



Freiburg, 20. Dezember 2007

Systeme I - WS 07/08 Zwischenklausur

Herzlich willkommen zur Nachklausur Systeme 1!

Die Klausur besteht aus fünf Aufgaben. Bitte überprüfen Sie ihr Exemplar auf Vollständigkeit!

Die Punkte dieser Klausur tragen zu 35 % zur Endnote Systeme 1 bei.

Die Bearbeitungszeit beträgt 90 Minuten.

Bitte lesen Sie alle Fragen genau durch, bevor Sie diese beantworten! Wir gehen davon aus, dass wenn mehrere Aspekte gefragt werden, diese auch beantwortet werden sollten. Dieses trifft auch zu, wenn mehrere Fragen zusammen gestellt werden, aber nicht einzeln mit a) - f) gekennzeichnet wurden.

Die zu erreichende Punktzahl ist für jede Frage angegeben. Bitte benantworten Sie alle Fragen nur auf den gehefteten Blätter der Klausur. Verwenden Sie ggf. die Rückseite der Blätter.

Soll eine Lösung nicht berücksichtigt werden, so streichen Sie diese deutlich durch. Ansonsten wird bei der Korrektur immer die erste Lösung berücksichtigt.

Hilfsmittel:

Als Hilfsmittel zugelassen ist ausschließlich ein **nicht** programmierbarer Taschenrechner. Mobiltelefone sind bei der Klausuraufsicht **ausgeschaltet** abzugeben.

Viel Erfolg!

Name, Vorname:

Matrikelnummer:

Unterschrift:

Aufgabe	1	2	3	4	5	Σ
Punkte	... / 6	... / 4	... / 10	... / 7	... / 8	... / 35

Aufgabe 1: (3 + 3 Punkte)

In der Vorlesung wurde das FAT32 Dateisystem vorgestellt. FAT32 nutzt 28 Bits zur Adressierung von Blöcken, bei einer Blockgröße von 32 Kb ($32 \cdot 2^{10}$ Bytes). Die restlichen 4 Bits der FAT-Einträge werden für andere Zwecke verwendet.

Weiterhin wurde das I-Node Konzept vorgestellt. Gehen Sie davon aus, dass der I-Node

- 10 Blöcke direkt adressieren kann,
- einfach, doppelt und dreifach indirekte Blöcke verwenden kann,
- eine Adresse 4 Byte groß ist,
- und jeder Block 1 Kb (2^{10} Bytes) groß ist.

a.) Beschreiben Sie wie bei einem FAT-Dateisystem der Zugriff auf das Byte Nummer 20000 einer Datei erfolgt, wenn Sie den FAT-Eintrag für den ersten Block dieser Datei kennen. Skizzieren Sie ggf. den Ablauf grafisch.

Sie können davon ausgehen, dass die Datei ausreichend groß ist.

b.) Ab welcher Dateigröße werden Einträge in dreifach indirekte Blöcke eines I-Node notwendig.

Aufgabe 2: (2 + 2 Punkte)

- a.) Nennen kurz Sie die Unterschiede zwischen präemptiven und kooperativen Multitasking.
- b.) Nennen sie jeweils ein Betriebssystem, das eine solche Variante implementiert.

Aufgabe 3: (4 + 2 + 2 Punkte)

a.) Vier Prozesse (P_1, P_2, P_3, P_4) werden zu $t_{P_1} = 0, t_{P_2} = 0, t_{P_3} = 3, t_{P_4} = 5$ lafbereit. Die erwarteten Laufzeiten seien 7, 5, 1 und X .

In welcher Reihenfolge (abhängig von der Laufzeit X) sollten Sie abgearbeitet werden, um die durchschnittliche Durchlaufzeit zu minimieren?

b.) Berechnen Sie die durchschnittliche Durchlaufzeit für $X = 2$!

c.) Zeigen Sie, dass bei einem Shortest Remaining Time (SRT) Scheduler, Prozesse verhungern können. Geben Sie dazu einen Schedule mit Ankunftszeit und Laufzeit der Prozesse an.

Aufgabe 4: (4 + 3 Punkte)

a.) In der Vorlesung wurde folgende Softwarelösung für wechselseitigen Ausschluss zweier Prozesse vorgeschlagen:

```
/* Prozess 0 */
wiederhole
{
    flag[0] := true;
    solange (flag[1] = true)
        tue nichts;

    /* kritischer Abschnitt */
    flag[0] := false;
    /* nichtkrit. Abschnitt */
}

/* Prozess 1 */
wiederhole
{
    flag[1] := true;
    solange (flag[0] = true)
        tue nichts;

    /* kritischer Abschnitt */
    flag[1] := false;
    /* nichtkrit. Abschnitt */
}
```

a.) Bei dieser Lösung kann ein Deadlock auftreten. Beschreiben Sie einen konkrete Ablaufsequenz, die zu einer solchen Situation führt.

b.) Wie können solche Probleme verhindert werden? Welche Maßnahmen müssen auf Hardwareebene und welche auf Betriebssystemebene getroffen werden? Wie kann aktives Warten verhindert werden? Skizzieren sie eine solche Lösung.

Aufgabe 5: (8 Punkte)

Schreiben Sie ein Programmfragment in Pseudocode. Benutzen Sie Semaphoren mit `down()` und `up()` Operationen. Folgendes soll realisiert werden:

- Ein Prozess P1 errechnet ein Ergebnis, das von Prozess P2 weiterbenutzt werden soll.
- Prozess P2 darf nicht ausgeführt werden, bevor das Ergebnis vorliegt.
- P1 wiederum muss warten bis P2 seine Berechnungen fertiggestellt hat, bevor er ein neues Ergebnis an P2 liefert.
- Prozess P2 liefert sein Ergebnis an Prozess P3 weiter. Dieser besitzt einen Puffer mit N freien Plätzen.

Bitte kommentieren Sie den Pseudocode ausführlich.