Klausur im Studiengang Allg.Informatik – Realzeitsysteme – WS05

(SL + PL)

Dozent: Döben-Henisch Zeit: Di, 7.Febr. 2006, 12:00 – 14:00h Raum AudiMax

NAME		***MATRIKELNUMMER***	
		SL oder PL	
Punkte		***Note***	

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. 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. 32 Pkt)

1. (2 Pkte) Welche Anforderung an ein System charakterisiert ein System als ein ***Realzeitsystem***?
2. (3 Pkte) Nennen Sie Faktoren, die die Ermittlung der ***Worst Case Execution Time (WCET)*** in heutigen Computersystemen erschweren bzw. beim heutigen Wissensstand letztlich unmöglich machen?
3. (5 Pkte) Was versteht man unter Systemen, die (i) als ***Fail-safe/Fail-operational*** charakterisiert werden bzw. (ii) als ***Resource-Adequate/ Resource Inadequate***?
4. (3 Pkte) Was unterscheidet ***periodische*** Ereignisse von ***spontanen*** Ereignissen?
5. (3 Pkte) Was unterscheidet ***abhängige*** von ***unabhängigen*** Tasks?
6. (3 Pkte) Was ist der Unterschied zwischen einem ***Ereignis*** und einem ***Task***?
7. (3 Pkte) Wie unterscheidet sich die ***relative*** von der ***absoluten*** Deadline? Inwiefern kann man mit einer Deadline eine ***Zeitfenster*** definieren?

8. (3 Pkte) Welche Faktoren beeinflussen den ***Worst Case Administration Overhead (WCAO)***?
9. (3 Pkte) Welche Bedeutung haben die Begriffe '***target system***' und '***host system***' im Kontext der Entwicklung von eingebetteten Systemen?
10. (4 Pkte) Wie ist ein ***Linux-Modul*** aufgebaut?

3. Aufgaben (Max. 80 Pkt)

1. (6 Pkte) Gegeben sei die folgende ***Taskmenge T1*** (siehe Anhang). Nehmen Sie mittels der Planbarkeitsformeln aus der Vorlesung eine Abschätzung der ***Planbarkeit*** dieser Taskmenge vor, sowohl für den ***DM-Algorithmus*** als auch für den ***EDF-Algorithmus***.
2. (18 Pkte) Erstellen Sie sowohl für die Berechnung mit dem ***DM-Algorithmus*** als auch für die Berechnung mit dem ***EDF-Algorithmus*** ein ***Zeitdiagramm***, das auf der Taskmenge 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. Im Falle des DM-Algorithmus ist das System nicht präemptiv, im Falle des EDF-Algorithmus ist es präemptiv.
3. (6 Pkte) Geben sie an, welche der ***aperiodischen Tasks*** aus der ***Taskmenge T2*** (siehe Anhang) zusammen mit den ***Tasks aus Taskmenge T1*** und bei Annahme des ***EDF-Algorithmus*** ausführbar wären und welche nicht.
4. (20 Pkte) Gegeben ein ***komplexes System*** mit den Komponenten G,S und M, die wie folgt angeordnet sind: ((G --> S) || M). Für die ***Fehlerraten lambda*** und ***Reparaturraten my*** siehe den Anhang. (1) (10 Pkte) Berechnen Sie die ***Zuverlässigkeit*** des Systems ((G --> S) || M). (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. Geben Sie für jeden Zustand des Diagramms an, ob der Zustand als ***'guter' ('good') Zustand*** oder als ***'fehlerhafter' ('failed')*** zu bezeichnen ist.
5. (30 Pkte) Versuchen Sie, soweit wie möglich, ein ***Programm in Form eines Flussdiagramms*** oder ***Struktogramms*** zu konstruieren, das die Scheduling-Regel des ***DM-Algorithmus*** umsetzt. Es soll gelten, dass ihr System 1 CPU hat und nicht präemptiv ist.Testen Sie ihren Algorithmus mit der ***Taskmenge aus Aufgabe 1***. Die Ausgabe ihres Programms auf dem Bildschirm soll wie folgt beschaffen sein:

   (i) Normalfall: es tritt kein Konflikt auf.
   n: $\{t_1, ..., t_k\}$ ==> t_i
   n := aktueller Zeitpunkt
   $\{t_1, ..., t_k\}$:= Alle Tasks, die warten
   t_i := Task, der zur Ausführung kommt.

   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 Konflitkfall bricht ihr Programm ab.

4. Anhang

Taskmenge T1 für Aufgabe 1

Taskmenge	C	D	T
t1(r0 = 0, C=3, D=7, T=20)	3	7	20
t2(r0 = 0, C=2, D=4, T=5)	2	4	5
t3(r0 = 0, C=2, D=9, T=10)	2	9	10

Taskmenge T2 für Aufgabe 3

Taskmenge	C	D	T
t4(r0 = 9, C=1, D=3)	1	3	-
t5(r0 = 8, C=2, D=3)	2	3	-
t6(r0 = 17, C=2, D=3)	2	3	-

Fehlerrate und Reparaturraten für Aufgabe 4

Angenommener Zeitraum in Jahren: 1

	Lambda	My
G	5	5
S	4	4
M	1	3

/* Anmerkung. die Zahlen bei Lambda sind die Fehler pro Zeiteinheit. Lambda ist zu errechnen. entsprechend bei My */