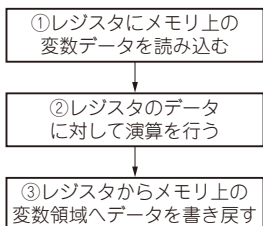


複数のタスク間での資源競合を防ぐ …セマフォによる排他制御

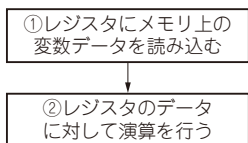
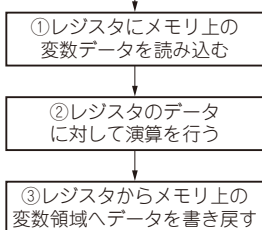
豊山 祐一

タスクAのグローバル変数への処理

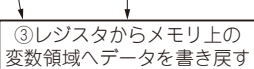


(a) グローバル変数へのアクセスが競合しない場合

タスクAのグローバル変数への処理

タスクBのグローバル変数への処理
ディスパッチタスクBの演算結果は、
タスクAの演算結果に
上書きされてしまう

ディスパッチ



(b) グローバル変数へのアクセスが競合した場合

図1 タスク間の競合(グローバル変数の演算処理)

タスク間の同期には、これまでに実装したタスク付属同期やイベント・フラグのようにおのおのタスクの動作を連携させる機能の他に、お互いの動作の干渉を防ぐ排他制御と呼ばれる機能があります。

排他制御には主にセマフォやミューテックスなどが使われます。本章ではTry Kernelにセマフォを実装します。

排他制御は競合を防ぐ機能

● 共有資源を使うときに必要となる

複数のタスクの動作がある場合に競合を防ぐのが排他制御です。

排他制御が必要となる例として、タスク間で共有する資源へのアクセスがあります。共有資源としては、複数のタスクから使用されるグローバル変数やI/Oデバイスなどがあげられます。

グローバル変数に対する単純な演算処理でも排他制御が必要となる場合があります。例えば変数を加算したり減算したりするだけでも、マイコンでは複数の機械語の命令が実行されます。

一般的な変数の演算では、まずメモリ上の変数領域からマイコンのレジスタにデータを読み取り、演算を行った後、レジスタからメモリ上の変数領域へ書き戻します[図1(a)]。この演算処理中にディスパッチが発生して他のタスクが実行され、ディスパッチ先のタスクで同じ変数に対し演算処理が行われると、その変数の値は正しくなくなります[図1(b)]。

タスク間の共有資源へのアクセスの競合を防ぐには、あるタスクが共有資源を使用中であればそれを示し、他のタスクは使用が終了するまで待つ仕組みが必要です。

タスク間の排他制御を実現する一般的な機能の1つがセマフォです。IEEE 2050-2018規格でもセマフォが規定されています。

● セマフォの仕組み

セマフォは共有する資源を表したカーネル・オブジェクトと言えます。

セマフォに対する基本的な操作は、資源の獲得と返却です。タスクは共有する資源を使用する前にセマフォから資源を獲得します。そして共有資源の使用が終わると、セマフォに資源を返却します[図2(a)]。もしセマフォから資源を獲得しようとしたときに、既