

Lösungsvorschläge der Klausur zu Echtzeitsysteme

Aufgabe 1 Wissensfragen (Lösungsvorschlag)

(30 Punkte)

- a) Fristverletzungen führen in harten Echtzeitsystemen zu einem Versagen des Systems. In weichen Echtzeitsystemen können Fristverletzungen zum Teil akzeptiert werden, es kommt eventuell einer Dienstverschlechterung.
- b) Virtual Memory kann zu unvorhersagbaren Verzögerungen führen.
- c) Echtzeitsysteme werden häufig in verteilten Systemen ausgeführt, zudem gibt es typischerweise mehrere Aufgaben mit unterschiedlichen Anforderungen an die Reaktionszeit.
- d) Mit Race Conditions werden Situationen bezeichnet, bei denen das Ergebnis von der Ausführungsreihenfolge der interagierenden Prozesse abhängt.
- e) Grundsätzliches Kriterium bei Echtzeitsystemen ist die Erfüllung von Fristen, in Standard-systemen sind typische Kriterien die allgemeine Performance (z.B. Durchsatz, durchschnittliche Wartezeit) und Fairness ausschlaggebend.
- f) Beispiele sind Semaphoren in PEARL, Rendezvous in Ada und Monitore in Java, Kanäle in Occam. Die Erläuterungen finden Sie in den Vorlesungsunterlagen.
- g) Einführung eines Echtzeitschedulers, von Real-time Threads/ No-Heap-Real-time Threads, von unterschiedlichen Speicherkonzepten (z.B. immortal memory)
- h) Die zentrale Einheit wäre ein Single-Point-of-Failure.
- i) Sicherheit bedeutet die Abwesenheit von Gefahren, Zuverlässigkeit die Erfüllung der Spezifikation (beim Auftritt eines Fehlers, muss der Ausfall aber nicht sicher sein). Beispiel: rote Ampel ist sicher, aber nicht zuverlässig.
- j) siehe Abbildung

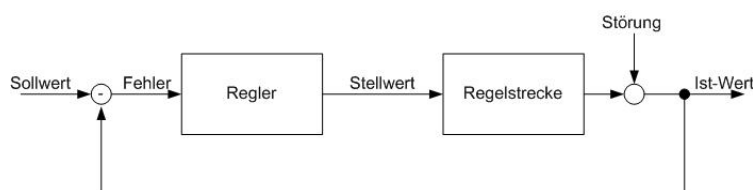


Abbildung 1: Ausgabe 1j

Aufgabe 2 Esterel (Lösungsvorschlag)

(10 Punkte)

- a) $T1 = (\{D\}, \{E\}), \{\{U\}, \{F\}\}$
 $T2 = (\{D\}, \{E\}), (\{D\}, \{N\})$
 $T3 = (\{D\}, \{E\}), (\{U\}, \{F\}), (\{U\}, \{\})$
 $T4 = (\{D\}, \{E\}), (\{D\}, \{N\})$
 $T5 = (\{U\}, \{\}), (\{D\}, \{E\}), (\{D\}, \{E\})$
- b) Modul xxx realisiert einen Semaphor, Module yyy betritt einen kritischen Bereich nach Anforderung des Semaphors.
- c) Atomare Eigenschaft des Await-Statements: es wird immer genau ein case-Statement ausgeführt, auch für den Fall, dass mehrere case-Statements gleichzeitig war werden.

Aufgabe 3 Nebenläufigkeit

(15 Punkte)

- a) `semWartebereich(1); semZapfsaeule(3), semEngstelle(1);`
- b) siehe Algorithmus

Algorithm 1 Prozess: tankendes Auto

```
1: down(semWartebereich);  
2: fahreInWartebreiche();  
3: down(semZapfsaeule);  
4: fahreAnZapfsaeule();  
5: up(semWartebereich);  
6: tanke();  
7:  
8: bezahle();  
9: down(semEngstelle);  
10: fahreInEngstelle();  
11: up(semZapfsaeule);  
12: verlasseEngstelle();  
13: up(semEngstelle);
```

- c) Ja. Bevor der Semaphor `semWartebereich` angefordert wird, muss noch der Semaphor für die Zapfsäule angefordert werden. Ansonsten kann es zur Verklemmung kommen.

Aufgabe 4 Scheduling (Lösungsvorschlag)

(10 Punkte)

- a) Nicht-präemptives, prioritätenbasiertes Schedulingverfahren.
- b) siehe Abbildung Präemptives, prioritätenbasiertes Scheduling.
- c) Problem: unbegrenzte Prioritätsinversion. Lösung: z.B. Prioritätsvererbung.
- d) siehe Abbildung mit Prioritätsvererbung/Immediate Priority Ceiling.

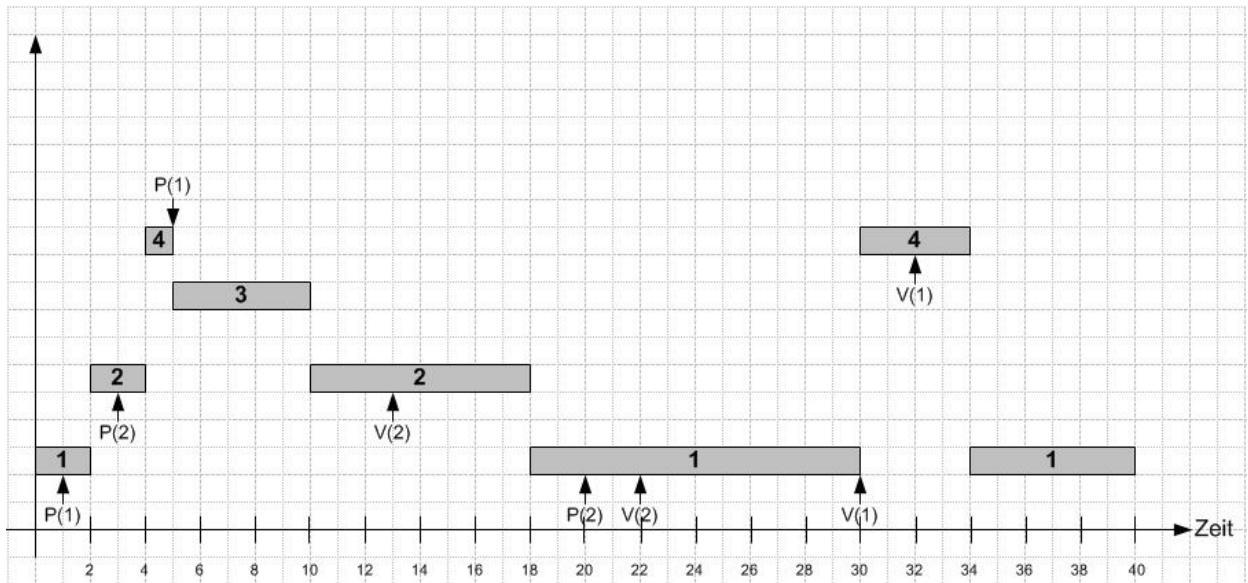


Abbildung 2: Präemptives, prioritätenbasiertes Scheduling

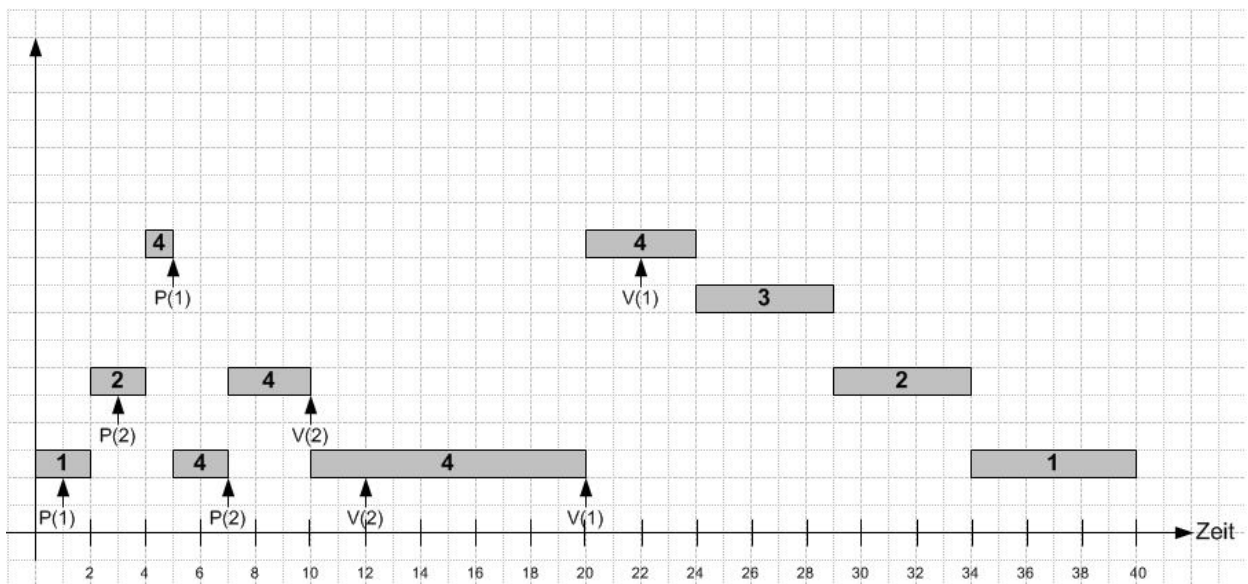


Abbildung 3: mit Prioritätsvererbung

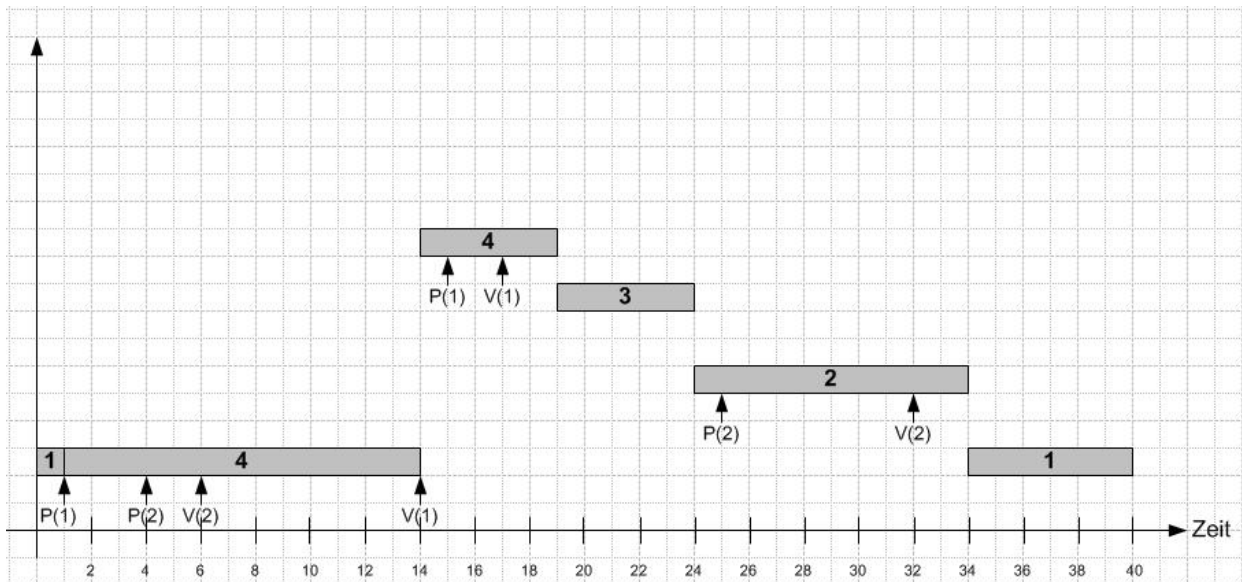


Abbildung 4: mit Immediate Priority Ceiling

e) Grundsätzlich sollten niederpriore Prozesse nicht lange kritische Bereiche blockieren.

Aufgabe 5 Echtzeitbetriebssysteme

(10 Punkte)

- TinyOS - ZebraNet: Auslegung auf Sensornetzwerke mit limitierter Stromversorgung.
- OsekTime - Zugsteuerung: Zeitgesteuertes Betriebssystem und damit klare Trennung einzelner Softwarekomponenten.
- QNX - Sicherheitsüberwachung: Hartes, stark skalierbares (kleiner Kernel) Echtzeitbetriebssystem.
- VxWorks - Fabrikanlagensteuerung: Hartes Echtzeitbetriebssystem mit guter Unterstützung unterschiedlichster Sensorik.
- WindowsCE - PDA: weiches Echtzeitbetriebssystem mit Schwerpunkt auf Benutzerfreundlichkeit (z.B. GUI), Kompatibilität mit Anwendungen von Standardbetriebssystemen

Aufgabe 6 Echtzeitfähige Kommunikation

(20 Punkte)

- a) Nachricht von T4 (einziger Rechner der zunächst senden will), Nachricht von T3 (Priorität), Nachricht von T2 (Priorität), Nachricht von T1
- b) Token, Nachricht von T4, Nachricht von T1 (Nachfolger von T4, andere Teilnehmer mit höherer Priorität konnten die Prioritätsreservierungsbits noch nicht setzen), Nachricht von T3 (aufgrund der Reservierung über die Priorität), Nachricht von T2

- c) Werden dominante und rezessive Bits gleichzeitig gesendet, so wird das dominante Bit auf dem Kanal beobachtet.
- d) Manchester-Code (Flankenwechsel immer zur Mitte des Takts, dadurch kann die Dauer eines einzelnen Bits extrahiert werden)
Bitstuffing: Die maximale Länge einer Folge gleicher Bits wird begrenzt.
- e) In CD wird die Kollision nur entdeckt, es kommt zur Wiederholung. Bei CA entdeckt der Sender mit niedrigerer Priorität (Sendung des rezessiven Bits) die Kollision und beendet die Übertragung.