

Institut
Forschung
Lehre

Bachelor und Master

Lehrangebot

Studentische Arbeiten

Hinweise

Studienbüro IEF

Vorlesungsverzeichnis

Bibliothek

Mitarbeiter
Presse und Jobs
Intranet
Sitemap

Fakultät IEF

Institute der Elektrotechnik

Projekte

Startseite » Lehre » Lehrangebot » Laborpraktikum » Software- und Echtzeittechnik » Echtzeittechnik » POSIX-Betriebssystem

POSIX-Betriebssystem

Preemptives prioritätsbasiertes Scheduling in einem Echtzeit-POSIX-Betriebssystem

1. Versuchsziel

- Kennenlernen verschiedener Prozesskonzepte am Beispiel des Echtzeitbetriebssystems RT-Linux.
- Programmierung von Threads zur Datenerzeugung und Datenverarbeitung.
- Manipulation der Schedulingereigenschaften von RT-Linux und Nutzung von modernen Monitoring-Tools.
- Vertiefung der Kenntnisse des Fixed-Priority-Schedulings und der zugehörigen Analyse.

2. Grundlagen

In prozessrechenstechnischen Anlagen erfolgt neben speicherprogrammierbaren Steuerungen der Einsatz von frei programmierbaren Steuerungen. Freiprogrammierbare Steuerungen bestehen in der Regel aus einem Prozessor in modularer Bauweise mit einem industriellen Bussystem (z.B. VMEbus oder Multibus) und einem kommerziellen Echtzeitbetriebssystem. Das im Versuch einzusetzende Betriebssystem ist ein Echtzeit-UNIX-Betriebssystem, ist mit dem neuen Echtzeit-POSIX-Standard 1003.1b kompatibel und ist auf mehreren Plattformen lauffähig. Es eignet sich zur Programmierung und Steuerung komplexer Echtzeitsysteme bzw. Applikationen. Die Analyse von harten Echtzeitsystemen erfolgt unter dem Gesichtspunkt, dass in einem Echtzeitsystem alle harten Zeitschranken erfüllt werden müssen. Ausgangspunkt einer solchen Analyse ist ein idealisiertes Prozessmodell, dass von periodischen Prozessen und bekannten Worstcase-Rechenzeiten ausgeht. Der Nachweis der Durchführbarkeit eines Prozesssatzes erfolgt auf der Basis der Berechnung der Gesamtauslastung bzw. auf der Ermittlung von Reaktionszeiten. Während mit der ersten Methode notwendige und hinreichende Bedingungen, dafür dass ein bestimmter Prozesssatz eingehalten wird, angegeben werden kann, ist mit der zweiten Methode ein exakter Nachweis möglich. [PDFversion](#)

(1) Hinreichende Bedingung:

$$n \left(2^{\frac{1}{n}} - 1 \right) \geq \sum_{i=1}^n \frac{C_i}{T_i}$$

(2) Notwendige Bedingung:

$$1 > \sum_{i=1}^n \frac{C_i}{T_i}$$

(3) Analyse der Reaktionszeiten

$$R_i^{n+1} = C_i + \sum_{j=1}^{i-1} \left\lceil \frac{R_i^n}{T_j} \right\rceil C_j$$

3. Studienfragen

- 3.1. Erläutern Sie die Begriffe harte und weiche Echtzeit.
- 3.2. Charakterisieren Sie statische Echtzeitschedulingverfahren (Fixed-Priority-Scheduling, Rate-Monotonic-Scheduling, Deadline-Monotonic) und dynamische Echtzeitschedulingverfahren (Earliest Deadline First, Least Laxity), und geben Sie für diese Verfahren hinreichende und notwendigen Bedingungen für das Scheduling von Prozesssätzen an.
- 3.3. Erläutern Sie die in den Gleichungen (1-3) angegebenen Sachverhalte.
- 3.4. Nach welchen Klassifizierungsmerkmalen können Echtzeitbetriebssysteme unterschieden werden?
- 3.5. Erläutern Sie den Prozess- und den Thread-Begriff.
- 3.6. Welche allgemeinen Methoden zur Synchronisation und Kommunikation zwischen UNIX-Prozessen sind Ihnen bekannt?
- 3.7. Welche Semaphore-Arten sind in POSIX Standard definiert?
- 3.8. Wozu dienen die IPC-Mechanismen Mutex und Condition?

4. Aufgaben

4.1. Hausaufgaben (schriftlich)

Gegeben sind drei periodische Prozesse P, Q und S. P hat eine Periode TP von 100 ms. Seine Rechenzeit RP beträgt 30 ms. Die entsprechenden Werte für Q sind (TP =5 ms, RP =1ms) und (TP=25 ms, RP =5ms). P soll der wichtigste Prozess im System sein, gefolgt von S und Q. Des weiteren soll gelten, dass der Endtermin (Deadline) EP gleich der Periode TP.

- 4.1.1. Zeigen Sie das Verhalten des Schedulers, wenn die Priorität auf Wichtigkeit basiert. (z.B.: mittels Diagramm).
- 4.1.2. Berechnen Sie die Prozessorauslastung von P, Q und S.
- 4.1.3. Zeigen Sie den Prozessverlauf, wenn die Prioritäten auf Basis des rangmonotonen Zuteilungsverfahrens (rate-monotonic scheduling) ermittelt werden. Werden die Endtermine im Fall 4.1.1 bzw. 4.1.3 eingehalten?
- 4.1.4. Berechnen Sie die notwendige und die hinreichende Bedingung für die Zuteilbarkeit periodischer Prozesse für diesen Prozesssatz. Begründen Sie auf dieser Basis das Ergebnis in Aufgabe 4.1.3.
- 4.1.5. Bestimmen Sie die Reaktionszeiten des niederpriorisierten Prozesses unter Anwendung der Gleichung (3) mit den Prioritäten aus der Aufgabenstellung und aus Aufgabe 4.1.3.
- 4.1.6. Machen Sie sich mit den Systemdiensten bzw. Systemfunktionen (POSIX-Systemcalls) zur Thread-Erzeugung und -Steuerung vertraut.
- 4.1.7. Erläutern Sie die Prinzipien der Interthread-Kommunikation.

4.2. Versuchsaufgaben

- 4.2.1. Simulieren Sie eine Messwertzeugung mittels Zufallszahlengenerator im Thread 1 und einen Messwertempfang in Thread 2. Thread 1 soll jeweils 10 Werte erzeugen und Thread 2 soll diese Werte empfangen und den Durchschnitt dieser Werte ausgeben. Realisieren Sie den wechselseitigen Ausschluss mit den POSIX-IPC-Mechanismen: Mutex, Condition, Semaphore.
- 4.2.2. Simulieren sie 2 Threads, die einen Zähler "count" hoch zählen. Ein dritter Thread "watch_count" fragt den Zähler ab und legt sich so lange schlafen, bis er von einem der beiden anderen Threads aufgeweckt wird. Der dritte Thread druckt dann aus, welchen Zählerstand er ausliest, nachdem er aufgeweckt wurde. Wichtig ist, dass jeder der drei Threads die gemeinsame Variable "count" exklusiv benutzt, d.h. der Zugriff auf die Variable "count" muss zusätzlich durch eine Sperre geschützt werden.

5. Literatur

- Vorlesungsunterlagen "Echtzeitsysteme", www.amd.e-technik.uni-rostock.de/ma/gol/ebssys.htm
- R. Kern, "Prozessauswahl und Ablaufplanung in Echtzeit-Systemen" Elektronik Heft 11/92, S. 26-32, 1992
- J. Zalewski: What Every Engineer Needs to Know About Rate-Monotonic Scheduling: A Tutorial, Real-Time Magazine, S. 6-24, 1/1995
- Tanenbaum, Andrew S.: Betriebssysteme, Carl Hanser Verlag München Wien, 1991
- Danneger, Gaugelin-Danneger: Parallele Prozesse unter UNIX, Carl Hanser Verlag München Wien, 1991
- Burns, A., Wellings, A., " Real-time Systems and their Programming Languages" Addison Wesley Publishing Company, 1990
- Gallmeister, B. O., "Programming for the Real World: POSIX.4", O'Reilly & Associates, Inc., 1995
- Klein, H.K., Ralya, T., Pollack, B. Obenza, R., Harbour, M.G., "A Practitioners Handbook for Real-Time Analysis"

Suchbegriff...



Mitarbersuche...



Versuchsbetreuer

Dr.-Ing. F. Gólatowski
Raum: W1323
Tel.: 498 7274

Aufgabenstellung

Versuchsziel
Grundlagen
Studienfragen
Aufgaben
Literatur

Schnelleinstieg

[Publikationen](#)
[Anfahrt](#)
[Kontakt](#)
[Laborpraktikum](#)
[Lehrangebot](#)
[Highlights](#)
[Projekte](#)