

Hochschule Esslingen

Wintersemester	2007/2008	Blatt-Nr.:	1 von 8
Fachbereich:	Informationstechnik	Semester:	IT4
Prüfungsfach:	Echtzeitsysteme	Fachnummer:	4061
Hilfsmittel:	Eigene handschriftliche Aufzeichnungen, Laborberichte, Taschenrechner, Programmlistings und Zeichnungen	Zeit:	90 Minuten

Bitte Ihren Namen nicht vergessen! **NAME:** _____

Aufgabe 1: Diverse Fragen

(25 Punkte)

Hinweis: 1.1 bis 1.11 können unabhängig voneinander bearbeitet werden.

- 1.1 Was versteht man unter relokierbaren Adressen?
Erläutern Sie anhand eines Beispiels.
Nennen Sie ein Objektdatei-Format, das relokierbare Adressen unterstützt.
- 1.2 Beschreiben Sie in Stichworten die Aufgaben von Linker und Locator.
- 1.3 Wozu werden Guards in Header-Dateien benötigt?
Erläutern Sie anhand eines kleinen Beispiels (C-Code).
- 1.4 Welche besonderen Eigenschaft besitzen die reentrante (wiedereintrittsfähige) Funktionen? Erläutern Sie an einem einfachen Beispiel.
- 1.5 In Software-Architekturen wird häufig der Begriff HAL eingesetzt.
Erläutern Sie den Begriff und nennen Sie 3 Vorteile, die sich ergeben.
- 1.6 Das wesentliche Verhalten von Systemen wird häufig durch Modelle beschrieben.
Unter welchen Voraussetzungen eignet sich ein Automatenmodell ganz besonders zur Beschreibung eines Systems?
- 1.7 Erläutern Sie den Unterschied zwischen 2 parallelen und 2 nebenläufigen Prozessen.
- 1.8 Nennen Sie die Zustände, die eine Task einnehmen kann und beschreiben Sie alle Zustände in Stichworten.
Zeigen Sie mit Hilfe einer Skizze die möglichen Zustandsübergänge auf und benennen Sie diese.
- 1.9 Nennen Sie drei zusätzliche Eigenschaften eines Mutex gegenüber einer binären Semaphore. Weshalb werden die von Ihnen genannten Eigenschaften benötigt?
- 1.10 Nennen Sie zwei Möglichkeiten zur Vermeidung einer Prioritätsinvertierung.
Beschreiben Sie jeweils kurz die Wirkungsweise.
- 1.11 Eine Synchronisierung kann oft durch Flags oder durch Semaphore erreicht werden.
Erläutern Sie die Unterschiede und nennen Sie ein Beispiel für eine Synchronisation, die über Flags nicht erreicht werden kann.

Hochschule Esslingen

Wintersemester	2007/2008	Blatt-Nr.:	2 von 8
Fachbereich:	Informationstechnik	Semester:	IT4
Prüfungsfach:	Echtzeitsysteme	Fachnummer:	4061
Hilfsmittel:	Eigene handschriftliche Aufzeichnungen, Laborberichte, Taschenrechner, Programmlistings und Zeichnungen	Zeit:	90 Minuten

Aufgabe 2: Synchronisation

(25 Punkte)

In zwei Anwendungsprozessen sollen durch Synchronisation der nebenläufigen Tasks TA und TB unterschiedliche Abläufe erzwungen werden.

Task TA wird wiederholt in einer Funktion des jeweiligen Anwendungsprozesses aktiviert, führt Aktion A einmalig aus und terminiert sich danach selbst, während Task TB beim Systemstart einmalig für die gesamte Laufzeit aktiviert wird und nach der Ausführung der Aktion B in den Wartezustand wechselt.

Für den Ablauf der Anwendung sind zwei Aktionen maßgeblich, die jeweils in einer Task ausgeführt werden:

Aktion A in Task TA,
Aktion B in Task TB.

Zur Synchronisation sollen ausschließlich zählende Semaphore eingesetzt werden.

2.1 Anwendungsprozess 1:

Aktion B soll jeweils nach einmaliger Ausführung der Aktion A ausgeführt werden (während Aktion B darf A bereits wieder ausgeführt werden).

Erstellen Sie ein Aktivitätsdiagramm, das den geforderten Ablauf darstellt. Benennen Sie von Ihnen verwendete Semaphore (Name).
Wie ist(sind) die Semaphore zu initialisieren?

Aktivitätsdiagramm zu 2.1:

Task A

Task B

Hochschule Esslingen

Wintersemester	2007/2008	Blatt-Nr.:	3 von 8
Fachbereich:	Informationstechnik	Semester:	IT4
Prüfungsfach:	Echtzeitsysteme	Fachnummer:	4061
Hilfsmittel:	Eigene handschriftliche Aufzeichnungen, Laborberichte, Taschenrechner, Programmlistings und Zeichnungen	Zeit:	90 Minuten

2.2 Anwendungsprozess 2:

Jetzt soll Aktion B jeweils dann ausgeführt werden, wenn zuvor Aktion A mindestens zweimal ausgeführt worden war. Dazu soll die Semaphore „SEM_2T“ verwendet werden. Zudem muss jetzt sichergestellt sein, dass die Aktionen A und B nicht nebenläufig ausgeführt werden (während Aktion B darf Aktion A nicht ausgeführt werden).

Erstellen Sie wieder ein Aktivitätsdiagramm, das den geforderten Ablauf darstellt. Benennen Sie von Ihnen verwendete Semaphore (Name). Wie ist(sind) die Semaphore zu initialisieren?

Aktivitätsdiagramm zu 2.2:

Task A

Task B

Hochschule Esslingen

Wintersemester	2007/2008	Blatt-Nr.:	4 von 8
Fachbereich:	Informationstechnik	Semester:	IT4
Prüfungsfach:	Echtzeitsysteme	Fachnummer:	4061
Hilfsmittel:	Eigene handschriftliche Aufzeichnungen, Laborberichte, Taschenrechner, Programmlistings und Zeichnungen	Zeit:	90 Minuten

Aufgabe 3: Scheduling

(25 Punkte)

Betrachten Sie eine Menge von Tasks τ bestehend aus (normierte Zeiten)

- $\tau_1(r_0 = 0, e = 3, D=x, p = 14)$
- $\tau_2(r_0 = 0, e = 2, D=4, p = 4)$
- $\tau_3(r_0 = 0, e = 1, D=6, p = 7)$

a) Berechnen Sie die Hyperperiode H des Tasksets.

$H =$ _____

b) Berechnen Sie die Prozessorauslastung U sowie den Lastfaktor CH des Tasksets in Abhängigkeit von x .

$CH =$ _____ $U =$ _____

c) Weisen Sie den Tasks für ein geeignetes Verfahren mit zur Laufzeit unveränderlichen Prioritäten geeignete Prioritäten zu, wenn $x=14$.

Gewähltes Verfahren: _____

$P(\tau_1) =$ _____ $P(\tau_2) =$ _____ $P(\tau_3) =$ _____

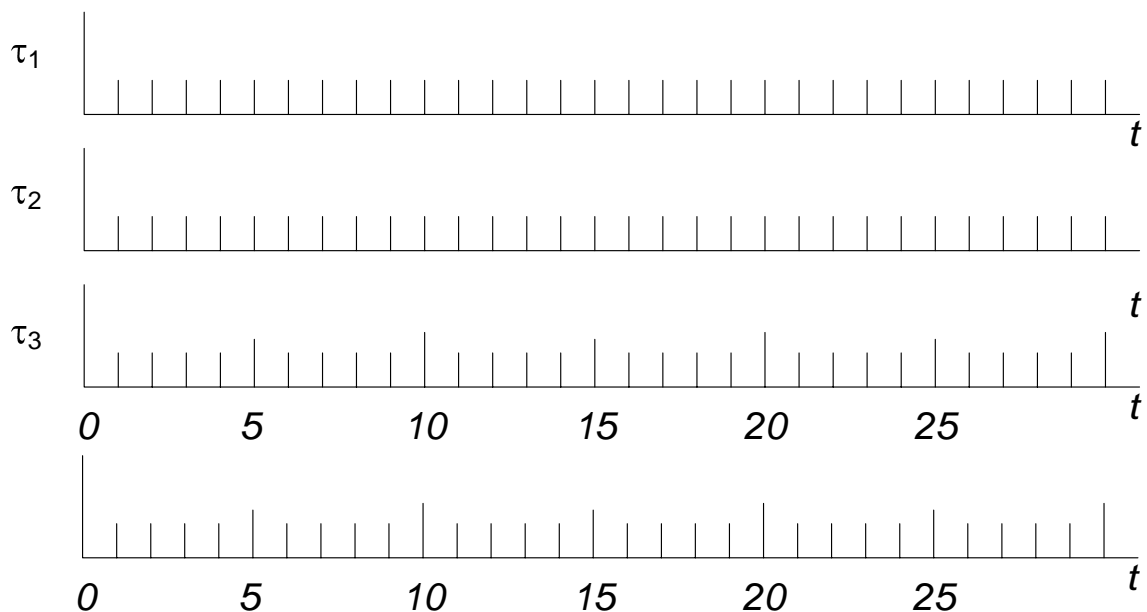
Hochschule Esslingen

Wintersemester	2007/2008	Blatt-Nr.:	5 von 8
Fachbereich:	Informationstechnik	Semester:	IT4
Prüfungsfach:	Echtzeitsysteme	Fachnummer:	4061
Hilfsmittel:	Eigene handschriftliche Aufzeichnungen, Laborberichte, Taschenrechner, Programmlistings und Zeichnungen	Zeit:	90 Minuten

- d) Wie groß darf x minimal sein, wenn das Taskset bei einem Verfahren mit zur Laufzeit unveränderlichen Prioritäten garantiert planbar sein soll?

$x_{\min} =$ _____

- e) Erstellen Sie den Plan nach dem unter d) gewählten Verfahren und zeichnen Sie ihn in das folgende Diagramm (eigene Diagramme werden nicht akzeptiert).



Hochschule Esslingen

Wintersemester	2007/2008	Blatt-Nr.:	6 von 8
Fachbereich:	Informationstechnik	Semester:	IT4
Prüfungsfach:	Echtzeitsysteme	Fachnummer:	4061
Hilfsmittel:	Eigene handschriftliche Aufzeichnungen, Laborberichte, Taschenrechner, Programmlistings und Zeichnungen	Zeit:	90 Minuten

- f) Wenn $x=11$, ist das Taskset dann noch ohne Verletzung von Echtzeitbedingungen planbar? Bitte begründen.

Antwort mit Begründung:

Aufgabe 4: Threads, reentrante Funktionen und race conditions

(25 Punkte)

Gegeben sind die folgenden zwei C-Module (rand() = ANSI-C Zufallsfunktion).

```
/* Code A */
#include <stdio.h>

static int a, d, f;
int e, b, c, g;

int computeA(i, j) {
    int a, e;
    static int g, b = 0;
    e = 5 + g + b;
    a = i + j + e;
    g = a;
    return a;
}

void theTaskA() {
    static int a = 1;
    b = 200;
    e = computeA(a, rand());
    printf("%d\n", e);
    printf("%d\n", a);
}
```

```
/* Code B */
#include <stdio.h>

static int a, d, f;
int e, b, c;

int computeB(i, j) {
    int a, g;
    static int b = 0;
    e = 5 + b + j;
    a = i + j + e;
    g = a;
    return a;
}

void theTaskB() {
    int a, b;
    a = 4;
    b = 200 + a + c;
    b = computeB(rand(), d);
    printf("%d\n", f);
    printf("%d\n", b);
}
```

Hochschule Esslingen

Wintersemester	2007/2008	Blatt-Nr.:	7 von 8
Fachbereich:	Informationstechnik	Semester:	IT4
Prüfungsfach:	Echtzeitsysteme	Fachnummer:	4061
Hilfsmittel:	Eigene handschriftliche Aufzeichnungen, Laborberichte, Taschenrechner, Programmlistings und Zeichnungen	Zeit:	90 Minuten

- a) Betrachten Sie die Funktionen `computeA` und `computeB`. Welche Funktionen sind reentrant, d.h. können von verschiedenen Threads aus gleichzeitig aufgerufen werden?

Lösung mit Begründung:

- b) Im Multitasking-Betrieb kann es durch quasi-parallel Abarbeitung von Programmteilen zu Konflikten kommen („race conditions“). Wo genau (bitte Nummern am Zeilenanfang von `theTaskA` und `theTaskB` markieren) in den obigen Beispielen können parallel ausgeführt werden (d. h. es läuft entweder `theTaskA` oder `theTaskB` in mehreren Threads auf einem Rechner). In die Software wird entweder Modul Code A eingebunden oder Modul Code B, aber nie beide.

Erklären Sie die Art des Konflikts für jede Nummer.

Lösung mit Begründung:

Hochschule Esslingen

Wintersemester	2007/2008	Blatt-Nr.:	8 von 8
Fachbereich:	Informationstechnik	Semester:	IT4
Prüfungsfach:	Echtzeitsysteme	Fachnummer:	4061
Hilfsmittel:	Eigene handschriftliche Aufzeichnungen, Laborberichte, Taschenrechner, Programmlistings und Zeichnungen	Zeit:	90 Minuten

- c) Ergänzen Sie den Code von theTaskA und theTaskB in geeigneter Weise so, dass race conditions in optimaler Weise vermieden werden. Bestehende Code-Teile dürfen Sie nicht löschen. Begründen Sie Ihr Vorgehen detailliert.

```
/* Code A */
#include <stdio.h>

static int a, d, f;
int e, b, c, g;

int computeA(i, j) {
    int a, e;
    static int g, b = 0;
    e = 5 + g + b;
    a = i + j + e;
    g = a;
    return a;
}

void theTaskA() {
    static int a = 1;
    b = 200;
    e = computeA(a, rand());
    printf("%d\n", e);
    printf("%d\n", a);
}
```

```
/* Code B */
#include <stdio.h>

static int a, d, f;
int e, b, c;

int computeB(i, j) {
    int a, g;
    static int b = 0;
    e = 5 + b + j;
    a = i + j + e;
    g = a;
    return a;
}

void theTaskB() {
    int a, b;
    a = 4;
    b = 200 + a + c;
    b = computeB(rand(), d);
    printf("%d\n", f);
    printf("%d\n", b);
}
```

Begründung der Lösung zu c):