

Lösung von Übungsblatt 5

Aufgabe 1 (Speicherverwaltung)

1. Kreuzen Sie an, bei welchen Konzepten der Speicherpartitionierung interne Fragmentierung entsteht.

- Statische Partitionierung
 Dynamische Partitionierung
 Buddy-Algorithmus

2. Kreuzen Sie an, bei welchen Konzepten der Speicherpartitionierung externe Fragmentierung entsteht.

- Statische Partitionierung
 Dynamische Partitionierung
 Buddy-Algorithmus

3. Geben Sie eine Möglichkeit an, um externe Fragmentierung zu beheben.

Durch Defragmentierung. Bei virtuellem Speicher spielt externe Fragmentierung allerdings keine Rolle.

4. Kreuzen Sie an, welches Speicherverwaltungskonzept im kompletten Adressraum den freien Block sucht, der am besten zur Anforderung passt.

- First Fit Next Fit Best fit Random

5. Kreuzen Sie an, welches Speicherverwaltungskonzept ab dem Anfang des Adressraums den ersten passenden freien Block sucht.

- First Fit Next Fit Best fit Random

6. Kreuzen Sie an, welches Speicherverwaltungskonzept den großen Bereich freien Speicher am Ende des Adressraums schnell zerstückelt.

- First Fit Next Fit Best fit Random

7. Kreuzen Sie an, welches Speicherverwaltungskonzept zufällig einen freien und passenden Block sucht.

- First Fit Next Fit Best fit Random

8. Kreuzen Sie an, welches Speicherverwaltungskonzept ab der Stelle der letzten Blockzuweisung einen passenden freien Block sucht.

- First Fit Next Fit Best fit Random

9. Kreuzen Sie an, welches Speicherverwaltungskonzept viele Minifragmente produziert und am langsamsten arbeitet.

- First Fit Next Fit Best fit Random

10. Statische Partitionierung erfordert zwingend Partitionen gleicher Größe.

- Wahr Falsch

11. Der folgende Speicherbereich gehört zu einem Speicher mit dynamischer Partitionierung. Geben Sie für jeden der drei Algorithmen First Fit, Next Fit und Best Fit die Nummer der freien Partition an, die der entsprechende Algorithmus verwendet, um einen Prozess einzufügen, der 21 MB Speicher benötigt.

a) First Fit: 2

b) Next Fit: 7

c) Best Fit: 8

letzter zugewiesener Bereich →

10 MB	0
22 MB	1
30 MB	2
2 MB	3
7 MB	4
17 MB	5
12 MB	6
45 MB	7
21 MB	8
39 MB	9

frei
belegt

Aufgabe 2 (Buddy-Verfahren)

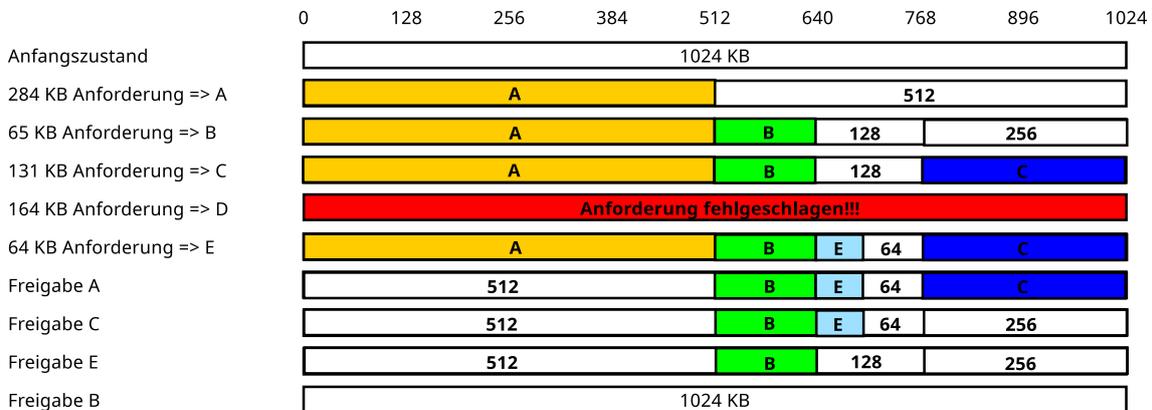
Das Buddy-Verfahren zur Zuweisung von Speicher an Prozesse soll für einen 1024 kB großen Speicher verwendet werden. Führen Sie die angegebenen Aktionen durch und

geben Sie den Belegungszustand des Speichers nach jeder Anforderung oder Freigabe an.

	1024 KB						
65 KB Anforderung => A	A	128 KB	256 KB	512 KB			
30 KB Anforderung => B	A	B	32	64 KB	256 KB	512 KB	
90 KB Anforderung => C	A	B	32	64 KB	C	128 KB	512 KB
34 KB Anforderung => D	A	B	32	D	C	128 KB	512 KB
130 KB Anforderung => E	A	B	32	D	C	128 KB	E
Freigabe C	A	B	32	D	128 KB	128 KB	E
	A	B	32	D	256 KB		E
Freigabe B	A	32	32	D	256 KB	E	256 KB
	A	64 KB		D	256 KB	E	256 KB
275 KB Anforderung => F <small>Nicht möglich, weil keine 275 kB am Stück frei</small>	A	64 KB		D	256 KB	E	256 KB
145 KB Anforderung => G	A	64 KB		D	G		E
Freigabe D	A	64 KB	64 KB	G		E	256 KB
	A	128 KB		G		E	256 KB
Freigabe A	128 KB	128 KB	G		E	256 KB	
	256 KB		G		E	256 KB	
Freigabe G	128 KB	128 KB	256 KB		E	256 KB	
	512 KB				E	256 KB	
Freigabe E	512 KB			256 KB	256 KB		
	512 KB			512 KB			
	1024 KB						

Aufgabe 3 (Buddy-Verfahren)

Wenden Sie das Buddy-Verfahren zur Zuweisung von Speicher an Prozesse an.



Aufgabe 4 (Real Mode und Protected Mode)

1. Beschreiben Sie die Arbeitsweise des Real Mode.

Jeder Prozess kann direkt auf den gesamten adressierbaren Speicher zugreifen.

2. Begründen Sie warum der Real Mode für Mehrprogrammbetrieb (Multitasking) ungeeignet ist.

Es gibt keinen Speicherschutz.

3. Beschreiben Sie die Arbeitsweise des Protected Mode.

Jeder Prozess darf nur auf seinen eigenen virtuellen Speicher zugreifen. Virtuelle Speicheradressen übersetzt die CPU mit Hilfe der MMU in physische Speicheradressen.

4. Beschreiben Sie was virtueller Speicher ist.

Jeder Prozess besitzt einen eigenen Adressraum. Der Adressraum ist eine Abstraktion des physischen Speichers. Es handelt sich dabei um virtuellen Speicher. Er besteht aus logischen Speicheradressen, die von der Adresse 0 aufwärts durchnummeriert sind und er ist unabhängig von der verwendeten Speichertechnologie und den gegebenen Ausbaumöglichkeiten.

5. Erklären Sie, warum mit virtuellem Speicher der Hauptspeicher besser ausgenutzt wird.

Die Prozesse müssen nicht am Stück im Hauptspeicher liegen. Externe Fragmentierung entsteht, spielt aber keine Rolle.

6. Beschreiben Sie was Mapping ist.

Abbilden des virtuellen Speichers auf den realen Speicher.

7. Beschreiben Sie was Swapping ist.

Prozess des Ein- und Auslagerns von Daten in den/vom Arbeitsspeicher vom/in den Hintergrundspeicher (Festplatten/SSDs).

8. Geben Sie den Namen der Komponente der CPU an, die virtuellen Speicher ermöglicht.

Memory Management Unit (MMU).

9. Beschreiben Sie die Aufgabe der Komponente aus Teilaufgabe 8.

Virtuelle Speicheradressen übersetzt die CPU mit Hilfe der MMU in physische Speicheradressen.

10. Beschreiben Sie das Konzept des virtuellen Speichers mit dem Namen Paging.

Virtuelle Seiten der Prozesse werden auf physische Seiten im Hauptspeicher abgebildet. Alle Seiten haben die gleiche Länge. Die Seitenlänge ist üblicherweise 4 kb. Das Betriebssystem verwaltet für jeden Prozess eine Seitentabelle. In dieser steht, wo sich die einzelnen Seiten des Prozesses befinden. Prozesse arbeiten nur mit virtuellen Speicheradressen. Virtuelle Speicheradressen bestehen aus zwei Teilen. Der werthöhere Teil enthält die Seitennummer. Der wertniedrigere Teil enthält den Offset (Adresse innerhalb einer Seite). Die Länge der virtuellen Adressen ist architekturabhängig und darum 16, 32 oder 64 Bits.

11. Beschreiben Sie wo beim Paging interne Fragmentierung entsteht.

Nur in der letzten Seite eines Prozesses.

12. Geben Sie die maximale Anzahl von Speicheradressen an, die mit einem 16-Bit-Computersystem adressiert werden können.

2^{16} Adressen.

13. Geben Sie die maximale Anzahl von Speicheradressen an, die mit einem 32-Bit-Computersystem adressiert werden können.

2^{32} Adressen.

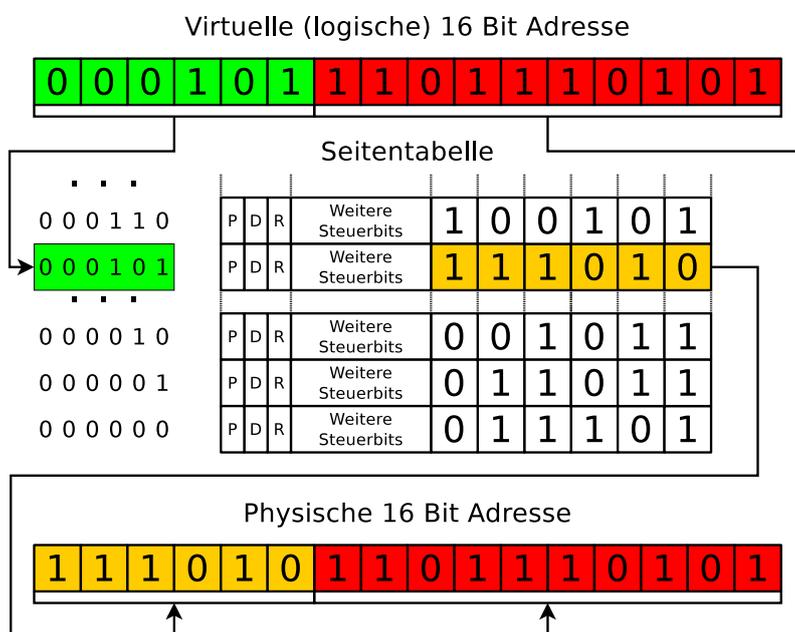
14. Geben Sie die maximale Anzahl von Speicheradressen an, die mit einem 64-Bit-Computersystem adressiert werden können.

Das hängt von der Adressbusbreite ab. Der Adressbus moderner CPUs (AMD64, RISC V, etc.) enthält 48 Leitungen. Dementsprechend umfasst der Adressraum 2^{48} Adressen.

15. Erklären Sie, warum in 32-Bit- und 64-Bit-Systemen mehrstufiges Paging und nicht einstufiges Paging verwendet wird.

Bei 32-Bit-Betriebssystemen mit einer Seitenlänge von 4 kB kann die Seitentabelle jedes Prozesses 4 MB groß sein. Bei 64-Bit-Betriebssystemen können die Seitentabellen wesentlich größer sein. Mehrstufiges Paging reduziert die Hauptspeicherbelegung, da einzelne Seiten der verschiedenen Stufen in den Auslagerungsspeicher verschoben werden können, um Speicherkapazität im Hauptspeicher freizugeben.

16. Berechnen Sie die physische 16-Bit-Speicheradresse unter Verwendung der Adressumrechnung mit einstufigem Paging. Ergänzen Sie die einzelnen Bits in der physischen 16-Bit-Adresse.



17. Beschreiben Sie die Aufgabe und den Inhalt des Page-Table Base Register (PTBR).

Das PTBR speichert die Speicheradresse, bei der die Seitentabelle des laufenden Prozesses anfängt.

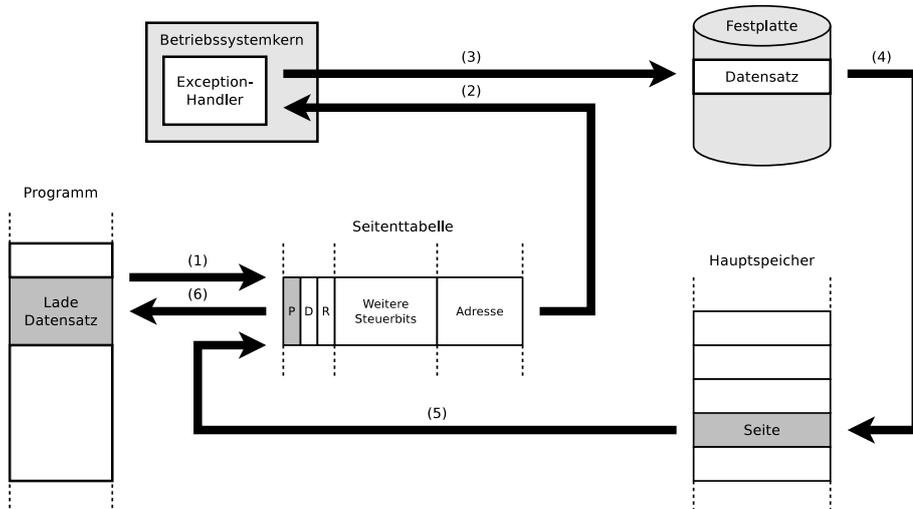
18. Beschreiben Sie die Aufgabe und den Inhalt des Page-Table Length Register (PTLR).

Das PTLT speichert speichert die Länge der Seitentabelle des laufenden Prozesses.

19. Beschreiben Sie wie eine Page Fault Ausnahme (Exception) entsteht.

Ein Prozess versucht auf eine Seite zuzugreifen, die nicht im physischen Hauptspeicher ist.

20. Die Abbildung zeigt eine Page Fault Ausnahme (Exception). Beschreiben Sie den Ablauf Schritt für Schritt.



- (1) Ein Prozess versucht auf eine Seite zuzugreifen, die nicht im physischen Hauptspeicher liegt.
 - (2) Ein Software-Interrupt (Exception) wird ausgelöst, um vom Benutzermodus in den Kernelmodus zu wechseln.
 - (3) Die Seite mit Hilfe des Controllers und Gerätetreibers auf dem Auslagerungsspeicher (SSD/HDD) lokalisieren.
 - (4) Die Seite in eine freie Hauptspeicherseite kopieren.
 - (5) Die Seitentabelle aktualisieren.
 - (6) Die Kontrolle an den Prozess zurückgeben.
21. Beschreiben Sie wie eine Access Violation Ausnahme (Exception) oder General Protection Fault Ausnahme (Exception) entsteht.

Ein Prozess versucht auf eine virtuelle Speicheradresse zuzugreifen, auf die er nicht zugreifen darf.

22. Beschreiben Sie die Auswirkung einer Access Violation Ausnahme (Exception) oder General Protection Fault Ausnahme (Exception).

Bei einigen Windows-Betriebssystemen aus der Vergangenheit waren Schutzverletzungen häufig ein Grund für Systemabstürze und hatten einen „Blue Screen“ zur Folge. Unter Linux wird als Ergebnis das Signal SIGSEGV erzeugt.

23. Geben Sie an, was der Kernelspace enthält.

Der Kernelspace enthält den Betriebssystemkern (Kernel) und Kernelerweiterungen (Treiber).

24. Geben Sie an, was de Userspace enthält.

Der Userspace enthält den aktuell ausgeführten Prozess, der um den Erweiterungsspeicher („Swap“, Windows: „Page-File“) vergrößert wird.

Aufgabe 5 (Speicherverwaltung)

Kreuzen Sie bei jeder Aussage zur Speicherverwaltung an, ob die Aussage wahr oder falsch ist.

1. Real Mode ist für Multitasking-Systeme geeignet.
 Wahr Falsch
2. Beim Protected Mode läuft jeder Prozess in seiner eigenen, von anderen Prozessen abgeschotteten Kopie des physischen Adressraums.
 Wahr Falsch
3. Bei statischer Partitionierung entsteht interne Fragmentierung.
 Wahr Falsch
4. Bei dynamischer Partitionierung ist externe Fragmentierung unmöglich.
 Wahr Falsch
5. Beim Paging haben alle Seiten die gleiche Länge.
 Wahr Falsch
6. Ein Vorteil langer Seiten beim Paging ist geringe interne Fragmentierung.
 Wahr Falsch
7. Ein Nachteil kurzer Seiten beim Paging ist, das die Seitentabelle sehr groß werden kann.
 Wahr Falsch
8. Die MMU übersetzt beim Paging logische Speicheradressen mit der Seitentabelle in physische Adressen.
 Wahr Falsch
9. Moderne Betriebssysteme (für x86) arbeiten im Protected Mode und verwenden ausschließlich Paging.
 Wahr Falsch

Aufgabe 6 (Seiten-Ersetzungsstrategien)

1. Erklären Sie, warum die optimale Ersetzungsstrategie OPT nicht implementiert werden kann.

Weil man nicht in die Zukunft schauen kann und damit ist die zukünftige Zugriffsfolge unbekannt.

2. Führen Sie die gegebene Zugriffsfolge mit den Ersetzungsstrategien Optimal, LRU, LFU und FIFO einmal mit einem Datencache mit einer Kapazität von 4 Seiten und einmal mit 5 Seiten durch. Berechnen Sie auch die Hitrate und die Missrate für alle Szenarien.

Optimale Ersetzungsstrategie (OPT):

Hinweis: Wenn bei der optimalen Ersetzungsstrategie eine Seite verdrängt werden muss, wird die Seite verdrängt, auf die am längsten in der Zukunft nicht zugegriffen wird.

Anfragen: **1 3 5 4 2 4 3 2 1 0 5 3 5 0 4 3 5 4 3 2 1 3 4 5**

Seite 1:	1	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	2	1	1	1	1
Seite 2:		3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3
Seite 3:			5	5	2	2	2	2	2	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5
Seite 4:				4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4

Hitrate: $15/24 = 0,625$

Missrate: $9/24 = 0,375$

Anfragen: **1 3 5 4 2 4 3 2 1 0 5 3 5 0 4 3 5 4 3 2 1 3 4 5**

Seite 1:	1	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
Seite 2:		3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	
Seite 3:			5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	
Seite 4:				4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	
Seite 5:					2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	1	1	1

Hitrate: $17/24 = 0,7083333$

Missrate: $7/24 = 0,2916666$

Ersetzungsstrategie Least Recently Used (LRU):

Hinweis: Wenn bei der Ersetzungsstrategie LRU eine Seite verdrängt werden muss, wird die Seite verdrängt, auf die am längsten nicht zugegriffen wurde.

Anfragen: **1 3 5 4 2 4 3 2 1 0 5 3 5 0 4 3 5 4 3 2 1 3 4 5**

Seite 1:	1	1	1	1	2	2	2	2	2	2	2	3	3	3	3	3	3	3	3	3	3	3	3			
Seite 2:		3	3	3	3	3	3	3	3	3	3	5	5	5	5	5	5	5	5	5	5	1	1	1	1	
Seite 3:			5	5	5	5	5	5	5	1	1	1	1	1	1	1	4	4	4	4	4	4	4	4		
Seite 4:				4	4	4	4	4	4	4	0	0	0	0	0	0	0	0	0	0	0	2	2	2	2	5

Queue:

1	3	5	4	2	4	3	2	1	0	5	3	5	0	4	3	5	4	3	2	1	3	4	5
	1	3	5	4	2	4	3	2	1	0	5	3	5	0	4	3	5	4	3	2	1	3	4
		1	3	5	4	2	4	3	2	1	0	5	3	5	0	4	3	5	4	3	2	1	3
			1	3	5	4	2	4	3	2	1	0	5	3	5	0	4	3	5	4	3	2	1
				1	3	5	4	2	4	3	2	1	0	5	3	5	0	4	3	5	4	3	2
					1	3	5	4	2	4	3	2	1	0	5	3	5	0	4	3	5	4	3
						1	3	5	4	2	4	3	2	1	0	5	3	5	0	4	3	5	4
							1	3	5	4	2	4	3	2	1	0	5	3	5	0	4	3	5
								1	3	5	4	2	4	3	2	1	0	5	3	5	0	4	3
									1	3	5	4	2	4	3	2	1	0	5	3	5	0	4
										1	3	5	4	2	4	3	2	1	0	5	3	5	0
											1	3	5	4	2	4	3	2	1	0	5	3	5
												1	3	5	4	2	4	3	2	1	0	5	3
													1	3	5	4	2	4	3	2	1	0	5
														1	3	5	4	2	4	3	2	1	0
															1	3	5	4	2	4	3	2	1
																1	3	5	4	2	4	3	2
																	1	3	5	4	2	4	3
																		1	3	5	4	2	4
																			1	3	5	4	2
																				1	3	5	4
																					1	3	5
																						1	3
																							1

Hitrate: 11/24 = 0,4583333%
Missrate: 13/24 = 0,5416666%

Anfragen: **1 3 5 4 2 4 3 2 1 0 5 3 5 0 4 3 5 4 3 2 1 3 4 5**

Seite 1:	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	2	2	2	2
Seite 2:		3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3
Seite 3:			5	5	5	5	5	5	5	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1
Seite 4:				4	4	4	4	4	4	4	4	5	5	5	5	5	5	5	5	5	5	5	5	5
Seite 5:					2	2	2	2	2	2	2	2	2	2	2	2	4	4	4	4	4	4	4	4

Queue:

1	3	5	4	2	4	3	2	1	0	5	3	5	0	4	3	5	4	3	2	1	3	4	5
	1	3	5	4	2	4	3	2	1	0	5	3	5	0	4	3	5	4	3	2	1	3	4
		1	3	5	4	2	4	3	2	1	0	5	3	5	0	4	3	5	4	3	2	1	3
			1	3	5	4	2	4	3	2	1	0	5	3	5	0	4	3	5	4	3	2	1
				1	3	5	4	2	4	3	2	1	0	5	3	5	0	4	3	5	4	3	2
					1	3	5	4	2	4	3	2	1	0	5	3	5	0	4	3	5	4	3
						1	3	5	4	2	4	3	2	1	0	5	3	5	0	4	3	5	4
							1	3	5	4	2	4	3	2	1	0	5	3	5	0	4	3	5
								1	3	5	4	2	4	3	2	1	0	5	3	5	0	4	3
									1	3	5	4	2	4	3	2	1	0	5	3	5	0	4
										1	3	5	4	2	4	3	2	1	0	5	3	5	0
											1	3	5	4	2	4	3	2	1	0	5	3	5
												1	3	5	4	2	4	3	2	1	0	5	3
													1	3	5	4	2	4	3	2	1	0	5
														1	3	5	4	2	4	3	2	1	0
															1	3	5	4	2	4	3	2	1
																1	3	5	4	2	4	3	2
																	1	3	5	4	2	4	3
																		1	3	5	4	2	4
																			1	3	5	4	2
																				1	3	5	4
																					1	3	5
																						1	3
																							1

Hitrate: 14/24 = 0,5833333%
Missrate: 10/24 = 0,4166666%

Ersetzungsstrategie Least Frequently Used (LFU):

Hinweis: Wenn bei der Ersetzungsstrategie LFU eine Seite verdrängt werden muss, wird die Seite verdrängt, auf die am wenigsten zugegriffen wurde. Es wird für jede Seite in der Seitentabelle ein Referenzzähler geführt, der die

3. Beschreiben Sie die Kernaussage der Anomalie von Laszlo Belady.

FIFO führt bei bestimmten Zugriffsmustern bei einem vergrößerten Speicher zu schlechteren Ergebnissen.

4. Zeigen Sie Belady's Anomalie, indem sie die gegebene Zugriffsfolge mit der Ersetzungsstrategie FIFO einmal mit einem Datencache mit einer Kapazität von 3 Seiten und einmal mit 4 Seiten durchführen. Berechnen Sie auch die Hitrate und die Missrate für beide Szenarien.

Anfragen: **3 2 1 0 3 2 4 3 2 1 0 4**

Seite 1:	3	3	3	0	0	0	4	4	4	4	4	4
Seite 2:		2	2	2	3	3	3	3	3	1	1	1
Seite 3:			1	1	1	2	2	2	2	2	0	0

Hitrate: $3/12 = 25\%$

Missrate: $9/12 = 75\%$

Anfragen: **3 2 1 0 3 2 4 3 2 1 0 4**

Seite 1:	3	3	3	3	3	3	4	4	4	4	0	0
Seite 2:		2	2	2	2	2	2	3	3	3	3	4
Seite 3:			1	1	1	1	1	1	2	2	2	2
Seite 4:				0	0	0	0	0	0	1	1	1

Hitrate: $2/12 = 16,66\%$

Missrate: $10/12 = 83,33\%$

Aufgabe 7 (Zeitgesteuerte Kommandoausführung, Sortieren, Umgebungsvariablen)

1. Erzeugen Sie in Ihrem Benutzerverzeichnis (Home-Verzeichnis) ein Verzeichnis `Entbehrlich` und schreiben Sie einen Cron-Job, der immer Dienstags um 1:25 Uhr morgens den Inhalt von `Entbehrlich` löscht.

Die Ausgabe des Kommandos soll in eine Datei `LöschLog.txt` in Ihrem Home-Verzeichnis angehängt werden.

```
$ mkdir ~/Entbehrlich  
$ crontab -e
```

Folgende Zeile eintragen:

```
25 1 * * 2 rm -rfv /home/USERNAME/Entbehrlich/* >>  
/home/USERNAME/LöschLog.txt
```

2. Schreiben Sie einen Cron-Job, der alle 3 Minuten zwischen 14:00 und 15:00 Uhr an jedem Dienstag im Monat November eine Zeile mit folgendem Aussehen (und den aktuellen Werten) an die Datei `Datum.txt` anhängt:

```
Heute ist der 30.10.2008  
Die Uhrzeit ist 09:24:42 Uhr  
*****
```

```
$ crontab -e
```

Folgende Zeile eintragen:

```
*/3 14 * 11 * date +"Heute ist der %x%nDie Uhrzeit ist  
%H:%M:%S Uhr%n*****" >> Datum.txt
```

3. Schreiben Sie einen at-Job, der um 17:23 Uhr heute eine Liste der laufenden Prozesse ausgibt.

*Das Kommandozeilenwerkzeug `at` müssen Sie evtl. erst installieren.
Unter Debian/Ubuntu geht das mit:
`$ sudo apt update && sudo apt install at`
Unter CentOS/Fedora/RedHat geht das mit:
`$ sudo yum install at`*

```
$ at 1723 today
```

Folgende Zeile eintragen:

```
ps -r
```

4. Schreiben Sie einen at-Job, der am 24. Dezember um 8:15 Uhr morgens den Text „Endlich Weihnachten!“ ausgibt.

```
$ at 0815 DEZ 24
```

Folgende Zeile eintragen:

```
echo "Endlich Weihnachten!"
```

5. Erzeugen Sie in Ihrem Home-Verzeichnis eine Datei `Kanzler.txt` mit folgendem Inhalt:

```
Willy      Brandt      1969
```

```
Angela   Merkel   2005
Gerhard  Schröder  1998
KurtGeorg Kiesinger 1966
Helmut   Kohl     1982
Konrad   Adenauer  1949
Helmut   Schmidt  1974
Ludwig   Erhard   1963
```

```
$ echo "Willy   Brandt   1969" >> ~/Kanzler.txt
$ echo "Angela  Merkel   2005" >> ~/Kanzler.txt
$ echo "Gerhard Schröder  1998" >> ~/Kanzler.txt
$ echo "KurtGeorg Kiesinger 1966" >> ~/Kanzler.txt
$ echo "Helmut  Kohl     1982" >> ~/Kanzler.txt
$ echo "Konrad  Adenauer  1949" >> ~/Kanzler.txt
$ echo "Helmut  Schmidt  1974" >> ~/Kanzler.txt
$ echo "Ludwig  Erhard   1963" >> ~/Kanzler.txt
```

6. Geben Sie die Datei `Kanzler.txt` sortiert anhand der Vornamen aus.

```
$ sort ~/Kanzler.txt
```

7. Geben Sie die Datei `Kanzler.txt` sortiert anhand des dritten Buchstabens der Nachnamen aus.

```
$ sort -k+2.4 ~/Kanzler.txt
```

8. Geben Sie die Datei `Kanzler.txt` sortiert anhand des Jahres der Amtseinführung aus.

```
$ sort -k3 ~/Kanzler.txt
```

9. Geben Sie die Datei `Kanzler.txt` rückwärts sortiert anhand des Jahres der Amtseinführung aus und leiten Sie die Ausgabe in eine Datei `Kanzlerdaten.txt`.

```
$ sort -k3 -nr ~/Kanzler.txt > ~/Kanzlerdaten.txt
```

10. Erzeugen Sie mit dem Kommando `export` eine Umgebungsvariable `VAR1` und weisen Sie dieser den Wert `Testvariable` zu.

```
$ export VAR01=Testvariable
```

11. Geben Sie den Wert von `VAR1` in der Shell aus.

```
$ printenv VAR01
```

12. Löschen Sie die Umgebungsvariable `VAR1`.

```
$ unset VAR01
```