

AUFGABE 7: ZUGRIFFSKONTROLLE IN ECOS

In dieser letzten Übung 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.

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

Aufgabensystem 1: Prioritätsumkehr

Aufgabe	Periode ms	Phase ms	WCET ms	Betriebsmittel ¹
T_1	20	4	6	($BM_1, 3, 1$)
T_2	50	3	4	
T_3	200	1	9	($BM_1, 1, 7$)

Aufgabensystem 2: Transitive Blockung

Aufgabe	Periode ms	Phase ms	WCET ms	Betriebsmittel ¹
T_4	20	7	2	($BM_2, 1, 1$)
T_5	50	5	6	($BM_2, 1, 5$) ($BM_3, 3, 2$)
T_6	100	3	6	($BM_3, 1, 5$) ($BM_4, 4, 2$)
T_7	200	1	10	($BM_4, 1, 9$)

Aufgabensystem 3: Verklemmung

Aufgabe	Periode ms	Phase ms	WCET ms	Betriebsmittel ¹
T_8	20	3	6	($BM_5, 1, 5$) ($BM_6, 4, 1$)
T_9	50	8	12	
T_{10}	200	1	6	($BM_6, 1, 5$) ($BM_5, 5, 1$)

Implementierungshinweise:

- Nutzen Sie die in der Vorgabe enthaltenen Vorlagen und implementieren Sie die o. g. Aufgabensysteme in separaten Dateien.

¹Notation Betriebsmittel: (Betriebsmittel, Anforderungszeitpunkt, Haltezeit)

2. Vergeben Sie die Prioritäten nach dem RMA, simulieren Sie die WCET wie angegeben.
3. Nutzen Sie das von eCos bereitgestellte Mutexkonzept zur Implementierung der Betriebsmittel.
4. Bei Verwendung des Priority Ceiling Protocol ist zu beachten, dass die Priorität des Betriebsmittels um eins höher sein muß als die des höchsprioren verwendenden Fadens und nicht, wie eigentlich üblich, gleich der Fadenpriorität.
5. Implementieren Sie anschließend die verschiedenen Varianten zur Synchronisation, nutzen Sie hierfür die in der Tafelübung vorgestellten Mechanismen von eCos.
6. Belassen Sie die Lösungen der vorangegangenen Teilaufgaben deaktiviert im Code für die spätere Abgabe.

Zentrale Fragen: Wie so häufig haben die verschiedenen Techniken ihre Vor- und Nachteile. Die zentrale Fragestellung für alle folgenden Aufgaben ist daher:

1. Welche maximale Blockadezeit (für die höchspriore Aufgabe) erwarten Sie und wie lässt sie sich messen?
2. Können Deadlocks auftreten?
3. Welche Nachteile erfahren unbeteiligte Aufgaben (die kein Betriebsmittel nutzen)?

Machen Sie sich bei den nachfolgenden Aufgaben zunächst mit Stift und Papier (oder einer anderen Darstellungsmöglichkeit Ihrer Wahl) klar, was bezüglich belegter Betriebsmittel, blockierter Aufgaben, Prioritäten etc. passieren sollte. Implementieren Sie die Aufgabe anschließend wie beschrieben.

Aufgabenstellung

1. Verdrängungssteuerung: Nutzen Sie das Konzept der Verdrängungssteuerung (NPCS) zur Synchronisation von Aufgabensystem 1. Beantworten Sie Frage A bis C (s. o.).
☞ cyg_scheduler_lock()
☞ ...unlock()
2. Prioritätsvererbung: Verwenden Sie als nächstes Prioritätsvererbung (PI) um Aufgabensystem 2 umzusetzen. Wählen Sie hierfür bei der Initialisierung der Mutexobjekte CYG_MUTEX_INHERIT als Protokoll aus (cyg_mutex_set_protocol()). Beantworten Sie Frage A bis C (s. o.).

3. Prioritätsobergrenzen: Setzen Sie als letztes noch die Prioritätsobergrenzen (PCP) ein um Aufgabensystem 3 zu realisieren. Wählen Sie hierfür bei der Initialisierung der Mutexobjekte CYG_MUTEX_CEILING als Protokoll. Beantworten Sie Frage A bis C (s. o.). Zeichnen Sie den Ablauf mittels Tracer auf und überlagern Sie diese Aufzeichnung mit der vermuteten Systemobergrenze. Welche Variante von PCP ist in eCos implementiert? Wieso kommt es zu keiner Verklemmung?

Erweiterte Aufgabe

1. Konzeptvergleich: Implementieren Sie nun noch die anderen Kombinationen (jedes Aufgabensystem mit jeder Synchronisationstechnik). Messen Sie wiederum die Blockadezeit. Was fällt Ihnen insbesondere beim Aufgabensystem 2 und 3 auf?

Hinweise

- Bearbeitung: Gruppe mit je zwei/drei Teilnehmern.
- Abgabezeit: 26.01.2015
- Fragen bitte an i4ezs@lists.cs.fau.de