

Klausur

Realzeitsysteme WS04

Dozent: Döben-Henisch

8. Februar 2005, 10:00 – 12:00h, Raum AudiMax (BCN 231)

NAME	
MATRIKELNUMMER	
Prüfungsart: SL oder PL	

Summe Pkte		Note	
-------------------	--	-------------	--

Minimale Punktezahl, um RT zu bestehen: SL: 12,5 Pkte + Übungen; PL 25 Pkt (keine Übungen)

1. Regularien

- i. Teilnehmer an der Klausur müssen sich durch Lichtbildausweis identifizieren.
 - ii. Es dürfen nur leere Blätter, Schreibgeräte und Taschenrechner benutzt werden. 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 120 Min. Man darf seinen Platz erst verlassen, wenn der Prüfungsleiter die Klausurarbeit in Empfang genommen hat.
 - iv. Es gilt folgende Punktetabelle:
 1. >69
 2. 55-69
 3. 40-54
 4. 25-39
 5. <25
 - v. Das aktive Abschreiben wie auch das Zulassen von Abschreiben wird als Täuschungsversuch bewertet.
 - vi. Jeder, der an der Klausur teilnimmt, erklärt zu Beginn, dass er sich durch keine Umstände beeinträchtigt empfindet, die seine Klausurleistung wesentlich beeinflussen könnten.
-

Fragen (Max. 18 Pkt):

1. (Pkt. 1,5) Welche speziellen Anforderungen ergeben sich durch Realzeitsysteme für Betriebssysteme?
2. (Pkt. 1) Welche Mechanismen in Programmiersprachen wie z.B. C++, Java, Perl, PHP stehen einem Einsatz in Realzeitsystemen entgegen? (Antwort für eine Sprache reicht)
3. (Pkt. 1,5) Welche Eigenschaften der neuronalen Informationsverarbeitung beeinflussen die Verarbeitung zeitlicher Ereignisse?
4. (Pkt. 1) Wie ist heute die Basiseinheit für Zeit definiert?
5. (Pkt. 2,5) Was bedeuten UT0, UT1, TAI und UTC? Wie hängen sie untereinander zusammen?
6. (Pkt. 0,5) Mit welcher Genauigkeit kann man im Rahmen von C99 Zeit messen?
7. (Pkt. 1,5) Was unterscheidet *periodische* Ereignisse von *spontanen* oder *aperiodischen*

Ereignissen?

8. (Pkt. 1) Was ist der Unterschied zwischen einem *Ereignis* und einem *Task*?
9. (Pkt. 1,5) Was versteht man unter einem *Jitter*, unter der *Jitter-Toleranz* und der relativen und der absoluten Jitter-Toleranz?
10. (Pkt. 1) Wie unterscheidet sich die *relative* von der *absoluten Deadline*?
11. (Pkt. 1,5) Wie berechnet man die *Auftretenszeit des k-ten Ereignisses*, wenn die Periode des Ereignisses T ist?
12. (Pkt. 1) Was bedeutet es, wenn man sagen kann, dass der EDF-Algorithmus *optimal in seiner Klasse* ist?
13. (Pkt. 1) Was ist ein FIFO im Kontext von RTAI und Linux?
14. (Pkt. 1,5) Wozu benötigt man einen FIFO-Handler? Wie arbeitet er?

Aufg.1 (Max.Pkt: 14):

Im Anhang dieser Klausuraufgabe ist eine Schaltung zur Erzeugung bzw. zum Messen von Impulsen abgebildet (mit kurzer Beschreibung). Beschreiben Sie, wie Sie unter Benutzung dieser Schaltung die folgenden Aufgaben lösen würden. Zeigen Sie in beiden Fällen durch eine *Zeichnung* der Signal- und Zählerverläufe, wie Sie dies mithilfe der Schaltung tun können. *Kommentieren* Sie die Zeichnung

A1.1)(7 Pkt) Sie wollen den Abstand zwischen zwei aufeinanderfolgende Rechtecksignalen messen.

A1.2)(7 Pkt). Sie wollen ein periodisches Rechtecksignal mit einer definierten Dauer und einem definierten Abstand erzeugen.

Aufg.2 (Max.Pkt: 18):

Gegeben sei das folgende komplexe System mit den Komponenten M1 – M9. Die Zahlen in eckigen Klammern repräsentieren die Fehlerrate Lambda (Fehler/h) für jede Komponente.

$(M1[0.000002] \rightarrow ((((M5[0.000205] \rightarrow M7[0.000041]) || (M6[0.000305] \rightarrow M8[0.000021])) \rightarrow ((M2[0.000223] || M3[0.000223]) || M4[0.000223]) || M9[0.000042]))$

(i) Zeichnen Sie ein Diagramm des Systems mit den Komponenten in sequentieller und paralleler Anordnung.

(ii) Berechnen Sie die Zuverlässigkeit dieses Systems für einen Zeitraum von 10 Jahren (1 Jahr = 365 Tage; 1 Tag = 24 h)(Formeln im anhang).

Aufg.3 (Max.Pkt: 18):

Gegeben sei die folgende Taskmenge:

t1(r0 = 0, C=1, D=4,T=4)

t2(r0 = 0, C=2, D=6,T=6)

t3(r0 = 0, C=2, D=8,T=8)

(i) Berechnen Sie die *notwendige* Bedingung für eine Planbarkeit dieser Taskmenge mit EDF (Formel im Anhang).

(ii) Erstellen sie ein *Zeitdiagramm* mit Hilfe des EDF-Algorithmus

(iii) Geben Sie die Zeitintervalle an, in denen die *CPU untätig* (idle) ist.

Gegeben seien zusätzlich die beiden folgenden aperiodischen Tasks:

- (a) $t_4(r_0 = 9, C=2, D=6)$
- (b) $t_5(r_0 = 10, C=2, D=7)$

- (iv) Fall (a): wäre t_4 planbar? Was sagt die hinreichende Planbarkeitsformel (siehe Anhang)?
- (v) Fall (b): wäre t_5 planbar? Was sagt die hinreichende Planbarkeitsformel (siehe Anhang)?

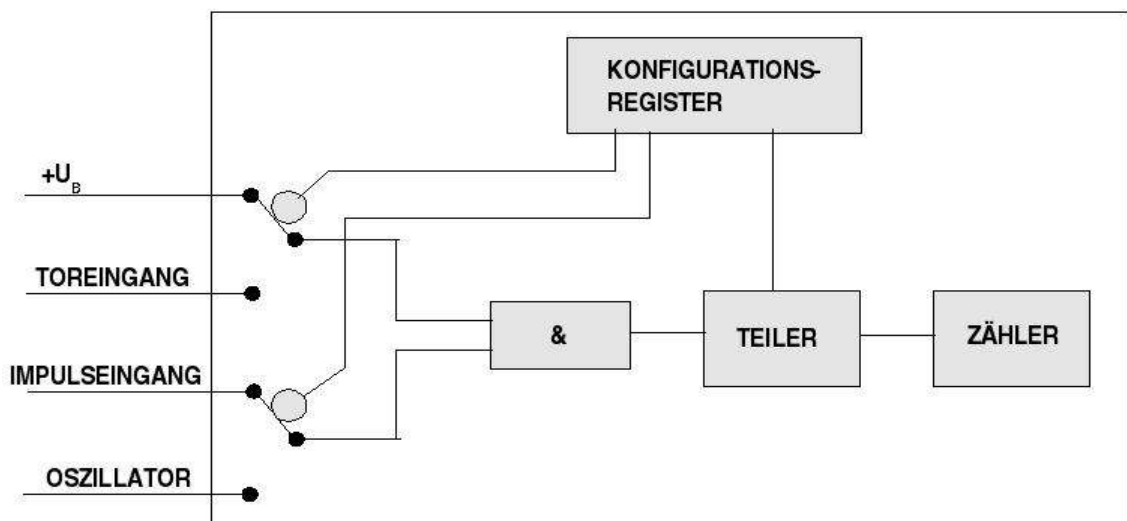
Aufg.4 (Max.Pkt:18):

Gegeben sei die folgende Taskmenge mit Prioritäten:

EREIGNIS (e)	PERIODE (T)	TASK ID	AUSFÜHRUNGS ZEIT (C)	PRIORITÄT (P)	BLOCKADE ZEITEN (B)	DEADLINE (D)
e1	12	a1	4	v.high	0	8
e2	24	a2	6	high	0	24
e3	16	a3	2	medium	0	14
e4	32	a4	1	low	3	16
e5	32	a5	1	v.low	0	32

Benutzen Sie die Formel aus dem Anhang, um die Planbarkeit des Tasks a_4 zum Ereignis e_4 zu berechnen.

ANHANG: Zeiterzeugung und -messung



Die Grundeigenschaften dieser Schaltung werden wie folgt angenommen: Bei *konstanter Vorspannung* durch $+U_B$ oder nur bei Vorliegen einer bestimmten *Torspannung* zählt ein Zähler einzelne *Impulse* oder die Impulse eines *Oszillators*.. Zusätzlich kann in allen vier Fällen der Zählvorgang durch einen *Teiler* modifiziert werden. Ein Zähler kann *aufwärts* oder *abwärts* laufen.

Beim *Zählerüberlauf* kann ein Zähler aus einem Register mit einem *bestimmten Wert geladen* werden einen *Interrupt* auslösen und/oder einen *Pegelwechsel* an einem Anschluss verursachen. Die Abfrage eines Zählers kann mittels *capture* erfolgen oder mittels *compare*. Im Falle von *capture* wird der Inhalt des Zählerregisters gelesen, ohne dass der Zähler angehalten wird; im Falle von *compare* wird der Inhalt des Zählerregisters mit einem anderen Wert verglichen. Falls ungleich läuft der Zähler weiter, falls gleich wird ein Interrupt oder ein Pegelwechsel ausgelöst.

Bei einem Zähler mit N-Bit Breite, einem Startwert S und einer Inkrementzeit T_i beträgt die Zeitdauer T_O zwischen zwei Überläufen: $T_O = (2^N - S) * T_i$. Bei einem Systemtakt von z.B. 1 MHz würde ein Takt $1/1 \text{ MHz} = 1 * 10^{-6} \text{ sec} = 1 \text{ Mikrosekunde}$ dauern.

ANHANG : Zuverlässigkeit

Einzelnes System: $(R(t) = e^{-\lambda * t} ; (t \text{ ist die Zeitdauer})$	$R(t) = e^{-\lambda * t}$
Systeme in sequentieller Anordnung:	$R(t) = \prod_{i=1}^N R_i(t)$
Systeme in paralleler Anordnung:	$R(t) = 1 - \prod_{i=1}^N [1 - R_i(t)]$

Anhang: Planbarkeit EDF

Bedingung	Formel
Notwendige Bedingung, damit eine Taskmenge mit $D=T$ mit EDF planbar ist	$U = \sum_{i=1}^n \frac{C_i}{T_i} \leq 1$
Hinreichende Bedingung, damit eine hybride Taskmenge mit $D \leq T$ mit EDF planbar ist	$U = \sum_{i=1}^n \frac{C_i}{D_i} \leq 1$

ANHANG: Planbarkeit eines einzelnen Ereignisses

Es soll geprüft werden, ob Ereignis e_i planbar ist.

1. Identifiziere eine Menge (H) von Ereignissen, deren Priorität höher oder gleich der Priorität P_i von e_i sind. Teile dann die Menge (H) auf in zwei Mengen (H1) und (Hn). (H1) enthält solche Ereignisse, deren Perioden T grösser oder gleich der Deadline D_i von e_i sind. (Hn) soll solche Ereignisse enthalten, deren Perioden T kleiner als die Deadline D_i von e_i sind.
2. Berechne die totale effektive Nutzungszeit f_i von Ereignis e_i . Diese totale Nutzungszeit setzt sich aus zwei Bestandteilen zusammen: (i) die totale Nutzungszeit der Ereignisse in (Hn), sowie (ii) die Ausführungszeit von e_i plus Verspätungszeiten durch Blockaden plus Präemptionszeiten verursacht durch Ereignisse aus (H1), alles dividiert durch die Periode von e_i :

$$f_i = [\text{SUM}(j \in Hn) C_j / T_j] + [1 / T_i (C_i + B_i + \text{SUM}(k \in H1) C_k)]$$

3. Berechne die Nutzbarkeitsgrenze $U(n, Dt_i)$ mit $n = \text{Anzahl Elemente in (Hn)} + 1$ sowie $Dt_i = D_i / T_i$. Es gilt:

$$U(n, Dt_i) =$$

$$(i) n * ((2Dt_i)^{1/n} - 1) + 1 - Dt_i ; \text{falls } 0.5 < Dt_i \leq 1.0$$

$$(ii) Dt_i ; \text{falls } 0 \leq Dt_i \leq 0.5$$

4. Vergleiche die effektive totale Nutzungszeit von e_i mit der oberen Schranke:

$$f_i \leq U(n, Dt_i)$$

Falls f_i kleiner oder gleich der Grenze ist, dann kann dieses Ereignis seine Deadline einhalten.