

Hochschule Ravensburg-Weingarten

Schriftliche Prüfung Betriebssysteme

Prof. Dr. M. Zeller

Datum, Zeit 20. Februar 2009, 08:00 – 09:30 Uhr (90 min)
Aufgabenblätter 12 Seiten (einschl. Deckblatt)
erreichbare Punktzahl 69
zugelassene Hilfsmittel A (s. Prüfungsplan)

Studiengang Prof. Nr. Raum
AI 3618 H061

Name: _____

Matrikelnummer: _____

Vorbemerkung Die Klausur ist ziemlich umfangreich. Lassen Sie sich nicht verunsichern, Sie benötigen nicht alle Punkte für die Note 1,0; Sie benötigen weniger als die Hälfte der Punkte für die Note 4,0.

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!

Vom Prüfer auszufüllen:

Aufgabe	1	2	3	4	5	6	Summe
Max. Punkte	17	7	21	8	6	10	69
Punkte							

Name:

Mat. Nr:

Aufgabe 1 Virtueller Speicher

Ein Betriebssystem verwendet Paging, um für die verschiedenen Prozesse jeweils einen virtuellen Hauptspeicher zu realisieren. Der virtuelle Speicher wird auf 48 MB Hauptspeicher und 8 MB der Festplatte abgebildet (Swap-Space). Die Gesamtlänge einer Adresse beträgt 26 Bit.

Das Betriebssystem verwendet eine zweistufige Seitentabelle. Die Länge der ersten Seitenadresse (PT1) beträgt 8 Bit; die Länge der zweiten Seitenadresse (PT2) beträgt 9 Bit; die Länge des Offsets beträgt 9 Bit.

8 Bit	9 Bit	9 Bit
PT1	PT2	Offset

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

1.1 (8 Punkte)

Ein Prozess belegt folgende Adressbereiche:

Prog. Teil	Adressbereich	Größe in Byte
TextSegment	0 - 390 000	390 001
HeapSegment	390 001 - 22 403 070	22 013 070
Stack Segment	268 291 437 – 268 435 456	144 020

(1 Punkt) Wie viele Einträge hat die Seitentabelle erster Stufe?

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

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

(1 Punkt) Wie viele Seiten belegt das StackSegment?

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

(1 Punkt) Wie viele Kacheln verwaltet das Betriebssystem?

(1 Punkt) Auf welche Größe kann der virtuelle Speicher maximal ausgebaut werden (Hauptspeicher plus Swap-Space)

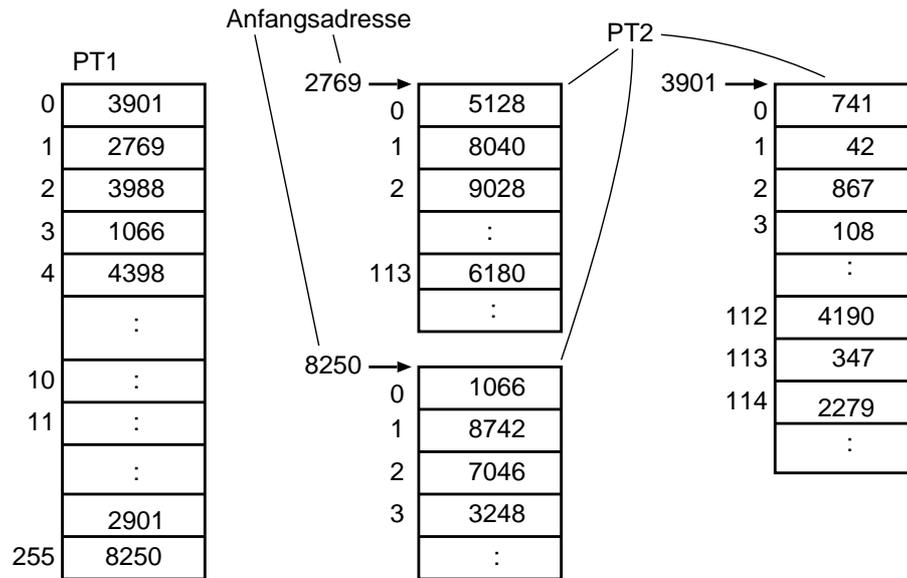
Name:

Mat. Nr:

1.2 (9 Punkte)

Im Weiteren soll eine virtuelle Adresse durch vier Dezimalzahlen für PT1, PT2, und Offset dargestellt werden. Beispiel: Die dezimalen Werte (33, 13, 116) stehen für die virtuelle Adresse 00100001 000001101 001110100.

Die folgende Abbildung zeigt einen Ausschnitt aus der Seitentabelle erster Stufe und einige Ausschnitte aus den Seitentabellen zweiter Stufe. Achtung: In den Seitentabellen zweiter Stufe stehen nur die signifikanten Bits, so dass der Offset lediglich angehängt werden muss!



Die physische Adresse soll in Form einer Dezimalzahl dargestellt werden.

Ergänzen Sie die fehlenden Werte in der Tabelle soweit möglich. Wenn Sie einen Wert nicht eintragen können, so begründen Sie dies bitte stichwortartig:

virt. Adresse			phys. Adresse
1	113	47	3 164 207
1	2	491	
0	1	21	
			56 496
			545 870
255	3	176	
			177 989
3	2	501	

Name:

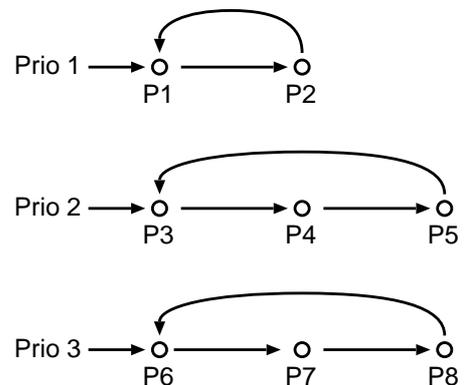
Mat. Nr:

Aufgabe 2 Scheduling

Ein Betriebssystem verwendet präemptive 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).

2.1 (7 Punkte)

Die folgende Tabelle soll die Zustände der Prozesse zu verschiedenen Zeitpunkten darstellen. Der Zustand unmittelbar vor dem 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. Die Zeitpunkte t_{-2} und t_{-1} fassen jeweils zwei Ereignisse zusammen.



Ergänzen Sie die Tabelle gemäß den auftretenden Ereignissen.

- t_0 : Ein Zeitscheibe für das präemptive Multitasking läuft ab
- t_1 : Prozess P1 wird blockiert
- t_2 : Prozess P2 wird blockiert
- t_3 : Prozess P5 wird blockiert
- t_4 : Prozess P4 wird blockiert
- t_5 : Prozess P5 wird deblockiert
- t_6 : Ein Zeitscheibe für das präemptive Multitasking läuft ab
- t_7 : Ein Zeitscheibe für das präemptive Multitasking läuft ab

	t_{-2}	t_{-1}	t_0	t_1	t_2	t_3	t_4	t_5	t_6	t_7
P1	BL	BL	BL	BR						
P2	BL	BL	RE	RE						
P3	BL	BL	BL	BL						
P4	BR	RE	BR	BR						
P5	BL	BL	BL	BR						
P6	BR	BR	BR	BR						
P7	RE	BR	BR	BR						
P8	BL	BL	BL	BL						

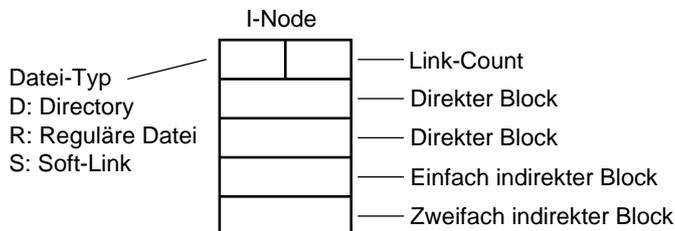
Name:

Mat. Nr:

Aufgabe 3 Datei System mit I-Nodes

Ein Dateisystem verwendet I-Nodes für die Verwaltung von Dateien. Für die Freispeicherung von I-Nodes und Blöcken verwendet das System je eine Bitmap.

Ein I-Node des Systems besitzt folgendes Format:



Die Daten sind also über zwei direkte Blöcke, einen einfach indirekten Block und einen zweifach indirekten Block erreichbar. Ein Block enthält 4096 Byte, ein Zeiger auf einen Block enthält 8 Byte. I-Nodes enthalten nie selbst Daten einer Datei.

3.1 (3 Punkte)

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

3.2 (2 Punkte)

Wie groß kann das Dateisystem maximal sein (Begründung)?

3.3 (3 Punkte)

Wie viele Blöcke belegt eine Datei, die 100 MB Daten enthält. Berücksichtigen Sie *nicht* den Platz, der im Datei-Verzeichnis (Directory) belegt wird und ebenfalls *nicht* den Platz, der durch den I-Node belegt wird.

Name:

Mat. Nr:

3.5 (4 Punkte)

Die Datei `text.bla` wird um 4 KB vergrößert. Welche Änderungen ergeben sich in dem gegebenen Dateisystem? Wenn das System zusätzliche Blöcke verwenden, so sind dies Block 12, 13, 14 ... Wenn das System zusätzliche I-Nodes verwenden, so sind dies I-Node 10, 11, 12 ...

Was wird angelegt, welche Werte werden wo eingetragen bzw. verändert?

3.6 FAT Datei-System (2 Punkte)

Eine Partition der Größe 2 GB soll durch ein FAT-Datei-System verwaltet werden. Die Größe eines Blocks beträgt 4 KB, ein Zeiger auf einen Block besteht aus 4 Byte

Wie groß ist die FAT für diese Dateisystem?

Name:

Mat. Nr:

Aufgabe 5 Funktionsaufruf (6 Punkte)

Ein Compiler verwendet nur den Stack, um Daten zwischen verschiedenen Funktionen eines Programms auszutauschen. Folgendes Programm ist gegeben:

```
1 int rofl (int *value, char line []) {
2     int result = *value;
3     if (line[0] < 'x'){
4         *value = result + 1;
5     }
6     return result;
7 }
8
9 int lol (int value, int nums[]){
10    char text[] = "abcd";
11    value = rofl(&nums[3], text);
12    text[1] = value + '0';
13    return value;
14 }
15
16 int main (void){
17    int values[ ] = {1, 2, 3, 4};
18    int test = 7;
19    test = lol(test, values) ;
20    return test;
21 }
```

Ergänzen Sie die Skizze des Stacks auf der nächsten Seite zu folgenden Zeitpunkten:

- t_1 Zeile 10 unmittelbar nach der Zuweisung zu `text`
- t_2 Zeile 6 unmittelbar vor der Anweisung `return`
- t_3 Zeile 13 unmittelbar vor der Anweisung `return`

Verwenden Sie dabei folgende Symbole:

- \longrightarrow Pointer
- $---$ Variable angelegt aber nicht initialisiert
- xxx Variable besitzt einen unbekanntem Wert
- $sp \rightarrow$ Stelle, auf die der Stackpointer zeigt

Name:

Mat. Nr:

Aufgabe 6 Synchronisation

Das folgende Petri-Netz zeigt die Synchronisation von drei Prozessen P_{AB} , P_C und P_{DE} . Es handelt sich um ein Bedingungs-Ereignis-Netz. Die Transitionen A, B gehören zu Prozess P_{AB} , die Transition C gehört zum Prozess P_C , die Transitionen D, E gehören zu Prozess P_{DE} .

6.1 (2 Punkte)

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

6.2 (3 Punkte)

Geben Sie den Quell-Code für die Prozesse P_{AB} , P_C und P_{DE} an. Sie können dazu PseudoPascal verwenden (s. Skript von Frau Keller) oder (Pseudo)Java.

Prozess P_{AB} {

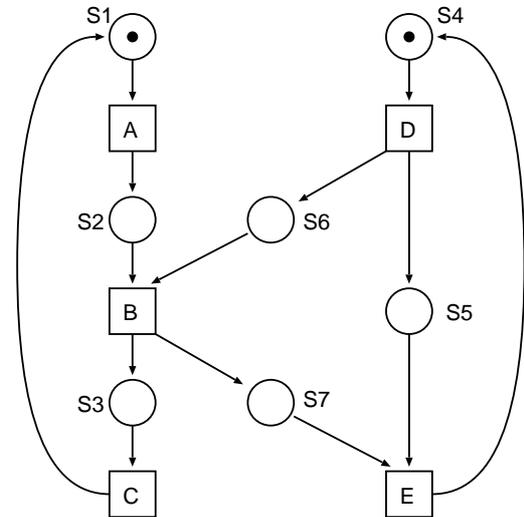


Abbildung 3: Ein paralleles System in Form eines Petri-Netzes

Prozess P_C {

}

}

Name:

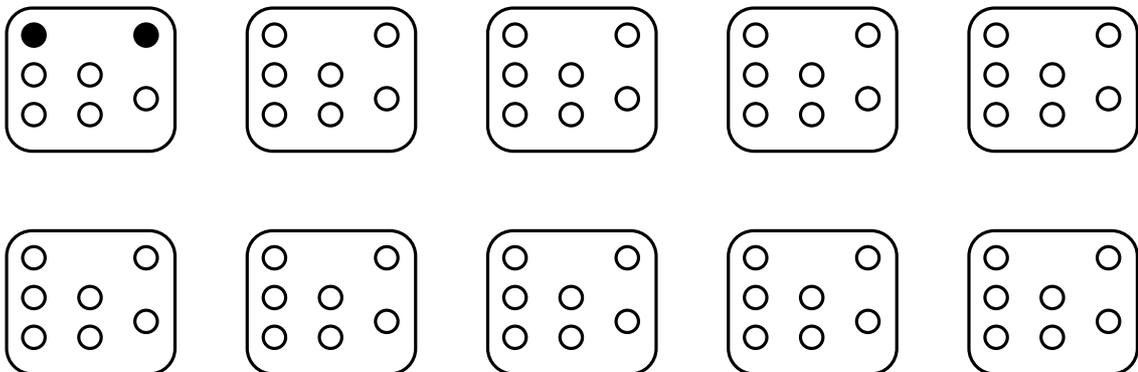
Mat. Nr:

Prozess P_DE{

}

6.3 (3 Punkte)

Zeichnen Sie den Ereignisgraphen des Petri-Netzes. Sie können die Vorlage unten verwenden oder eine eigene Skizze anfertigen. Geben Sie zu jedem Übergang die Transition, die ihn auslöste, an.



6.4 (2 Punkte)

Kann das System, das in dem oben angegebenen Petri-Netz (s. Abb. 3) dargestellt ist, in einen Deadlock geraten; wenn ja, wie; wenn nein, warum nicht?
