

シーケンサ・イベントカウンタ機能による リングバッファ使用の簡易化の提案と検証

山崎 拓也[†]南角 茂樹[†][†] 大阪電気通信大学総合情報学部情報学科

1. はじめに

複数の並列/並行処理間でデータの引き渡しによく使われるデータ構造としてリングバッファがある。リングバッファは、データアクセス時の排他制御が不要であるが、リード/ライトポインタの制御が必要である。そのため、優れたデータ構造であるにも関わらず、アプリケーションソフトウェアで実際に使用されることが少ない。そこで、リングバッファを想定した排他制御を行うための排他制御方式としてシーケンサ・イベントカウンタ[1][2]機能が提案されている。しかしながら、この提案方式は、提案はされているが、検証はまだ行われていない。そこで、シーケンサ・イベントカウンタ機能を REMON[3]に実装し、検証を行った。

2. シーケンサ・イベントカウンタ

シーケンサはチケットの発券機のような役割を持ち、イベントカウンタはシーケンサによってタスクに配れたチケット値によってタスクの起動を順序化させる役割を持つ。シーケンサ・イベントカウンタには sequencer, await, .advance の3つのシステムコールが存在する。図1に3つのシステムコールの処理の図を示す。sequencer は呼ばれたタスクにチケットを渡し、sequencer チケットをインクリメントする処理となっている。await は呼ばれたタスクのチケットとイベントカウンタを比較し、チケットの方が高い場合、待ち配列にそのタスクのチケットの値を格納し、タスクを待ち状態にする処理となっている。advance はイベントカウンタをインクリメントし、待ち状態のタスクがある場合、そのタスクのチケットとイベントカウンタを比較し、イベントカウンタの方が高ければその待ち配列のチケットの値を-1にし、そのタスクを待ち状態から実行可能状態にする処理となっている。

3. 検証内容

検証として、典型的な排他制御問題の生産者/消費者問題を4つでできたリングバッファを想定した場合で解決する。内容としては、1つの生産者タスクが連続で5回実行し、その後消費者タスクを1回実行する。この場合、生産者タスクが4回処理を終えると、リングバッファがいっぱいになり、これ以上データの書き込みができなくなるので生産者タスクの5回目の実行で待ち状態に入る。その後、消費者タスクが実行され、データの読み込みが終了し、データに空きができるので生産者タスクが実行可

能になり、再びデータの書き込みをするという処理になることが予想される。

svc_sequencer

1. 呼ばれたタスクにシーケンサの現在の値を渡す
2. シーケンサの値をインクリメントする

svc_await

1. 呼ばれたタスクのチケットの値とイベントカウンタの値を比べる
チケットの方が高かった場合
 2. タスクのチケットの値を待ち配列に格納
 3. タスクの状態を待ち状態にする
それ以外の場合
- svc_awaitを抜ける

svc_advance

1. イベントカウンタの値をインクリメントする
 2. 待ち配列に格納されている全てのチケットの値と
イベントカウンタの値を比較
イベントカウンタの値がチケットの値以上の場合
 3. その待ち配列に格納されているチケットの値を-1にする
 4. そのタスクの状態を実行可能状態にする
それ以外の場合
- svc_advanceを抜ける

図1 各システムコールの処理

4. まとめと今後の課題

シーケンサ・イベントカウンタ機能の検証として、REMONにシーケンサ・イベントカウンタ機能を実装し、検証を行った。今後の課題として、シーケンサ・イベントカウンタの値リセットの検討が考えられる。

参考文献

- [1] 前川守, 佐藤文考, 星野康夫, 渡辺清, オペレーティングシステムの先端的概念, 1990.
- [2] John M Mellor-Crummey, Michael L Scott 「Algorithms for Scalable Synchronization on SharedMemory Multiprocessors」, (1991-1)
- [3] 南角 茂樹, 水篠 公範, 小泉 寿男, 福田 晃: 「組込みシステム用割り込みスケジューラ REMON」, 電気学会論文誌 C (電子・情報・システム部門誌), Vol.133 No.2 pp.316-325 (2013-2)