

# Hochschule Ravensburg-Weingarten

## Schriftlich Prüfung Systemsoftware

Prof. Dr. M. Zeller

Datum, Zeit 1. Februar 2006, 14:00 – 15:30 Uhr (90 min)  
Aufgabenblätter 11 Seiten (einschl. Deckblatt)  
erreichbare Punktzahl 46  
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 des Datenschutzes keine telefonischen Auskünfte gegeben werden können.**

---

Vom Prüfer auszufüllen:

Aufgabe	1	2	3	4	Summe
Max. Punkte	15	14	12	5	46
Punkte					

Name:

Mat. Nr:

## Aufgabe 1 Speicherverwaltung, Paging

Betriebssystem X verwendet Paging um für die verschiedenen Prozesse einen virtuellen Hauptspeicher zu realisieren. Der virtuelle Speicher wird auf 8 MB Hauptspeicher und einen Teil der Festplatte abgebildet. Die Gesamtlänge einer Adresse beträgt 24 Bit.

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

### 1.1 (4 Punkte)

Gegeben ist das Format einer virtuelle Adresse:

PT	Offset
----	--------

Die Länge der Seitenadresse ist 13 Bit; die Länge des Offsets beträgt 11 Bit. Das Betriebssystem verwendet eine einstufige Seitentabelle.

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

---

---

1 Punkt: Seitengröße = Kachelgröße = 2048 Byte (2 KB)

Wie viele Kacheln verwaltet das Betriebssystem?

---

---

1 Punkt: Anzahl Kacheln = Größe Hauptspeicher / Kachelgröße = 8 MB/2 KB = 4 K Seiten (4096)

Wie groß (in Kilobyte, KB) ist die Seitentabelle, wenn ein Eintrag aus 3 Byte besteht?

---

---

2 Punkte: Anzahl adressierbare Seiten \* Größe eines Eintrags = 8 K \* 3 = 24 KB (24576)

### 1.2 (11 Punkte)

Das Betriebssystem verwendet nun ein anderes Format für die virtuellen Adressen:

PT1	PT2	Offset
-----	-----	--------

Das Betriebssystem verwendet eine zweistufige Seitentabelle. Die Länge der ersten Seitenadresse beträgt 6 Bit; die Länge der zweiten Seitenadresse beträgt 7 Bit; die Länge des Offsets beträgt 11 Bit.

Ein Prozess belegt folgende Adressbereiche:

Name:

Mat. Nr:

Prog. Teil	Adressbereich	Größe in Byte
Text-Segment	0 - 65 000	65 001
Heap-Segment	65 001 - 227 348	162 348
Stack Segment	15 915 072 - 16 777 215	862 144

Wie viele Einträge hat die Seitentabelle erster Stufe?

---

1 Punkt: PT1:  $2^6 = 64$  Einträge

Wie viele Einträge hat eine Seitentabelle zweiter Stufe?

---

1 Punkt: PT1:  $2^7 = 128$  Einträge

Wie viele Adressen lassen sich durch eine Seitentabelle zweiter Stufe erreichen/ansprechen?

---

1 Punkt: PT1:  $2^7 * 2KB = 256KB$

Wie viele Seiten belegen das Text- und das Heap-Segment?

---

2 Punkte:  $227\,348 / 2\,KB = 111.009765625 \rightarrow 112$  Seiten

Wie viele Seitentabellen zweiter Stufe werden für Text- und Heap-Segment benötigt?

---

2 Punkte:  $112 \text{ Seiten(Kacheln)} / 128 \text{ Einträge} < 1 \rightarrow 1$  Seitentabellen zweiter Stufe.

Wie viele Seiten belegt das Stack-Segment?

---

2 Punkte:  $16\,777\,215 - 15\,915\,072 = 862\,143 \text{ Byte} / 2\,KB = 420.968261719 \rightarrow 421$  Seiten .

Wie viele Seitentabellen zweiter Stufe werden für das Stack-Segment benötigt?

---

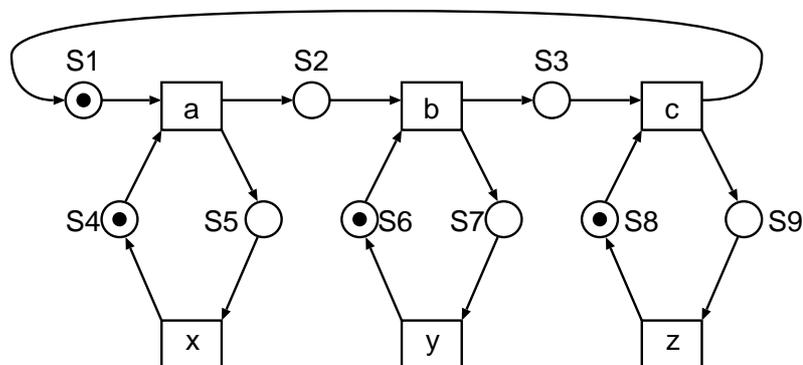
2 Punkte:  $421 \text{ Seiten(Kacheln)} / 128 \text{ Einträge} = 3.289 \rightarrow 4$  Seitentabellen zweiter Stufe.

Name:

Mat. Nr:

## Aufgabe 2 Synchronisation

Folgendes Petri-Netz zeigt die Synchronisation von 4 Prozessen. Die Transitionen a, b und c gehören zu Prozess  $P_1$ . Die Transition x gehört zu Prozess  $P_2$ . Die Transition y gehört zu Prozess  $P_3$ . Die Transition z gehört zu Prozess  $P_4$ .



### 2.1 (4 Punkte)

Welche Stellen müssen Sie als Semaphore realisieren um die vier Prozesse gemäß dem obigen Petri-Netz zu synchronisieren?

---

S4, S5, S6, S7, S8, S9

### 2.2 (4 Punkte)

Geben Sie den Quell-Code für die Prozesse  $P_1$ ,  $P_2$ ,  $P_3$  und  $P_4$  an. Sie können dazu Pseudo-Pascal verwenden (s. Skript von Frau Keller) oder (Pseudo-)Java.

Prozess P1{

Prozess P2{

}

}

Name:

Mat. Nr:

---

Prozess P3{

Prozess P4{

}

}

```
Prozess P1{
  while (true){
    S4.p();
    a();
    S5.v();
    S6.p();
    b();
    S7.v();
    S8.p();
    c;
    S9.v();
  }
}
```

```
Prozess P2{
  while (true){
    S5.p();
    x();
    S4.v()
  }
}
```

```
Prozess P3{
  while (true){
    S7.p();
    y();
    S6.v()
  }
}
```

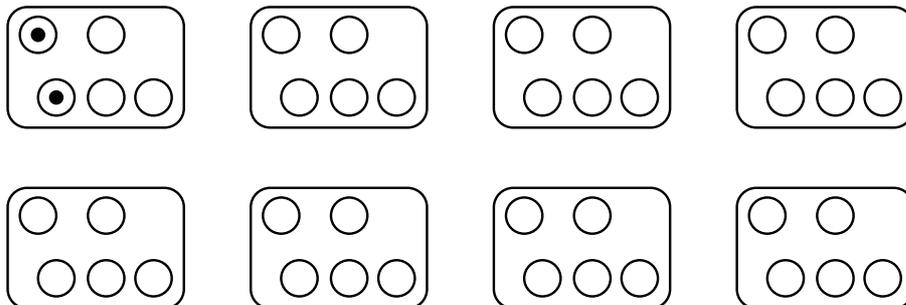
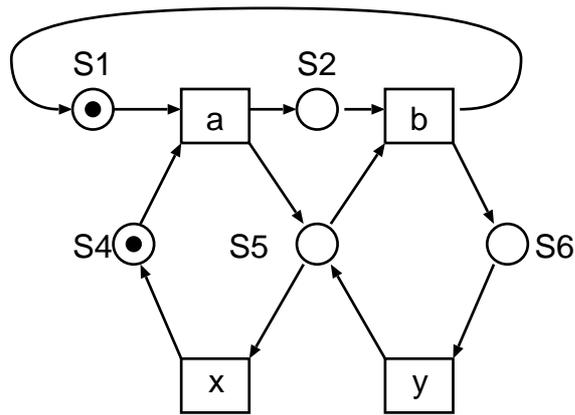
```
Prozess P4{
  while (true){
    S9.p();
    z();
    S8.v()
  }
}
```

### 2.3 (4 Punkte)

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

Name: \_\_\_\_\_

Mat. Nr: \_\_\_\_\_



**2.4 (2 Punkte)**

Kann das System in einen Deadlock geraten? Wie können Sie diese Frage anhand der Ereignis-  
 grafen entscheiden?

---



---

1 Punkt: Ja 1 Punkt: Es gibt mindestens einen Zustand, aus dem keine Kante herausführt.



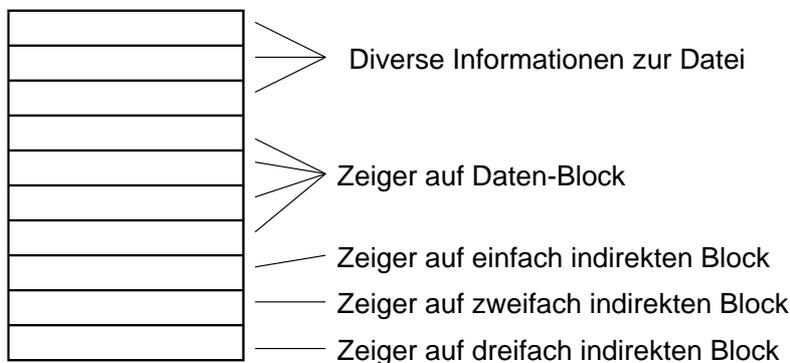
Name:

Mat. Nr:

---

### Aufgabe 3 Datei System

Ein Datei System verwendet I-Nodes, um Dateien zu verwalten. Ein I-Node besitzt folgendes Format:



Die Daten sind also über 4 direkte Blöcke, und je einen einfach, zweifach bzw. dreifach indirekten Block erreichbar. Ein Block enthält 512 Byte, ein Zeiger auf einen Block enthält 4 Byte.

Hinweis: Evtl. können Sie die Fragen leichter beantworten, wenn Sie eine Skizze der verwendeten Blöcke anfertigen.

#### 3.1 (1 Punkte)

Wieviel Zeiger auf Blöcke kann ein Block aufnehmen?

---

Blockgröße / Zeigergröße = 128 Zeiger

#### 3.2 (3 Punkte)

Wieviel Blöcke belegt eine Datei, die 72 000 Byte Daten enthält. Berücksichtigen Sie *nicht* den Platz, der im Datei-Verzeichnis (Directory, Katalog) belegt wird.

---

---

Dateigröße/Blockgröße =  $140.625 \rightarrow 141$  Daten-Blöcke. 4 direkte Blöcke, 128 einfach indirekt adressierte Blöcke (ein zusätzlicher Block), 9 zweifach indirekt adressierte Blöcke (zwei zusätzliche Blöcke). Es werden also drei zusätzliche Blöcke benötigt, die Datei belegt insgesamt 144 Blöcke.

#### 3.3 (3 Punkte)

Ein Programm vergrößert die Datei, indem es an das Ende der Datei einen 1 KB langen Datensatz schreibt.

Name:

Mat. Nr:

---

In welchen Blöcken (abgesehen vom I-Node) muss das System eine Änderung eintragen? Woher weiß das Betriebssystem, welche Blöcke der Platte/Partition nicht belegt sind?

---

---

---

Es werden zwei neue Datenblöcke belegt. Außerdem wird der Zeiger-Block, der am zweifach indirekten Block hängt, um die zwei Einträge erweitert.

Die neuen Blöcke werden der Freispeicherliste entnommen, alternativ: Die entsprechenden Bits der Speicher-Bitmap werden auf belegt gesetzt.

Name:

Mat. Nr:

### 3.4 (5 Punkte)

Ein Dateisystem wird per File Allocation Table (FAT) verwaltet. Ein Block enthält 512 Byte, ein Zeiger auf einen Block enthält 3 Byte. Nehmen Sie an, die unten abgebildete FAT ist bereits in den Hauptspeicher geladen. Ein Programm möchte die Datei C: \Max \test.txt einlesen. Die Blockadresse des Wurzelverzeichnis C: ist 2. Die Blockadresse des Verzeichnis C: \Max ist 4, der erste Block der Datei C: \Max \test.txt ist der Block 9.

0	
1	
2	12
3	14
4	5
5	EOF
6	13
7	18
8	7
9	3
10	8
11	EOF
12	EOF
13	10
14	15
15	11

Welche Plattenblöcke muss das Programm einlesen, um die Datei zu lesen?

---

---

Blöcke 2, 4, 9, 3, 14, 15, 11

Wie groß kann die Platte bzw. Partition für C: maximal sein?

---

---

Blockadress: 3 Byte = 24 Bit.  $2^{24}$  Blöcke \* 512 Byte pro Block = 8GB

Wie groß ist die FAT, wenn eine Partition mit 2 GB verwaltet werden soll.

---

---

$2\text{GB} / 512 \text{ Byte pro Block} = 4\,194\,304 \text{ Blöcke} * 3 \text{ Byte pro Block} = 12\text{MB}$

oder:  $2^{31} \text{ Byte} / 2^9 \text{ Byte pro Block} = 2^{22} \text{ Blöcke} * 3 \text{ Byte pro Block} = 12\text{MB}$

Name:

Mat. Nr:

---

## Aufgabe 4 Ein-/Ausgabe

Ein Programm auf einem Windows-, Linux- oder Mac-OS-Rechner liest eine Datei komplett ein. Anschließend speichert es die Datei dann leicht verändert aber mit gleicher Größe wieder ab. Um zu prüfen, ob die Daten korrekt gespeichert wurden, liest das Programm die Datei dann wieder komplett ein.

Das Programm misst für jede der oben genannten Aktivitäten die benötigte Zeit:

erstes Lesen	25 ms
Schreiben	12 ms
zweites Lesen	14 ms

### 4.1 (1 Punkte)

Warum dauert das erste Lesen länger, als das Schreiben?

---

---

Das Programm schreibt erst in einen Betriebssystem-Puffer, die Daten werden erst später auf die Platte geschrieben.

### 4.2 (1 Punkte)

Warum dauert das zweite Lesen nicht so lange wie das erste?

---

---

Die Daten befinden sich noch (teilweise) im Betriebssystem-Puffer; es sind daher weniger Plattenzugriffe nötig.

### 4.3 (1 Punkte)

Ist es eine gute Idee, die geschriebenen Daten – wie oben erklärt – zu überprüfen?

---

---

Nein: die Daten befinden sich noch (teilweise) im Betriebssystem-Puffer; es wird also i. Allg. nicht die Datei auf der Platte geprüft, sondern die Datei im Puffer.

Name:

Mat. Nr:

---

#### 4.4 (2 Punkte)

Ein Treiber im Kernel und ein Anwendungsprogramm wollen Daten austauschen. Ein Entwickler macht folgenden Vorschlag: Der Treiber allokiert Hauptspeicher als Puffer und gibt dem Anwendungsprogramm einen Pointer darauf.

Nehmen Sie an, dass sich die Prozesse korrekt synchronisieren.

Würden Sie den Vorschlag so umsetzen oder modifizieren (Begründen Sie Ihre Entscheidung)?

---

---

---

Das Anwendungsprogramm hat keine Zugriffsberechtigung für den Puffer des Treibers. Es muss daher umgekehrt das Anwendungsprogramm den Puffer einrichten. Der Treiber darf auf diesen Speicher zugreifen (Kernel-Mode).