

Übungen zur Vorlesung

Allgemeine Fragen

1. Nennen und erläutern Sie kurz den Unterschied zwischen einem Prozess und einem „technischen Prozess“.
2. Wie lässt sich ein Rechenprozess als ein technischen Prozess einordnen?
3. Was ist der Unterschied zwischen einem Programm und einem Rechenprozess (Task)?
4. Was versteht man unter dem Begriff des Prozesses? (kurze Definition)
5. Was sind Prozessgrößen und was Prozessdaten?
6. Was versteht man unter einem Automaten (EVA-Struktur)?
7. Charakterisieren sie ein „embedded System“ (Merkmale, Bedeutung).
8. Was versteht man darunter, wenn ein „embedded System“ deterministisch arbeitet?
9. Was ist der in der Vorlesung aufgezeigte Unterschied zwischen „harter“ und „weicher“ Echtzeit?
10. Was ist WCET und was ist BCET?
11. Im Arbeitsspeichermanagement eines Standard-Rechnerbetriebssystems ist Swapping/Paging und Garbagecollection durchaus üblich. Wieso nicht bei einem Echtzeitbetriebssystem?
12. Erklären Sie kurz die Bedeutung der in der Vorlesung verwendeten Begriffe Multitasking, Kontextswitch und Overhead in einem Einprozessorensystem?
13. Durch welche Maßnahmen läßt sich der rechnerische Overhead beim Kontextswitch zwischen Rechenprozessen reduzieren. Was ist bei der Realisierung in der Praxis zu beachten?
14. Was versteht man unter Zuverlässigkeit und was unter der Sicherheit einer technischen Anlage? Warum muß eine zuverlässig arbeitende technische Anlage noch lange nicht sicher sein? (Antwort in kurzen Sätzen).
15. Was versteht man unter einem Realzeitbetrieb eines Prozeßrechners?
16. Welche besondere Aufgabe hat ein Realzeitbetriebssystem in Bezug auf die Rechenzeiten der Rechenprozesse?
17. Was versteht man unter Prioritätsscheduling von Tasks?
18. Was versteht man unter round-robin -Scheduling von Tasks?
19. Wozu dienen Events? Erläutern Sie die Verwendung an einem selbstgewählten Beispiel mittels OS9-Events.
20. Was ist eine Semaphore? Wo werden Semaphore eingesetzt und wie werden sie benutzt? Was ist eine Deadlock-Situation bei der Verwendung von Semaphoren?

21. Zeigen Sie anhand einer geeigneten Graphik: Welche Zustände eine Task im Realzeitbetriebssystem einnehmen kann und wodurch die jeweiligen Zustandsübergänge ausgelöst werden?
22. Was versteht man unter einem präemptiven Betriebssystem?
23. Was passiert im Realzeitbetriebssystem OS9 wenn der Benutzer einen neuen Rechenprozeß startet? Was muß das RBS im einzelnen tun? (Stichworte genügen)
24. Wozu dient ein Petri-Netz ? Was ist ein SDL-Diagramm und wozu dient es?
25. Was versteht man unter dem Begriff Tasksynchronisation durch Prioritätensteuerung?
26. Was versteht man unter dem Begriff Tasksynchronisation durch Eventsteuerung?
27. Nennen und erklären Sie drei gebräuchliche Arten der Intertaskkommunikation? Nennen Sie Vor- und Nachteile der einzelnen Arten.
28. In einem RBS mit ausschließlichem Prioritätenscheduling besitzt zu einem Zeitpunkt die in diesem Augenblick höchstpriorre rechenwillige Echtzeittask die CPU. Durch welche Bedingungen wird diese Echtzeittask durch eine andere Task abgelöst? Wodurch kann der Scheduler, der die CPU zuteilt, aktiviert werden?
29. Bei der Überprüfung des Echtzeitverhaltens eines Systems tauchen zwei zu überprüfende Bedingungen auf: Gleichzeitigkeit und Rechtzeitigkeit. Was versteht man darunter und was muß wie überprüft werden?
30. Was versteht man unter der maximalen Reaktionszeit eines Rechenprozesses? Was ist die Verarbeitungszeit und was die Wartezeit ? Was ist die minimale Prozeßzeit ?
31. Was ist der Unterschied zwischen dem Signal- und dem Event-Mechanismus zur Tasksynchronisation?
32. Welche charakteristischen Eigenschaften zeichnet eine Echtzeittask in einem Echtzeitbetriebssystem aus (was steht im Taskkontrollblock) ?
33. Erklären Sie den Unterschied zwischen Tasks und Threads.
34. Wozu wird eine MMU i.d.R. in Echtzeitsystemen gebraucht?
35. Wie wird der Zugriff auf Hardwareschnittstellen im Echtzeitbetriebssystem organisiert und wie sieht die (Standard-) Schnittstelle zum Applikations-Programmierer aus?
36. Was spricht für die Verwendung eines Echtzeitbetriebssystems mit Hardwaretreibern anstelle eines direkten Hardwarezugriffs?
37. Was versteht man unter einer (Antwort-)Latenzzeit einer Echtzeittask? Wie kommt sie zustande?
38. Das Modulesystem in OS9 gestattet es Programme „in-place“ auszuführen, ohne sie laden zu müssen. Welchen Vorteil hat man dadurch?
39. Wieso mussten Sie im Praktikum vor dem Laden desselben Programms <programmname> vom PC ins Zielsystem OS9 immer das Kommando „unlink <programmname>“ am Zielsystem OS9 aufrufen?
40. Was ist der Unterschied zwischen synchronem und asynchronem Zugriff auf Geräte?
41. Was muß man bei der Programmierung mit synchronem Zugriff auf Geräte beachten?
42. Vor- und Nachteile bei der Verwendung von „shared libs“ im Vergleich zum statischen Linken?

43. Wie funktioniert ein in Hardware implementierter Watchdogtimer?
44. Warum reichen Zeitmessungen die aus der Systemzeit abgeleitet sind und auf TICKS basieren i.d.R. nicht aus?
45. Bei der Uhrensynchronisation zweier Rechner kommt es dazu, dass eine der beiden Uhr entweder vor- oder rückgestellt werden muß. Ist dies jederzeit unproblematisch möglich? Warum? (Erklärung beider Möglichkeiten erforderlich)
46. Was ist der Unterschied zwischen einem Standardbetriebssystem und einem Echtzeitbetriebssystem?
47. Was ist der Vorteil wenn ein Echtzeitbetriebssystem den POSIX Standard 1003.b unterstützt und implementiert?
48. Was ist Scheduling? Was ist der Unterschied zwischen statischen und dynamischen Scheduling?
49. Erklären Sie anhand von einem Beispiel mit drei Tasks A,B und C wie FCFS/round-robin/prioritätengesteuertes Scheduling funktioniert?
50. Wie funktioniert das Deadline-Scheduling?
51. Wie funktioniert das Scheduling unter OS9? (Nicht-Echtzeittasks,Echtzeittasks)
52. Wozu ist ein sog. Systembus gut?
53. Wie kann über den Systembus die analoge-/digitale-Peripherie angebunden (Zugriffsmethode,Schnelligkeit,DMA) werden?
54. Wie funktioniert prinzipiell ein sog. DMA Transfer? Was ist der Vorteil der Verwendung eines DMA-Transfers?
55. Was ist ein Dual Ported RAM? Wozu dient es und wie funktioniert es?
56. Was ist der Unterschied zwischen Zuverlässigkeit und Sicherheit?
57. MTBF und MTRR was ist das?
58. Wozu sind Petri-Netze gut geeignet?
59. Was versteht man unter Prioritätsinversion? Wann tritt sie auf und wie minimiert man sie?
60. Wodurch unterscheidet sich das Signalkonzept vom Eventkonzept? (Idee, Gemeinsamkeiten, Unterschiede)
61. „Shared memory“ was ist das und wofür braucht man es? Was ist beim Zugriff auf das „shared memory“ zu beachten? Wie kann eine Task, erfahren, dass neue Daten im „shared memory“ bereitstehen?
62. „Event-triggered“ versus „Time triggered“ Programmierung von Echtzeitaufgaben? (Ideen,Unterschiede)
63. Bei der Systementwicklung kamte der Begriff „self-hosted System“ vor. Was ist das?
64. Gegeben: Zwei Tasks, eine Sendertask S und eine Empfängertask E, sollen Nachrichten austauschen. Eine Nachricht darf maximal 80 Zeichen lang sein und soll in einem Shared-memory Bereich von S an E übergeben werden. Der Bereich ist maximal 80 Zeichen groß. S liest eine neue Nachricht vom Terminal A erst dann wieder ein, wenn E die letzte Nachricht auf Terminal B ausgegeben hat. Eine Leernachricht, ausser Zeilenvorschub kein anderes Zeichen eingegeben, führt dazu daß E und S sich beendigen, mit der Ausgabe am Terminal A/B "Ende!". Es stehen Ihnen Events, Semaphore und Signale zur Verfügung.

Zeigen Sie graphisch anhand geeigneter Zustandereignisdiagramme oder SDL-Diagramme den Ablauf des Nachrichtenaustausches und der nötigen Tasksynchronisation beider Tasks unter Einhaltung der Randbedingungen. Wählen Sie selbst den Synchronisationsmechanismus.

65. Angabe wie 32. Erweiterung: Nach Eingabe der Lernnachricht kein Abbruch, sondern Wechsel der Übertragungsrichtung: S wird zu E und E zu S bis zur nächsten Lernnachricht.

Aufgaben zu Themengebieten Prozessorbelegung und Echtzeit

1. Aufgabe:

Durch einen Prozessor mit Realzeitbetriebssystem und Prioritätsscheduling (7 Taskprioritätsebenen verfügbar, Ebene 7 höchste Priorität) werden vier Rechenprozesse (P1,P2,P3 und P4) bearbeitet, die zusammen einen technischen Prozeß steuern und überwachen.

Jeder Rechenprozeß P_i ist verantwortlich für die jeweilige schritthaltende Bearbeitung eines Prozeßereignisses, das im Zeitabstand T_{pi} periodisches auftritt, und der Rechenprozeß benötigt hierfür eine Verarbeitungszeit T_{vi} .

<u>Rechenprozeß P_i</u>	<u>Prozeßzeit T_{pi}</u>	<u>max. T-Reaktionszeit</u>	<u>Verarbeitungszeit T_{vi}</u>
P1	180 ms	160 ms	25 ms
P2	30 ms	30 ms	10 ms
P3	20 ms	10 ms	5 ms
P4	60 ms	60 ms	15 ms

1.1 Ist es von der Auslastung her gesehen möglich die vier Tasks von einem Prozessor bearbeiten zu lassen?

1.2 Wie müssen die Prioritäten verteilt werden, damit jede Task schritthaltend verarbeitet werden kann?

Die Kontextswitchzeit von 10 usec ist zu vernachlässigen.

a. Graphisches Verfahren: Zeichnen Sie maßstabsgerecht (Zeitauflösung der Diagramme pro Teilstrich 5 ms) :

- ein Diagramm für den Zeitverlauf der Prozessorbelegung mit der Kennzeichnung wann welcher Prozeß P_i die CPU besitzt
- Jede Task P_i muß sicher beendet sein, bevor sein zugehöriges Prozeßereignis T_{pi} wieder eintrifft oder als härtere Bedingung -falls angegeben- die max. Taskreaktionszeit abgelaufen ist.
- Jede Task P_i darf ansonsten jederzeit beliebig unterbrochen werden.

Ausgangsbedingung zum Startzeitpunkt der Diagramme ist, daß alle vier Prozeßereignisse gleichzeitig zur Bearbeitung eintreffen.

b. Rechnerisches Verfahren: Weisen Sie anhand des „mathematischen“ Verfahrens der Vorlesung nach, dass die Echtzeitbedingung eingehalten werden

2.Aufgabe

Messwerterfassung und Übertragung in Echtzeit mittels eines Microcontroller

Zur Erfassung von kontinuierlich anfallenden Analogdaten wird ein Prozeßrechner (embedded Microcontroller) verwendet. Beschreibung der Aufgabe:

1. Die zu erfassenden analogen Werte werden automatisch von einem auf dem Microcontroller integrierten Analog/Digitalwandler mit einer Rate von 1 kHz digitalisiert (gewandelt).
2. Nach jedem gewandeltem Wert wird vom A/D-Wandler ein Interrupt erzeugt, um die Abholung durch den Prozessor anzustoßen.
3. Nach Erhalt des Interrupts verzweigt der Prozessor in eine Interruptserviceroutine und diese holt den Wert ab und legt ihn in einem vereinbarten Speicherbereich ab.
4. Vom Auftreten des Interrupts bis zur abgeschlossenen Übernahme und Speicherung eines Wertes dauert es einschließlich der nötigen A/D-Wandlersteuerungen 0,2 ms.
5. Nach Abholen des 128. gewandelten Analogwertes sollen die bis dahin abgespeicherten Daten an den entfernten Zentralrechner übertragen werden.
6. Für die Übertragung sind in Summe Netto! 4 ms Rechenzeit (Verarbeitungszeit) nötig.
7. Die Übertragung kann beliebig unterbrochen und dann wieder aufgenommen werden.

2.1 Aus Gründen der Vereinfachung der Programm- und Datenstrukturen ist die erste Idee alles in der Interruptserviceroutine zu erledigen. Warum kann dann die Analogwerterfassung nicht in Echtzeit erfolgen? Zeichnen Sie hierzu das Zeitdiagramm der Rechnerauslastung und stellen Sie die erforderliche Echtzeitbedingung auf.

Weitere Angaben:

Um die Entwicklungszeit zu reduzieren wird ein Realzeitbetriebssystemkern (RBS) eingesetzt, das Tasks nach dem Prioritätsscheduling-Verfahren aktiviert.

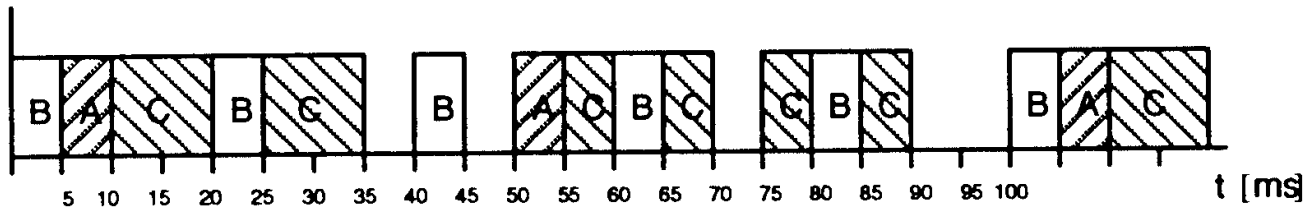
- Es stehen 16 Prioritätsstufen -0 bis 15- zu Verfügung, wobei 15 die höchste Stufe ist.
- Ferner stellt das RBS zur Prozesskommunikation zwei shared-memory-Speicherbereiche namens BUF1 und BUF2, in denen jeweils 128 Werte abwechselnd abgelegt werden können, zur Verfügung (Wechselpuffer).
- Als Taskychronisationshilfsmittel stehen drei **binäre** Events EV1, EV2 und EV3 zur Verfügung (Ein binäres Event kann nur den Wert 0 oder 1 annehmen).
- Der Interrupt des A/D-Wandlers wird jetzt von einem verfügbaren Treiber im RBS entgegengenommen und in ein transientes Event TR umgewandelt. Das transiente Event TR wird beim Eintreffen eines A/D-Interrupts vom Treiber auf den Wert 1 gesetzt. Durch Auslesen des Events TR -z.B. durch Wecken einer auf den TR Wert 1 wartenden Task- wird der Wert automatisch durch das RBS auf 0 zurückgesetzt.
- Zur besseren Übersicht soll von einer Task SYSTASK alle anderen notwendigen Tasks zu Beginn gestartet werden.

2.2 Zeigen Sie anhand geeigneter Zustands/Ereignisdiagramme der Tasks:

- welche Tasks mit welchen Prioritäten notwendig sind, wer wie mit wem zusammenarbeitet, um die Erfassung - unter der Ausnutzung der Puffer und Events- der Analogwerte, das Ablegen im Speicher und die Überspielung der 128-Werte Blöcke aufgabengerecht zu erledigen.
- Zeichnen Sie das Zeitdiagramm der Rechnerauslastung durch die Tasks.
- Wie groß ist die mittlere Rechnerauslastung?

3. Aufgabe:

Bei einem Prozeßrechner liegt folgende Prozessorbelegung durch die drei Tasks Task A, Task B und Task C vor:



- Leiten Sie aus dem Diagramm die Prozeßzeiten t_{p_i} und die Verarbeitungszeiten t_{v_i} der einzelnen Tasks Task A, Task B und Task C her. Gehen Sie davon aus, daß die Prozeßzeiten der einzelnen Tasks konstant sind.
- Geben Sie die Prioritäten (von 1 bis 3) der Tasks an, wobei 1 der höchsten und 3 der niedrigsten Priorität entspricht.
- Geben Sie die relative Belastung durch die einzelnen Tasks, sowie die Gesamtbelastung für den Prozeßrechner an.
- Wie groß sind im Beispiel die jeweiligen maximalen erreichten Task-Reaktionszeiten?

Aufgabe 4 (Zuverlässigkeit und Sicherheit):

4. Eine Prozeßrechenanlage besteht aus den in folgender Tabelle aufgelisteten Systemkomponenten mit der jeweiligen (Dauer-)Unverfügbarkeit q :

Komponente:	Unverfügbarkeit q:	Komponente:	Unverfügbarkeit q:
CPU	0.01	RAM	0.01
DMA-Interface	0.02	Analog-E/A Baugruppe	0.03
Terminal	0.02	Drucker	0.005

4.a: Die Anlage sei verfügbar, wenn die CPU, das RAM, das DMA-Interface, die Analog-E/A Baugruppe, der Drucker und das Terminal funktioniert. Zeichnen Sie das Zuverlässigkeitsersatzschaltbild und berechnen sie die Unverfügbarkeit q_{ges} des Gesamtsystems!

4.b: Um eine höhere Verfügbarkeit zu erhalten, sollen einige (oder eine einzelne) Komponenten der Anlage als Doppelsubsystem ausgeführt. (Für die dazu notwendige Koppel- und Vergleichselektronik wird eine Verfügbarkeit von 1 angenommen und eine automatische Fehlererkennbarkeit sei vorausgesetzt!)

Welche Komponenten zu verdoppeln bringt dabei den größten Beitrag zur Erhöhung der Verfügbarkeit?

Welche maximale verbesserte Verfügbarkeit läßt sich erreichen, wenn zwei Komponenten (welche?) verdoppelt werden können?

4.c: Um die Anlage bei einer Störung in einen **sicheren** Zustand zu bringen, seien die CPU und das DMA-Interface notwendig. Welche Strategie der Komponentenverdoppelung ist nun am sinnvollsten und unter welchen Umständen muß die Anlage in den sicheren Zustand gebracht werden, um eine Verbesserung der Sicherheit gegenüber 4.b zu erreichen? Zeichnen Sie das neue Ersatzschaltbild ! Wie groß ist jetzt die Verfügbarkeit der Anlage?