Sie die Strategien knapp

- **first fit** - der erste Speicherbereich in der Liste der freien Speicherbereiche, der passt, wird verwendet.
- **next fit** - ähnlich wie first fit. Nur wird hier an der Stelle fortgesetzt, wo bei der letzten Suche aufgehört wurde.
- **best fit** - der kleinste freie Speicherbereich, der noch für die Anfrage groß genug ist, wird verwendet.
- **worst fit** - der größte freie Speicherbereich wird verwendet.

Aufgabe 1.2 Lösungsvorschlag, first fit

Anfrage	Liste freier Speicherbereiche				
	100kB	400kB	250kB	200kB	50kB
30kB	70kB	400kB	250kB	200kB	50kB
60kB	10kB	400kB	250kB	200kB	50kB
120kB	10kB	280kB	250kB	200kB	50kB
20kB	10kB	260kB	250kB	200kB	50kB
100kB	10kB	160kB	250kB	200kB	50kB
250kB	10kB	160kB	0kB	200kB	50kB

Aufgabe 1.2 Lösungsvorschlag, next fit

Anfrage	Liste freier Speicherbereiche				
	100kB	400kB	250kB	200kB	50kB
30kB	70kB	400kB	250kB	200kB	50kB
60kB	10kB	400kB	250kB	200kB	50kB
120kB	10kB	280kB	250kB	200kB	50kB
20kB	10kB	260kB	250kB	200kB	50kB
100kB	10kB	160kB	250kB	200kB	50kB
250kB	10kB	160kB	0kB	200kB	50kB

Aufgabe 1.2 Lösungsvorschlag, best fit

Anfrage	Liste freier Speicherbereiche				
	100kB	400kB	250kB	200kB	50kB
30kB	100kB	400kB	250kB	200kB	20kB
60kB	40kB	400kB	250kB	200kB	20kB
120kB	40kB	400kB	250kB	80kB	20kB
20kB	40kB	400kB	250kB	80kB	0kB
100kB	40kB	400kB	150kB	80kB	0kB
250kB	40kB	150kB	150kB	80kB	0kB

Aufgabe 1.2 Lösungsvorschlag, worst fit

Anfrage	Liste freier Speicherbereiche				
	100kB	400kB	250kB	200kB	50kB
30kB	100kB	370kB	250kB	200kB	50kB
60kB	100kB	310kB	250kB	200kB	50kB
120kB	100kB	190kB	250kB	200kB	50kB
20kB	100kB	190kB	230kB	200kB	50kB
100kB	100kB	190kB	130kB	200kB	50kB
250kB	Kein geeigneter Speicherbereich vorhanden				

Aufgabe 2.1 Lösungsvorschlag

Frage: Wann kann es zu einem sogenannten Seitenfehler kommen?
Was passiert dabei?

Antwort:

Zu Seitenfehler kommt es, wenn eine Seite, auf die zugegriffen wird, nicht im Speicher vorhanden ist, sondern im Hintergrundspeicher (da der virtuelle Adressraum größer sein kann als der physische). Dann wird ein Interrupt ausgelöst, der dem Betriebssystem ermöglicht, die betreffende Seite in den Speicher zu holen. Dabei wird entweder ein freier Rahmen im Speicher dafür verwendet, oder es wird nach einem bestimmten Algorithmus einer der bereits im Speicher vorhandenen Seiten verdrängt. Die verdrängte Seite kommt dann wieder in den Hintergrundspeicher.

Frage: Erklären Sie folgende Begriffe knapp

- **FIFO** - First In First Out. Die Seite, die am längsten im Speicher ist, wird ersetzt. So werden im Prinzip bei Seitenfehlern alle Kacheln der Reihe nach verwendet.
- **LIFO** - Last In First Out. Die Seite, die zu letzt in den Speicher geladen wurde, wird ersetzt. Im Grunde werden also erst die leeren Kacheln befüllt, und bei Seitenfehlern nur die letzte Kachel verwendet.
- **LRU** - Least Recently Used. Die Seite, auf die am längsten nicht zugegriffen wurde, wird ersetzt.
- **Second Chance** - ist ein ähnlicher Algorithmus wie FIFO. Vermeidet jedoch, das ältere, häufig genutzte Seiten ausgelagert werden. Seiten, deren Zugriffsbit (r) gesetzt ist, werden wieder an das Ende der FIFO Liste gestellt.
- **Working Set** - beschreibt eine Menge von Seiten, die ein Prozess momentan (in den letzten k Zugriffen) verwendet. Diese Seiten werden nicht ausgelagert. Wird mehr Speicher benötigt wird k verkleinert.

Aufgabe 2.3 Lösungsvorschlag

Frage: Erklären Sie, warum ein Veränderungs-Bit im Seitendescrptor enthalten ist. Was hat es zur Folge, wenn dieser Bit nicht vorhanden wäre?

Antwort:

Das Veränderungs-Bit markiert Seiten, auf den ein Schreibzugriff stattgefunden hat. Beim Auslagern der Seite wird entsprechend die auf dem Hintergrundspeicher vorhandene Kopie der Seite aktualisiert. Wenn hingegen der Veränderungs-Bit nicht gesetzt ist, muss beim Auslagern keine Änderung an der Kopie auf dem Hintergrundspeicher vorgenommen werden.

Würde dieser Bit fehlen, so müsste bei jeder Auslagerung der Rahmeninhalt auf den Hintergrundspeicher geschrieben werden, auch wenn kein Schreibzugriff vorlag und es damit keine Änderung gegeben hat. Enthält die Seite z.B. Programm-Code oder statische Daten, die sich nicht geändert haben, so ist ein Neubeschreiben der Hintergrundkopie nicht sinnvoll.

Aufgabe 2.4 Lösungsvorschlag, FIFO

Anfrage	f1	f2	f3	f4	pagefault
1	1				!
3	1	3			!
5	1	3	5		!
0	1	3	5	0	!
2	2	3	5	0	!
4	2	4	5	0	!
6	2	4	6	0	!
5	2	4	6	5	!
3	3	4	6	5	!
4	3	4	6	5	

Aufgabe 2.4 Lösungsvorschlag, FIFO - Fortsetzung

Anfrage	f1	f2	f3	f4	pagefault
4	3	4	6	5	
2	3	2	6	5	!
1	3	2	1	5	!
5	3	2	1	5	
6	3	2	1	6	!
5	5	2	1	6	!
1	5	2	1	6	
0	5	0	1	6	!
6	5	0	1	6	
3	5	0	3	6	!
1	5	0	3	1	!

16 Pagefaults

Aufgabe 2.4 Lösungsvorschlag, LIFO

Anfrage	f1	f2	f3	f4	pagefault
1	1				!
3	1	3			!
5	1	3	5		!
0	1	3	5	0	!
2	1	3	5	2	!
4	1	3	5	4	!
6	1	3	5	6	!
5	1	3	5	6	
3	1	3	5	6	
4	1	3	5	4	!

Aufgabe 2.4 Lösungsvorschlag, LIFO - Fortsetzung

Anfrage	f1	f2	f3	f4	pagefault
4	1	3	5	4	!
2	1	3	5	2	!
1	1	3	5	2	
5	1	3	5	2	
6	1	3	5	6	!
5	1	3	5	6	
1	1	3	5	6	
0	1	3	5	0	!
6	1	3	5	6	!
3	1	3	5	6	
1	1	3	5	6	

12 Pagefaults

Aufgabe 2.4 Lösungsvorschlag, LRU

Anfrage	f1	f2	f3	f4	pagefault
1	1				!
3	1	3			!
5	1	3	5		!
0	1	3	5	0	!
2	2	3	5	0	!
4	2	4	5	0	!
6	2	4	6	0	!
5	2	4	6	5	!
3	3	4	6	5	!
4	3	4	6	5	

Aufgabe 2.4 Lösungsvorschlag, LRU - Fortsetzung

Anfrage	f1	f2	f3	f4	pagefault
4	3	4	6	5	
2	3	4	2	5	!
1	3	4	2	1	!
5	5	4	2	1	!
6	5	6	2	1	!
5	5	6	2	1	
1	5	6	2	1	
0	5	6	0	1	!
6	5	6	0	1	
3	3	6	0	1	!
1	3	6	0	1	

15 Pagefaults

Aufgabe 2.5 Lösungsvorschlag, F1

Anfrage	f1	f2	f3	pagefault
0	0			!
1	0	1		!
2	0	1	2	!
3	3	1	2	!
0	3	0	2	!
1	3	0	1	!
4	4	0	1	!
0	4	0	1	
1	4	0	1	
2	4	2	1	!
3	4	2	3	!
4	3	4	1	

9 Pagefaults

Aufgabe 2.5 Lösungsvorschlag, F2

Anfrage	f1	f2	f3	f4	pagefault
0	0				!
1	0	1			!
2	0	1	2		!
3	0	1	2	3	!
0	0	1	2	3	
1	0	1	2	3	
4	4	1	2	3	!
0	4	0	2	3	!
1	4	0	1	3	!
2	4	0	1	2	!
3	3	0	1	2	!
4	3	4	1	2	!

10 Pagefaults

Aufgabe 2.5 Lösungsvorschlag

Es fällt auf, dass mit F1 trotz weniger Rahmenanzahl weniger Seitenfehler ausgelöst werden als mit F2. Dies wird Belady's Anomalie bezeichnet und zeigt, dass mit der FIFO-Strategie mehr Rahmenanzahl in seltenen Fällen auch zu schlechterer Performanz führen kann.

Aufgabe 3.1 Lösungsvorschlag

Die niedrigwertigsten Bits werden für den Blockindex CO verwendet. Wir haben Blöcke der Grösse 4 Bytes d.h wir brauchen zwei Bits um jedes Byte in diesen Blöcken eindeutig adressieren zu können.

Die nächsten drei Bits bilden den Mengenindex. Mit drei Bit können wir die 8 Mengen eindeutig adressieren.

Die restlichen (höchstwertigen) acht Bits bilden das *Tag*, das Kollisionen, die mit Mengenindex und Blockindex auftreten eindeutig auflöst (weil alle Bits zusammen eben eine eindeutige Adresse bilden).

Aufgabe 3.2/3.3 Lösungsvorschlag

Der Blockindex ist 0x0 der Mengenindex ist 0x5, das *Tag* 0x71, der Block ist gültig daher handelt es sich um einen Cache Hit, Rückgabewert ist 0x0B.

Der Blockindex ist 0x1 der Mengenindex ist 0x5, das *Tag* 0x6E, der Block ist aber ungültig daher handelt es sich um einen Cache Miss.