

**Nach-Klausur im Studiengang Allg.Informatik – Realzeitsysteme – WS05**  
(SL + PL)

Dozent: Döben-Henisch Zeit: Do, 18.Mai 2006, 12:00 – 14:00h Raum: AudiMax

<b>NAME</b>		<b>MATRIKELNUMMER</b>	
<b>SL oder PL</b>			
<b>Punkte</b>		<b>Note</b>	

### 1. Regularien

- i. Teilnehmer an der Klausur, die eine PL schreiben, haben die Meldung zur studienbegleitenden Prüfungsleistung ausgefüllt.
- ii. Es dürfen benutzt werden: 1 einseitig handschriftlich beschriebener Zettel mit Name und Matr.Nr., Schreibgeräte (nicht rot), 1 Taschenrechner sowie leere unbeschriebene Blätter. Alle in Gebrauch befindlichen Blätter müssen mit Name und Matr.Nummer gekennzeichnet sein. Arbeitsblätter ohne Kennzeichnung gelten als Täuschungsversuch. Blätter ohne Kennzeichen werden bei der Auswertung nicht berücksichtigt.
- iii. Die Klausurzeit beträgt 90 Min für SL und 120 Min für PL. Man darf seinen Platz erst verlassen, wenn der Prüfungsleiter die Klausurarbeit in Empfang genommen hat.
- iv. Es gilt folgende Punktetabelle:

NOTE	PUNKTE
1	ab 85
2	75 bis < 85
3	65 bis < 75
4	50 bis < 65
5	<50

- v. Das aktive Abschreiben wie auch das Zulassen von Abschreiben wird als Täuschungsversuch bewertet.

---

### 2. Fragen (Max. 33 Pkt)

- 1.(2 Pkte) Wodurch unterscheidet sich ein *Realzeitsystem* von einem System, das kein Realzeitsystem ist?
- 2.(4 Pkte) Nennen sie alle wichtigen Komponenten eines Realzeitsystems und beschreiben Sie, wie diese zusammenspielen.
- 3.(2 Pkte) Welche Faktoren beeinflussen den *WCAO*?
- 4.(4 Pkte) Was versteht man unter Systemen, die (i) als *Guaranteed-Response/ Best-Effort* bzw. (ii) als *Event-triggered/ Time-triggered* charakterisiert werden?
- 5.(3 Pkte) Was unterscheidet *periodische* Ereignisse von *sporadischen* Ereignissen?
- 6.(3 Pkte) Was bedeutet es, dass ein System *präemptiv* ist?
- 7.(3 Pkte) Was versteht man unter dem *Jitter* und der *Jitter-Toleranz*?
- 8.(4 Pkte) Wie verhalten sich die Begriffe '*cluster*', '*ftu*', '*node*' und '*task*' zueinander?
- 9.(2 Pkte) *Was ist der Unterschied zwischen einem Ereignis und einem Task?*

- 10.(3 Pkte) Was unterscheidet im Betriebssystem Linux den *Kernel-Mode* vom *User-Mode*?
- 11.(3 Pkte) Was bedeutet es, wenn man sagen kann, dass ein *Algorithmus optimal* in seiner Klasse ist?

### 3. Aufgaben (Max. 80 Pkt)

1. (9 Pkte) Gegeben sei die folgende *Taskmenge T1 für den RM-Algorithmus*

Ereignis	Task	C	D	T
e1	t1	1	2	2
e2	t2	1	4	4
e3	t3	2	8	8

Nehmen Sie mittels der **Planbarkeitsformeln** aus der Vorlesung eine Abschätzung der *Planbarkeit* dieser Taskmenge vor. Benutzen Sie sowohl die notwendige wie auch die hinreichende Bedingung (soweit vorhanden).

Gegeben sei die folgende *Taskmenge T2 für den EDF-Algorithmus*:

Ereignis	Task	C	D	T
e1	t1	1	2	2
e2	t2	1	3	4
e3	t3	2	6	8

Nehmen Sie mittels der **Planbarkeitsformeln** aus der Vorlesung eine Abschätzung der *Planbarkeit* dieser Taskmenge vor. Benutzen Sie sowohl die notwendige wie auch die hinreichende Bedingung (soweit vorhanden).

2. (24 Pkte) Erstellen Sie sowohl für die Berechnung mit dem *RM-Algorithmus* als auch für die Berechnung mit dem *EDF-Algorithmus* ein **Zeitdiagramm**, das auf den Taskmengen T1 und T2 aus Aufgabe 1 basiert. Beachten sie die notwendige Länge des Zeitfensters. Benutzen sie für die Markierung von eintreffenden Ereignissen sowie den zugehörigen Deadlines die bekannten Pfeilsymbole. Markieren Sie diejenigen Zeiten, in denen die **CPU unbeschäftigt ('idle')** ist. Betrachten Sie im Falle des RM-Algorithmus sowohl den Fall, dass das System **nicht präemptiv ist**, wie auch den Fall, **dass es präemptiv ist**; welche Unterschiede können Sie beobachten? Im Falle des EDF-Algorithmus nehmen Sie **Präemption an**.
3. (20 Pkte) Gegeben ein **komplexes System** mit den Komponenten A,B,C,D, die wie folgt angeordnet sind: ((A || B) --> (C ||D)). Für die *Fehlerraten lambda* und *Reparaturraten my* sollen folgende Werte gelten:

	<b>Jahre</b>	<b>Tage</b>	<b>Stunden</b>	
Betriebszeit	1	365	8760	
Systeme	A	B	C	D
Fehler in 1 Jahr	6	3	4	2
Reparatureinheiten in 1 Jahren	30	21	24	20

(1) (10 Pkte) Berechnen Sie die **Zuverlässigkeit** des Systems ((A || B) --> (C || D)). Stellen Sie auch die Zwischenschritte dar.

(2) (10 Pkte) Erstellen Sie für ihr System ein vollständiges **Zustands-Übergangsdiagramm**, dessen Kanten sowohl mit der Fehlerrate Lambda als auch mit der Reparaturrate My gekennzeichnet sind (Achtung: Lambda und My müssen erst berechnet werden!). Geben Sie für jeden Zustand des Diagramms an, ob der Zustand als 'guter' ('good') Zustand oder als 'fehlerhafter' ('failed') zu bezeichnen ist.

4. (30 Pkte) Versuchen Sie, soweit wie möglich, einen Algorithmus *in Form eines Flussdiagramms oder Struktogramms* zu konstruieren, der die **Scheduling-Regel des EDF-Algorithmus umsetzt**. Es soll gelten, dass ihr System 1 CPU hat und präemptiv ist. Testen Sie ihren Algorithmus mit der *Taskmenge aus der vorausgehenden Aufgabe Nr.2*. Für die Ausgabe ihres Algorithmus auf dem Bildschirm nehmen Sie folgendes Format an:

$n: \{t_1, \dots, t_k\} ==> t_i$

Erklärung:

$n$  := aktueller Zeitpunkt

$\{t_1, \dots, t_k\}$  := Alle Tasks, die warten

$t_i$  := Task, der zur Ausführung kommt.

(i) Normalfall: es tritt *kein Konflikt* auf.

Beispiel:

1: {2,4} ==> 2

2: {2,4} ==> 2

3: {3,4} ==> 4

4: {3,4} ==> 3

...

(ii) Konfliktfall: Ein Task verletzt seine Deadline

Beispiel:

1: {2,4} ==> 2

2: {2,4} ==> 2

3: {3,4} ==> 4

4: {3,4} ==> !3!

...

Die Ausrufezeichen sollen den Konflikt anzeigen. Im Konfliktfall bricht ihr Programm ab.