

Lösung von Übungsblatt 2

Aufgabe 1 (Digitale Datenspeicher)

1. Nennen Sie einen digitalen Datenspeicher, der mechanisch arbeitet.

Lochstreifen, Lochkarte, CD/DVD beim Pressen.

2. Nennen Sie zwei rotierende magnetische digitale Datenspeicher.

Festplatte, Trommelspeicher, Diskette.

3. Nennen Sie zwei nichtrotierende magnetische digitale Datenspeicher.

Kernspeicher, Magnetband, Magnetstreifen, Magnetkarte, Compact Cassette (Datasette), Magnetblasenspeicher.

4. Nennen Sie vier Vorteile von Datenspeicher ohne bewegliche Teile gegenüber Datenspeichern mit beweglichen Teilen.

Weniger Energieverbrauch, weniger Abnutzung, weniger Abwärme, unempfindlichkeit gegen Stöße, keine Laufgeräusche.

5. Beschreiben Sie was wahlfreier Zugriff ist.

Wahlfreier Zugriff heißt, dass das Medium nicht - wie z.B. bei Bandlaufwerken - von Beginn an sequentiell durchsucht werden muss, um eine bestimmte Stelle (Datei) zu finden.

6. Nennen Sie einen nicht-persistenten Datenspeicher.

Hauptspeicher (DRAM).

7. Der Speicher eines Computersystems wird in die Kategorien Primärspeicher, Sekundärspeicher und Tertiärspeicher unterschieden. Auf welche Kategorie(n) kann der Prozessor direkt zugreifen?

Nur auf den Primärspeicher.

8. Nennen Sie die Kategorie(n) aus Teilaufgabe 7, auf die der Prozessor nur über einen Controller zugreifen kann.

Auf den Sekundärspeicher und den Tertiärspeicher.

9. Nennen Sie für jede Kategorie aus Teilaufgabe 7 zwei Beispiele.

Primärspeicher: Register, Cache, Hauptspeicher.

Sekundärspeicher: Festplatte, SSD, CF-Karte.

Tertiärspeicher: CD/DVD-Laufwerk, MO-Laufwerk, Magnetband.

10. Erklären Sie, warum Speicherseiten in den oberen Schichten der Speicherhierarchie ständig ersetzt werden.

Da die oberen Speicherebenen praktisch immer voll belegt sind, müssen Speicherseiten ersetzt werden.

Aufgabe 2 (Cache-Schreibstrategien)

1. Nennen Sie die beiden grundsätzlichen Cache-Schreibstrategien.

Write-Through und Write-Back.

2. Nennen Sie die Cache-Schreibstrategie aus Teilaufgabe 1, bei der es zu Inkonsistenzen kommen kann.

Write-Back.

3. Nennen Sie die Cache-Schreibstrategie aus Teilaufgabe 1, bei der die System-Geschwindigkeit geringer ist.

Write-Through.

4. Nennen Sie die Cache-Schreibstrategie aus Teilaufgabe 1, bei der sogenannte „Dirty Bits“ zum Einsatz kommen.

Write-Back.

5. Beschreiben Sie die Aufgabe der „Dirty Bits“.

Für jede Seite im Cache wird ein Dirty Bit im Cache gespeichert, das angibt, ob die Seite geändert wurde.

Aufgabe 3 (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. 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
4. Kreuzen Sie an, welches Speicherverwaltungskonzept ab dem Anfang des Adressraums den ersten passenden freien Block sucht.
☒ First Fit ☐ Next Fit ☐ Best fit ☐ Random
5. Kreuzen Sie an, welches Speicherverwaltungskonzept den großen Bereich freien Speicher am Ende des Adressraums schnell zerstückelt.
☐ First Fit ☒ Next Fit ☐ Best fit ☐ Random
6. Kreuzen Sie an, welches Speicherverwaltungskonzept zufällig einen freien und passenden Block sucht.
☐ First Fit ☐ Next Fit ☐ Best fit ☒ Random
7. Kreuzen Sie an, welches Speicherverwaltungskonzept viele Minifragmente produziert und am langsamsten arbeitet.
☐ First Fit ☒ Next Fit ☐ Best fit ☐ Random
8. Welches Konzept zur Speicherverwaltung produziert viele Minifragmente und arbeitet am langsamsten?
☐ First Fit ☐ Next Fit ☒ Best fit ☐ Random
9. 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 4 (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 256 KB
Freigabe C	A	B 32 D	128 KB	128 KB
	A	B 32 D	256 KB	E 256 KB
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 256 KB
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 5 (Real Mode und Protected Mode)

1. Beschreiben Sie wie der Real Mode arbeitet.

Jeder Prozess kann direkt auf den gesamten adressierbaren Speicher zugreifen.

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

Es gibt keinen Speicherschutz. Jeder Prozess kann auf den gesamten adressierbaren Speicher zugreifen. Das wäre inakzeptabel für Multitasking-Betriebssysteme. Zudem können Intel-kompatible Prozessoren im Real Mode maximal 1 MB Hauptspeicher adressieren.

3. Beschreiben Sie wie der Protected Mode arbeitet.

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. Darum ist die Fragmentierung des Hauptspeichers kein Problem.

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. Nennen Sie die Komponente der CPU, 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 Seitentabel.

le. 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.

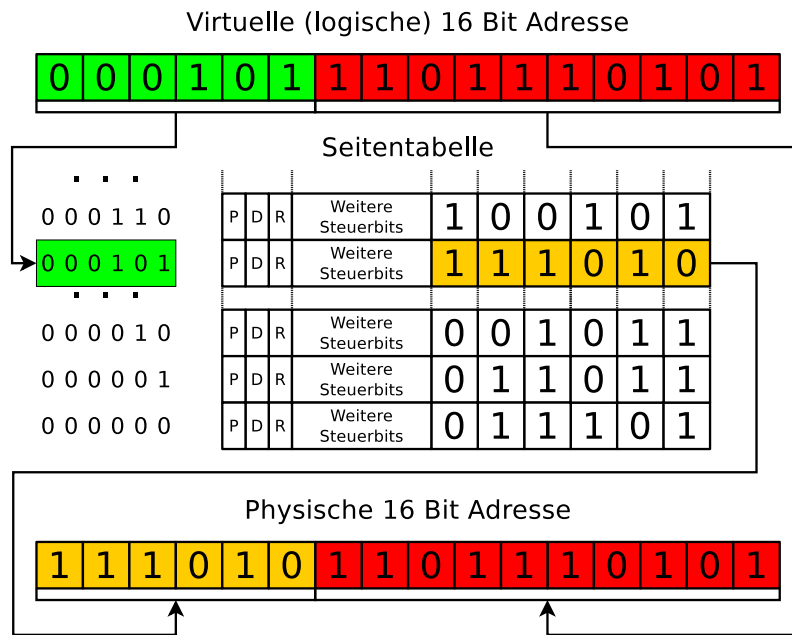
13. Geben Sie die maximale Anzahl von Speicheradressen an, die mit einem 32-Bit-Computersystem adressiert werden können.

2^{32} Adressen.

14. 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.

15. 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.



16. Beschreiben Sie den Zweck des Page-Table Base Register (PTBR).

Es speichert die Hauptspeicheradresse, bei der die Seitentabelle des aktuellen Prozesses beginnt.

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

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

18. Beschreiben Sie wie das Betriebssystem auf eine Page Fault Ausnahme (Exception) reagiert.

Das Betriebssystem behandelt die Ausnahme mit folgenden Schritten:

- *Daten auf dem Sekundärspeicher (SDD/HDD) lokalisieren.*
- *Freie Seiten im Hauptspeicher lokalisieren.*
- *Die Daten in die Seiten laden.*
- *Seitentabelle aktualisieren.*
- *Kontrolle an das Programm zurückgeben. Dieses fährt die Anweisung, die zum Page Fault führte, erneut aus.*

19. 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.

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

Crash des Betriebssystems.

Aufgabe 6 (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, dass 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 Paging.
☒ Wahr ☐ Falsch

Aufgabe 7 (Seiten-Ersetzungsstrategien)

1. Die beste Seitenersetzungsstrategie ist die optimale Strategie. Beschreiben Sie, wie sie funktioniert.

Sie verdrängt die Seite, auf die am längsten in der Zukunft nicht zugegriffen wird.

2. Begründen 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.

3. Beschreiben Sie ein Szenario, in dem die optimale Strategie in der Praxis hilfreich ist.

Mit OPT bewertet man die Effizienz anderer Ersetzungsstrategien.

4. 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	2	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
Seite 3:			5	5	2	2	2	2	2	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

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

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

Hitrate: $14/24 = 0,5833333\%$

Missrate: $10/24 = 0,4166666\%$

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 Anzahl der Zugriffe speichert. Ist der Speicher voll und kommt es zum Miss, wird die Seite entfernt, deren Referenzzähler den niedrigsten Wert hat.

[illegible]

Missrate: $12/24 = 0,5\%$

Seite 1:	1 ₁	1 ₁	1 ₁	1 ₁	1 ₁	1 ₁	1 ₁	1 ₁	1 ₂	1 ₂	1 ₂	1 ₂	1 ₂	1 ₂	0 ₁	0 ₁	0 ₁	0 ₁	0 ₁	0 ₁	1 ₁	1 ₁	1 ₁	1 ₁
Seite 2:	3 ₁	3 ₁	3 ₁	3 ₁	3 ₁	3 ₁	3 ₂	3 ₂	3 ₂	3 ₂	3 ₃	3 ₃	3 ₃	3 ₃	3 ₃	3 ₄	3 ₄	3 ₅	3 ₅	3 ₅	3 ₆	3 ₆	3 ₆	3 ₆
Seite 3:			5 ₁	5 ₁	5 ₁	5 ₁	5 ₁	5 ₁	5 ₁	0 ₁	5 ₁	5 ₁	5 ₂	5 ₂	5 ₂	5 ₂	5 ₃	5 ₃	5 ₃	5 ₃	5 ₃	5 ₃	5 ₄	5 ₄
Seite 4:			4 ₁	4 ₁	4 ₂	4 ₂	4 ₂	4 ₂	4 ₂	4 ₂	4 ₂	4 ₂	4 ₂	4 ₂	4 ₂	4 ₃	4 ₃	4 ₃	4 ₄	4 ₄	4 ₄	4 ₄	4 ₅	4 ₅
Seite 5:					2 ₁	2 ₁	2 ₁	2 ₂	2 ₂	2 ₂	2 ₂	2 ₂	2 ₂	2 ₂	2 ₂	2 ₂	2 ₂	2 ₂	2 ₂	2 ₂	2 ₃	2 ₃	2 ₃	2 ₃

Missrate: $15/24 = 0,625\%$

Ersetzungsstrategie FIFO:

Hinweis: Wenn bei der Ersetzungsstrategie FIFO eine Seite verdrängt werden muss, wird die Seite verdrängt, die sich am längsten im Speicher befindet.

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

Hitrate: $15/24 = 0,625\%$

Missrate: $9/24 = 0,375\%$

- Beschreiben Sie die Kernaussage der Anomalie von Laszlo Belady. *FIFO führt bei bestimmten Zugriffsmustern bei einem vergrößerten Speicher zu schlechteren Ergebnissen.*
- 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: $10/12 = 83,33\%$