

# 論文の内容の要旨

論文題目： Interrupt and Critical Section Arrangements  
For The Real-Time Systems  
(実時間システムのための割り込み処理と排他期間処理)

氏名： 戴 毛兵

リアルタイムオペレーティングシステムは広い範囲で使われている。その環境は複雑な大規模システムでもあり、小さな組み込みシステムでもある。この上、システムにおける実時間タスクの時間制限も多様である。そのため、静的に作り上げたリアルタイムオペレーティングシステムをすべてのハードウェア環境とソフトウェア環境に適用するには、無理がある。本論文では、一つ動的なリアルタイムシステムを提示する。このシステムでは、ユーザはシステムが使われている環境によって、動的にシステムの時間精度と優先度の指定ができ、最適化したリアルタイムシステムを作ることができる。また、このシステムは実時間タスクのデッドライン時間内の実行終了を保証する上、実時間タスクの実行によって、非実時間タスクへの性能影響を最大限に抑える。

リアルタイムオペレーティングシステムの開発には、一つ重要な目標は実時間タスクのデッドライン時間内にその実行を終了させることである。そのため、カーネルは実時間タスクを迅速にスタートさせる必要がある。実時間タスクのスタートは一般にタイマー割り込みを指定した時間とおりカーネルに来ることによって実現する。しかし、通常カーネルには、多くの割り込み禁止期間がある。タスクは割り込み禁止期間中にいるとき、ハードウェア割り込みが禁止され、割り込みハンドラの実行も遅延される。この原因で、タイマー割り込みは即時にカーネルに入らず、実時間タスクのスタートも遅延される。この問題を解決するため、本論文では、遅延した割り込み処理方式を提案する。この方式では、周辺デバイスからの割り込みは禁止されず、割り込みは即時にカーネルに入ることができる。また、共有リソースへの同時アクセスを回避するため、元の割り込みハンドラの実行は割り込み禁止期間の後に回される。この方式を用いて、タイマー割り込みが軽量の遅延時間でカーネルに入ることができ、システムがスタートラインに立っている実時間タスクのありを迅速に知ることができる。遅延した割り込み処理方式は、ただタイマー割り込みを迅速にカーネルに入らせることだけでなく、システムに精確な時間サービスを提供する仕組みも作り上げた。実時間タスクの時間関連の要求は、この仕組みによって、精確に提供することができる。本論文では、追加したシステムコールによって、実時間タスクが指定した時間に、優先度変更とタスクのウェークアップの例を示した。

カーネルはスタートラインに立っている実時間タスクがあることを知った後、実時間タスクを迅速にスタートさせるため、カーネルスケジューラの実行が期待される。しかし、カーネル内に多くのノンプリエンプション期間がある。タスクがノンプリエンプション期間中にいるとき、共有リソースへの同時アクセスを排除するため、一つのコアに他タスクへの実行は許されていない。このとき、カーネルスケジューラの実行も許されず、結果としては、実時間タスクのスタートが遅延される。この問題を解決するため、本論文では、遅延したロック処理方式を提案する。この方式では、実時間タスクの先行実行を実現させるため、通常タスクは以下の条件でノンプリエンプション期間の前でウェイトする：実時間タスクのスタート時間から、通常タスクが現在のノンプリエンプション期間の実行を終了する予測時間までの長さは、システム精度時間MTV (Maximum Tolerable Value) より大きい。通常カーネルでは、多くのノンプリエンプション期間の長さは短いものであり、実時間タスクへの影響は小さい。したがって、通常タスクが多くのノンプリエンプション期間の前に待たされることはなく、実時間タスクの実行により、通常タスクへの影響は従来の手法より軽減することができる。また、ユーザはシステムが使われている環境によって動的にMTVを設定することができ、システムは実時間タスクのデッドライン時間内の終了を保証することと同時に、非実時間タスクの実行への影響を最大限に抑える。

実時間タスクがカーネルスケジューラに選択され、その実行が開始される。しかし、実時間タスクの実行は頻繁にハードウェア割り込みハンドラの実行に阻害される。この問題を解決するため、本論文では、動的な優先度空間を提案する。この機構では、従来の独立したハードウェア優先度空間とソフトウェア優先度空間が統合される。割り込みは優先度ベースに三つのタイプに分けられる。それぞれは、緊急わりこみ、実時間割り込み、非実時間割り込みである。緊急割り込みはシステム内で重要な役割があり、その優先度がもっとも高い。タイマー割り込みがその一つである。実時間割り込みは実時間タスクとの関連があり、その優先度が実時間タスクから継承される。非実時間タスクはその他の一般的な割り込みであり、その優先度が実時間タスクにより低く、通常タスクにより高くに設定される。このアプローチを用いて、実時間タスクは一部割り込みの優先度よりも高い優先度を得ることができる。また、実時間タスクとハードウェア割り込みの関係によって、ユーザは動的に割り込みの優先度を決定することができる。したがって、実時間タスクの実行は優先度の低い割り込みハンドラの実行に阻害されない。

以上の手法と機構を用いて、本論文で作り上げたシステムは、動的にハードウェア環境とソフトウェア環境に適合することができる。また、本論文は、ただ従来の手法で残されている問題を解決するだけでなく、非実時間タスクの実行性能にも目を向き、最適化したリアルタイムシステムを成功した。