

Hochschule Esslingen

Wintersemester 2011/2012	Blatt-Nr. 1 von 10
Fakultät: Informationstechnik	Semester: IT4
Prüfungsfach: Echtzeitsysteme	Fachnummer: 4061/4062
Hilfsmittel: 2 DIN-A4-Blätter, Taschenrechner	Zeit: 90 Minuten

Name, Vorname: _____

Aufgabe 1: Diverse Fragen (30 Minuten)

Hinweis: Die folgenden Teilaufgaben können unabhängig voneinander bearbeitet werden.

1.1 Welche wesentlichen Funktionen muss ein typisches Echtzeit-Computersystem erfüllen?

Antwort:

1.2 Welchen Einfluss hat Jitter auf die Latenzzeit bei der Fehlererkennung?

Antwort:

Wintersemester 2011/2012	Blatt-Nr. 2 von 10
Fakultät: Informationstechnik	Semester: IT4
Prüfungsfach: Echtzeitsysteme	Fachnummer: 4061/4062
Hilfsmittel: 2 DIN-A4-Blätter, Taschenrechner	Zeit: 90 Minuten

- 1.3 Erläutern Sie die wesentlichen Unterschiede zwischen einem „soft real-time“-System und einem „hard real-time“-System. Kann es bei einem „soft real-time“-System einen „critical failure mode“ geben (bitte begründen)?

Antwort:

- 1.4 Geben Sie eine Daumenregel an, die die Abtastrate bei einem quasi-kontinuierlichen System in Beziehung zu der Anstiegszeit (rise time) der Sprungantwort eines Controlled Objects setzt.

Wintersemester 2011/2012	Blatt-Nr.	3 von 10
Fakultät: Informationstechnik	Semester:	IT4
Prüfungsfach: Echtzeitsysteme	Fachnummer:	4061/4062
Hilfsmittel: 2 DIN-A4-Blätter, Taschenrechner	Zeit:	90 Minuten

1.5 Benennen und beschreiben Sie kurz die fünf Prozessgebiete nach dem V-Modell XT.

Antwort:

1.6 Wozu dient das Projekthandbuch? Was ist der Unterschied zum Projektplan?

Antwort:

Wintersemester 2011/2012	Blatt-Nr. 4 von 10
Fakultät: Informationstechnik	Semester: IT4
Prüfungsfach: Echtzeitsysteme	Fachnummer: 4061/4062
Hilfsmittel: 2 DIN-A4-Blätter, Taschenrechner	Zeit: 90 Minuten

1.7 Es gibt drei Möglichkeiten, Nebenläufigkeit (Concurrency) auf einem Computersystem zu implementieren. Benennen Sie diese mit jeweils einer kurzen Erklärung.

Antwort zu 1.7:

1.8 Ist es möglich, dass eine Task, die sich im Wartezustand befindet, sich selbst wieder aufweckt (resume)? Bitte erklären.

Antwort zu 1.8:

Wintersemester 2011/2012	Blatt-Nr. 5 von 10
Fakultät: Informationstechnik	Semester: IT4
Prüfungsfach: Echtzeitsysteme	Fachnummer: 4061/4062
Hilfsmittel: 2 DIN-A4-Blätter, Taschenrechner	Zeit: 90 Minuten

1.9 Der Einspritzbeginn muss bei einem modernen Verbrennungsmotor auf genauer als 1° berechnet werden, um die Abgasvorschriften einhalten zu können. Die Leerlaufdrehzahl des Motors betrage 800 Umdrehungen/min.

- Wenn die maximale zeitliche Genauigkeit $37 \mu\text{s}$ beträgt, und der Einspritzzeitpunkt damit auf exakt 1° genau gehalten werden, wie groß ist die maximale Drehzahl des Motors in Umdrehungen pro Minute?
- Wie genau kann der Einspritzzeitpunkt in $^\circ$ bei der Leerlaufdrehzahl gehalten werden, wenn die zeitliche Genauigkeit $37 \mu\text{s}$ beträgt?
- Nehmen Sie an, die Leerlaufdrehzahl kann um $\pm 10\%$ des Nominalwerts schwanken. Wie groß ist die maximale Leerlaufdrehzahl?

Antwort mit Rechenweg:

a) Maximale Drehzahl in U/min = _____

b) Einspritzgenauigkeit in $^\circ$ im Leerlauf = _____

c) Maximale Leerlaufdrehzahl = _____

Wintersemester 2011/2012	Blatt-Nr. 6 von 10
Fakultät: Informationstechnik	Semester: IT4
Prüfungsfach: Echtzeitsysteme	Fachnummer: 4061/4062
Hilfsmittel: 2 DIN-A4-Blätter, Taschenrechner	Zeit: 90 Minuten

Aufgabe 2: Interprozesskommunikation (30 Minuten)

Ein System besteht aus zwei nebenläufigen Tasks TA und TB, die über einen Datenspeicher **Mailbox** miteinander kommunizieren. Das Schreiben in die Mailbox wird durch eine nicht-blockierende Funktion **send()** realisiert, das Lesen durch eine nicht-blockierende Funktion **receive()**.

- 2.1 Die Mailbox kann genau ein Datum halten. Signalisieren Sie über ein Event-Flag, wenn die Mailbox gefüllt ist. Modellieren Sie das System in einem Aktivitätsdiagramm für den Fall, dass TA eine run-to-completion-Struktur hat und bei jedem Durchlauf ein Datum schreibt. TB hat eine endless-loop-Struktur und liest ein Datum aus der Mailbox, wenn diese gefüllt ist; wenn nicht, wartet TB. TA darf nur in eine leere Mailbox schreiben, und muss ansonsten auf die Leerung durch TB warten. Verwenden Sie zur Synchronisation und zum Schutz von kritischen Abschnitten eine minimale Anzahl von Semaphoren. Nehmen Sie keine unnötigen Einschränkungen vor. Geben Sie an, wie die Semaphoren und das Flag initialisiert sein müssen.

Lösung zu Aufgabe 2.1:

Wintersemester 2011/2012	Blatt-Nr. 7 von 10
Fakultät: Informationstechnik	Semester: IT4
Prüfungsfach: Echtzeitsysteme	Fachnummer: 4061/4062
Hilfsmittel: 2 DIN-A4-Blätter, Taschenrechner	Zeit: 90 Minuten

- 2.2 Die Mailbox kann beliebig viele Daten halten. Signalisieren Sie den Füllstand der Mailbox über eine zählende Semaphore. Modellieren Sie das System in einem Aktivitätsdiagramm für den Fall, dass TA eine run-to-completion-Struktur hat und bei jedem Durchlauf einmal schreibt. TB hat eine endless-loop-Struktur und liest ein Datum aus der Mailbox, wenn diese gefüllt ist; wenn nicht, wartet TB. TA darf beliebig oft schreiben, und TB darf solange lesen und konsumieren, bis die Mailbox leer ist. Verwenden Sie zur Synchronisation und zum Schutz von kritischen Abschnitten eine minimale Anzahl von Semaphoren. Nehmen Sie keine unnötigen Einschränkungen vor. Geben Sie an, wie die Semaphoren initialisiert sein müssen.

Lösung zu Aufgabe 2.2:

Wintersemester 2011/2012	Blatt-Nr. 8 von 10
Fakultät: Informationstechnik	Semester: IT4
Prüfungsfach: Echtzeitsysteme	Fachnummer: 4061/4062
Hilfsmittel: 2 DIN-A4-Blätter, Taschenrechner	Zeit: 90 Minuten

- 2.3 Die Mailbox kann genau ein Datum von entweder TA oder TB halten. Die beiden Tasks TA und TB sollen im Ping-Pong-Verfahren Daten austauschen (z.B. TA schreibt, TB liest, TB schreibt, TA liest, etc). Beide Tasks haben eine run-to-completion-Struktur und können beliebig oft in beliebiger Reihenfolge aufgerufen werden. Eine Task darf nie ihr eigenes Datum aus der Mailbox lesen. Wenn nicht gelesen oder geschrieben werden kann, wird nichts gemacht. Koordinieren Sie die beiden Tasks mit insgesamt vier Event-Flags *TAcanWrite*, *TAcanRead*, *TBcanWrite* und *TBcanRead*, und schützen Sie kritische Abschnitte mit einer minimalen Anzahl von Semaphoren. Zeichnen Sie das Aktivitätsdiagramm. Geben Sie an, wie die Semaphoren und Event-Flags initialisiert sein müssen, wenn zuerst TA schreiben darf. Hinweis: Event-Flags können auch nicht-blockierend gelesen werden.

Lösung zu Aufgabe 2.3:

Wintersemester 2011/2012	Blatt-Nr. 9 von 10
Fakultät: Informationstechnik	Semester: IT4
Prüfungsfach: Echtzeitsysteme	Fachnummer: 4061/4062
Hilfsmittel: 2 DIN-A4-Blätter, Taschenrechner	Zeit: 90 Minuten

Aufgabe 3: Scheduling (30 Minuten)

Gegeben sind drei Tasks:

$T1(r_0 = 0, C = 1, T = 3)$

$T2(r_0 = 0, C = 1, T = 6)$

$TS(r_0 = 0, C = 2, T = 8)$ (Polling Server)

Alle Tasks sind präemptiv.

- 3.1 Berechnen Sie die Prozessorauslastung U und die Hyperperiode H . Ist das Task-Set so garantiert mit dem RM-Verfahren planbar, unter der Annahme, dass der Polling Server voll ausgelastet wird (bitte begründen)?

Antwort:

$U =$ _____

$H =$ _____

Task-Set garantiert planbar mit RM-Verfahren? _____

- 3.2 Wie sind die Prioritäten der drei Tasks nach dem RM-Verfahren einzustellen, und warum?

Antwort:

Prio _____ > Prio _____ > Prio _____

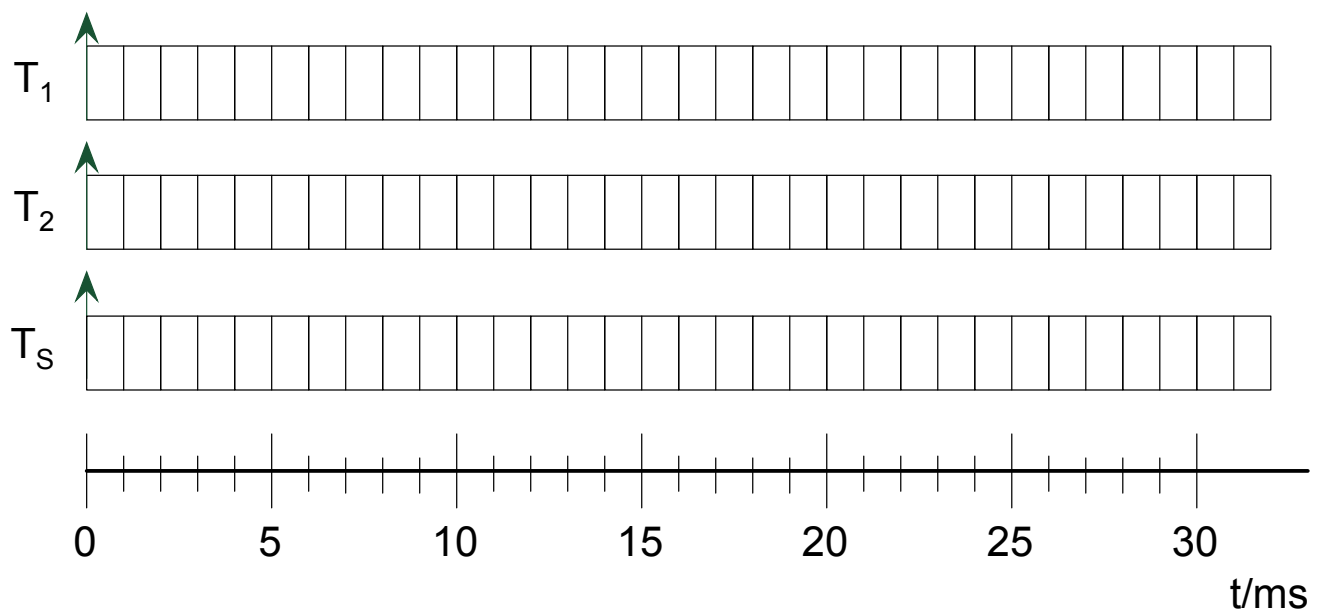
Begründung:

- 3.3 Zum Zeitpunkt $t=3$ steht ein sporadischer Job mit einer Rechenzeit von $C=2$ zur Bearbeitung an. Zeichnen Sie in die untenstehende Diagrammvorlage bis zum Zeitpunkt $t=8$ ein, wann unter Annahme eines präemptiven RM-Scheduling welche Task läuft (Kreuz in Kästchen, wenn Task läuft).

Wintersemester 2011/2012	Blatt-Nr. 10 von 10
Fakultät: Informationstechnik	Semester: IT4
Prüfungsfach: Echtzeitsysteme	Fachnummer: 4061/4062
Hilfsmittel: 2 DIN-A4-Blätter, Taschenrechner	Zeit: 90 Minuten

- 3.4 In Fortsetzung zu 3.3: Zum Zeitpunkt $t=8$ steht ein sporadischer Job mit einer Rechenzeit von $C=3$ zur Bearbeitung an. Zeichnen Sie in die untenstehende Diagrammvorlage bis zum Zeitpunkt $t=17$ ein, wann unter Annahme eines präemptiven RM-Scheduling welche Task läuft (Kreuz in Kästchen, wenn Task läuft).
- 3.5 In Fortsetzung zu 3.4: Zum Zeitpunkt $t=16$ steht ein sporadischer Job mit einer Rechenzeit von $C=1$ zur Bearbeitung an. Zum Zeitpunkt $t=19$ steht ein weiterer sporadischer Job mit einer Rechenzeit von $C=2$ an. Zeichnen Sie in die untenstehende Diagrammvorlage bis zum Zeitpunkt $t=30$ ein, wann unter Annahme eines präemptiven RM-Scheduling welche Task läuft (Kreuz in Kästchen, wenn Task läuft).

Diagramm für 3.3 bis 3.5:



- 3.6 Wird die Serverkapazität im oben geschilderten Szenario voll ausgenutzt oder überschritten? Bitte begründen.

Antwort mit Begründung:

- 3.7 Wie hoch ist für den Fall nach 3.3 bis 3.5 die durchschnittliche Antwortzeit (response time) für die vier oben beschriebenen sporadischen Jobs?

Antwort:

Durchschnittliche Antwortzeit = _____