Matrikelnummer:

FB Informatik und Mathematik
Prof. R. Brause

Klausur + Musterlösung

Betriebssysteme WS 2012/2013

07.02.2013

Vorname:	
Nachname:	
Matrikelnummer:	
Geburtsdatum:	
Studiengang:	

Bitte tragen Sie auf jeder Seite Ihre Matrikelnummer ein und überprüfen Sie diese Klausur auf Vollständigkeit (18 Seiten!). Tragen Sie auf dem Deckblatt Ihre Daten in Druckbuchstaben ein.

Verwenden Sie ausschließlich die beigefügten 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 14 Aufgaben mit insgesamt maximal 140 Punkten. Dabei erreichen Sie bereits mit 120 Punkten die Note 1,0.

Die Anzahl der Punkte entspricht ungefähr der Bearbeitungszeit in Minuten. Die maximale Bearbeitungszeit für die gesamte Klausur beträgt 180 Minuten.

Viel Erfolg !!!

1 10	**2** 10	**3** 5	**4** 15	**5** 10	**6** 10	**7** 10	**8** 7	**9** 6	**10** 12	**11** 5	**12** 20	**13** 10	**14** 10	**Σ** 140	**Übungs-punkte**

Matrikelnummer:

Aufgabe 1: Prozesse und virtuelle Maschinen (VM) 10 Punkte

(a) Zeichnen Sie die Schichten von einem MACH Modell und einem UNIX oder WinNT Modell. Benennen Sie die Unterschiede. (**4 Punkte**)

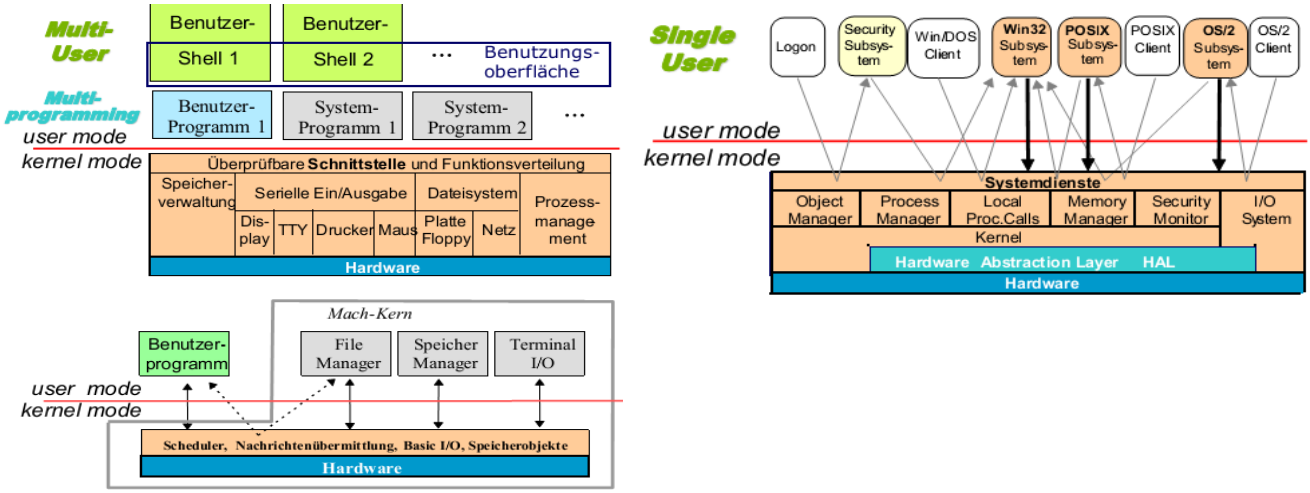
(b) Wie viele und welche Schnittstellen haben eine VM und eine abstrakte Maschine? (**1 Punkt**)

(c) Vervollständigen Sie folgenden Satz: „ Die Zustandsüberführung von Prozessen im Betriebssystem wird vom ..... ausgeführt“. (**1 Punkt**)

(d) Nennen Sie alle Prozesszustände von einem UNIX oder WinNT-System. Eine annotierte Zeichnung genügt. (**4 Punkte**)

Lösungen

(a)

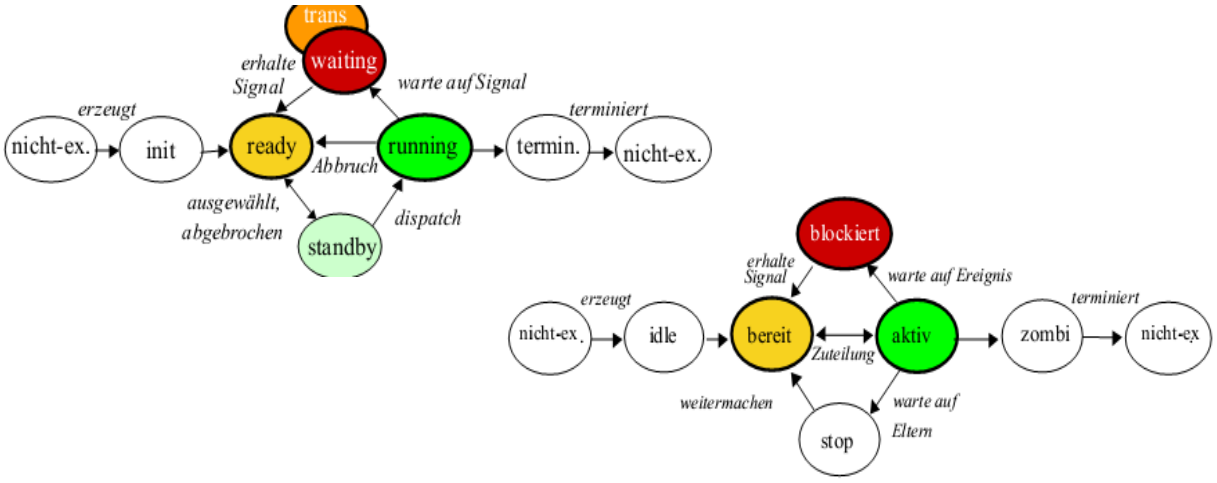


Unterschiede: Filemanager, Speichermanager u.d.gl. laufen bei Mach im *user mode* und nicht im *kernel mode* wie bei Unix und WinNT. Damit ist der Mach-Kern zwar deutlich kleiner, aber wegen der häufigen Systemcall-Aufrufe auch langsamer.

(b) VM hat 2 Schnittstellen (*import, export*), abstrakte Maschine hat nur eine (*export*).

(c) dispatcher

(d)



oder

Matrikelnummer:

Aufgabe 2: Prozess-Scheduling 10 Punkte

Gegeben sind folgende Abhängigkeiten zwischen den Prozessen A1, A2, A3, A4. In Klammern ist die jeweilige Jobdauer angegeben.

A1 >> A2	A3 >> A4
A1 >> A3	A2 >> A4
A1(1)	A2(2)
A3(3)	A4(1)

(a) Zeichnen Sie den zugehörigen Präzedenzgraphen und bestimmen Sie den kritischen Pfad darin sowie seine Länge. (**4 Punkte**)

(b) Zeichnen Sie das jeweils zu *earliest* und *latest scheduling* gehörige Gantt-Diagramm, (**4 Punkte**)

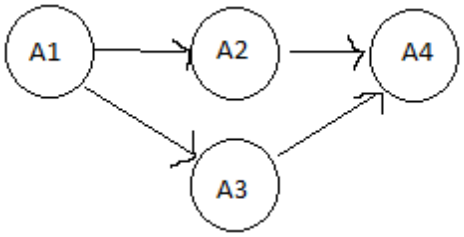
(c) Welcher der Prozesse verhungert? Begründen Sie ihre Antwort. (**1 Punkt**)

(d) Kann hier eine Verklemmung entstehen? Begründen Sie Ihre Antwort. (**1 Punkt**)

Hinweis: Wenn keine Angaben zur Prozessorzahl vorhanden ist, haben Sie beliebig viele zur Verfügung. Nehmen Sie auch an, dass die Prozessoren extra für diese Operationen reserviert sind.

Lösungen

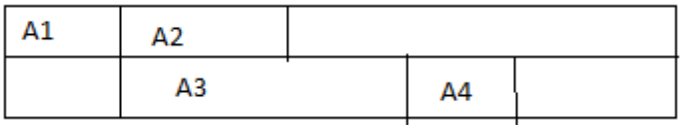
(a)



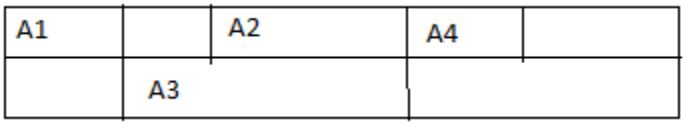
kritischer Pfad = 1+3+1 = 5 Zeiteinheiten über A1 nach A3 nach A4

(b)

earliest:



latest:



(c) Keiner, da der Ablaufplan feststeht und jede Prozessdauer vorgegeben ist.

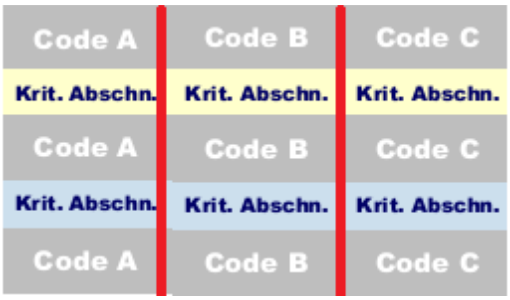
(d) Nein, keiner der Prozesse kann sich verklemmen, da ein Ablaufplan mit festen Jobdauern vorliegt und keine Nebenbedingungen über Betriebsmittel existiert.

Matrikelnummer:

Aufgabe 3: Synchronisation 5 Punkte

(a) Erklären Sie was ein Semaphor ist, wann man ihn benutzt und benennen Sie seine Operationen. (**2 Punkte**)

(b) Gegeben sei folgende Grafik:



Prozess1 Prozess2 Prozess3

Wie viele Semaphore werden benötigt, um einen fehlerfreien Ablauf zu garantieren, und warum? (**3 Punkte**)

Hinweis: Alle gelben Abschnitte korrespondieren zueinander, analog gilt dies für die blauen Abschnitte.

Lösungen

(a) Ein Semaphor ist eine Datenstruktur, die dafür sorgt dass nur eine bestimmte Anzahl an Prozessen / Threads auf einen kritischen Abschnitt zugreifen können. Operationen sind: Passieren P(s) und Verlassen V(s). Sie dienen zum Blockieren und reaktivieren von Prozessen, die kontrolliert unter gegenseitigem Ausschluss den kritischen Abschnitt betreten dürfen.

(b) Es werden 2 Semaphore gebraucht, da es nur 2 kritische Abschnitte gibt, die nicht zueinander korrespondieren.

Matrikelnummer:

Aufgabe 4: Verklemmungen 15 Punkte

(a) Sie und ihr Freund leihen sich bei der örtlichen Bibliothek jeweils ein Buch aus. Nun stellen Sie beide fest, dass sie auch noch das Buch des jeweils anderen dazu benötigen und reservieren es. Auf Nachfrage bei der Bibliothek ist jedoch kein weiteres Exemplar beschaffbar. Liegt hier ein Deadlock vor? Wenn ja, warum? Wenn nein, warum nicht? (**3 Punkte**)

(b) Lösen Sie mit Hilfe des Banker-Algorithmus folgenden Fall: Es gibt vier Prozesse im System, die einige Betriebsmittel haben und weitere benötigen.

Zusätzl. benötigte Betriebsmittel

	B1	B2	B3
P1	4	1	3
P2	5	4	4
P3	2	0	1
P4	5	3	4

Schon belegte Betriebsmittel

P1	1	1	1
P2	0	1	0
P3	2	1	2
P4	1	0	0

Frei	2	0	1

Falls eine Verklemmung entsteht, beseitigen Sie diese. Schreiben Sie den ganzen Rechenweg auf. Zeichnen Sie zum Abschluss noch den zum Anfangszustand gehörigen Betriebsmittelgraph. (**10 Punkte**)

(c) Nennen Sie mindestens 2 Möglichkeiten, wie man eine Verklemmung unmöglich machen kann. (**2 Punkte**)

Matrikelnummer:

Lösungen

(a) Nein es liegt kein Deadlock vor, da beide einen Timeout haben (das Buch muss nach einer Frist zurück gegeben werden) und damit Bedingung 3 nicht gilt.

(b) 2 0 1 + (P3) 2 1 2 = 4 1 3

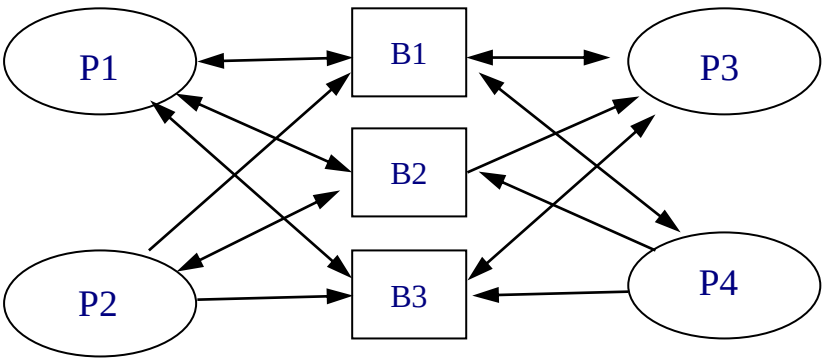
4 1 3 + (P1) 1 1 1 = 5 2 4

5 2 4 - > Verklemmung, P2 wird beendet →5 2 4 + 0 1 0 = 5 3 4

5 3 4 + (P4) 1 0 0 = 6 3 4

(c) Eine oder mehrere dieser vier Bedingungen verhindern:

1. Beschränkte Belegung (mutal exclusion)
2. Zusätzliche Belegung (hold-and-wait)
3. Keine vorzeitige Rückgabe (no preemption)
4. Gegenseitiges Warten (circular wait)



Zeichenerklärung:

Pfeil zum BM: BM wird benötigt

Pfeil zum Prozess: BM belegt vom Prozess

Matrikelnummer:

Aufgabe 5: Speicher 10 Punkte

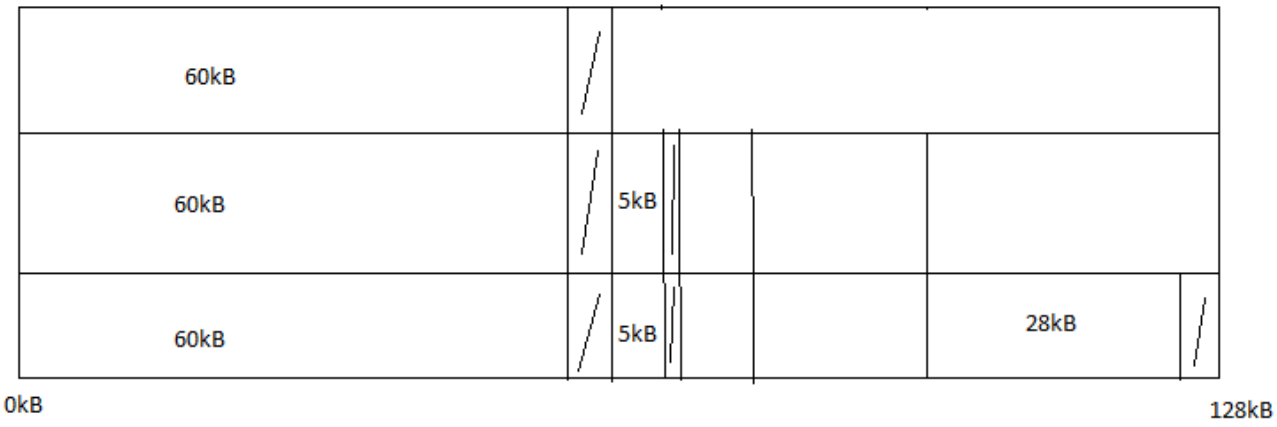
1. Bestimmen Sie die *worst case*-Laufzeit der Speicherverwaltungsstrategie „NextFit“. Ihre geordnete Liste mit freiem Speicher hat n Einträge, wobei n eine natürliche Zahl größer 0 ist. Beachten Sie, dass die Anzahl der durchsuchten Speicherstücke höchstens n beträgt und mindestens ein geeignetes in der Liste zu finden ist. Begründen Sie ihre Antwort. (**2 Punkte**)
2. Gegeben sei ein Buddy-System. Der freie Speicher sei anfangs 128kB groß. Fügen Sie zuerst 60kB, dann 5kB und zum Schluss 28kB ein.
   a. Zeichnen Sie, wie der Speicher bei jedem Schritt gefüllt wird, **(6 Punkte)**
   b. und bestimmen Sie den restlichen freien Speicher exklusive Verschnitt. (**2 Punkte**)

Lösungen

5.1) Die Laufzeit bei einer Suche beträgt $O(n)$ da die Liste nur einmal bis maximal zum n-ten Element durchlaufen werden muss.

5.2

(a)



(b) Freier Speicher: 128 – 64 – 8 – 32 = 24 kB

64kB 80kB 96kB

Matrikelnummer:

Aufgabe 6: Virtuelle und phys. Speicheradressen (10 Punkte)

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

Basis			Indx1		Indx2		*offset*					
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 der mehrstufigen Adresserzeugung die benötigten Tabellen angegeben. Man beachte, dass von der Hexadezimalzahl in den Tabellen von indx2 nur die letzten 7 Bits verwendet werden können (das höchstwertigste Bit fällt weg).

8	1B3B
7	2C87
6	2BD5
5	0ACE
4	1BDA
3	0ACE
2	2C87
1	5BD5
0	7BA3

Basistabelle

3	–
2	9378
1	4142
0	B035

0ACE

3	–
2	4D
1	FA
0	04

B035

3	4142
2	864A
1	B035
0	–

1BDA

3	71
2	20
1	–
0	62

9378

3	9378
2	–
1	B035
0	4142

2C87

3	–
2	33
1	AC
0	DC

4142

Gegeben seien die folgenden zwei virtuellen 16 Bit Speicheradressen in Hexadezimaldarstellung:

(a) C85C **(5 Punkte)** (b) 7464 **(5 Punkte)**

Bestimmen Sie die dazugehörigen physikalischen 16 Bit-Adressen in Hexadezimaldarstellung. Schreiben Sie den Rechenweg auf.

Lösungen *je 3 Pkte Lösungsweg+2 Pkte korrektes Ergebnis*

(a) C85C = 110 | 010 | 00 | 01 | 011100

→Basisindex=2, Index1-Tabellenadresse=2C87 mit Index1=0, Index2-Tabellenadresse=4142 mit Index2=1. Also header = AC = 1010 1100, davon 7 Bit = 0 1011 00 mit angefügtem *offset* = 01 1100 → 0 1011 0001 1100

→ phys. Adresse = 0B1C

(b) 011 101 00 01 100100 →1111 1010 (FA) 100100 →111 1010 100100 →1EA4

Matrikelnummer:

Aufgabe 7: Seitenersetzung 10 Punkte

Gegeben seien folgende Seiten im Hauptspeicher.

Name der Seite	Zuletzt benutzt vor	Zuletzt hinzugefügt vor	R	M
Page 3	5 sek	5 sek	1	1
Page 8	3 sek	3 sek	0	0
Page 7	8 sek	10 sek	0	1
Page 1	9 sek	9 sek	0	1
Page 6	4 sek	4 sek	1	0
Page 4	2 sek	2 sek	1	1

(a) Welche Seite wird bei FIFO zuerst aus dem Hauptspeicher entfernt? (**2 Punkte**)

(b) Welche Seite wird bei LIFO zuerst aus dem Hauptspeicher entfernt? (**2 Punkte**)

(c) Welche Seite wird bei NRU zuerst aus dem Hauptspeicher entfernt? (**2 Punkte**)

(d) Welche Seite wird bei LRU zuerst aus dem Hauptspeicher entfernt? (**2 Punkte**)

(e) Was ist die optimale Seitenersetzungsstrategie? Ist sie zu realisieren, wenn man zur Compilezeit eine Liste aller benötigten Seiten aufstellt? Gehen Sie davon aus, dass der Hauptspeicher generell zu klein ist, um ein ganzes Programm zu fassen. (**2 Punkte**)

Lösungen

(a) Page 7

(b) Page 4

(c) Page 8

(d) Page 1

(e) Bei der optimalen Seitenersetzungsstrategie wird immer die Seite entfernt, die man am längsten nicht braucht. Nein, sie kann nicht zu realisiert werden, denn man weiß i.A. vorher nicht, welches Ergebnis bei Entscheidungen während des Programmablaufs getroffen und welche Seiten deshalb benötigt werden, oder wie lange sie nicht benötigt werden.

Matrikelnummer:

Aufgabe 8: Anti-Thrashing 7 Punkte

(a) Benennen Sie die 4 gängigsten Anti-Thrashing-Maßnahmen und erklären Sie kurz, wie diese funktionieren. (**4 Punkte**)

(b) Was verbirgt sich hinter dem Nutzungsgradmodell? Erklären Sie es kurz. Sie dürfen dazu gerne eine Zeichnung verwenden. (**3 Punkte**)

Lösungen

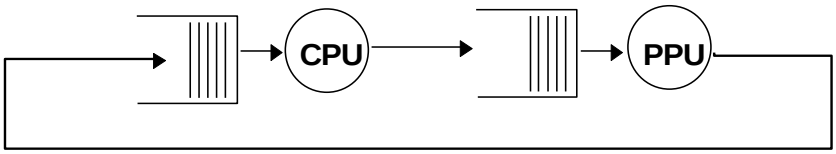
(a) (1) Hardware-Maßnahmen: Große Seiten, mehrere parallele *swap*-Festplatten.

(2) Programmier-Maßnahmen: lokaler Code (lokale Kopie benötigter Methoden), lokale Adressreferenzen (Methoden bei den Daten, die sie bearbeiten), kleinere nötige Seitenanzahl

(3) Working Set-Modell: nur so viele Prozesse, dass der Speicher für alle *working sets* fester Größe reicht.

(4) Page Faul Frequency-Modell (PFF): dynamisch Prozesse je nach *page fault* Aktivität über/unter der Schwelle stilllegen oder aktivieren.

(b) Adaptive Regelung der Auslastung basierend auf Messungen von der Gesamtbelastung L(n,t) aus Prozessor und Seitenauslagerung.



Strategie: Abfallen von L →Prozesse aktivieren. Ansteigen von L →Prozesse deaktivieren.

Matrikelnummer:

Aufgabe 9: Dateisysteme 6 Punkte

a) Erklären Sie den Unterschied zwischen *hard links* und *symbolic links*. (**2 Punkte**)

b) Welche Dateioperation kann man als Semaphor verwenden? (**2 Punkte**)

c) Ist die Operation *open*/*close file* grundsätzlich notwendig? Kennen Sie andere Konzepte, um ohne *open*/*close* auf Daten zuzugreifen? (**2 Punkte**)

Lösungen

(a) *hard links* sind *device*-intern (nur gültig für ein Gerät) und die *symbolic links* sind *device*-übergreifend (gültig über Geräte- und Netzgrenzen hinweg).

(b) Die *create_file()*- Operation. Auch eine *lock_file()*-Operation (falls existent) ist brauchbar.

(c) Nein ist sie nicht. Es gibt noch die Option der verbindungslosen Kommunikation, bei der bei jedem *read/write* alle Dateiangaben (Name, Blocknr., etc) mit übergeben werden.

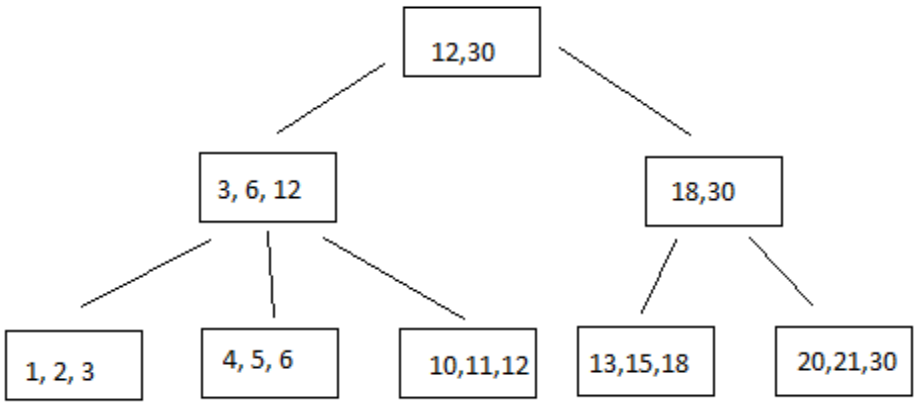
Matrikelnummer:

Aufgabe 10: Index- und B-Bäume 12 Punkte

(a) Seien maximal 3 Schlüssel pro Container möglich. Gegeben sind folgende Schlüssel: 12, 30, 6, 18, 3, 2, 5, 1, 4, 20, 15, 11, 10, 13, 21. Zeichnen Sie den dazugehörigen **Indexbaum**. (**4 Punkte**)

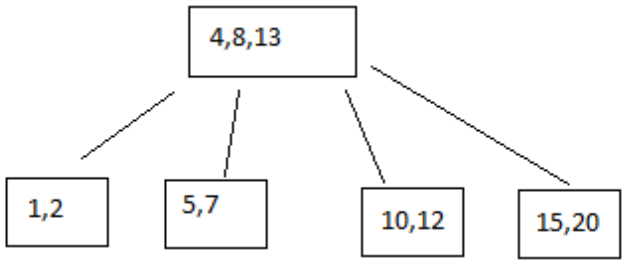
(b) Seien maximal 4 Schlüssel pro Container möglich. Gegeben sind folgende Schlüssel: 7, 4, 8, 10, 12, 15, 2, 5, 20, 13, 1. Zeichnen Sie den dazugehörigen **B-Baum** und geben Sie alle wichtigen Zwischenschritte an. (**8 Punkte**)

Lösungen



(a) Indexbaum

(b) B-Baum Endstand:



Matrikelnummer:

Matrikelnummer:

Aufgabe 11: Treiber 5 Punkte

(a) Was ist I/O-*mapping* und wozu dient es? (**2 Punkte**)

(b) In welchem Systemmodus laufen Treiber? Warum? (**1 Punkt**)

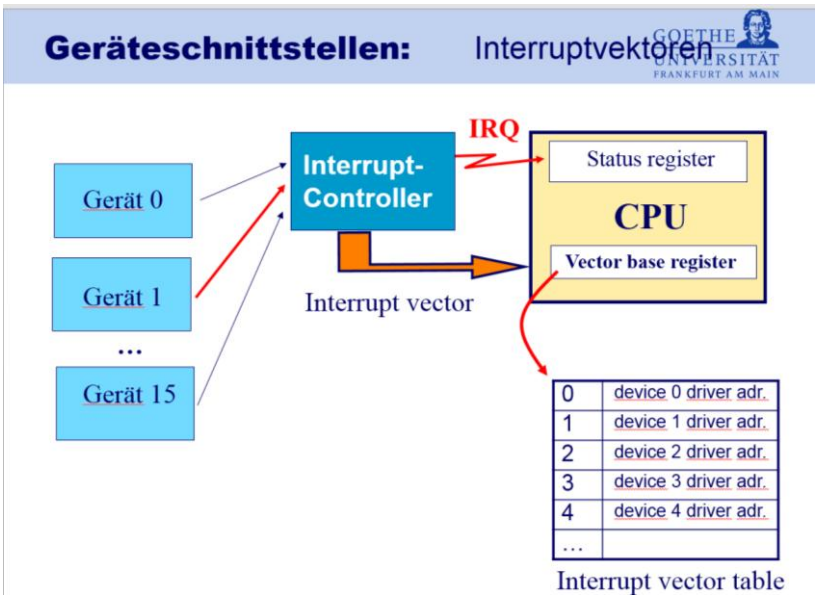
(c) Wie wird ein Treiber bei einem Interrupt eines Geräts ausgewählt? (**2 Punkte**)

Lösungen

(a) Beim I/O-mapping werden die Geräteregister wie Speicher an bestimmten Adressen im Adressraum angesprochen. Damit wird die Benutzung von speziellen, hardwareabhängigen I/O-Maschinenbefehlen vermieden.

(b) Kernel-Mode, um möglichst rasch und privilegiert auf alle Hardwareadressen und Daten zugreifen zu können.

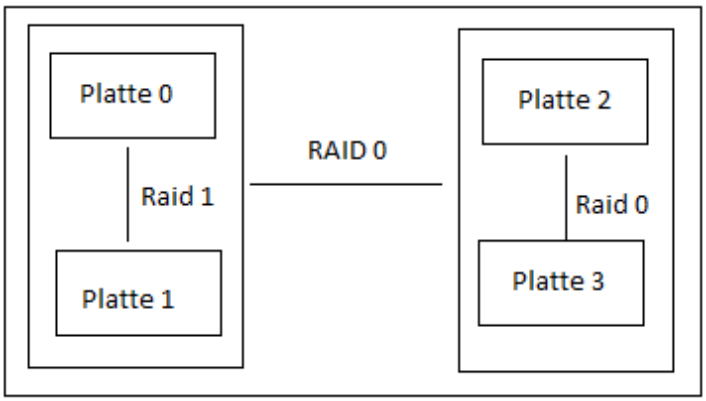
(c) Der Interrupt liefert durch Hardware die Interruptnummer (Interruptvektor), der einem Index in einer Interrupt-Tabelle entspricht. In diesem Tabelleneintrag ist die Treiberadresse gespeichert, die dann benutzt wird, um den Treiber anzuspringen.



Matrikelnummer:

Aufgabe 12: RAID-Systeme 20 Punkte

Gegeben ist folgendes RAID-System:



sowie die Information über die einzelnen Ausfallwahrscheinlichkeiten pro Zeitraum

Name	Ausfallwahr-scheinlichkeit	Kapazität
Platte 0	1,00%	5TB
Platte 1	2,00%	5TB
Platte 2	1,50%	8TB
Platte 3	0,50%	3TB

(a) Berechnen Sie genau, welches der beiden folgenden Systeme zuverlässiger läuft: Eine einzelne Platte mit einer Ausfallwahrscheinlichkeit von 2,0%, oder das obige RAID-System? **(10 Punkte)**

(b) Was bietet mehr effektiv nutzbaren Speicher? Das RAID-System oder eine einzelne Festplatte von 20TB? (**5 Punkte**)

(c) Erklären Sie wie ein RAID2-System funktioniert (Normalfunktion und Fehlerausgleich). Eine aussagekräftige Zeichnung mit Annotationen genügt. (**5 Punkte**)

Matrikelnummer:

Lösungen

(a) Ausfall von Subsystem P01 mit RAID1 nur, wenn beide gespiegelten Platten ausfallen. Dies ist mit der Wahrscheinlichkeit P01defekt = 0.01*0.02 = 0.0002 der Fall.

Ausfall von Subsystem P23 mit RAID0, sobald eine der beiden oder beide Platten ausfallen; es müssen beide Platten funktionieren. Eine Platte funktioniert mit (1-1,5%) bzw. (1-0,5%), so dass beide mit (1-1,5%)*(1-0,5%) funktionieren. Eine oder beide fallen also mit P23defekt = 1 - ((1 – (0.015)) * (1 – (0.005))) = 0,019925 aus.

Ausfall von Gesamtsystem P0123, wenn eines oder beide Subsysteme ausfallen, ist also = 1- (1-P01defekt)*(1-P23defekt) = 1 – ((1 – 0,0002)*(1 – 0,019925)) = 0,020121015 = 2,012% und damit leicht mehr als die zuverlässigere Einzelplatte,

Die Ausfallwahrscheinlichkeit des RAID ist somit etwas größer als die der Festplatte. (Allerdings sind die RAIDS auch deutlich billiger als eine solche zuverlässige Festplatte).

(b) Effektiver Speicher des RAIDs : 5TB + 8TB + 3TB = 16TB. Die einzelne Festplatte ist also um 4 TB größer. (Allerdings sind die RAIDS deutlich billiger als eine solche große Festplatte)

(c) *Normalfunktion*: Ein RAID2-System speichert die Datenworte auf n verschiedenen Festplatten und bildet außerdem Paritätsbits über die jeweils n Worte. Diese werden auf p extra Festplatten gesichert. *Fehlerfall*: Sollte eine der $n+p$ Platten ausfallen, so kann mit Hilfe der Paritätsbits die fehlende Information der ausgefallenen Platte wiederhergestellt werden.

Matrikelnummer:

Aufgabe 13: Dienste/Arbeitsverteilung 10 Punkte

(a) Nennen Sie den Unterschied zwischen einem Dienst, einem Dämon, einem Prozess und einem Thread. Berücksichtigen Sie dabei die Fragen: Ist ein Dienst ein Dämon? Aus wie vielen Prozessen besteht ein Dämon? (**4 Punkte**)

(b) Zeichnen und benennen Sie die Betriebssystemteile eines NC-Computers. Welche Teile fehlen, verglichen zu denen eines Standard Desktop PCs? (**6 Punkte**)

Lösungen

(a) Ein Dienst kann aus einem oder mehreren Prozessen/Dämonen bestehen. Ein Dämon ist ein Hintergrundprozess. In einem Prozess können ein oder mehrere Threads laufen, die den gleichen (Prozess-)kontext nutzen, also denselben Speicher und dieselben Dateien.

(b) Ein NC-Computer ist als „dummer“ Bildschirm aufzufassen, der nur als Cache für Programme und Daten gedacht sind, die von einem zentralen Server stammen. CPU, Hauptspeicher, Tastatur, Bildschirm und Maus sind vorhanden, aber alle anderen Funktionen nicht.

Es fehlen insbesondere die Dateiverwaltung, die Nutzerrechteverwaltung, alle Ein- und Ausgabegeräte wie Massenspeicher, Drucker etc. Dies sind gesonderte Geräte, die vom Server bedient werden und nicht vom NC.

Matrikelnummer:

Aufgabe 14: Sicherheit 10 Punkte

(a) Gegeben sind 4 Benutzer und 8 Objekte (Dateien, Prozesse). Wie viele ACL-Listen werden benötigt? Wie viele Einträge hat maximal jede Liste? (**3 Punkte**)

(b) Woraus besteht der *buffer overflow*-Angriff? Wie ist ein solcher zu verhindern? (**4 Punkte**)

(c) Nennen Sie den Unterschied zwischen einem *Rootkit* und einem Virus. (**3 Punkte**)

Lösungen

(a) Es werden 8 ACL-Listen mit maximal jeweils 4 Einträgen benötigt.

(b) Ein *buffer overflow*-Angriff versucht bei der Argumentübergabe den Pufferplatz über seine Grenzen hinaus zu füllen und dabei wichtige Adressen (Startadresse, Rücksprungadresse) zu überschreiben und so die Programmausführung in den eigenen Schadcode zu lenken. Man kann einen solchen Angriff verhindern, indem man durch Prüfen der Argumentgröße nicht zulässt, dass der Speicher überlaufen kann.

(c) Ein Rootkit versteckt nur die Dateien und Prozesse des Virus vor dem Benutzer oder dem Antivirenprogramm, der Virus selbst führt den Schaden aus. Ein Rootkit ist also eine „Tarnkappe“ eines Schädlings und nicht der Schädling selbst.