

Name:  Vorname:  Matr.-Nr.:

**Technische Universität München**  
**Fakultät für Informatik**  
**Prof. Dr. A. Knoll**

**WS 2006/2007**  
**15. Februar 2007**

### Klausur zu Echtzeitsysteme

#### Aufgabe 1 Wissensfragen

**(30 Punkte)**

Beantworten Sie folgende Fragen in ein bis zwei Sätzen:

- a) Was ist der Unterschied zwischen harten und weichen Echtzeitsystemen?
- b) Wieso sollte Virtual Memory nicht in Echtzeitsystemen verwendet werden?
- c) Nennen Sie zwei Gründe, wieso nebenläufige Programmierung häufig in Echtzeitsystemen angewandt wird.
- d) Was sind Race Conditions?
- e) Erläutern Sie den Unterschied zwischen dem Scheduling in Echtzeitsystemen und Standard-systemen (z.B. anhand der zugrundeliegenden Kriterien).
- f) Nennen Sie zwei Konzepte von Programmiersprachen zur Prozesssynchronisation und erläutern Sie diese kurz.
- g) Nennen Sie drei veränderte, grundlegende Konzepte von Real-Time Java gegenüber Java.
- h) Wieso kann ein zentraler Synchronisationsalgorithmus nicht fehlertolerant implementiert werden?
- i) Was ist der Unterschied zwischen Sicherheit und Zuverlässigkeit. Nennen Sie ein Beispiel.
- j) Zeichnen Sie einen Regelkreis auf und benennen Sie die einzelnen Komponenten (ohne weitere Erläuterung).

**Aufgabe 2 Esterel**

**(10 Punkte)**

Gegeben ist der Esterel-Code für folgende Module:

```
module xxx:
  input U, D;
  output E,F,N;
  var V=1;
  loop
    await
      case U do
        if (?V>0)
          V:=V+1;
        else
          V:=V+1;
          emit F;
        case D do
          if (?V>0) then
            V:=V-1;
            emit E;
          else
            emit N;
          end await;
        end loop;
      end await;
  end loop;
.

module yyy:
  input E,F,N;
  output U,D;
  loop
    emit D;
    await
      case E do
        ...
        emit U;
      case N
        await F;
      end await;
  end loop;
.
```

a) Vervollständigen Sie folgende Testfälle, so dass das Modul xxx diese Testfälle erfüllt:

- T1=({D} , \_\_\_ ) , ( \_\_\_ , { F } )
- T2=({D} , \_\_\_ ) , ( { D } , \_\_\_ )
- T3=( \_\_\_ , { E } ) , ( \_\_\_ , { F } ) , ( \_\_\_ , { } )
- T4=( \_\_\_ ) , ( \_\_\_ , { N } )
- T5=( \_\_\_ ) , ( \_\_\_ , { E } ) , ( \_\_\_ , { E } )

Zur Erinnerung: (A,B),(C,D) bedeutet: im ersten Moment erfolgt das Ereignis A als Eingabe, die Reaktion des Moduls ist B, im zweiten Moment erfolgt das Ereignis C als Eingabe mit der Reaktion D.

- b) Welcher Mechanismus wird in den Modulen xxx bzw. yyy implementiert.
- c) Welche ist die zentrale Eigenschaft des Await-Statements, die besonders im Module xxx von Bedeutung ist?

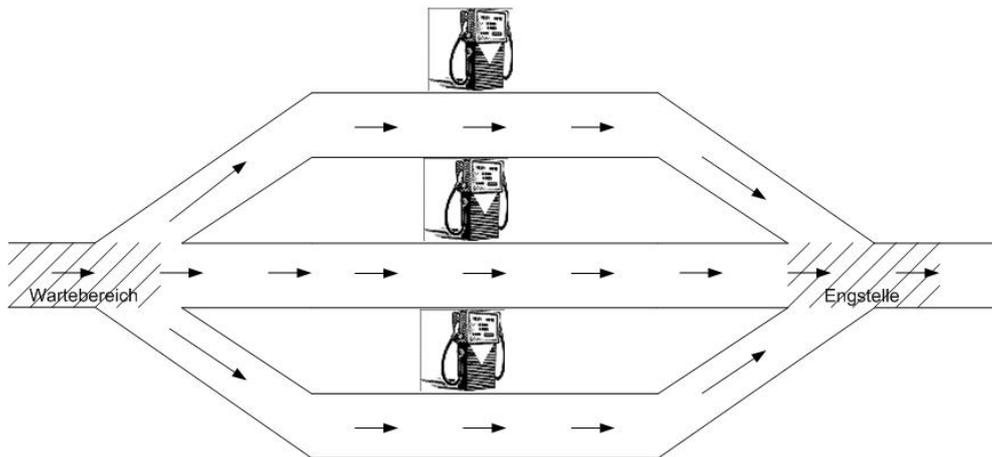


Abbildung 1: Aufgabe a,b

### Aufgabe 3 Nebenläufigkeit

(15 Punkte)

Gegeben Sie folgendes Szenario (siehe auch Abbildung): eine Tankstelle hat drei Tanksäulen, wobei jede Parksäule nur den Platz für maximal ein Auto bietet. Vor der Tankstelle befindet sich eine Engstelle, die genau von einem Auto als Wartebereich benutzt werden kann. Hinter den Tankstellen befindet sich ebenfalls eine Engstelle, so dass immer nur ein Auto gleichzeitig losfahren kann.

In den folgenden Teilaufgaben sollen Sie nun das Problem der Synchronisation lösen. Die Synchronisation der Autos vor der ersten Engstelle und nach der zweiten Engstelle ist nicht Teil der Aufgabe. Außerdem fährt das Auto selbstständig an eine freie Zapfsäule, es muss nur sichergestellt werden, dass mindestens eine der drei Zapfsäulen frei ist.

- Geben Sie die notwendigen Semaphore (mitsamt Initialisierung) an, um das gegebene Problem zu lösen.  
Beispiel: `semAuto(1)` würde bedeuten, Sie verwenden einen Semaphor `semAuto`, der mit 1 initialisiert ist.
- Ergänzen Sie den folgenden Autoprozess mit passenden `up()` und `down()`-Methoden, um Kollisionen zu vermeiden. Achten Sie darauf, dass es zu keiner Verklemmung kommt. **Anmerkung:** Es muss nicht an jeder freien Stelle Code eingefügt werden.  
Beispiel: `1: down(semAuto); up(semAuto);` bedeutet das Einfügen der beiden Operationen in Zeile 1.
- Aufgrund einer Baustelle ist die Ausfahrt blockiert (siehe Abbildung), so dass die Wartebereich sowohl zur Einfahrt, als auch zur Ausfahrt genutzt werden muss. Ergeben sich notwendige Änderungen im Vergleich zur Lösung der Aufgabe b) und wenn ja welche?

---

**Algorithm 1** Prozess: tankendes Auto

---

```
1:  
2: fahreInWartebereich();  
3:  
4: fahreAnZapfsaeule();  
5:  
6: tanke();  
7:  
8: bezahle();  
9:  
10: fahreInEngstelle2();  
11:  
12: verlasseEngstelle2();  
13:
```

---

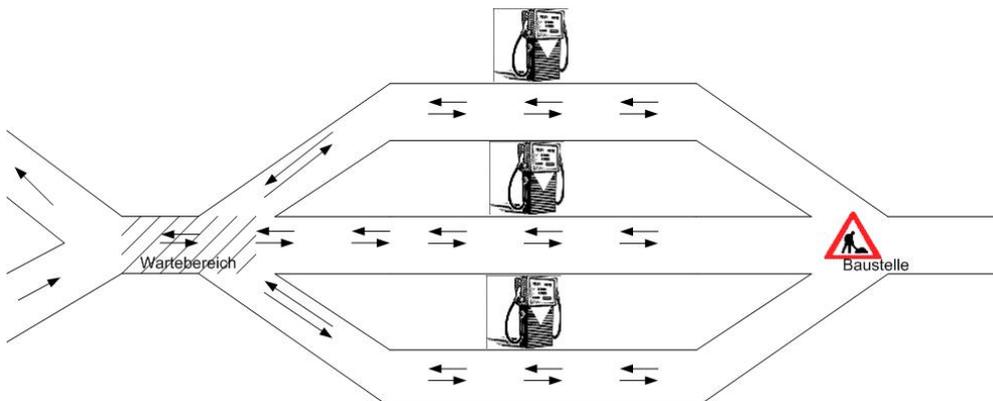


Abbildung 2: Aufgabe c

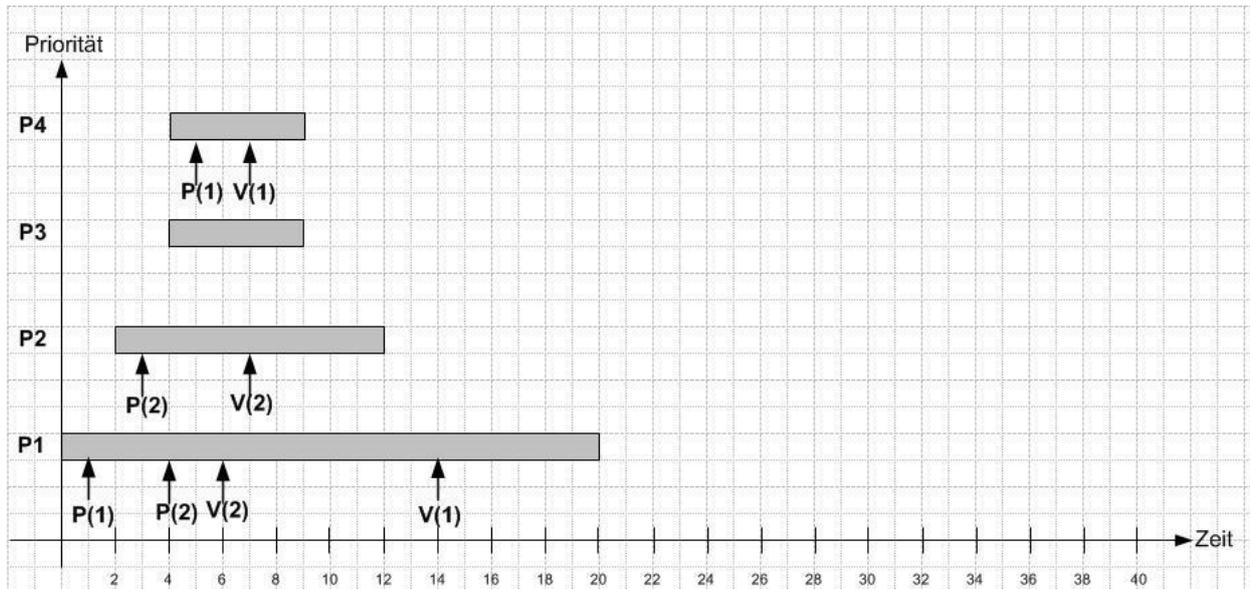


Abbildung 3: Szenario

**Aufgabe 4 Scheduling**

**(15 Punkte)**

Gegeben sei das folgende Szenario entsprechend der Abbildung. 4 Prozesse werden nebenläufig ausgeführt und fordern binäre Semaphore an. Die Abbildung zeigt die Ausführung jedes einzelnen Prozesses, falls dieser als einziger ausgeführt werden würde:

- Startzeiten  $s$ :  $s(P1)=0$ ;  $s(P2)=2$ ;  $s(P3)=4$ ;  $s(P4)=4$ ;
- Ausführungszeiten  $e$ :  $e(P1)=20$ ;  $e(P2)=10$ ;  $e(P3)=5$ ;  $e(P4)=5$ ;
- Prioritäten  $prio$ :  $prio(P1)=1$  (niedrigste);  $prio(P2)=2$ ;  $prio(P3)=3$ ;  $prio(P4)=4$  (höchste);

Zudem fordern die Prozesse auch binäre Semaphore an (zu Beginn sind alle Semaphore freigegeben). Die Zeitpunkt für Anforderung (P) und Freigabe (V) sind dabei in der nachfolgenden Tabelle relativ zur eigenen Ausführungszeit angegeben.

	S1		S2	
	P	V	P	V
P1	1	14	4	6
P2	-	-	1	5
P3	-	-	-	-
P4	1	3	-	-

- Ein möglicher Ausführungsplan für ein Ein-Prozessorsystem ist in Abbildung Scheduling xxx gezeichnet. Welches Schedulingverfahren wurde hier angewandt. **Hinweis:** Die Zahl im jeweiligen Ausführungsblock bezeichnet dabei die Priorität.
- Zeichnen Sie den Ausführungsplan (analog zu dem angegebenen) für ein Ein-Prozessor-System mit präemptiven, prioritätenbasierten Scheduling.

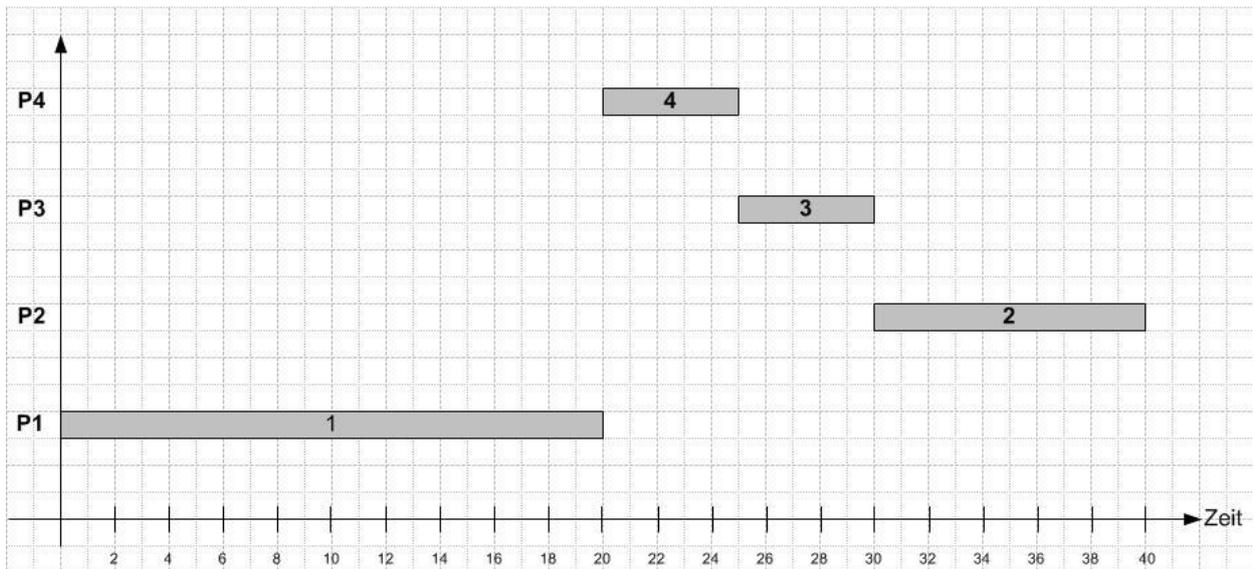


Abbildung 4: Ausführungsplan bei Scheduling xxx

- c) Welches Problem tritt auf und wie kann es beispielsweise gelöst werden.
- d) Zeichnen Sie den Ausführungsplan mit ihrer Lösung entsprechend Teilaufgabe b. **Hinweis:** Zeichnen Sie die Ausführungszeiten zur einfacheren Zuordnung weiterhin in der entsprechenden Zeile des Prozesses.
- e) Betrachten Sie nun die Anwendung: welche prinzipielle Designregel wurde hier verletzt?

## **Aufgabe 5 Echtzeitbetriebssysteme**

**(10 Punkte)**

Gegeben seien folgende fünf Echtzeitbetriebssysteme:

1. TinyOS
2. OsekTime
3. QNX
4. VxWorks
5. WindowsCE

und folgende fünf Anwendungen:

- a) Sicherheitsüberwachung für Robotersteuerung: auf Basis eines Controllers mit geringen Rechen- und Speicherkapazitäten wird eine Anwendung zur Überwachung der Robotersteuerung implementiert. Dringt eine Person in den Arbeitsbereich des Roboters ein, so wird dieses Ereignis durch Lichtschranken detektiert und der Roboter unverzüglich gestoppt.
- b) ZebraNet: Zur Erforschung der Wanderwege von Zebras, werden kleine Funkmodule ausgestattet mit GPS-Sensoren zur Lokalisation und Solarzellen zur Stromversorgung an Zebras angebracht. Nähern sich zwei Zebras, so tauschen die Funkmodule die gesammelten Daten aus.
- c) Zugsteuerung: Zur Steuerung und Überwachung eines Zuges werden in einem verteilten System verschiedene periodische Regelungsfunktionen ausgeführt. Diese Funktionen werden dabei als Komponenten von unterschiedlichen Entwicklergruppen zur Verfügung gestellt. Insbesondere eine reibungslose Integration dieser Komponenten steht im Vordergrund.
- d) Fabrikanlagensteuerung: In einer Chemiefabrik wird die Produktion von leistungsfähigen Feldrechnern gesteuert. Diese Rechner sind mit einer Vielzahl von Sensorik verbunden. Auf kritische Ereignisse muss schnell reagiert werden.
- e) PDA: Auf einem Handheld werden unterschiedlichste Anwendungen ausgeführt: unter anderem Routenplaner, Browser, Musik-/Videospiele und Programme zur Terminverwaltung.

Ordnen Sie jedem der fünf Anwendungsgebiete ein Echtzeitbetriebssystem zu und begründen Sie Ihre Zuordnung knapp mit einem Satz.

## Aufgabe 6 Echtzeitfähige Kommunikation

(20 Punkte)

Gegeben sei ein ringförmiges Netzwerk mit 4 Teilnehmern, T1 bis T4, die im Uhrzeigersinn angeordnet sind. Die Prioritäten der Nachrichten der einzelnen Sender steigt von T1 (Nachrichten mit niedrigster Priorität) zu T4 (Nachrichten mit höchster Priorität) an.

Für die Aufgabe kann davon ausgegangen werden, dass die Übertragung eines Bits eine Zeiteinheit dauert. Die für die Ausbreitung der Signale, sowie zur Verarbeitung der Signale in den einzelnen Knoten benötigten Zeiten können in dieser Aufgabe vernachlässigt werden.

Szenario:

- Teilnehmer T1 will zum Zeitpunkt 48 eine Nachricht mit 16-bit Nutzdaten schicken.
  - Teilnehmer T2 will zum Zeitpunkt 48 eine Nachricht mit 16-bit Nutzdaten schicken.
  - Teilnehmer T3 will zum Zeitpunkt 48 eine Nachricht mit 16-bit Nutzdaten schicken.
  - Teilnehmer T4 will zum Zeitpunkt 16 eine Nachricht mit 16-bit Nutzdaten schicken.
- a) Geben Sie die **Reihenfolge** der Nachrichten an, die im Netzwerk bei Verwendung des CAN-Protokolls gesendet werden und begründen Sie ihre Antwort. Zur Erinnerung: Zusätzlich zu den Nutzdaten sind bei CAN 46 Bit Steuerungsdaten pro Nachricht notwendig. Zwischen den einzelnen Nachrichten ist eine Lücke von mindestens 3 Bit.
- b) Geben Sie die **Reihenfolge** der Nachrichten an, die im Netzwerk bei Verwendung des TokenRing-Protokolls gesendet werden und begründen Sie ihre Antwort. Zum Zeitpunkt 0 soll dabei der Teilnehmer T1 im Besitz des Tokens sein. Zur Erinnerung: Ein Token besteht aus insgesamt 3 Byte (8 Bit Startbegrenzer, 8 Bit Zugriffskontrolle mit Zugriffspriorität und Reservierungspriorität, 8 Bit Endbegrenzer).
- c) Was ist der Unterschied zwischen dominanten und rezessiven Bits.
- d) Nennen Sie zwei Mechanismen zur Bitsynchronisierung und erklären Sie diese.
- e) Was ist der Unterschied zwischen CSMA/CD und CSMA/CA?