

Name:



Klausur+Musterlösung

Betriebssysteme SS 2014

11.07.14

Vorname:	
Nachname:	
Matrikelnummer:	
Studienfach:	

Bitte tragen Sie auf jeder Seite Ihre Matrikelnummer ein und überprüfen Sie diese Klausur auf Vollständigkeit (16 Seiten!).

Tragen Sie auf dem Deckblatt Ihre Daten in Druckbuchstaben ein.

Verwenden Sie ausschließlich die beigelegten Blätter. Die Rückseite können Sie als Schmierpapier für Ihre Notizen verwenden. Sollten diese nicht ausreichen, so wenden Sie sich bitte an die Aufsicht. Ein Taschenrechner (kein Handy!) ist erlaubt. Andere Hilfsmittel (z. B. Handys, Bücher, eigenes Papier, etc...) sind verboten. Die Benutzung gilt als Täuschungsversuch und führt zum Ausschluss von der Klausur.

Die Klausur enthält 15 Aufgaben mit insgesamt maximal 165 Punkten. Davon sind für eine volle Leistung (100 Prozent) 120 Punkte zu erreichen. Die Anzahl der Punkte entspricht ungefähr der Bearbeitungszeit in Minuten. Die maximale Bearbeitungszeit für die gesamte Klausur beträgt 180 Minuten.

Viel Erfolg !!!

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	Σ	Ü	Note
10	16	4	6	14	6	14	10	6	18	10	6	22	15	8			

Aufgabe 1: Prozesse und virtuelle Maschinen (10 Punkte)

1. Benennen Sie den Unterschied zwischen *dispatcher* und *scheduler*. Erläutern Sie was die beiden jeweils machen. (2 Pkte)

Der Dispatcher überführt von einem Zustand in den anderen, der Scheduler entscheidet über die Reihenfolge in der *bereit*-Schlange.

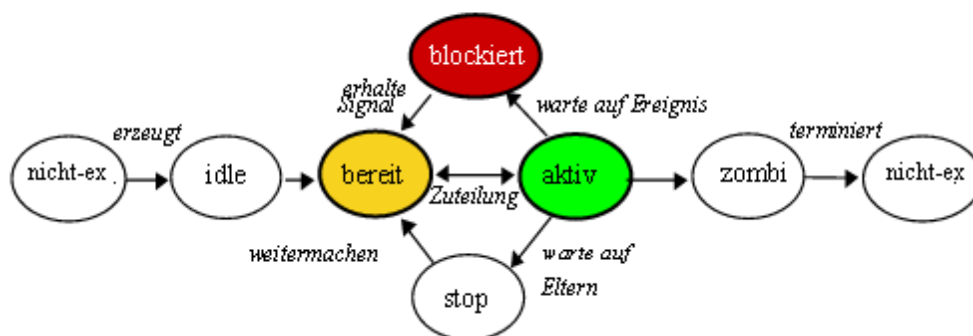
2. Erklären Sie was Software - Hardware-Migration ist und nennen Sie ein Beispiel, was nicht in der Vorlesung erwähnt wurde. (2 Pkte)

Ein Programm wird in Hardware gegossen, d.h. das, was per Software gelöst wurde, wird in eine Schaltung überführt. Diese kann nicht mehr verändert werden, ist aber wesentlich schneller als die Software-Lösung, da hier nicht mehr über den Prozessor gearbeitet wird sondern der Chip selbst die Operation übernimmt. Beispiel: Shader-Chips auf einer Grafikkarte.

3. Wann ist es sinnvoll, Virtuelle Maschinen zu benutzen? (2 Pkte)

Es ist immer sinnvoll eine VM zu benutzen, wenn man verschiedene Hardware darstellen muss (z.B. wenn man Software für ein Handy schreibt, das man nicht physisch besitzt). Auch um Software (Viren) zu analysieren in einer gesicherten Umgebung. Des weiteren bietet es sich für Arbeitsplätze an, bei denen viele Systeme auf einmal aktualisiert und gesichert werden müssen.

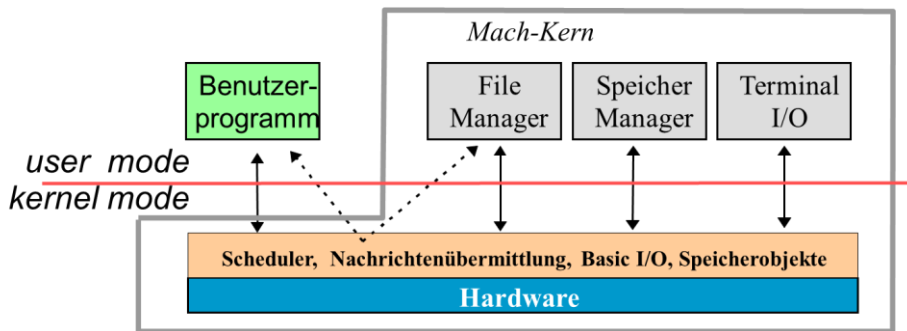
4. Nennen Sie die typischen Prozesszustände von einem Unix-System. (2 Pkte)



Name:

5. Erläutern Sie anhand einer Zeichnung, was typisch für das MACH-Modell ist. (2 Pkte)

Typisch für das MACH-Modell ist die Auslagerung aller Manager aus dem Kern in den *user*-Bereich, so dass sie sie leicht ausgetauscht werden können. Der Kern hat nur noch *message-passing*-Funktionalität. Damit ist der Mach-Kern zwar deutlich kleiner, aber wegen der häufigen Systemcall-Aufrufe auch langsamer.



Aufgabe 2: Threads & Prozess-Scheduling (16 Punkte)

1. Was ist der Unterschied zwischen einem *heavyweight* und einem *kernel* Thread? (2 Pkte)

Heavyweight threads werden im Kern registriert und laufen im *user mode*. Kernel-threads werden nur für Systemzwecke verwendet, laufen nur im Kern und deshalb immer im *kernel mode*.

2. Wie viele Prozesse können für einen Thread laufen? (2 Pkte)

Genau einer, da ein *thread* nur einem Prozess zugewiesen ist.

3. Gegeben sind folgende Präzedenzrelationen: (10 Punkte)

A1 >> A3	A2 >> A4	A4 >> A5	A3 >> A5	A5 >> A6	A5 >> A7	
----------	----------	----------	----------	----------	----------	--

mit den Prozessen und Laufzeiten

A1(1), A2(2), A3(1), A4(3), A5(1), A6(2), A7(1).

Zeichnen Sie die dazu gehörigen Gantt-Diagramme für *earliest* und *latest scheduling*. Gehen Sie davon aus, dass sie beliebig viele Prozessoren zur Verfügung haben.

- *earliest*

A1	A3		A5	A6
A2		A4		A7

Name:

-latest:

	A1	A3	A5	A6
A2	A4			A7

4. Benennen Sie den kritischen Pfad und seine Länge. (2 Punkte)

- A2, A4, A5, A6 mit der Länge 8

Aufgabe 3: Multiprozessorbetrieb (4 Punkte)

Welche Eigenschaften hat ein Betriebssystem im symmetrischen Multiprozessorbetrieb im Unterschied zum asymmetrischen Betrieb?

Ein Betriebssystem, das im symmetrischen Multiprozessorbetrieb funktioniert, erlaubt jedem Prozessor den direkten, parallelen Zugriff auf die Systemaufrufe. Es erlaubt also eine parallele Ausführung unterschiedlicher Aufrufe.

Im Unterschied dazu kann beim asymmetrischen Betrieb immer nur ein Prozessor auf das Betriebssystem zugreifen, da dort nur ein Stack usw. für die Aufrufe existiert. Diese Anforderung wird abgearbeitet, erst dann kann der nächste Prozessor einen Systemaufruf durchführen.

Aufgabe 4: Synchronisation (6 Punkte)

1. Angenommen, Sie haben 5 Prozesse mit einer Codeauslegung, wie in der Abbildung angegeben. Wie viele Semaphore benötigen Sie zur Synchronisation, wenn die Abschnitte der Typen A bis G korrespondierend kritisch sind? (2 Punkte)

P1	P2	P3	P4	P5
A	G	F	A	C
H	A	A	A	C
B	G	B	B	A
D	H	F	G	F

A benötigt eine, B benötigt eine, C nicht, D nicht, E gibt es nicht, F benötigt eine, G benötigt eine, H benötigt keine → 4 Semaphore werden benötigt.

2. Erklären Sie was *race conditions* sind und wodurch sie verursacht werden. (2 Punkte)

Eine *race condition* entsteht, wenn zwei oder mehr Prozesse auf dieselbe globale Ressource lesend und schreibend zugreifen und dabei sich stören und Fehler verursachen.

Name:

3. Erklären Sie was ein *kritischer Abschnitt* ist und was ihn ausmacht. (2 Punkte)

Ein kritischer Abschnitt ist ein Codeabschnitt, in dem sowohl Lesen als auch Schreiben auf eine globale Ressource (z.B. Variable oder Speicherabschnitte) durchgeführt wird. Dabei wird die Variable auch von anderen Prozessen lesend und schreibend verwendet.

Aufgabe 5: Verklemmungen (14 Punkte)

1. Nennen Sie die vier Bedingungen, bei denen eine bei Nichterfüllung ausreicht, um Verklemmungen unmöglich werden zu lassen. (4 Punkte)

mutual exclusion (Beschränkte Belegung)

hold and wait (Zusätzliche Belegung)

no preemption (Keine vorzeitige Rückgabe)

circular wait (Gegenseitiges Warten)

2. Testen Sie mit Hilfe des Banker-Algorithmus folgenden Fall: (10 Punkte)

Prozesse	Noch benötigte Betriebsmittel				
P1	2	0	5	1	0
P2	0	1	1	0	0
P3	1	0	4	0	0
P4	3	2	6	1	2
Schon belegte Betriebsmittel					
P1	0	1	1	2	2
P2	2	2	1	1	1
P3	2	1	0	1	2
P4	4	1	3	2	1
Frei	1	0	5	1	0

Sollte eine Verklemmung entstehen, lösen Sie diese auf. Schreiben Sie ihren Rechenweg nachvollziehbar auf.

Wir starten mit Frei = 1 0 5 1 0

Darauf kann Prozess P3 abschließen und gibt zurück = 2 1 0 1 2

Es ist Frei = 3 1 5 2 2

Dann kann P2 abschließen mit Rückgabe = 2 2 1 1 1

Es ist Frei = 5 3 6 3 3

Dann P4 mit Rückgabe = 4 1 3 2 1

Es ist Frei = 9 4 9 5 4

Und damit ist P1 zuende -> fertig

Name:

Aufgabe 6: Echtzeitsysteme (6 Punkte)

1. Nennen Sie den Unterschied zwischen einem *soft* und *hard*-Echtzeitsystem. (1 Punkte)

Ein Soft-Echtzeitsystem sollte, ein hard-Echtzeitsystem muss den Zeitschranken genügen.

2. Wie ist die CPU-Last in *real time* -Systemen definiert? (2 Punkte)

Die CPU-Last H ist der Quotient aus Ausführungsdauer und Ausführungsabstand.

3. Erläutern Sie die Funktionsweise des *Guaranteed Percentage*-Schedulings GPS. (3 Punkte)

Beim *Guaranteed Percentage*-Scheduling wird die Anzahl der *time slices* proportional zur CPU-Last des Jobs gewählt.

Aufgabe 7: Speicher (14 Punkte)

1. Wie ist eine HDD grundsätzlich aufgebaut? Wieso können index-sequentielle Bäume darauf abgebildet werden und warum ist das praktisch? (2 Punkte)

Eine HDD hat mehrere über einander hängende Platten, jeweils einen Lese/ Schreibkopf und einen Arm, an dessen Ende die Köpfe hängen. Die Spuren aller Köpfe auf den Platten bei einer Stellung werden als Zylinder bezeichnet.

Bei einem Zylinder sind schnell ohne eine weitere Armbewegung verschiedene Spuren übereinander liegender Platten und ihre Sektoren lesbar. Die verschiedenen Baumebenen werden deshalb auf Zylinder (Wurzel, oberste Ebene), Spur (mittlere Ebene), und Sektor (Blätter) abgebildet, so dass man schnell von der Wurzel bis zum Blatt den Baum durchsuchen kann.

2. Erläutern Sie den Unterschied zwischen *NextFit* und *Quickfit*. (2 Punkte)

Nextfit: nimmt das erste ausreichend große Speicherstück und führt einen Index der Speicherliste mit, ab dem bei der nächsten Anforderung gesucht wird.

Quickfit: eine Liste pro Objektgröße, so dass die Speicheranforderungen immer exakt bedient werden.

Name:

Sie haben ein Buddy-System, welches 256kB groß ist. Fügen Sie folgende Stücke ein: 80kB, 8kB, 1kB, 19kB, 3kB und 5kB und zeichnen sie die Speicherauslegung nach jedem Schritt. Wieviel Verschnitt ist dabei entstanden? (**10 Punkte**)

0kB	128kB						256kB
80							
80	8						
80	8	1					
80	8	1			19		
80	8	1	3			19	
80	8	1	3	5			19
80	8	1	3	5			19
80	8	1	3	5			19
80	8	1	3	5			19
80	8	1	3	5			19
128kB	8kB	4kB	4kB	8kB	8kB	32kB	64kB

Verschnitt (nicht ausgenutzter Speicherplatz): 48kB + 0kB + 0kB + 13kB + 1kB + 3kB = 65kB

Aufgabe 8: virtuelle Speicheradressen (10 Punkte)

Es wird ein virtuelles Speichersystem mit drei Seitentabellen verwendet. Die Zuordnung der Bits zu den drei Seitentabellen und zum *offset* ist durch folgende Zeichnung gegeben:

Basis			Indx1		Indx2		<i>offset</i>						
12	11	10	9	8	7	6	5	4	3	2	1	0	

Neben der Basis-Seitentabelle sind in der folgenden Abbildung für jede Stufe die benötigten Tabellen angegeben. Man beachte, dass von der Hexadezimalzahl in den Tabellen von *indx2* nur die letzten 10 Bits verwendet werden können (die höchstwertigsten Bits fallen weg).

<table border="1" style="width: 50px; height: 50px; border-collapse: collapse;"> <tr><td>7</td><td>4</td></tr> <tr><td>6</td><td>2</td></tr> <tr><td>5</td><td>1</td></tr> <tr><td>4</td><td>3</td></tr> <tr><td>3</td><td>1</td></tr> <tr><td>2</td><td>4</td></tr> <tr><td>1</td><td>4</td></tr> <tr><td>0</td><td>7</td></tr> </table> <p>Basistabelle</p>	7	4	6	2	5	1	4	3	3	1	2	4	1	4	0	7	<table border="1" style="width: 50px; height: 50px; border-collapse: collapse;"> <tr><td>3</td><td>–</td></tr> <tr><td>2</td><td>5</td></tr> <tr><td>1</td><td>7</td></tr> <tr><td>0</td><td>0</td></tr> </table> <p>indx1-Tabelle1</p>	3	–	2	5	1	7	0	0	<table border="1" style="width: 50px; height: 50px; border-collapse: collapse;"> <tr><td>3</td><td>–</td></tr> <tr><td>2</td><td>A4D</td></tr> <tr><td>1</td><td>1FA</td></tr> <tr><td>0</td><td>804</td></tr> </table> <p>indx2-Tabelle0</p>	3	–	2	A4D	1	1FA	0	804	<table border="1" style="width: 50px; height: 50px; border-collapse: collapse;"> <tr><td>3</td><td>7</td></tr> <tr><td>2</td><td>2</td></tr> <tr><td>1</td><td>7</td></tr> <tr><td>0</td><td>–</td></tr> </table> <p>indx1-Tabelle3</p>	3	7	2	2	1	7	0	–	<table border="1" style="width: 50px; height: 50px; border-collapse: collapse;"> <tr><td>3</td><td>971</td></tr> <tr><td>2</td><td>520</td></tr> <tr><td>1</td><td>–</td></tr> <tr><td>0</td><td>162</td></tr> </table> <p>indx2-Tabelle5</p>	3	971	2	520	1	–	0	162	<table border="1" style="width: 50px; height: 50px; border-collapse: collapse;"> <tr><td>3</td><td>2</td></tr> <tr><td>2</td><td>0</td></tr> <tr><td>1</td><td>0</td></tr> <tr><td>0</td><td>7</td></tr> </table> <p>indx1-Tabelle4</p>	3	2	2	0	1	0	0	7	<table border="1" style="width: 50px; height: 50px; border-collapse: collapse;"> <tr><td>3</td><td>–</td></tr> <tr><td>2</td><td>666</td></tr> <tr><td>1</td><td>FAC</td></tr> <tr><td>0</td><td>BBC</td></tr> </table> <p>indx2-Tabelle7</p>	3	–	2	666	1	FAC	0	BBC
7	4																																																																					
6	2																																																																					
5	1																																																																					
4	3																																																																					
3	1																																																																					
2	4																																																																					
1	4																																																																					
0	7																																																																					
3	–																																																																					
2	5																																																																					
1	7																																																																					
0	0																																																																					
3	–																																																																					
2	A4D																																																																					
1	1FA																																																																					
0	804																																																																					
3	7																																																																					
2	2																																																																					
1	7																																																																					
0	–																																																																					
3	971																																																																					
2	520																																																																					
1	–																																																																					
0	162																																																																					
3	2																																																																					
2	0																																																																					
1	0																																																																					
0	7																																																																					
3	–																																																																					
2	666																																																																					
1	FAC																																																																					
0	BBC																																																																					

Name:

1. Gegeben seien die folgenden zwei virtuellen 16-Bit Speicheradressen in Hexadezimaldarstellung: (8 Punkte)

(1) C6AA (2) 7135

Bestimmen Sie die dazugehörigen physikalische Adressen in Hexadezimaldarstellung. Schreiben Sie den Rechenweg auf.

(1) 1100 0110 1010 1010 → 1 2 2 → A4D → 1010 0100 1101 → 10 0100 1101
→ 1001 0011 0110 A → 936A

Basis			Indx1		Indx2		<i>offset</i>					
0	0	1	1	0	1	0	1	0	1	0	1	0

(2) 0111 0001 0011 0101 → 4 1 0 → BBC → 1011 1011 1100 → 11 1011 1100
→ 1110 1111 0011 5 → EF35

Basis			Indx1		Indx2		<i>offset</i>					
1	0	0	0	1	0	0	1	1	0	1	0	1

2. Erklären Sie was ein *page fault* ist und wozu er benutzt wird. (2 Punkte)

Bei einem *page fault* fehlt eine Seite im Hauptspeicher, die benötigt wird. Er sorgt dafür, dass evtl. eine Seitenersetzung vorgenommen wird, um die fehlende Seite zu laden und ein neuer Tabelleneintrag in die *page table* erfolgt.

Name:

Aufgabe 9: Thrashing (6 Punkte)

1. Wie lautet die Bedingung, damit es zu Thrashing kommt? (2 Punkte)

Ausführungszeit aller Prozesse < Seitenaustauschzeit aller Prozesse

2. Nennen Sie jeweils eine Hardware- und eine Software-Lösung, um Thrashing zu verhindern. (2 Punkte)

Hardware: Swap-Bereiche auf mehrere Platten verteilen.

Software: Inline-Code benutzen, um eine Lokalität des Codes zu erreichen.

3. Was ist der Unterschied zwischen Seitenflattern und Thrashing? (2 Punkte)

Seitenflattern lässt sich beobachten, wenn die Bedingung zum Einlagern einer Seite nach der Erfüllung nicht mehr zutrifft und die Seite deshalb wieder ausgelagert wird. Tritt dann wieder die Einlagerungsbedingung ein, so erfolgt wieder eine Einlagerung und so fort: die Seite „flattert“. Im Gegensatz dazu ist die Thrashing-Bedingung (s.o) von der tatsächlichen Last abhängig und nicht von den Bedingungen eines Algorithmus.

Aufgabe 10: Index-, B- und B*-Bäume (18 Punkte)

Gegeben sei $m = 4$ der Verzweigungsgrad. Es werden folgende Schlüssel in dieser Reihenfolge in einen leeren Baum eingefügt: 3, 5, 7, 15, 18, 20, 22, 33, 30, 50, 1, 8, 2, 10, 11, 17, 13, 23, 25.

Zeichnen sie den dazugehörigen Index-, B- und B*-Baum inklusive jedem Schritt, der den Baum signifikant verändert.

(3+5+6 Punkte)

In den folgenden Zeichnungen sind die Knoten der Bäume so gezeichnet, dass sie nicht den Containergrößen entsprechen, sondern Kontakt mit den Containern haben, mit denen sie verbunden sind. Neu hinzuzufügende Schlüssel sind in den Containern in Klammern gesetzt.

1. Indexbaum

13, 33, 50						
3, 8, 13			18, 23, 33			50
1, 2, 3,	5, 7, 8,	10, 11, 13,	15,17,18	20,22,23,	25,30,33,	50

2. B-Baum hat maximal $m-1 = 3$ Schlüssel pro Container.

3, 5, 7, (15)

5	
3	7,15,18 (20)

5, 15		
3	7	18, 20, 22 (33)

Name:

5, 15, 20			
3	7	18	22, 30, 33 (50)

15				
5		20,30		
1,2,3	7,8,10 (11)	18	22	33,50

15					
5,8			20,30		
1,2,3	7	10,11,13	17,18	22,23,25	33,50

3. **B*-Baum** → Wurzel hat max $2 * \lfloor (2*4 - 2)/3 \rfloor = 2 * \lfloor (8-2)/3 \rfloor = 2*2 = 4$ Schlüssel, pro Container maximal 3 und mindestens 2.

3, 5, 7, 15, (18)

7	
3,5	15, 18, 20 (22)

15	
3,5,7	18, 20, 22 (33)

7, 20		
3, 5	15, 18	22, 30, 33 (50)

7, 22		
1, 3, 5	15, 18, 20 (8)	30, 33, 50

5, 15, 22			
1, 2, 3	7, 8, 10 (11)	18, 20	30, 33, 50

Name:

5, 11, 22			
1, 2, 3	7, 8, 10	15, 18, 20 (17)	30, 33, 50

5, 10, 17, 22				
1, 2, 3	7, 8	11, 15,	18, 20	30, 33, 50

5, 10, 17, 23				
1, 2, 3	7, 8	11, 13, 15	18, 20, 22	(25), 30, 33, 50

5, 10, 17, (22, 30)					
1, 2, 3	7, 8	11, 13, 15	18, 20	23, 25	33, 50

17					
5, 10			22, 30		
1, 2, 3	7, 8	11, 13, 15	18, 20	23, 25	33, 50

4. Angenommen, Sie haben ein Dateisystem, das in Multilevel-Tabellen die Speicherzuordnung organisiert. Es gibt 4 Container der Stufe 0 sowie jeweils einen für die Stufen 1, 2 und 3. Jede Tabelle der Stufen enthält 16 Einträge. Wie viele Speicherblöcke lassen sich damit insgesamt ansprechen? (4 Punkte)

Stufe 0: 4

Stufe 1: 16

Stufe 2: $16 \cdot 16 = 256$

Stufe 3: $16 \cdot 16 \cdot 16 = 4096$

Insgesamt also $4 + 16 + 256 + 4096 = 4372$ Speicherblöcke.

Name:

Aufgabe 11: Überfließen der Behälter (10)

Beim B*-Baum werden durch Überfließen aus einem vollen Behälter mit $m-1$ Schlüsseln und einem nicht-vollem Behälter mit k Schlüsseln zwei halbvolle Behälter zu je $\lfloor (k+m)/2 \rfloor$ und $\lfloor (k+m-1)/2 \rfloor$ Schlüssel erstellt. Zeigen Sie, dass dabei für die Anzahl der Schlüssel gilt:

$$k+m-1 = \lfloor (k+m)/2 \rfloor + \lfloor (k+m-1)/2 \rfloor$$

Dies bedeutet, dass die zwei Behälter tatsächlich wie oben angegeben durch Überfließen entstehen.

Beweis:

Sei p eine ganze Zahl. Angenommen, p ist ohne Rest durch 2 teilbar. Dann ist auch $p-2$ ohne Rest durch 2 teilbar, da mit $p=n \cdot 2$ auch $n \cdot 2 - 2 = (n-1) \cdot 2 = p-2$ ohne Rest teilbar ist. Da dies für beliebige p gilt, muss also immer eine der zwei Zahlen $k+m$ oder $k+m-1$ ohne Rest durch 2 teilbar sein.

Betrachten wir nun die zwei Fälle näher.

a) Sei $p=k+m$ durch zwei teilbar. Dann gilt

$$\lfloor (k+m)/2 \rfloor + \lfloor (k+m-1)/2 \rfloor = (k+m)/2 + \lfloor (k+m)/2 - 1/2 \rfloor = (k+m)/2 + (k+m)/2 - 1 = k+m-1$$

b) Sei $p=k+m-1$ durch zwei teilbar. Dann gilt

$$\lfloor (k+m)/2 \rfloor + \lfloor (k+m-1)/2 \rfloor = \lfloor (k+m-1)/2 + 1/2 \rfloor + (k+m-1)/2 = (k+m-1)/2 + (k+m-1)/2 = k+m-1$$

Damit gilt in allen zwei Fällen und damit für beliebige Zahlen $k+m$ die Gleichung, was zu beweisen war.

Aufgabe 12: Treiber (6 Punkte)

1. Was ist ein Treiber? Warum laufen Geräte-Treiber am besten im Kernel-Mode? (2 Punkte)

Ein Treiber ist eine Software-Schicht, die Funktionen und Anforderungen an ein Gerätemodell in weitere Aufrufe umsetzt. Speziell Gerätetreiber sind sehr hardware-spezifisch und müssen meist die Gerätefunktionen direkt an der Hardware anfordern. Dies geht aber nur im Kernel-Mode.

2. Was ist eine atomare Aktion und warum müssen sie in fehlertoleranten Dateisystemen implementiert sein? (2 Punkte)

Atomare Aktionen werden ganz oder gar nicht durchgeführt. Können Sie nicht komplett ausgeführt werden, gibt es einen Rollback auf den Ursprungszustand. In fehlertoleranten Dateisystemen werden sie gebraucht, damit Inkonsistenz und halb geschriebene Dateien verhindert werden.

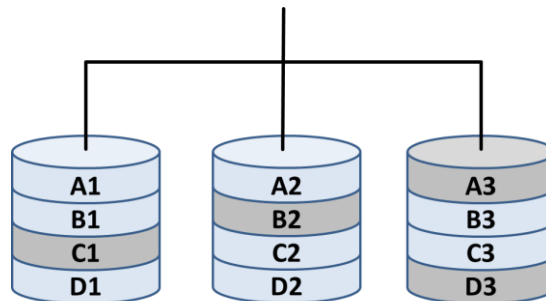
3. Benennen Sie die Aufgabe eines *wear level* Algorithmus. (2 Punkte)

Der Algorithmus sorgt dafür dass die Speicherzellen einer SSD gleichmäßig abgenutzt werden, keine Zellen vollständig ausfallen und damit der maximale Speicherplatz solange wie möglich erhalten bleibt.

Aufgabe 13: RAID (22 Punkte)

1. Wie funktioniert das RAID5 System mit 3 Platten? Verwenden Sie eine Zeichnung mit Erläuterung. (3 Punkte)

Ein RAID5-System besteht aus mehreren Platten, auf die Daten aufgeteilt wird; in diesem Fall in zwei Stücke für zwei Platten. Eine zusätzlich erstellte Fehlertoleranzinformation wird dabei auf eine dritte Platte gespeichert. Für jedes neue Datenstück wird diese Funktion von einer anderen Platte wahrgenommen. Im Falle eines Plattenausfalls wird nun mit Hilfe der Fehlertoleranzinformation, die auf den beiden intakten Platten vorhanden ist, die defekte Platte wieder rekonstruiert. Anhand der Zeichnung soll dies erläutert werden. Angenommen, wir haben Daten A,B,C.. die geschrieben werden sollen, so wird A in A1 und A2 aufgeteilt und in A3 die Fehlertoleranzinformation (ECC) geschrieben. In der Zeichnung ist dies grau schraffiert. Auch die Information B wird in zwei Teile geteilt und ECC aber auf Platte 2 geschrieben; analog auch für C bei dem die ECC auf Platte eins gespeichert wird. Bei D kann man dann wieder bei Platte drei weitermachen, so dass Originaldaten und mit korrespondierenden ECC-Daten gleichmäßig auf allen Platten verteilt sind.



2. Wann ist es sinnvoll RAID0 zu benutzen, wann ist es sinnvoller RAID1 zu benutzen? Benennen Sie Vor- und Nachteile beider Systeme. (2 Punkte)

Beide Systeme lassen sich am besten mit ihren Vorteilen vergleichen. Die Vorteile des einen Systems bedeuten die Nachteile des anderen.

RAID 0	RAID 1
Schnelles, paralleles Lesen und Schreiben durch <i>striping</i>	Schnelles, paralleles Lesen durch <i>mirror</i>
Ausnutzung der vollen Speichermöglichkeiten für Dateien	Nur halber Speicher sichtbar
Kein Fehlertoleranz: Jeder Plattenausfall führt zum Ausfall des ganzen Systems	Fehlertoleranz: Ausfall einer Platte tolerierbar

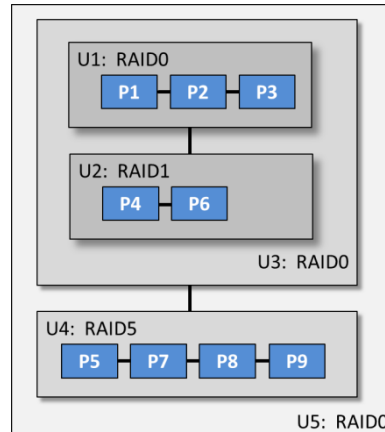
Name:

3. Berechnen Sie die **Ausfallwahrscheinlichkeit** p_s des folgenden Systems: **(15 Punkte)**

Man beachte: P1, P2 und P3 der Untereinheit U1 laufen im RAID0-Modus. P4 und P6 von U2 im RAID1. Bei U4 ist P5, P7, P8 und P9 in RAID5 mit 1 Platte Fehlertoleranz. Einheiten U1 und U2 laufen zusammen in RAID0 als U3 und das gesamte System U5 aus U3 und U4 läuft auch im RAID0-Modus.

Die Ausfallraten dabei sind

Platte	Ausfall-rate	Größe in GB
P1	2%	6
P2	4%	3
P3	7%	4
P4	1%	5
P5	5%	7
P6	3%	5
P7	6%	7
P8	2%	7
P9	4%	7



Lösung:

Bezeichnen wir mit p die Ausfallwahrscheinlichkeiten der Platten und mit u die der Untereinheiten, so ist:

RAID0 aus P1, P2, P3: Untereinheit U1 fällt aus, sobald nur eine Platte ausfällt. Also ist das System nur intakt, wenn alle Platten intakt sind. Dies ist mit der Wahrscheinlichkeit $(1-p_1)(1-p_2)(1-p_3)$ der Fall. Also ist $u_1 = 1 - (1-p_1)(1-p_2)(1-p_3) = 1 - 0.98 \cdot 0.96 \cdot 0.93 = 0.125056$

RAID1 aus P4 und P6: Untereinheit U2 fällt nur aus, wenn beide Platten ausgefallen sind. Dies ist mit $u_2 = p_4 \cdot p_6 = 0.0003$ der Fall.

RAID0 aus U1 und U2: Untereinheit U3 ist nur intakt, wenn U1 und U2 intakt sind. Also ist $u_3 = 1 - (1-u_1)(1-u_2) = 1 - 0.874944 \cdot 0.9997 = 0.125318483$

RAID5 aus P5, P7, P8, P9: U4 ist solange intakt, wie nur eine Einheit ausgefallen ist. Also ist die Wahrscheinlichkeit s für „intakt“ die Summe aller derartigen Fälle

$$\begin{aligned}
 s &= (1-p_5)(1-p_7)(1-p_8)(1-p_9) &&= 0.95 \cdot 0.94 \cdot 0.98 \cdot 0.96 \\
 &+ p_5(1-p_7)(1-p_8)(1-p_9) &&+ 0.05 \cdot 0.94 \cdot 0.98 \cdot 0.96 \\
 &+ (1-p_5)p_7(1-p_8)(1-p_9) &&+ 0.95 \cdot 0.06 \cdot 0.98 \cdot 0.96 \\
 &+ (1-p_5)(1-p_7)p_8(1-p_9) &&+ 0.95 \cdot 0.94 \cdot 0.02 \cdot 0.96 \\
 &+ (1-p_5)(1-p_7)(1-p_8)p_9 &&+ 0.95 \cdot 0.94 \cdot 0.98 \cdot 0.04 \\
 &= 0.84 + 0.044 + 0.0536 + 0.0171 + 0.035 = 0.99013
 \end{aligned}$$

Damit ist $u_4 = 1 - s = 0.0098712$.

RAID0 aus U3 und U4: Das System U5 aus U3 und U4 fällt mit $u_5 = 1 - (1-u_3)(1-u_4) = 1 - 0.8746815 \cdot 0.9901288 = 0.133952656$ aus.

Das Gesamtsystem U5 hat also eine Ausfallwahrscheinlichkeit von ca. **0,134**.

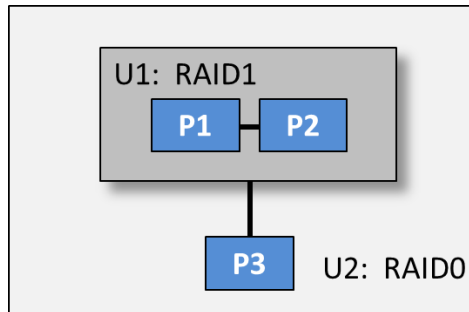
Name:

4. Wie viel Speicher zeigt das Datei-System an? (2 Punkte)

U1 hat 13GB, U2 hat 5GB als Spiegelsystem, zusammen also 18GB in RAID0 für U3. Einheit U4 hat statt 28 GB nur 21GB zur Dateinutzung, da 7GB Fehlerredundanz ebenfalls gespeichert werden muss. Zusammen mit U3 haben sie im RAID0-Modus für U5: $18\text{GB} + 21\text{GB} = 39\text{GB}$ fürs Dateisystem sichtbaren Speicher.

Aufgabe 14: Gemeinsame Ausfälle (15 Punkte)

Berechnen Sie die Ausfallwahrscheinlichkeit des folgenden Systems:

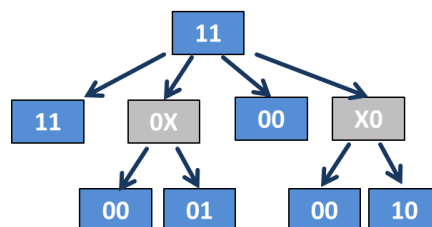


Dabei nehmen Sie an, dass P1 und P2 in einem gemeinsamen Schacht stecken und deshalb (durch Wärmeschäden) eine Platte in $p_0 = 15\%$ der Fällen ausfällt, wenn die andere Platte ausfällt. Die Ausfallwahrscheinlichkeiten von P1, P2 und P3 seien mit $p_1 = 1\%$, $p_2 = 2\%$ und $p_3 = 3\%$ gegeben.

Hinweis: Berechnen Sie zuerst die Wahrscheinlichkeiten für alle auftretenden Fälle im U1-Untersystem. Dabei können Sie von einem Zustand (1,1) (alles intakt) ausgehen und alle möglichen Endzustände betrachten, die mit einer Wahrscheinlichkeit angenommen werden. Beachten Sie dabei, dass die Zustände (X,0) bzw. (0,X) mit einer bedingten Wahrscheinlichkeit in die Zustände (1,0) und (0,0) sowie (0,1) und (0,0) übergehen. Die Addition der Wahrscheinlichkeiten aller Endzustände (die Blätter des Baumes mit der Wurzel (1,1)) ergibt eins. Für die Ausfallwahrscheinlichkeit von U1 müssen Sie also nur die Wahrscheinlichkeiten aller Zustände aufaddieren, bei denen das System U1 ausfällt.

Lösung

Für die Einheit U1 gilt folgender Baum von Zustandsübergängen



Nur im Zustand (0,0) ist das System U1 ausgefallen. Dieser Zustand kann auf drei Wegen erreicht werden, so dass sich $u_1 = \text{Prob}(0,0) = p_1 p_2 + p_1 (1-p_2) p_0 + p_2 (1-p_1) p_0 = 0.01 * 0.02 + 0.01 * 0.98 * 0.15 + 0.02 * 0.99 * 0.15 = 0.00464$ ergibt.

Für das Gesamtsystem ist bei RAID0 nur ein Funktionieren gegeben, wenn beide Systeme, U1 und P3, funktionieren. Das System U2 ist also intakt mit $(1-u_1)(1-p_3)$ und mit der Ausfallwahrscheinlichkeit $u_3 = 1 - (1-u_1)(1-p_3) = 1 - 0.99536 * 0.97 = 0.0345008$.

Name:

Aufgabe 15: Netzwerkdienste (8 Punkte)

1. Nennen Sie die vier bekannten Zugriffssemantiken (2 Punkte)

- Read only - Semantik
- Operationssemantik
- Sitzungssemantik
- Transaktionssemantik

2. Was hat der Unterschied zwischen verbindungsloser und verbindungsorientierter Kommunikation mit zustandsbehafteten und zustandslosen Dateiservern zu tun? (2 Punkte)

Eine verbindungsorientierte Kommunikation findet sich nur bei zustandsbehafteten Servern wieder, weil die Eröffnung und das Schließen eines Kommunikationskanals immer mit einer Zustandsänderung beim Dateiserver verbunden sind. Ein zustandsloser Server kann deshalb nur eine verbindungslose Kommunikation implementieren, mehr nicht.

3. Was ist ein Schattenserver und wie arbeitet er? (2 Punkte)

Auf dem Client eines Schattenservers werden nur die Dateien und Programme gehalten, die vom Nutzer dort zuletzt benötigt wurden. Wird mehr benötigt, so wird es bei Netzwerkverbindung vom Server geladen. Ein Schattenserver spiegelt also immer nur Teile eines Client wider, die beim jeweiligen Client relevant sind; der Client hat die Rolle eines Puffers bzw. Cache. Damit lässt sich eine automatische, zentrale Datenhaltung und Wartung (Backup und Versionskontrolle) etablieren.

4. Wie kann man eine Website im Internet erreichen, wenn der DNS-Server defekt ist? (1 Punkt)

Indem man direkt für die URL die IP-Adresse als Zahl eingibt.

5. Was ist ein Backbone-System? (1 Punkt)

Ein Backbone-System ist die Menge der Rechner eines Unternehmens, die im Netzwerk zusammengeschaltet ohne Peripherie zentrale Aufgaben wahrnehmen. Massenspeicher, Peripheriegeräte, Anzeigegeräte, Backup-Geräte usw. sind nicht direkt an den Backbone-Rechnern, sondern am Netzwerk angeschlossen.