
Zentralübung Echtzeitsysteme

Wintersemester 2010/2011

Dr. Christian Buckl

fortiss GmbH – Cyber-Physical Systems

TU München - Lehrstuhl VI Robotics and Embedded Systems

Zentralübung - Hintergrund

- Ziele:
 - Klärung offener Fragen
 - Anwendung der gelernten Inhalte
 - Ansprechpartner für Probleme
- Methodik:
 - Interaktive Gestaltung: Ihre Mitwirkung ist wichtig
 - Hands-On Übungen: Werkzeuge, Programmierung
 - Durchsprache von Klausuraufgaben

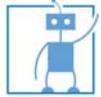


Kapitel 1

Einführung Echtzeitsysteme

Klausurfragen

- Klausur WS 06/07
 - Was ist der Unterschied zwischen harten und weichen Echtzeitsystemen? (3 Punkte = 3 min)
 - Wieso sollte Virtual Memory nicht in Echtzeitsystemen verwendet werden? (3 Punkte = 3 min)
- Wiederholungsklausur WS 06/07 (5 Punkte = 5 min)
 - Ordnen Sie folgende Anwendungen in die Kategorien harte bzw. weiche Echtzeitsysteme ein und begründen Sie Ihre Antwort:
 - Ampelsteuerung
 - Flugzeugregelung
 - Internettelefonie



Kapitel 2

Modellierung von Echtzeitsystemen und Werkzeuge

Fragen zur Vorlesung

- Was ist der Vorteil von ereignisgesteuerten Applikationen gegenüber zeitgesteuerten?

Zeitgesteuerte Applikationen	Ereignisgesteuerte Applikationen
Zeitlicher Systemablauf wird zur Übersetzungszeit festgelegt.	Ausführungen werden durch das Eintreten von Ereignissen angestoßen.
Präzise und globale Uhr erforderlich (inkl. Uhrensynchronisation)	Garantierte Antwortzeiten sind erforderlich.
Einzelberechnungen werden in einem jeweils reservierten Zeitslot durchgeführt. Ableitung der max. Laufzeit notwendig. (worst case execution time)	Das Scheduling erfolgt dynamisch. Keine Aussage über zeitlichen Ablauf zur Übersetzungszeit möglich.
<u>VORTEIL:</u> Statisches Scheduling möglich. Vorhersagbares deterministisches Verhalten.	<u>VORTEIL:</u> ???

Häufige Fragen zum Stoff der Vorlesung

- Was sind Aktoren?
 - Unterscheidung zum Begriff Aktoren aus der Mechatronik?
- Wie zuverlässig ist der generierte Code bei modellbasierten Entwicklungswerkzeugen?
- Wie gut ist der generierte Code zu lesen?
- Frage zu dieser Vorlesung:
 - Wann verwendet man synchrone Sprachen, wann synchronen Datenfluss?
 - Unterscheidung zwischen kontrollorientierten (Fragestellung: wie reagiert das System auf Ereignisse) und datenorientierten (Fragestellung: wie werden eingehende Daten verarbeitet) Anwendungen.

Fragen zur letzten Vorlesung

- Wieso wird Rekursion in Esterel nicht unterstützt?
- Wie funktioniert die Umsetzung des Await-Statements?

```
module Temperature:
input IN1,IN2;
output OUT1,OUT2;
  loop
    await IN1; emit OUT1;
  end loop;
||
  loop
    await IN2; emit OUT2;
  end loop;
```



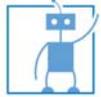
Auszug aus Klausur WS 06/07 (7 Punkte = 7 min)

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

1. T1=({D},___),(___,{F})
2. T2=({D},___),({D},___)
3. T3=(___,{E}),(___,{F}),(___,{})
4. T4=(____),(___,{N})
5. 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.

```
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;
```



Kapitel 3

Nebenläufigkeit

Klausur WS06/07 – Nebenläufigkeit (15 Punkte = 15min)

Prozess: tankendes Auto

fahreInWartebereich();

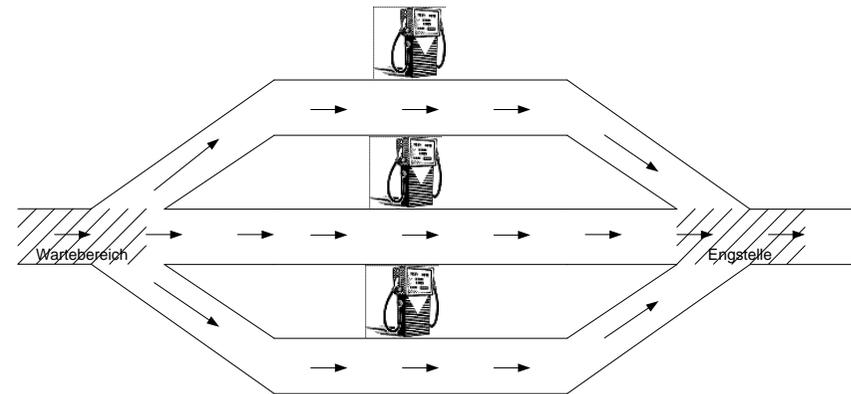
fahreAnZapfsaeule();

tanke();

bezahle();

fahreInEngstelle2();

verlasseEngstelle2();



- 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(semaAuto); up(semaAuto);` bedeutet das Einfügen der beiden Operationen in Zeile 1.

Klausur WS06/07 - Nebenläufigkeit

- c) 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?

Prozess: *tankendes Auto*

fahreInWartebereich();

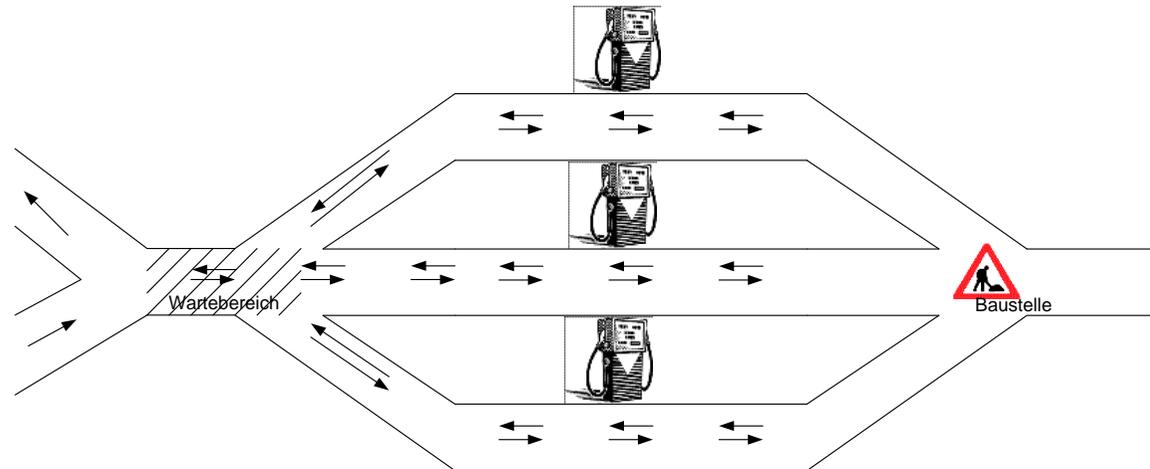
fahreAnZapfsaeule();

tanke();

bezahle();

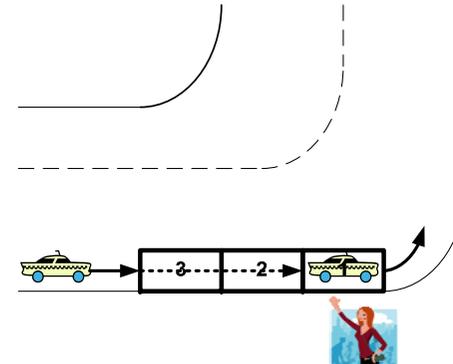
fahreInEngstelle2();

verlasseEngstelle2();

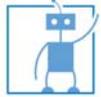


Klausur WS07/08 – Nebenläufigkeit (20 Punkte = 20min)

- Gegeben Sie folgendes Szenario: am Münchner Odeonsplatz gibt es eine Wartebucht für Taxis. Zur Vereinfachung gehen wir davon aus, dass die Wartebucht aus drei Plätzen besteht und immer nur ein Passagier gleichzeitig auf ein Taxi wartet. Passagiere steigen an der ersten Wartebucht ein, die Taxis rücken nach, sobald das Taxi vor ihnen losgefahren ist. Implementieren Sie nun schrittweise eine Prozesssynchronisation, so dass es zu keinen Auffahrunfällen kommt, die Taxis in der Ankunftsreihenfolge auch wieder losfahren, Taxis nur mit Passagier losfahren, Passagiere nicht aus Versehen ein nicht-existentes Taxi betreten und es zu keinen Verklemmungen kommt.



- Notieren Sie die wichtigen Programmabschnitte des Taxiprozesses und des Passagierprozesses. Lassen Sie genügend Platz für spätere Synchronisationsoperationen.
Beispiel: `fahreInErsteWartebucht()` ;
- Geben Sie die zur Synchronisation der Taxis und Passagiere benötigten Semaphore, sowie der Initialwerte an. Gehen Sie dabei davon aus, dass zu Beginn kein Taxi in der Wartebucht und keine wartenden Passagiere vorhanden sind.
Beispiel: `semTaxi(1)` würde bedeuten, Sie verwenden einen Semaphor `semTaxi`, der mit 1 initialisiert ist.
`int i=0`; wenn sie eine ganzzahlige Variable mit Initialisierungswert 1 benutzen wollen.
- Ergänzen Sie den Taxiprozess und Passagierprozess mit passenden `up()` und `down()`-Methoden, um die Aufgabenstellung zu erfüllen.
Beispiel: `down(semTaxi)` ; bedeutet das Anfordern des Semaphors `semTaxi`
Beispiel: `up(semTaxi)` ; bedeutet das Freigeben des Semaphors `semTaxi`
- Der Wartebereich am Odeonsplatz ist begrenzt. Stellen Sie sicher, dass maximal 3 Taxis auf Fahrgäste warten und kein Rückstau entsteht. Die Überprüfung ob der Wartebereich belegt ist, soll dabei so schnell wie möglich erfolgen um den Straßenverkehr nicht zu behindern. Andererseits, sollen die Taxifahrer auf jeden Fall in den letzten Wartepplatz fahren, falls dieser frei ist.



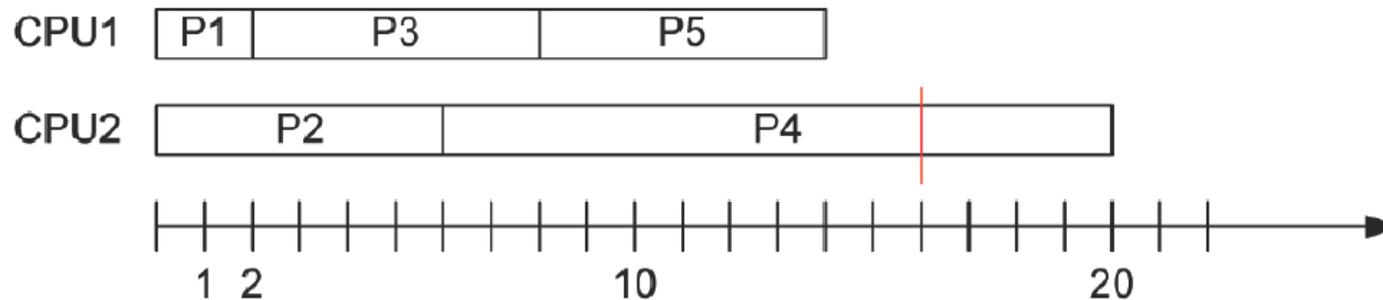
Kapitel 4

Scheduling

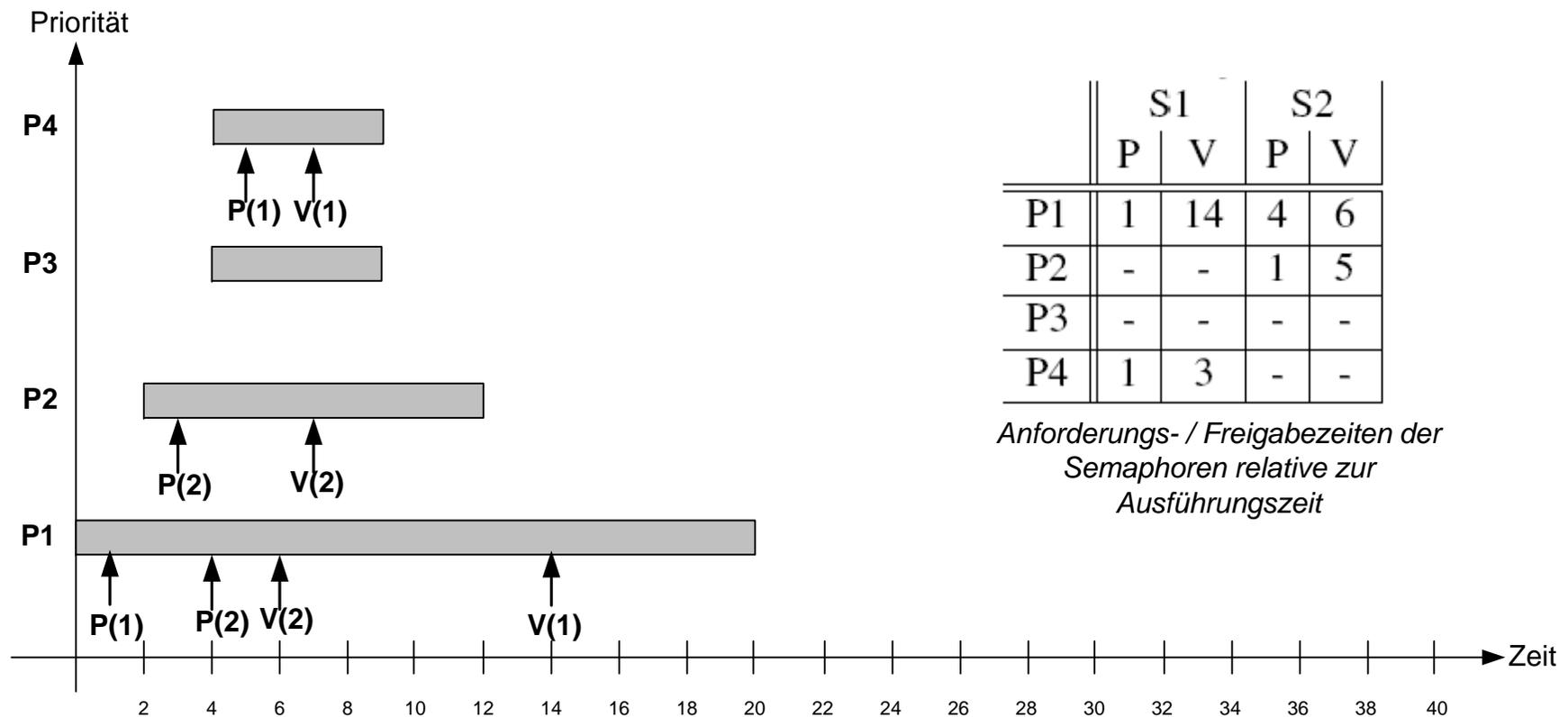
Klausur SS 07 – Szenario (20 Punkte = 20 min)

- Welches Schedulingverfahren wurde verwendet? Welche Änderungen würden sich ergeben, wenn das Verfahren präemptiv wäre?
- Welche optimalen Schedulingverfahren existieren für Mehrprozessorsysteme?
- Welche Voraussetzungen müssen für ein optimales Schedulingverfahren in Mehrprozessorsystemen erfüllt sein?
- Zeichnen Sie einen unter Zuhilfenahme eines optimalen Schedulingplanes einen korrekten Ausführungsplan.
- In der Praxis werden diese Schedulingverfahren nicht angewandt. Was spricht dagegen und welcher Ansatz wird stattdessen gewählt?

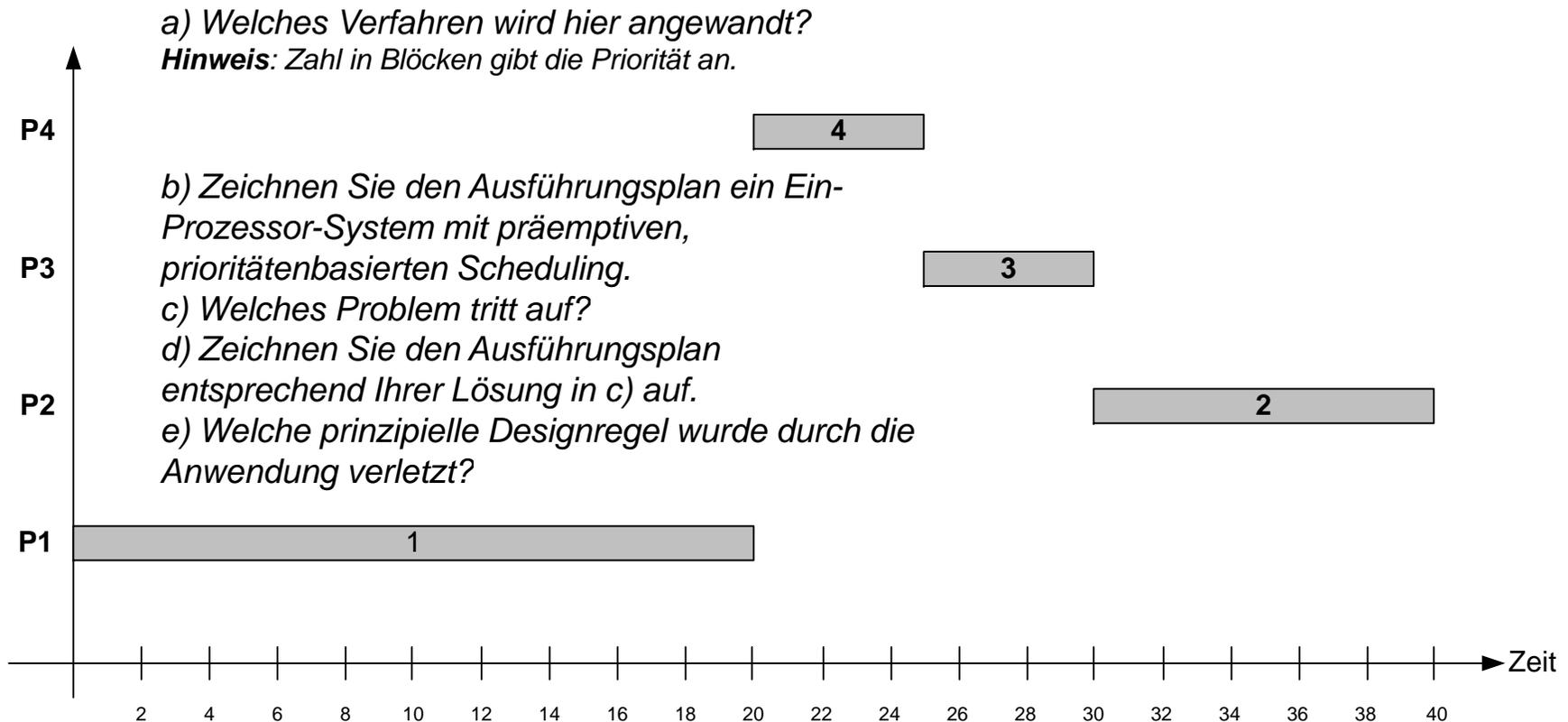
Startzeiten s : $s(P1)=0$; $s(P2)=0$; $s(P3)=0$; $s(P4)=0$; $s(P5)=0$;
 Ausführungszeiten e : $e(P1)=2$; $e(P2)=6$; $e(P3)=6$; $e(P4)=14$; $e(P5)=6$;
 Deadlines d : $d(P1)=4$; $d(P2)=8$; $d(P3)=12$; $d(P4)=16$; $d(P5)=18$;



Klausur WS 06/07 – Szenario (15 Punkte = 15min)



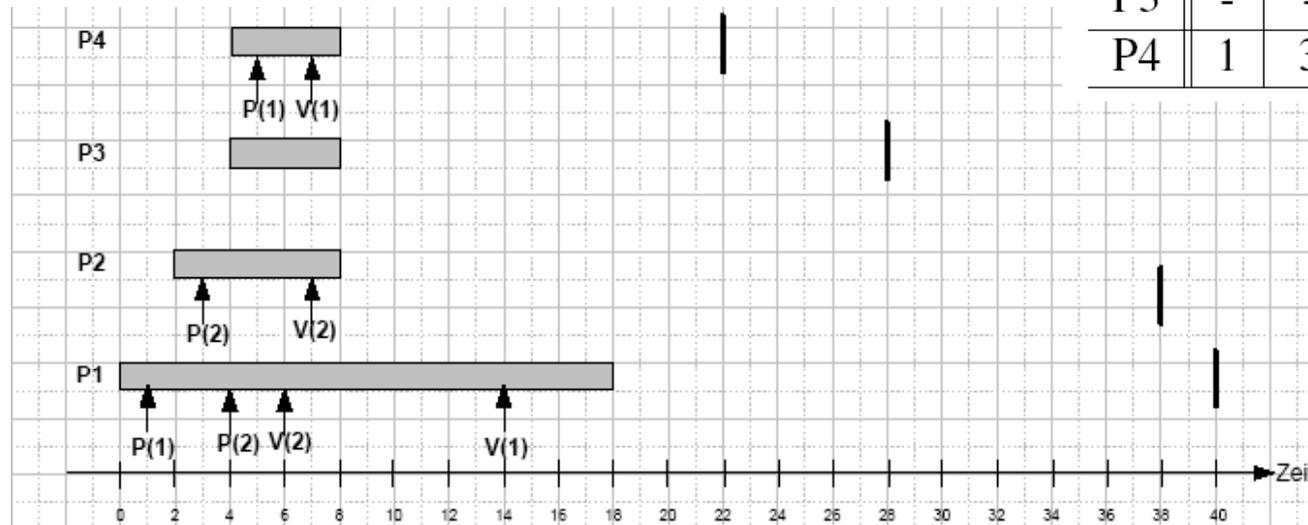
Fortsetzung – Möglicher Ausführungsplan für ein 1-Prozessor-System



Klausur WS 07/08 (20 Punkte = 20min)

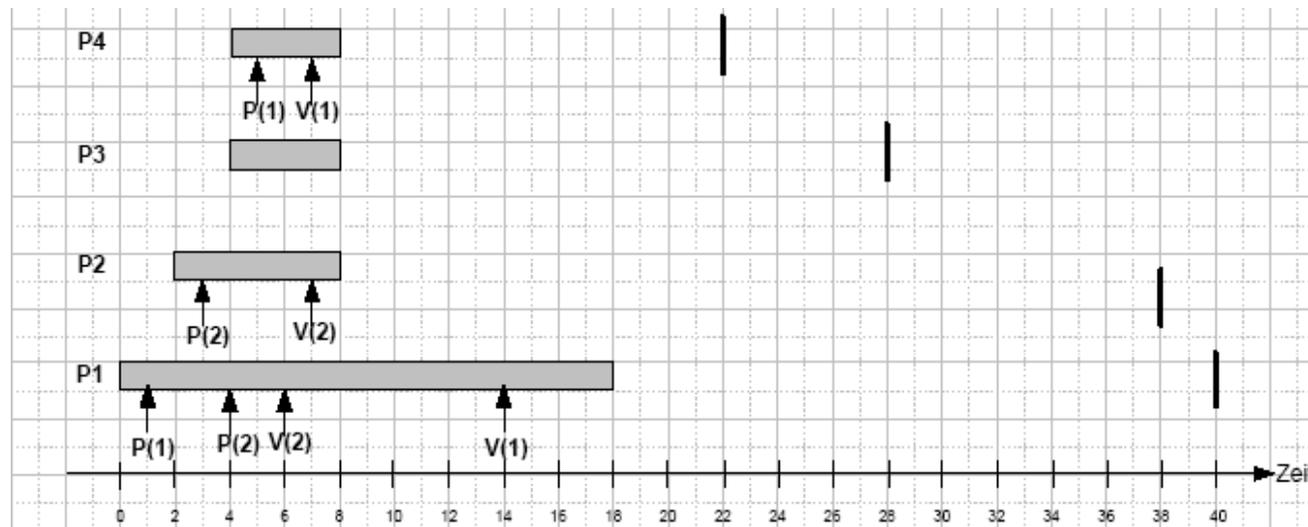
- Startzeiten s : $s(P1)=0$; $s(P2)=2$; $s(P3)=4$; $s(P4)=4$;
- Ausführungszeiten e : $e(P1)=18$; $e(P2)=6$; $e(P3)=4$; $e(P4)=4$;
- Fristen d : $d(P1)=40$; $e(P2)=38$; $e(P3)=28$; $e(P4)=22$;

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



Klausur WS 07/08

- Ignorieren Sie zunächst die Semaphore. Zeichnen Sie einen Ausführungsplan für das Schedulingverfahren Earliest-Deadline-First.
- Ignorieren Sie zunächst die Semaphore. Zeichnen Sie einen Ausführungsplan für das Schedulingverfahren Least-Slack-Time (Zeitscheiben: 1).
- Zeichnen Sie nun den Ausführungsplan für das Schedulingverfahren Earliest-Deadline-First unter Berücksichtigung der Semaphoren.
- Wie könnte man das Schedulingverfahren modifizieren, um das in Teilaufgabe c) aufgetretene Problem zu beheben.

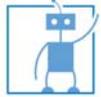


Übungsaufgabe Scheduling

Prozess	Startzeit	CPU-Zeit	Statische Priorität
P1	0	8	1 (niedrig)
P2	2	2	3
P3	2	5	4 (hoch)
P4	4	3	2

Gegeben seien die in der Tabelle angegebenen Prozesse:

1. Zeichnen Sie einen Ausführungsplan für nicht präemptives, prioritätenbasiertes Scheduling (FIFO)?
2. Zeichnen Sie einen Ausführungsplan für präemptives, prioritätenbasiertes Scheduling (FIFO).
3. Zeichnen Sie einen Ausführungsplan für präemptives, prioritätenbasiertes Scheduling (Round Robin, Zeitscheiben 0.5).
4. Zeichnen Sie einen Ausführungsplan für präemptives, prioritätenbasiertes Scheduling (FIFO), wenn jeder Prozess zu Beginn den Semaphor S anfordert und bei Beendigung der Ausführungszeit freigibt und das Betriebssystem Prioritätsvererbung unterstützt.
5. Zeichnen Sie einen Ausführungsplan für präemptives, prioritätenbasiertes Scheduling (FIFO), wenn jeder Prozess zu Beginn den Semaphor S anfordert und bei Beendigung der Ausführungszeit freigibt und das Betriebssystem immediate priority ceiling unterstützt.

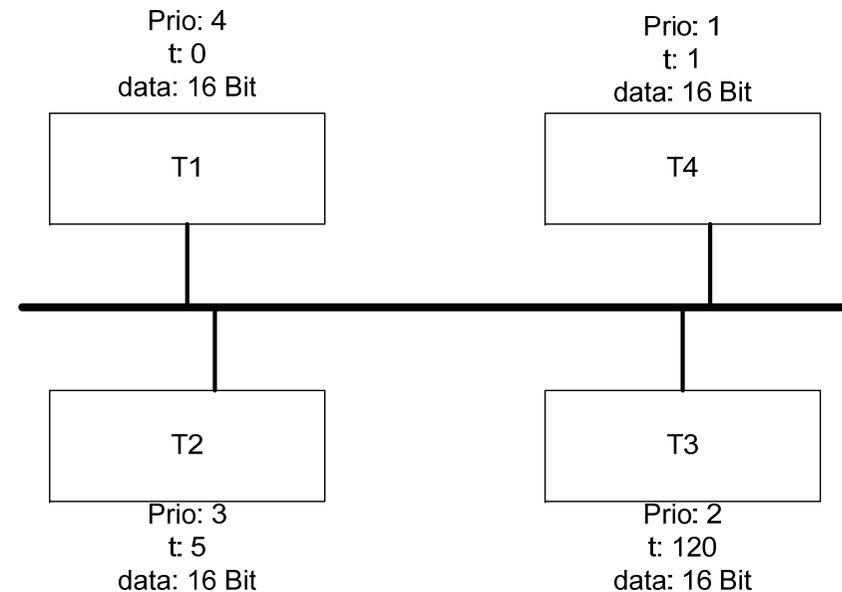


Kapitel 8

Echtzeitfähige Kommunikation

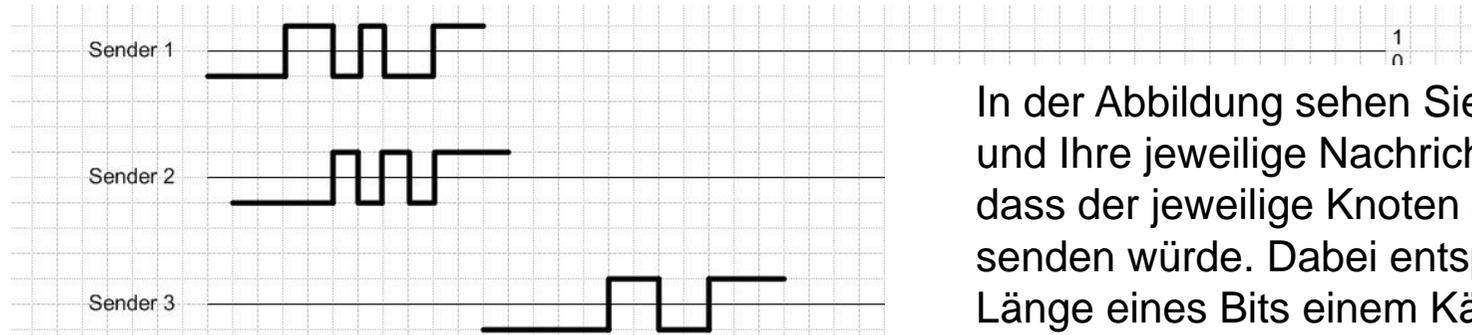
Klausur 06/07 (modifiziert) – CAN (8 Punkte = 8 min)

- a) Geben Sie die Reihenfolge der Nachrichten an, die im Netzwerk bei Verwendung des CANProtokolls 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.

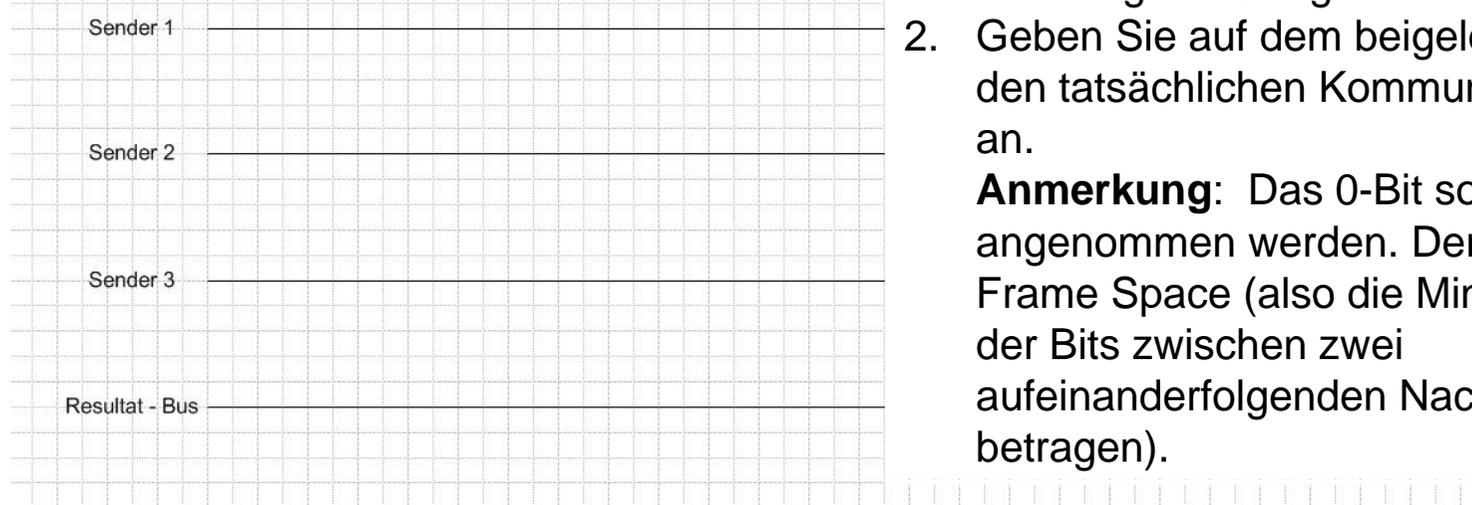


*Annahmen: Bitsendedauer 1 Zeiteinheit
Priorität: 1 – hoch, 4 – niedrig*

Wiederholungsklausur SS 07 – CAN-Protokoll (15 Punkte = 15min)



Tatsächlicher Kommunikationsablauf:

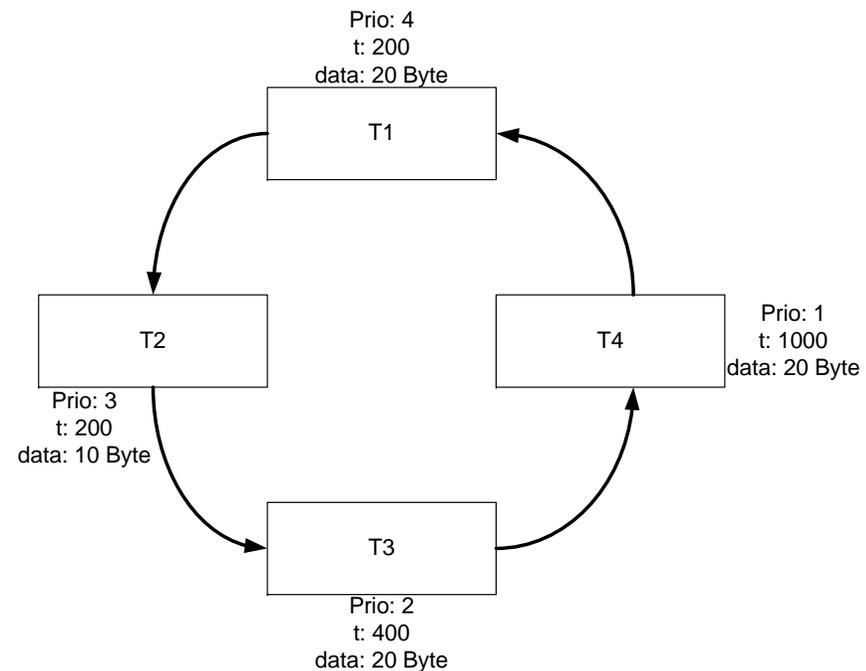


- In der Abbildung sehen Sie drei Knoten und Ihre jeweilige Nachricht für den Fall, dass der jeweilige Knoten als einziger senden würde. Dabei entspricht die Länge eines Bits einem Kästchen.
1. In welcher Reihenfolge würden die Nachrichten gesendet werden, wenn alle Knoten gleichzeitig senden würden?
 2. Geben Sie auf dem beigelegten Blatt den tatsächlichen Kommunikationsablauf an.

Anmerkung: Das 0-Bit soll als dominant angenommen werden. Der Intermission Frame Space (also die Mindestanzahl der Bits zwischen zwei aufeinanderfolgenden Nachrichten soll 3 betragen).

Klausur 06/07 (modifiziert) – TokenRing (8 Punkte = 8 min)

- a) 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).
Der Header für ein Datenpaket besteht aus mindestens 20 Byte.



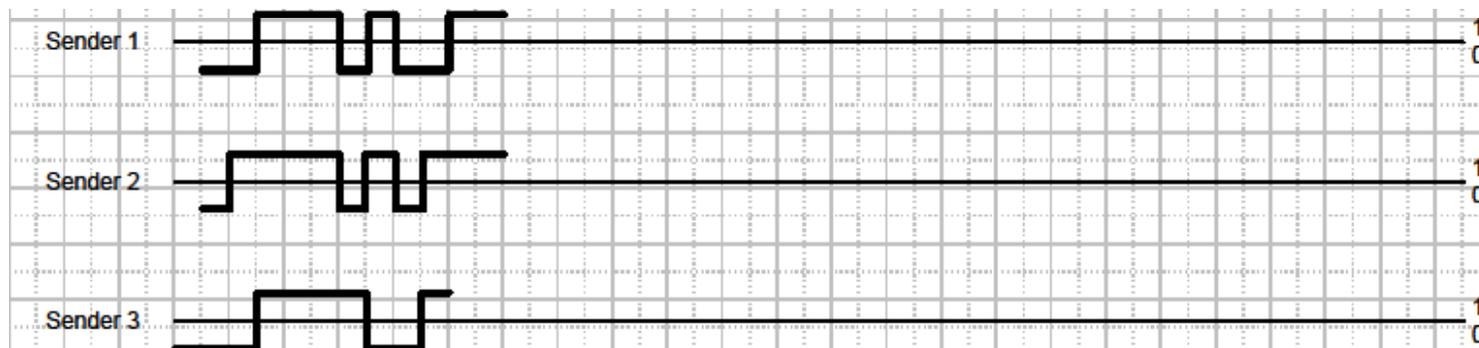
*Annahmen: Bitsendedauer 1 Zeiteinheit
Laufzeit zwischen 2 Knoten 200 Zeiteinheiten
Priorität: 1 – hoch, 4 – niedrig*

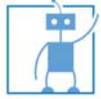
Klausur Wintersemester 07/08 (20 Punkte = 20min)

In der Abbildung sehen Sie drei Knoten und Ihre jeweilige Nachricht für den Fall, dass der jeweilige Knoten als einziger senden würde. Dabei entspricht die Länge eines Bits einem Kästchen.

Gehen Sie davon aus, dass für die Lösung der Aufgabe alle Daten bitsynchron übertragen werden. Das JAM-Signal soll aus einer Folge von 5 0-Bits bestehen. Das 0-Bit ist dominant. Zwischen zwei Nachrichten gibt es eine Pause (interframe gap) von mindestens 3 Bits.

- Zeigen Sie für die angegebenen Nachrichten einen möglichen Ablaufplan in CSMA-CD.
- Geben Sie den entsprechenden Plan in CSMA-CA an.
- Für ein konkretes Netzwerk ist die maximale Signallaufzeit mit einer Zeiteinheit angegeben. Welche der angegebenen Bitübertragungsdauern würden Sie für CSMA/CA auswählen. Geben Sie eine knappe Begründung für Ihre Antwort.
 - 0,5 Zeiteinheiten
 - 1 Zeiteinheit
 - 4 Zeiteinheiten
 - 10 Zeiteinheiten



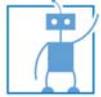


Kapitel 5

Echtzeitbetriebssysteme

Klausurfragen

- Wiederholungsklausur WS 2006/2007 (5 Punkte = 5 min)
 - Erläutern Sie die Unterschiede zwischen Betriebssystemen mit kooperativem Scheduling, mit präemptiven Scheduling und präemptiblen Betriebssystemen.
- Klausur WS 2007/2008 (4 Punkte = 4 min)
 - Erläutern Sie kurz (jeweils 1-2 Sätze) die Hauptkonzepte von TinyOS, QNX und PikeOS.



Fragestunde Klausur

Offene Fragen

- Zur Klausur WS 06/07 Semaphoren mit priority ceiling: Warum bekommt P1 die Semaphore 2 zugeteilt? Er hat in diesem Moment die Priorität 4 erhalten ($\text{actprio}(P1)=4$) und das $\text{aktceil} = 4$, da P1 Semaphore 1 blockiert. Als dürfte er die Semaphore doch nicht bekommen, da $\text{actprio}(P1) > \text{aktceil}$ nicht erfüllt ist.
- CSMA/CA und somit CAN und tokenbasierte Verfahren: Kann von Echtzeitfähigkeit gesprochen werden? Auf Folie 340 steht, dass CSMA/CA und Token nicht deterministisch sind. Kann etwas echtzeitfähig sein obwohl es nicht deterministisch ist?
- Folien 120 und 121: Was bedeutet formale Methode und funktionale Methode? Ist "Testen" beides?

Übungsaufgabe Esterel (basierend auf Klausur WS 07/08)

1. Modellieren Sie einen Automaten und schreiben Sie ein Esterel-Programm, der das Rendezvouskonzept umsetzt. Die Prozesse, die diese Komponente benutzen, müssen Sie nicht modellieren.
2. Modellieren Sie einen Automaten und schreiben Sie ein Esterel-Programm, der das Leser-Schreiber-Problem für 1 Schreiber und beliebig viele Leser mit Schreiberpriorität umsetzt.

Hinweise zur Vorgehensweise:

- Definieren und erläutern Sie die Signale zur Kommunikation.
- Definieren Sie die möglichen Zustände.
- Spezifizieren Sie die Übergänge. Zur Vereinfachung können Sie davon ausgehen, dass zu einem Zeitpunkt immer nur ein Signal an den entsprechenden Automaten gesendet wird.