

AUFGABE 7: ZUGRIFFSKONTROLLE

In dieser letzten Übungsaufgabe werden Sie sich mit gegenseitigem Ausschluss und Zugriffskontrolle befassen. Diese Übungsaufgabe zielt auf die Probleme der Prioritätsumkehr und der Verklemmung ab und dient dazu die in der Vorlesung vorgestellten echtzeitfähigen Synchronisationsprotokolle (vgl. VII 11 ff.) praktisch anzuwenden.

In dieser Übung greifen wir der Einfachheit halber auf die folgenden synthetischen Aufgabensysteme zurück:

Tabelle 1: Aufgabensystem 1 – „Pathfinder“

| Aufgabe | Periode ms | Phase ms | WCET ms | Betriebsmittel ¹ |
|---------|---------------|-------------|------------|-----------------------------|
| T_1 | 20 | 4 | 6 | $(R_1, 3, 1)$ |
| T_2 | 50 | 3 | 4 | |
| T_3 | 200 | 1 | 9 | $(R_1, 1, 7)$ |

Tabelle 2: Aufgabensystem 2

| Aufgabe | Periode ms | Phase ms | WCET ms | Betriebsmittel ¹ |
|---------|---------------|-------------|------------|-----------------------------|
| T_4 | 20 | 7 | 2 | $(R_2, 1, 1)$ |
| T_5 | 50 | 5 | 6 | $(R_2, 1, 5) (R_3, 3, 2)$ |
| T_6 | 100 | 3 | 6 | $(R_3, 1, 5) (R_4, 4, 2)$ |
| T_7 | 200 | 1 | 10 | $(R_4, 1, 9)$ |

Tabelle 3: Aufgabensystem 3

| Aufgabe | Periode ms | Phase ms | WCET ms | Betriebsmittel ¹ |
|----------|---------------|-------------|------------|-----------------------------|
| T_8 | 20 | 3 | 6 | $(R_5, 1, 5) (R_6, 4, 1)$ |
| T_9 | 50 | 8 | 12 | |
| T_{10} | 200 | 1 | 6 | $(R_6, 1, 5) (R_5, 5, 1)$ |

¹Notation Betriebsmittel: (Betriebsmittel, relativer Anforderungszeitpunkt, Haltezeit)

Implementierungshinweise:

1. Nutzen Sie die in der Vorgabe enthaltenen Vorlagen und implementieren Sie die o. g. Aufgabensysteme in separaten Dateien.
2. Verändern sie die Datei `CMakeLists.txt` um zwischen den einzelnen Aufgabenimplementierungen umzuschalten.
3. Vergeben Sie die Prioritäten nach dem RMA, simulieren Sie die WCET wie angegeben.
4. Nutzen Sie das von eCos bereitgestellte Mutexkonzept zur Implementierung der Betriebsmittel.
5. Belassen Sie die Lösungen der vorangegangenen Teilaufgaben deaktiviert im Code für die spätere Abgabe.
6. Zeichnen Sie die Abläufe Ihrer Implementierung für die Abgabe auf.

☞ aufgabe_{1,2,3}.c

1 Aufgabenstellung

1.1 Grundlagen

1. Aufgabe Welche Bedingungen müssen erfüllt sein, damit es zu einer Verklemmung (engl. Deadlock) kommen kann?

Antwort:

2. Aufgabe Welcher zusätzliche Parameter muss bei der Laufzeitanalyse einer Aufgabe mit gegenseitigem Ausschluss beachtet werden?

3. Aufgabe Wie wirkt sich dieser Parameter auf die Laufzeitanalyse von niederpriorien/höherpriorien Aufgaben im Allgemeinen aus, welche selbst eine Abhängigkeit zu dem kritischen Abschnitt aufweisen?

Antwort:

4. Aufgabe Wie sieht dies bei niederprioren/höherprioren Aufgaben im Allgemeinen aus, die selbst in **keiner** Abhängigkeit zu dem kritischen Abschnitt stehen?

Antwort:

1.2 Verdrängungssteuerung

5. Aufgabe Machen Sie sich zunächst mit Stift und Papier (oder einer anderen Darstellungsmöglichkeit Ihrer Wahl) klar, was bei der Koordinierung von Aufgabensystem 1 mit Verdrängungssteuerung (NPCS) bezüglich belegter Betriebsmittel, blockierter Aufgaben, Prioritäten etc. passieren sollte. Zeichnen Sie den entstehenden Ablaufplan.

6. Aufgabe Welche maximale Blockadezeit (für die höchstpriorie Aufgabe) erwarten Sie?

Antwort:

7. Aufgabe Können Verklemmungen auftreten?

Antwort:

8. Aufgabe Welche Nachteile erfahren unbeteiligte Aufgaben (die kein Betriebsmittel nutzen)?

Antwort:

9. Aufgabe Nutzen Sie nun das Konzept der Verdrängungssteuerung (NPCS) zur Synchronisation von Aufgabensystem 1 in eCos. Messen Sie die Blockadezeit der höchstpriorien Aufgabe und archivieren Sie einen Screenshot Ihres der zeitlichen Abfolge für die Abgabe.

ESP cyg_scheduler_lock()

ESP ...unlock()

ESP make trace

1.3 Prioritätsvererbung

10. Aufgabe Machen Sie sich zunächst mit Stift und Papier (oder einer anderen Darstellungsmöglichkeit Ihrer Wahl) klar, was bei der Koordinierung von Aufgabensystem 2 mithilfe der Prioritätsvererbung, bezüglich belegter Betriebsmittel, blockierter Aufgaben, Prioritäten etc. passieren sollte. Zeichnen Sie den entstehenden Ablaufplan.

11. Aufgabe Welche maximale Blockadezeit (für die höchstpriorre Aufgabe) erwarten Sie?

Antwort:

12. Aufgabe Können Verklemmungen auftreten?

Antwort:

13. Aufgabe Welche Nachteile erfahren unbeteiligte Aufgaben (die kein Betriebsmittel nutzen)?

14. Aufgabe Verwenden Sie Prioritätsvererbung (PI) um Aufgabensystem 2 in eCos umzusetzen. Wählen Sie hierfür bei der Initialisierung der Mutexobjekte CYG_MUTEX_INHERIT als Protokoll aus. Messen Sie die Blockadezeit der höchstpriorren Aufgabe und archivieren Sie einen Screenshot Ihres der zeitlichen Abfolge der Aufgaben für die Abgabe.

Ⓢ cyg_mutex_set_protocol

Ⓢ make trace

1.4 Prioritätsobergrenzen

15. Aufgabe Machen Sie sich zunächst mit Stift und Papier (oder einer anderen Darstellungsmöglichkeit Ihrer Wahl) klar, was bei der Koordinierung von Aufgabensystem 3 mithilfe von Prioritätsobergrenzen, bezüglich belegter Betriebsmittel, blockierter Aufgaben, Prioritäten etc. passieren sollte. Zeichnen Sie den entstehenden Ablaufplan.

16. Aufgabe Welche maximale Blockadezeit (für die höchstpriorre Aufgabe) erwarten Sie und wie lässt sie sich messen?

Antwort:

17. Aufgabe Können Verklemmungen auftreten?

18. Aufgabe Welche Nachteile erfahren unbeteiligte Aufgaben (die kein Betriebsmittel nutzen)?

Antwort:

19. Aufgabe Setzen Sie als nun noch die Prioritätsobergrenzen (PCP) ein um Aufgabensystem 3 in eCos zu realisieren. Wählen Sie hierfür bei der Initialisierung der Mutexobjekte CYG_MUTEX_CEILING als Protokoll und konfigurieren Sie die Prioritätsobergrenzen der Betriebsmittel passend – beachten Sie hierbei, dass die Priorität des Betriebsmittels um eins höher sein muss als die des höchstprioreren verwendenden Fadens und nicht, wie eigentlich üblich, gleich der Fadenpriorität. Messen Sie die Blockadezeit der höchstprioreren Aufgabe und archivieren Sie einen Screenshot Ihres der zeitlichen Abfolge der Aufgaben für die Abgabe. ☞ make trace

20. Aufgabe Zeichnen Sie den Ablauf mittels Tracer auf und überlagern Sie diese Aufzeichnung mit der *vermuteten* Systemobergrenze. Welche Variante von PCP ist in eCos implementiert?

Antwort:

21. Aufgabe Welchem anderen Verfahren ähnelt die von eCos verwendete Variante des PCP? Worin liegt der entscheidende Unterschied zur der zweiten, in der Vorlesung vorgestellten, Variante?

Antwort:

22. Aufgabe Wieso kommt es zu keiner Verklemmung?

Antwort:

2 Erweiterte Aufgabe

23. Aufgabe Implementieren Sie Aufgabensystem 2 mithilfe von Prioritätsobergrenzen. Messen Sie wiederum die Blockadezeit. Sichern Sie auch hierfür Ihren Ablaufplan. Welches Verhalten stellen Sie fest?

Antwort:

24. Aufgabe Implementieren Sie Aufgabensystem 2 mithilfe von Verdrängungssteuerung. Sichern Sie auch hierfür Ihren Ablaufplan. Wie verhält sich diese Implementierung im Vergleich zu den vorherigen mit Prioritätsobergrenzen und Prioritätsvererbung?

Antwort:

25. Aufgabe Implementieren Sie Aufgabensystem 3 mithilfe von Prioritätsvererbung. Welches Verhalten stellen Sie fest? Sichern Sie auch hierfür Ihren Ablaufplan.

Antwort:

Hinweise

- Bearbeitung: Gruppe mit je drei Teilnehmern.
- Abgabefrist: 31.01.2019
- Fragen bitte an i4ezs@lists.cs.fau.de