Hochschule Ravensburg-Weingarten
Schriftlich Prüfung Systemsoftware

Prof. Dr. M. Zeller

Datum, Zeit	5. Juli 2006, 10:30 – 12:00 Uhr (90 min)
Aufgabenblätter	14 Seiten (einschl. Deckblatt)
erreichbare Punktzahl	51
zugelassene Hilfsmittel	A (s. Prüfungsplan)

Studiengang	Prf. Nr.	Raum
AI	1825	C004
WI	4021	C004

Name: _____ Matrikelnummer: _____

Hinweise:

- Schreiben Sie bitte Name und Matrikelnummer auf jedes Aufgabenblatt.
- Schreiben Sie Ihre Lösung zu den Aufgaben auf den freien Platz, direkt anschließend an die Fragestellungen. Wenn Sie zusätzliche Blätter verwenden, so schreiben Sie bitte Name und Matrikelnummer auf jedes Blatt.
- Schreiben Sie lesbar!

Falls Sie es wünschen, dass Ihr Prüfungsergebnis auf einer Liste mit Matrikelnummern und Zensuren ausgehängt bzw. per Internet veröffentlicht wird, unterschreiben Sie bitte folgende Erklärung.

Ich bin damit einverstanden, dass mein Klausurergebnis auf diese Weise veröffentlicht wird.

Unterschrift: _____

Bitte haben Sie dafür Verständnis, dass aus Gründen das Datenschutzes keine telefonischen Auskünfte gegeben werden können.

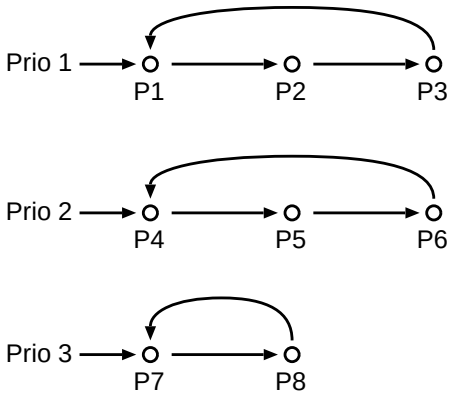
Vom Prüfer auszufüllen:

Aufgabe	1	2	3	4	5	Summe
Max. Punkte	7	7	11	11	15	51
Punkte						

Aufgabe 1 Scheduling

Ein Betriebssystem verwendet preemptive Multitasking mit einer Kombination aus Round-Robin und prioritätsbasiertem Scheduling. Es gibt drei Prioritätsstufen: 1, 2 und 3, wobei die Stufe 1 die höchste Priorität darstellt und die Stufe 3 die niedrigste. Der Scheduler wird aktiv, wenn eine Zeitscheibe abläuft oder wenn ein Prozess blockiert wird. Das System verwaltet 8 Prozesse (P1 ... P8).

Ein Prozess ist rechnend (RE), einige Prozesse sind bereit (BR) einige sind blockiert (BL).



1.1 (7 Punkte)

Die folgende Tabelle soll die Zustände der Prozesse zu verschiedenen Zeitpunkten darstellen. Der Zustand zum Zeitpunkt t_0 ist gegeben. Es treten nun der Reihe nach Ereignisse auf. Mit dem Ausdruck "Prozess P7 wird deblockiert" ist gemeint, dass die Ursache für die Blockade aufgehoben ist. Z. B. weil eine Aus- oder Eingabe des Prozess beendet wurde.

Ergänzen Sie die Tabelle gemäß den auftretenden Ereignissen (s. u.).

	t_0	t_1	t_2	t_3	t_4	t_5	t_6	t_7
P1	BL							
P2	BL							
P3	BL							
P4	BR							
P5	RE							
P6	BL							
P7	BL							
P8	BR							

Zeitpunkt: Ereignis

t_1: Prozess P7 wird deblockiert
t_2: Prozess P6 wird deblockiert
t_3: Prozess P5 wird blockiert
t_4: Ein Zeitscheibe für das preemptive Multitasking läuft ab
t_5: Prozess P1 wird deblockiert
t_6: Prozess P4 wird blockiert
t_7: Ein Zeitscheibe für das preemptive Multitasking läuft ab

	t_0	t_1	t_2	t_3	t_4	t_5	t_6	t_7
P1	BL	BL	BL	BL	BL	BR	RE	RE
P2	BL	BL	BL	BL	BL	BL	BL	BL
P3	BL	BL	BL	BL	BL	BL	BL	BL
P4	BR	BR	BR	BR	RE	RE	BL	BL
P5	RE	RE	RE	BL	BL	BL	BL	BL
P6	BL	BL	BR	RE	BR	BR	BR	BR
P7	BL	BR	BR	BR	BR	BR	BR	BR
P8	BR	BR	BR	BR	BR	BR	BR	BR

Aufgabe 2 Ersetzungsstrategien

Ein Betriebssystem verwendet Paging zur Speicherverwaltung. Als Seitenersetzungsstrategie verwendet das System den Clock-Algorithmus. Der Hauptspeicher ist in 8 Kacheln unterteilt, die Prozesse, die gerade laufen, verwenden die Seiten 0 ... 16.

2.1 (7 Punkte)

Zum Zeitpunkt t_0 sind die Seiten 1 ... 8 im Hauptspeicher eingelagert. In Klammer steht das "R-Bit", das angibt, ob auf die Seite seit dem letzten Durchlauf lesend zugegriffen wurde. Schreibende Zugriffe werden hier nicht berücksichtigt. Der Zeiger des Clock-Algorithmus ($*$) steht auf der Kachel 4.

Die Prozesse greifen nun (lesend) wie folgt auf den Hauptspeicher zu:

Zeitpunkt	t_1	t_2	t_3	t_4	t_5	t_6	t_7
Seite	15	7	12	6	15	7	9

Welche Seiten werden ersetzt? Ergänzen Sie die leeren Felder der Tabelle.

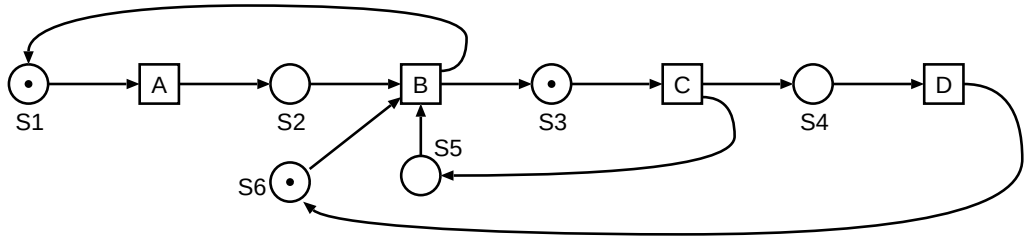
	t_0	t_1	t_2	t_3	t_4	t_5	t_6	t_7
K0	1 (0)							
K1	2 (1)							
K2	3 (1)							
K3	4 (0)							
K4	5 (1) *							
K5	6 (0)							
K6	7 (0)							
K7	8 (1)							

	t_0	t_1	t_2	t_3	t_4	t_5	t_6	t_7
K0	1 (0)	1 (0)	1 (0)	12 (1)	12 (1)	12 (1)	12 (1)	12 (1)
K1	2 (1)	2 (1)	2 (1)	2 (1) *	2 (0)	2 (0)	2 (0)	2 (0)
K2	3 (1)	3 (1)	3 (1)	3 (1)	3 (0)	3 (0)	3 (0)	3 (0)
K3	4 (0)	4 (0)	4 (0)	4 (0)	6 (1)	6 (1)	6 (1)	6 (1)
K4	5 (1) *	5 (0)	5 (0)	5 (0)	5 (0) *	5 (0) *	5 (0) *	9 (1)
K5	6 (0)	15 (1)	15 (1)	15 (1)	15 (1)	15 (1)	15 (1)	15 (1) *
K6	7 (0)	7 (0) *	7 (1) *	7 (0)	7 (0)	7 (0)	7 (1)	7 (1)
K7	8 (1)	8 (1)	8 (1)	8 (0)	8 (0)	8 (0)	8 (0)	8 (0)

Aufgabe 3 Synchronisation

Folgendes Petri-Netz zeigt die Synchronisation von 3 Prozessen. Die Transitionen A und B sind jeweils eigenen Prozesse. Die Transitionen C und D bilden zusammen einen Prozess.

Die Transition A gehört zu Prozess P_A. Die Transition B gehört zu Prozess P_B Die Transitionen C und D gehören zu Prozess P_{CD}.



3.1 (2 Punkte)

Welche Stellen müssen Sie als Semaphor realisieren, um die drei Prozesse gemäß dem obigen Petri-Netz zu synchronisieren?

S1, S2, S3, S5, S6

3.2 (4 Punkte)

Geben Sie den Quell-Code für die Prozesse P_A, P_B und P_{CD}. an. Sie können dazu Pseudo-Pascal verwenden (s. Skript von Frau Keller) oder (Pseudo-)Java.

```
Prozess P_A{

}
```

```
Prozess P_B{

}
```

```
Prozess P_CD{




}
```

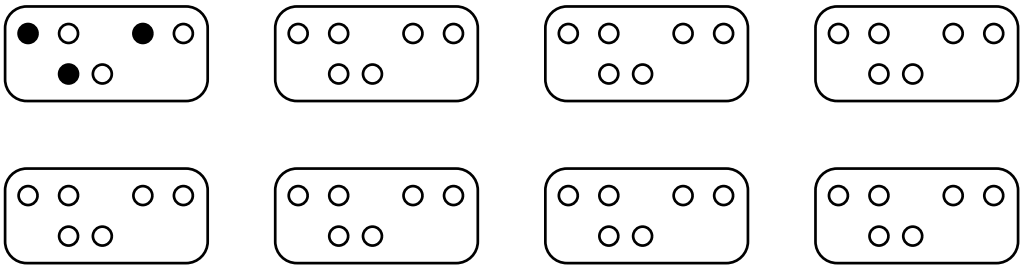
```
Prozess P_A{
 while(true){
    S1.p();
    A();
    S2.v();
  }
}
```

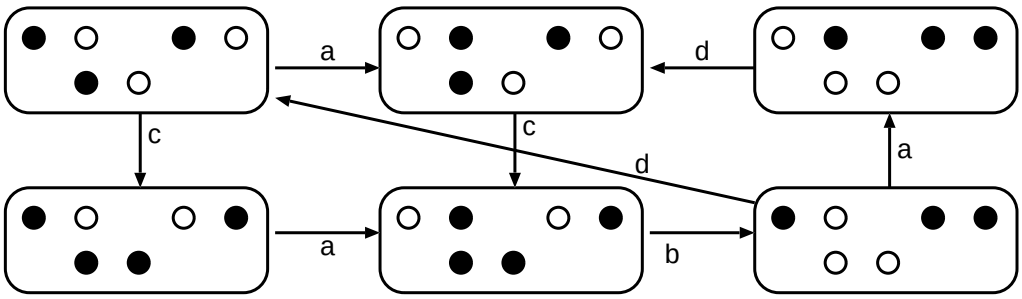
```
Prozess P_B{
 while(true){
    S2.p();
    S5.p();
    S6.p();
    B();
    S1.v();
    S3.v();
  }
}
```

```
Prozess P_CD{
 while(true){
    S3.p();
    C();
    S5.v();
    D();
    S6.v();
  }
}
```

3.3 (3 Punkte)

Zeichnen Sie den Ereignisgrafen des Petri-Netzes. Sie können die Vorlage unten verwenden oder eine eigene Skizze anfertigen. Hinweis: Die Vorlage enthält mehr Zustände als nötig.





3.4 (2 Punkte)

Kann das System in einen Deadlock geraten? Wie können Sie diese Frage anhand des Ereignisgrafen entscheiden?

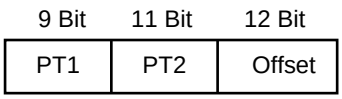
---

1 Punkt: Nein 1 Punkt: Aus jedem Zustand führt mindestens eine Kante heraus.

Aufgabe 4 Virtueller Speicher

Ein Betriebssystem verwendet Paging, um für die verschiedenen Prozesse einen virtuellen Hauptspeicher zu realisieren. Der virtuelle Speicher wird auf 12 MB Hauptspeicher und einen Teil der Festplatte abgebildet. Die Gesamtlänge einer Adresse beträgt 32 Bit.

Das Betriebssystem verwendet eine zweistufige Seitentabelle. Die Länge der ersten Seitenadresse (PT1) beträgt 9 Bit; die Länge der zweiten Seitenadresse (PT2) beträgt 11 Bit; die Länge des Offsets beträgt 12 Bit.



Allg. Hinweis: Schreiben immer den Rechenweg auf, z. B. "Größe des Speicherbereich XY dividiert durch Anzahl Z".

4.1 (7 Punkte)

Ein Prozess belegt folgende Adressbereiche:

Prog. Teil	Adressbereich	Größe in Byte
Text-Segment	0 - 126 000	126 001
Heap-Segment	126 001 - 408 388 260	408 262 260
Stack Segment	2 145 095 485 - 2 147 483 648	2 388 164

Wie viele Einträge hat die Seitentabelle erster Stufe?

$2^9 = 512$ Einträge

(1 Punkt) Wie viele Einträge hat eine Seitentabelle zweiter Stufe?

$2^{11} = 2048$ Einträge

(1 Punkt) Wie groß (in Kilobyte, KB) ist ist eine Seite, wie groß ist eine Kachel?

$2^{12} = 4096$ Einträge

(1 Punkt) Wie viele Seiten belegt das Stack-Segment?

$2\,388\,164/4096 = 583.047851562 \rightarrow 584$ Seiten

(1 Punkt) Wie viele Seitentabellen zweiter Stufe werden für das Stack-Segment benötigt?

Eine Seite

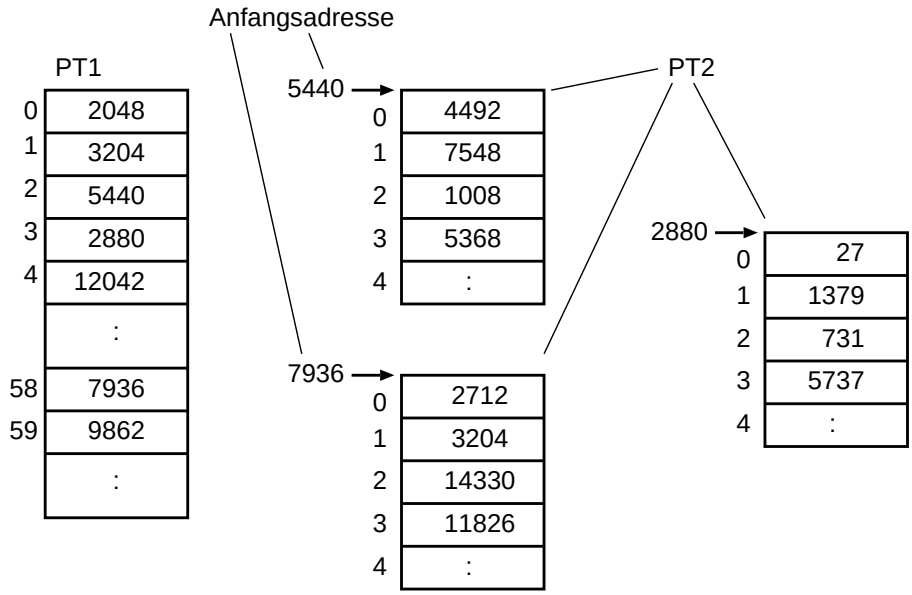
(2 Punkte) Wie viele Seitentabellen zweiter Stufe werden für das Text- und das Heap-Segment benötigt?

$408\,388\,260/4096 = 99704.1650 \rightarrow 99\,705$ Seiten. $99\,705/2048 = 48,684 \rightarrow 49$ Seitentabellen zweiter Stufe.

4.2 (4 Punkte)

Ein Betriebssystem verwendet Paging, um für die verschiedenen Prozesse einen virtuellen Hauptspeicher zu realisieren. Die Gesamtlänge einer Adresse beträgt 20 Bit. Das Betriebssystem verwendet eine zweistufige Seitentabelle. Die Länge der ersten Seitenadresse beträgt 6 Bit; die Länge der zweiten Seitenadresse beträgt 6 Bit; die Länge des Offsets beträgt 8 Bit.

Die folgende Abbildung zeigt einen Ausschnitt aus der Seitentabelle erster Stufe und einige Ausschnitte aus Seitentabellen 2. Stufe.



Im Weiteren soll eine virtuelle Adresse durch drei Dezimalzahlen für PT1, PT2 und Offset dargestellt werden. Beispiel: Die dezimalen Werte (47, 34, 205) stehen für die Adresse 101111 100010 11001101.

Die physische Adresse soll als *eine* Dezimalzahl dargestellt werden.

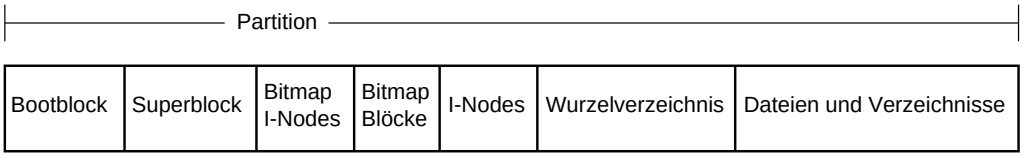
Ergänzen Sie die fehlenden Werte in der Tabelle:

virtuelle Adresse	physische Adresse
2, 0, 168	4660
3, 1, 231	
58, 3, 117	
	1020
	14 400

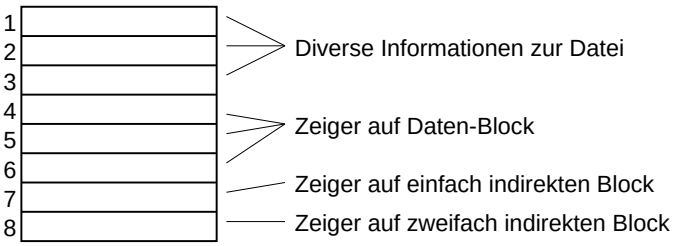
virtuelle Adresse	physische Adresse
2, 0, 168	4660
3, 1, 231	1610
58, 3, 117	11 943
2, 2, 12	1020
58, 2, 70	14 400

Aufgabe 5 Datei System

Ein Dateisystem verwendet I-Nodes für die Verwaltung von Dateien. Für die Freispeicherverwaltung von I-Nodes und Blöcken verwendet das System je eine Bitmap. Es gibt keine Kopie der Blöcke für die Freispeicherverwaltung.



Ein I-Node des Systems besitzt folgendes Format:



Die Daten sind also über 3 direkte Blöcke, einen einfach und einen zweifach indirekten Block erreichbar. Ein Block enthält 1024 Byte, ein Zeiger auf einen Block enthält 4 Byte. I-Nodes enthalten nie selbst Daten einer Datei.

5.1 (3 Punkte)

Beim Start des Systems stellt das Dateisystem fest, dass die Blöcke mit den Freispeicher-Bitmaps für I-Nodes und Blöcke nicht mehr gelesen werden können.

Alle anderen Blöcke der Platte sind nicht beschädigt. Kann die Information wiederhergestellt werden? Wenn ja, skizzieren Sie stichwortartig das Vorgehen.

Ja, die Information aus den anderen Blöcken wieder gewonnen werden. Das System liest rekursiv alle Verzeichnisse ein und ermittelt auf diese Weise, welche I-Nodes belegt sind. Das System untersucht zu jedem belegten I-Node, welche Blöcke von diesem I-Node aus erreicht werden und ermittelt auf diese Weise, welche Blöcke belegt sind.

5.2 (3 Punkte)

Wie groß kann eine Datei in diesem Dateisystem maximal sein? Bitte geben Sie alle Rechenschritte an.

Ein indirekter Block enthält 256 Zeiger (Blockgröße/Zeigergröße). Über den einfach indirekten Block sind also 256 Blöcke adressierbar. Über den zweifach indirekten Block sind 256^2 also 65536 Blöcke adressierbar. Insgesamt kann ein I-Node also 3 + 256 + 65536 Blöck adressieren. Das sind 65795 Blöcke d. h. 65795 KB oder 67374080 Byte oder 64.252929688 MB.

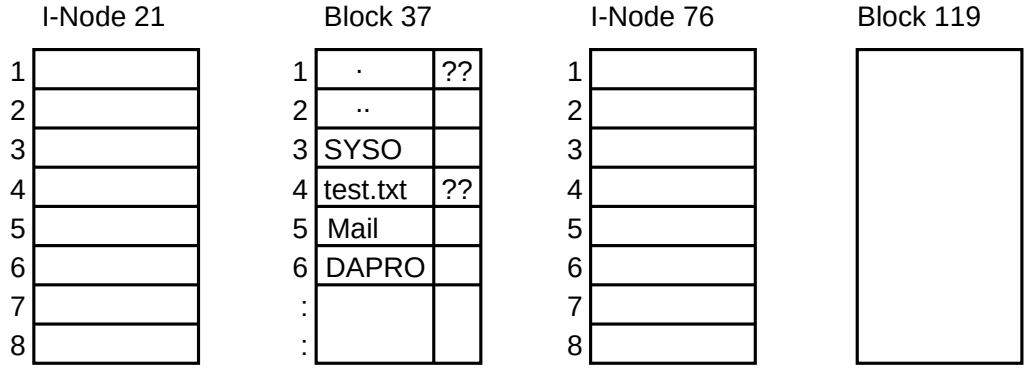
5.3 (2 Punkte)

Wieviel Blöcke belegt eine Datei, die 86 272 Byte Daten enthält. Berücksichtigen Sie *nicht* den Platz, der im Datei-Verzeichnis (Directory, Katalog) belegt wird und ebenfalls *nicht* den Platz, der durch den I-Node belegt wird.

Dateigröße/Blockgröße = 84.25 → 85 Daten-Blöcke. 3 direkte Blöcke, 82 einfach indirekt adressierte Blöcke (ein zusätzlicher Block). Die Datei belegt insgesamt 86 Blöcke.

5.4 (7 Punkte)

Der I-Node 21 verwaltet das Verzeichnis `/home/stud/tom`. Die Daten d. h. die Einträge des Verzeichnis liegen in Block 37. Ein Verzeichniseintrag ist 128 Byte groß. Das Verzeichnis enthält 6 Einträge (einschließlich `.` und `..`). Im Verzeichnis `/home/stud/tom` liegt u. a. die Datei `test.txt` (Größe: 632 Byte). Der I-Node 76 verwaltet diese Datei. Die Daten von `test.txt` liegen in Block 119.



Welche Werte müssen im Block 37 anstelle der ?? stehen?

Zeile 1: _____

Zeile 4: _____

Zeile 1: 21
Zeile 4: 76

In den I-Nodes 21 und 76 können Sie aus den obigen Angaben zwei Werte ableiten:

I-Node 21: Zeile _____, Wert _____

I-Node 76: Zeile _____, Wert _____

I-Node 21: Zeile 4, Wert 37
I-Node 76: Zeile 4, Wert 119

Ein Terminal-Programm (Shell, Eingabeaufforderung) verwendet als aktuelles Arbeitsverzeichnis `/home/stud/tom`.

Der Benutzer legt über das Terminal-Programm einen Hard-Link auf `test.txt` mit dem Namen `foo.txt` an (Kommando: `ln test.txt foo.txt`).

Wird ein zusätzlicher I-Node belegt?

---

nein

Wird ein belegter I-Node verändert, wenn ja, welcher I-Node, welche Veränderung?

Ja, der Ref-Count in I-Node 76 wird um 1 erhöht.

Wird ein zusätzlicher Block belegt?

---

nein

Wird ein belegter Block verändert, wenn ja, welcher Block, welche Veränderung ?

Ja, Block 37, zusätzlicher Eintrag: foo.txt, 76

Der Benutzer legt nun über das Terminal-Programm einen Soft-Link (symbolischer Link) auf `test.txt` mit dem Namen `bar.txt` (Kommando: `ln -s test.txt bar.txt`).

Wird ein zusätzlicher I-Node belegt?

---

ja

Wird in ein belegter I-Node verändert, wenn ja, welcher I-Node, welche Veränderung?

---

nein

Wird ein zusätzlicher Block belegt?

Ja, ein neuer Block, der den Pfadnamen der Zieldatei enthält.

Wird ein belegter Block verändert, wenn ja, welcher Block, welche Veränderung ?

Ja, Block 37, zusätzlicher Eintrag: bar.txt, *Nummer des neuen I-Nodes*

Der Benutzer legt nun über das Terminal-Programm eine Kopie der Datei test.txt an (Kommando: `cp test.txt fooBar.txt`).

Wielviel Blöcke werden dadurch zusätzlich belegt?

---

2 Blöcke werden zusätzlich belegt: Einer für das Verzeichnis (9. Eintrag), einer für die Datei.