Ostfalia Hochschule für angewandte Wissenschaften Fakultät Fahrzeugtechnik Prof. Dr.-Ing. V. von Holt Institut für Fahrzeuginformatik und Fahrzeugelektronik	Modulprüfung Embedded Systems WS 2011/2012 20.01.2012	Name:..... Vorname..... Matr.Nr.:..... Unterschrift.....

Zugelassene Hilfsmittel: **Keine**
Zeit: 60 Minuten

Punkte:

1 (20)	**2 (20)**	**3 (20)**	**Summe (60)**

Prozente Klausur (50%)	**Prozente Labor (50%)**	**Gesamtnote**

Aufgabe 1 (20 Punkte) – Kurzfragen

a) (5 P) Bei Echtzeitsystemen müssen Aktionen u.a. „**rechtzeitig**“ erfolgen. Was versteht man unter dem Begriff „**Rechtzeitigkeit**“ und welche **Varianten** der „**Rechtzeitigkeit**“ gibt es (Skizze)?

b) (5 P) Was versteht man unter „**Synchroner Programmierung**“? Nennen Sie den **Hauptvorteil** und den **Hauptnachteil** der „Synchronen Programmierung“!

c) (5 P) Nennen Sie mindestens **3 Aufgaben** eines (allgemeinen) „**Betriebssystem**“? Welche **zusätzlichen Eigenschaften** müssen „**Echtzeitbetriebssysteme** besitzen?

d) (5 P) Was unterscheidet Systeme mit **Preemptiven** bzw. **Nicht-Preemptivem** Multitasking voneinander? Wann bezeichnet man ein **Schedulingverfahren** als „**optimal**“?

Aufgabe 2 (20 Punkte) – Scheduling

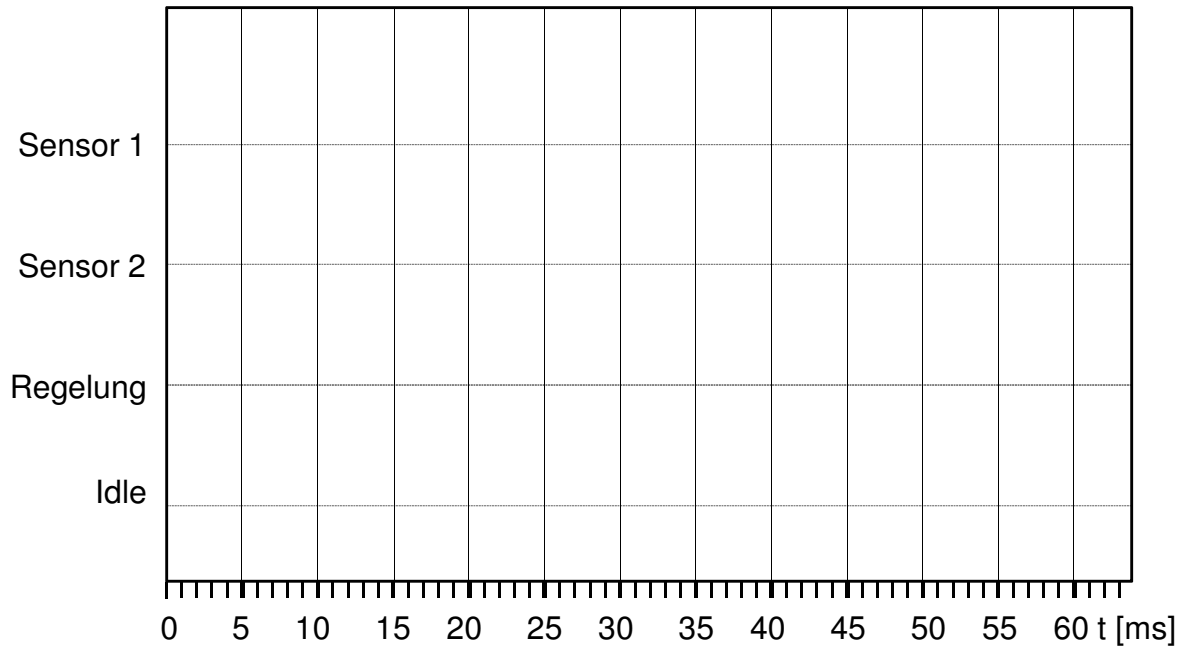
Ein Regelungssystem verfügt zur Messwerterfassung über 2 Sensoren, die in unterschiedlichen Intervallen Messwerte zur Zustandserfassung des Systems liefern. Die Sensormesswerte sowie die Regelung sollen in jeweils eigenen Tasks ablaufen. Die folgende Tabelle enthält die Zykluszeiten sowie die Laufzeiten der einzelnen Tasks:

Tasks	Zykluszeit [ms]	Laufzeit[ms]
Sensor 1	10	2…3
Sensor 2	5	1…2
Regelung	20	4

(Die Deadline der Tasks entspricht der Periodendauer/Zykluszeit.)

a) (4 P) Berechnen Sie die **Prozessorlast**, die durch das **Taskset** verursacht wird! Ist das gegebene Taskset **realisierbar**?

b) (10 P) Es soll versucht werden, das Taskset durch ein **FIFO-Scheduling** zu realisieren. Welches ist das **Worst-Case-Szenario** für das gegebene Problem?
Gehen Sie zur **Überprüfung der Schedulebarkeit** von diesem Worst-Case-Szenario aus und weisen Sie die Schedulebarkeit nach bzw. widerlegen Sie diese anhand eines **Scheduldiagramms**!



c) (6 P) Wenn das Taskset alternativ durch ein **Rate-Monotonic-Scheduling** realisiert werden soll, welche **Prioritäten** müssen den **Tasks** dann jeweils zugewiesen werden? (**Höchste Priorität : 0**) Nach welcher **Regel** werden die **Prioritäten vergeben**?
Ist das **Taskset** in jedem Fall mit RMS-Scheduling **umsetzbar**?

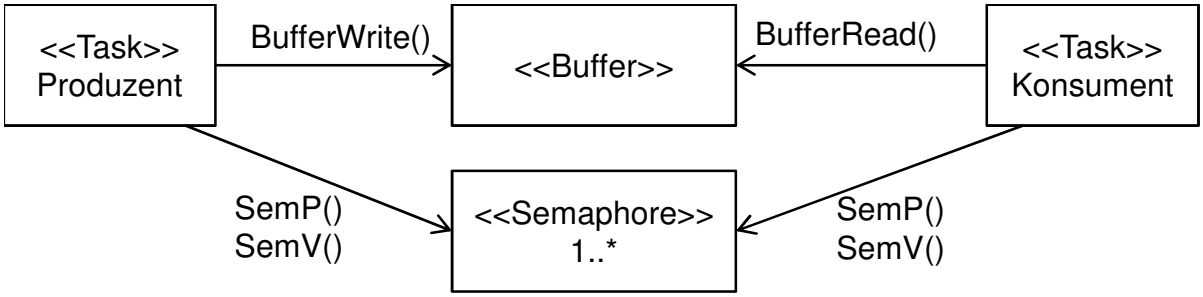
Aufgabe 3 (20 Punkte) – Semaphore / Synchronisation / Kommunikation

a) (10 P) Erläutern Sie die **Funktionsweise** eines **(Zähl-)Semaphors**! Aus welchen **Bestandteilen** besteht ein Semaphor und welche **Wirkung** haben die zugehörigen **Operationen** (Aktivitätsdiagramme)?

b) (10 P) Zwischen einer **„Produzenten“-Task** und einer **„Konsumenten“-Task** sollen Daten über einen **Puffer** ausgetauscht werden. Der Puffer fasst genau **ein Datenpaket**. Die **Reihenfolge** des **Zugriffs** auf den Puffer soll durch **Semaphore** abgesichert sein. D.h., es soll sichergestellt sein, dass stets **zuerst** der **Produzent** Daten in den Puffer **schreibt**, **dann** der **Konsument** diese Daten **liest** und erst dann der Produzent neue Daten schreiben kann.

Folgende Funktionen stehen zur Verfügung:

```
void SemP(SemId)   Passieren des Semaphors SemId.
void SemV(SemId)   Verlassen des Semaphors SemId.
void BufferWrite() Schreibt eine Nachricht in den Puffer.
int  BufferRead()  Liest eine Nachricht aus dem Puffer. Liefert die
                   Anzahl der noch im Puffer befindlichen Nachrichten
                   zurück.
void Erzeuge()     Funktion auf Produzentenseite zur Erzeugung neuer
                   Werte.
void Verarbeite()  Funktion auf Konsumentenseite zur Verarbeitung neuer
                   Werte.
```



Geben Sie den **Ablauf des Zugriffs** auf den **gemeinsamen Puffer** aus der **Produzenten-Task** und aus der **Konsumenten-Task** in Form eines **Aktivitätsdiagramms** oder in Form von **Pseudocode** an!
(Sowohl Produzent wie Konsument sollen in einer **Endlosschleife** ablaufen.)