

Windows NT オペレーティングシステムによるリアルタイムソフトウェア制御 - その虚構と真実

白書



NT

「Windows NT
オペレーティング
システムの限られ
たデータをコンテ
キストから切り離
して提示して、制
御には向かないと
主張しているベン
ダーがあります。
これらのベンダー
は独自の拡張機能
を採用し、
Microsoft の標準
テクノロジーを利用
する機会を取り逃
がしています」

この文書は、発行日時点での論点について、ロックウェル・オートメーションの最新の見解をまとめたものです。市場の状況はたえず変化しているため、この文書に記載している情報はロックウェル・オートメーションの公式の表明ではなく、情報の正確さを保証するものでもありません。この文書はあくまでも参考用として作成されています。ロックウェル・オートメーションは、この文書に関する明示的または暗黙的な保証を一切致しません。

© 1998 Rockwell Software Inc. All rights reserved.

ControlWare, Rockwell Automation, および Allen-Bradley は、Rockwell International, Inc.の商標です。

Alpha AXP は、Digital Equipment Corporation の商標です。

Component Integrator, RTX, および VenturCom は、VenturCom, Inc.の商標です。

DEC は、Digital Equipment Corporation の商標です。

Hyperkernel は、Imagination Systems Inc.の商標です。

INtime および iRMX は、Radisys Corporation の商標です。

Intel は、Intel Corporation の登録商標です。

IBM および OS/2 は、International Business Machines Corporation の登録商標です。

Microsoft および Windows は Microsoft Corporation の登録商標です。また、Windows 95 および Windows NT は Microsoft Corporation の商標です。

OS-9 は、Microware Systems Corporation の登録商標です。

Pentium は、Intel Corporation の登録商標です。

PowerPC は、International Business Machines Corporation の商標です。

QNX は、QNX Software Systems, Ltd.の登録商標です。

SCO は、Santa Cruz Operation, Inc.の登録商標です。

TNT は、PharLap, Inc.の商標です。

UNIX は、米国およびその他諸国の The Open Group の登録商標です。

はじめに

当社が Microsoft® Windows® NT オペレーティングシステムを選択した理由

当社は、オープンな PC アーキテクチャベースのソフトウェア制御に、どのオペレーティングシステムを採用するかという点で重大な決断に迫られていました。市場には以下のさまざまな選択肢が投入されていました。

- リアルタイム・オペレーティング・システム(iRMX™, QNX®, OS-9®)
- DOS エクステンダー(ControlWare™, PharLap TNT™など)
- Windows NT オペレーティングシステム用の拡張機能(Radisys INtime™, VenturCom™ RTX™, Imagination Systems Hyperkernel™)
- 標準のオペレーティングシステム(Windows 95®, Windows NT®, IBM™ OS/2®, SCO®, UNIX®)

これらの選択肢は、それぞれ以下のような利点を備えていました。

- 下位スケラビリティ
- リアルタイムパフォーマンス
- テクノロジーの使いやすさ
- 開発の生産性
- サードパーティとの互換性
- 市場での認知度

市場での状況を勘案して、当社は初期の段階から Win32 ベースのソリューションの採用に傾いていました。顧客からのフィードバックによると、本当の意味で「オープン」と見なせるオペレーティングシステムは、ソフトウェアベンダーからの十分なサポートがある Microsoft Windows プラットフォームだけでした。当社の顧客にとっては、エンドユーザがソフトウェアベンダーを自由に選択できなければ、オープンなプラットフォームとはいえません。自動化ソフトウェア市場では、Microsoft Windows の勢いが際だっており、オープンなアプリケーションのためのソリューションとして十分な資格を備えていました。さらに、複数のベンダーから提供されているソフトウェアを、プラットフォーム上で支障なく共存させる必要があることも、もう一つの重要な条件として顧客から繰り返し指摘されていました。オープンな制御を実現することが、当社の決断にとって決定的な要因となりました。

この文書では、当社がソフトウェア制御の基本的なプラットフォームとして、標準版の Windows NT オペレーティングシステムを採用した理由を説明し、Windows NT オペレーティングシステムの制御アプリケーション向けプラットフォームとしての適性や、そのパフォーマンスについての当社の調査結果を報告します。当社が標準の Windows NT オペレーティングシステムを採用したことは、競合他社の最近の製品アプローチと比較すると異彩を放っており、ある意味では論争的になっています。ソフトウェア制御の普及に伴い、Windows NT オペレーティングシステムのパフォーマンス、デザイン、信頼性は数々の批判にさらされており、ソフトウェア制御アプリケーションには不向きだという結論にエンドユーザを駆り立てています。

これらの批判の多くにはそれなりの根拠があることは事実ですが、コンテキストが適切ではなく、エンドユーザに誤解を生じさせる原因となっています。また、Windows NT オペレーティングシステムに取ってかわる代替ソリューションが詳細に検証されているわけでもありません。この文書では、Windows NT オペレーティングシステムのパフォーマンスを分析し、代替ソリューションとして提案されているリアルタイム拡張システムの主な問題点を取り上げます。Windows NT オペレーティングシステムとリアルタイム拡張システムについてのテストと分析、および現場での制御経験の両面から、当社は真のオープンシステムのメリットを活かすことができるのは、そのままの Windows NT オペレーティングシステムであると確信しています。

背景情報

ハードウェア/ソフトウェア・リアルタイム・システムによる確定的制御

確定的制御とハードウェア/ソフトウェア・リアルタイム・システムのご概念は、ソフトウェア制御を語るうえで中核をなす事項です。説明を始める前に、典型的な制御システムを考へてみましょう。一般に、制御システムは入力データを受け取り、何らかの処理を行ない、出力データを送出します。この制御シーケンスには以下の実行方法があります。

- I/O スキャナの割り込みをはじめとする外部イベントによって実行
- 20msec おきなど、一定の時間間隔で定期的に実行
- 連続ループの中でできるだけ頻繁に実行

たとえば、アレン・ブラドリーの PLC-5 プロセッサでは、以下の手段によって上記の 3 種類の実行方法を実現しています。

- プロセス入力割り込みタスク(PII)
- 選択時間割り込みタスク(STI)
- メイン・コントロール・プログラム(MCP)

コントローラが一定の時間内に必ず処理を実行できる場合に、その制御を「確定的制御」といいます。たとえば、あるコントローラで、20msec 間隔の定期的タスクが実際には 20 ~ 24msec 間隔で実行されたとします。このタスクでは、スキャン時間の反復性が 4msec の範囲で変動します。ワーストケースのパフォーマンスが判明している場合に、この制御を確定的制御といいます。コントローラにウォッチドッグスレッドを組み込むと、制御が所定の時間間隔で実行されたかどうかを確認することができます。この機能を取り入れると、実行間隔が限界値を超過した場合にプロセッサをフォルト状態にし、I/O をフェールセーフ状態に切り換えることができます。

理論上と実際のワーストケースの確定性を考えると、ハードウェア/ソフトウェア・リアルタイム・システムのさらに深遠な問題が明らかになります。ハードウェア・リアルタイム・システムでは、緊密な制御を行なうことで、ワーストケースのパフォーマンスを保証しようとしています。この手法により、ワーストケースのパフォーマンスを理論的に証明できるはずですが、ただし、ハードウェア・リアルタイム・システムのパフォーマンスは、制御アプリケーションや I/O サブシステムだけでなく、オペレーティングシステムにも依存しています。

オペレーティングシステムは、ハードウェア・リアルタイム・システムにおいてさえ、優先順位が最高のリアルタイム・アプリケーション・タスク、すなわち割り込みの確定性しか保証できません。その他のタスクは、この最高優先順位のタスクに優先される可能性があるため、その確定性はオペレーティングシステムと、最高優先順位のタスクのワーストケースのコードパス長(プログラミングされた処理の最悪の実行時間)の両方に依存します。このような確定性の低下は、優先順位がさらに低いタスクに伝搬していきます。結論として、リアルタイム・オペレーティング・システムを採用しても、リアルタイムシステムのパフォーマンスは保証されません。その意味では、実質上既存のオペレーティングシステムとリアルタイム・オペレーティング・システムを区別する明確な基準はありません。

一方、ソフトウェア・リアルタイム・システムという用語は、コントローラのワーストケースのパフォーマンスを理論的に証明するのではなく、観測によって決定する考え方から生まれています。ソフトウェア・リアルタイム・システムでは、環境の複雑さ、すなわち既成のコンポーネントを混在使用することから、厳密な証明を行なうことは不可能であり、代わりにシステムを実験環境で負荷を与えてテストし、長期間にわたってパフォーマンスを観測します。実際の負荷条件よりも過酷な負荷を与え、分析精度を高めるために十分に長期間にわたってパフォーマンスを観測することで、システムの信頼性を確認します。同様のテストは、ハードウェア・リアルタイム・システムを検証する際にも必要になります。

ソフトウェア・リアルタイム・システムでは、リアルタイム確定性を特に考慮していないコンポーネントを使用するため、システムパフォーマンスが制約される場合があります。ただし、高性能とリアルタイム確定性の間には密接な関係があるため、高性能であれば、確定性を考慮していないシステムでも通常は制御システムとして十分に機能します。

確定的制御の必要条件

現実には、完璧な反復性を備えたリアルタイム制御システムはありませんが、幸運なことに、ほとんどのリアルタイム制御プロセスでは、ある程度まではスキャンの変動が許容されます。当社では、何年にもわたって制御上の各種の問題に取り組み、スキャン時間と反復性の条件に関する膨大な専門知識を蓄積しています。厳密さが要求されるアプリケーションも多数ありますが、一般的なアプリケーションの多くは変動に対する耐性があります。スキャン時間の条件を決定するうえで有用な概念に、プロセス時定数とマシンサイクル時間の2つがあります。

プロセス時定数の概念は、連続プロセス制御の分野から生まれたもので、通常はPID(比例/積分/微分)アルゴリズムと関連付けて考えられます。システムを1次のプロセスとしてモデル化した場合は、単純な開ループテストによってプロセス時定数を推測することができます。コントローラの出力をステップ関数状に変更したときに、制御量が最終値の63%に達するまでの時間が、そのプロセスの時定数の推測値となります。この時定数はプロセスによって大きく異なります。たとえば、産業用ガスタービンの燃料バルブ開度を変更した場合の高圧スプール速度の時定数は420msec程度かもしれませんが、水処理沈殿槽の供給液の組成を変更した場合のpHの時定数は2日にも達することがあります。

プロセス制御ループのスキャン時間の条件を決定する際には、最大スキャン時間をプロセス時定数の 6 分の 1 以下にすることが目安となります。したがって、先ほどのガスタービンの例では、少なくとも 70msec ごとに 1 回のスキャンが必要になり、沈殿槽の例では 8 時間に 1 回のスキャンですむこととなります。また、可変時間間隔の形式で記述された PID アルゴリズムは、スキャン時間の中～大規模な超過に対する耐性にも優れています。スキャン時間が設定値を 50% 以上超過することが時おり発生する程度であれば、通常は問題は起きません。

PID 制御では、測定値と PID アルゴリズムの出力値に現れるむだ時間とその変動を最小限に抑える必要もあります。非常に高速なアプリケーションでは、専用の密結合アナログモジュールを使用し、通常は I/O バックプレーンからこれらのモジュールに直接アクセスする必要があります。

マシンサイクル時間は、ディスクリート制御アプリケーションに対する同様の指標となります。ディスクリート制御では、スキャン時間によって入力イベントと出力イベントの同期状態が決定されます。大まかな目安として、スキャン時間の変動をマシンサイクル時間の 5% 以内に抑える必要があります。そのために専用の高速コントローラが必要になる場合もありますが、それほど厳密さは要求されない場合もあります。巨大な中間リレー、ソレノイドバルブなどの電子機械素子によって制御動作に遅延が生じる場合は、状況はさらに複雑になります。その場合のシステムの条件は、アプリケーションのタイプに応じて大きく変動します。

PLC の一般的なパフォーマンス

現代的なコントローラの能力に着目することも、制御システムの条件を決定するうえで有効な方策となります。プログラマブルロジックコントローラ(PLC)は、すでに 20 年以上の使用実績があり、マイクロプロセッサ技術の急速な進展と歩調を合わせて、たえず進化しつづけています。今日の PLC プロセッサは 5 ~ 10msec のスキャン時間を実現し、STI 型のタスクでのスキャン時間の変動も、機種によっては 1 ~ 4msec にまで短縮されています。特殊用途向けの専用 PLC では、さらに高速なスキャン時間を実現されています。さらに、最近の PLC プロセッサの中には、PLC-5 プロセッサの PII のような「割り込みタスク」をサポートしているものもあり、0.5 ~ 2.0msec 以内にイベントを検出し、処理を開始するプログラムを作成することが可能となっています。

プロセッサのパフォーマンスは、そのアーキテクチャとアプリケーションに応じて変化します。スキャンの反復性は、システムタイマや通信インターフェイス用の割り込み処理の影響を受けます。さらに、以下の不定期イベントも反復性に影響を及ぼします。

- シリアル入力の処理
- キーボードからの入力の処理
- 通信エラーの処理
- ネットワーク要求の受け付け
- コントローラの雑用処理

スキャンの反復性は、PLC アプリケーションのコーディングスタイルによっても変動します。条件分岐、ループ、サブルーチンコールをサポートしている PLC プロセッサでは、コードパス長が状況に応じて変化するため、スキャン時間の変動が避けられません。通常、大規模な PLC アプリケーションのスキャン時間は、15 ~ 100msec 程度となります。時間に対する依存性が低い MCP 型のタスクでは、25 ~ 50msec 程度のスキャン時間の変動は許容されるのが普通です。PLC アプリケーションや単純な汎用モーション制御(GMC)アプリケーションの多くでは、スキャン時間の変動は問題にはなりません。

より高速なアプリケーションでは、STI 型のタスクや小規模な専用 PLC プロセッサを使用した方が無難です。これらのアプリケーションでは、比較的単純なプログラムと、I/O ハードウェアの緊密な接続が必要になります。さらに、アプリケーションによっては、汎用の PLC プロセッサでは実現不可能な性能が要求される場合があります。これらのアプリケーションには以下のようなものがあります。

- 高度な汎用モーション制御(GMC)アプリケーション
- 多くのコンピュータ数値制御(CNC)アプリケーション
- 高速な協調駆動制御

コントローラに加えて、制御システムの I/O モジュール、アダプタ、ラック、ケーブルも、スキャン時間と反復性に影響を及ぼします。コントローラよりも I/O モジュールの方が、スキャン時間と反復性に大きな影響を与える場合があります。たとえば、10Mbps の Ethernet™ ベースの I/O システムで、マスタ/スレーブプロトコルを使用してネットワーク上の I/O アダプタをスキャンする場合は、一周のスキャンに通常 8 ~ 10msec の時間がかかります。スキャン時間の変動する原因の一つに、巡回冗長検査(CRC)によってパケットの伝送エラーが検出された場合の回復処理があります。一回の再試行で回復処理が成功したとしても、I/O スキャン時間の 8 ~ 10msec 程度の変動は避けられません。

システムの完全性と I/O のフェールセーフ動作

確定的制御のパフォーマンス以外に、制御システムを安全に使用するために以下の 3 つの点を考慮する必要があります。

- システムの完全性
- フォルトの迅速な検出
- I/O のフェールセーフ動作

正常動作時には、制御スキャンの変動パターンはある程度決まっていますが、制御システムのコンポーネントに障害が発生した場合は、出力をフェールセーフ状態に移行させる必要があります。コンポーネントごとに多数の障害が発生する可能性があるため、この処理は極めて厄介です。

コントローラの主要な障害には、以下のようなものがあります。

- ホールト状態(バグを含んだコードを実行したときに発生)
- 電源の故障
- メモリのパリティエラーによるマスク不能割り込み(NMI)の発生
- 無限ループの発生(バグを含んだコードを実行したときに発生)
- プロセッサの予期しない例外処理(バグを含んだコードを実行したときに発生)

システムの完全性を高めるために、PLC プロセッサは雑用処理サイクルの内部で、さらに継続的にシステムの広範な完全性チェックを実行し、コンポーネントの障害が検出された場合に即座に実行を停止できるようにします。これらの障害の多くは、プロセッサの例外処理によってトラップされます。通常、PLC システムでは、これらの例外に対して致命的フォルト(PLC プロセッサの赤ランプフォルトなど)が発令されます。

さらに、I/O インターフェイススキャナにも、PLC プロセッサの障害を確実に検出し、出力をフェールセーフ状態に移行させる機能が必要になります。通常、PLC プロセッサの障害はウォッチドッグタイマによって検出されます。このウォッチドッグタイマは、I/O インターフェイススキャナモジュールの内部に設けられ、PLC プロセッサによって定期的のリセットしないと I/O 処理が継続されない仕組みになっています。これでもまだ十分ではありません。I/O インターフェイススキャナモジュール自身に障害が発生する可能性もあるので、I/O リンクアダプタにも I/O インターフェイススキャナ用のウォッチドッグタイマを組み込む必要があります。

制御プロセッサは前触れなくホールト状態に移行することがあります。したがって、I/O サブシステムは PLC プロセッサの動作に依存することなく、フェールセーフ状態に移行すべきかどうかを決定できなければなりません。どのマイクロプロセッサにも、特権モードで不適切なコードが実行された場合に、割り込みを解消/無効化し、ホールト状態に移行する機能が用意されています。また、リターン命令が実行される前にスタックが破壊されると、カーネル内のどのアドレスに実行ポイントが移されるか保証されないため、境界配置が不適切なコードが実行される場合があります。プロセッサはこの場合にもホールト状態に移行します。

まとめ

この節では、制御システムの条件を理解し、Windows NT オペレーティングシステムのデザイン、パフォーマンス、信頼性を評価するのに必要な基礎知識と背景情報を説明しました。後ほど Windows NT オペレーティングに関する当社の分析結果を解説する際に、実際の制御システムと照らし合わせてこれらの問題を再び取り上げ、確定的制御のベンチマークとしての現実の PLC の機能と比較することにします。さらに、システムの完全性と I/O のフェールセーフ動作に関しても、Windows NT オペレーティングシステムのソフトウェア・リアルタイム・パフォーマンスを評価するうえで重要な問題を取り上げます。

次の節では、Windows NT オペレーティングシステムの評価結果を詳細に解説し、Windows NT オペレーティングシステムがソフトウェア制御アプリケーション用のプラットフォームとして適切かどうかについての一般的な結論を導き出すことにします。

Windows NT オペレーティングシステムの評価

Windows NT オペレーティングシステムのディスパッチアルゴリズム

Windows NT オペレーティングシステムのソフトウェア・リアルタイム・パフォーマンスを評価する前に、このオペレーティングシステムの基本的なディスパッチアルゴリズムを理解する必要があります。Windows NT オペレーティングシステムでは、以下の3種類のオブジェクトがディスパッチされます。

- **割り込みサービスルーチン(ISR)** 主にデバイス割り込みによってディスパッチされるドライバルーチンです。
- **遅延手続き呼び出し(DPC)** あまり急がない処理を実行するドライバルーチンです。ISRによってキューに格納されます。
- **ディスパッチされた実行スレッド(スレッド)** Windows NT オペレーティングシステムのスケジューラによって管理される基本的な実行単位です。カーネル、デバイスドライバ、プロセスはスレッドを所有することができます。

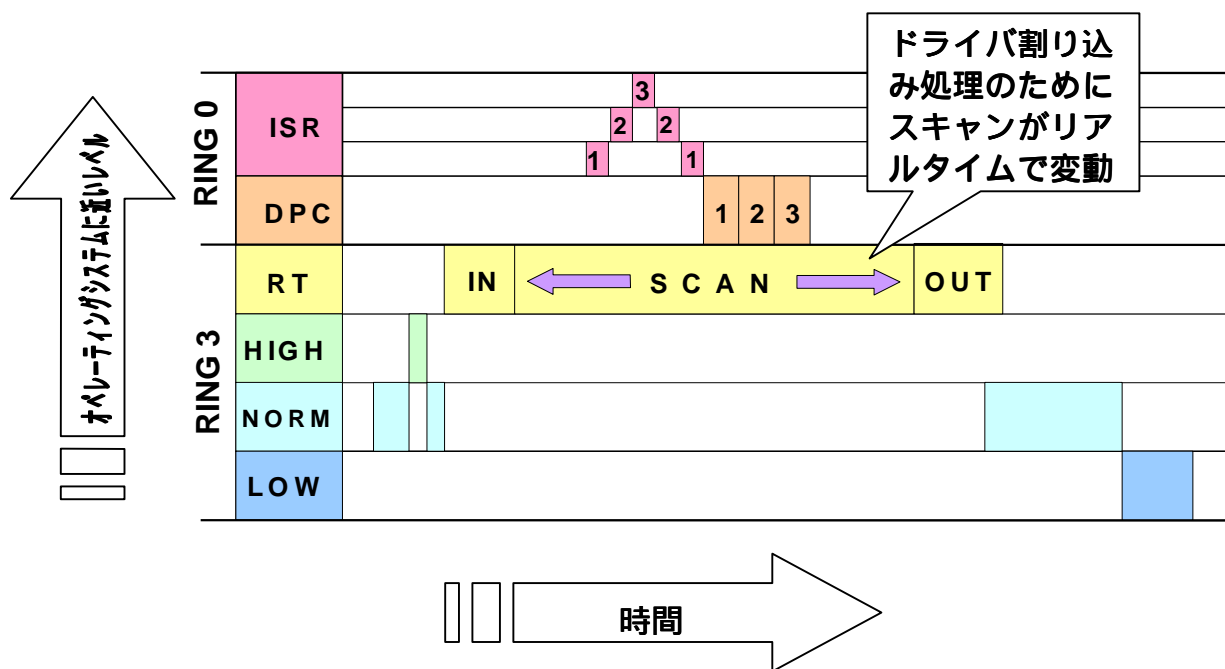


図1：Windows NT オペレーティングシステムにおけるディスパッチアルゴリズムの概略図。リアルタイム(RT)クラスでソフトウェア制御を実行すると、その他のアプリケーションによる確定性の大幅な変動を防止することができます。ただし、確定性の変動を完全に防止することはできません。優先順位の低いアプリケーションによってリング0のデバイスドライバが呼び出され、ディスクの読み書きなどの処理が実行されることがあるからです。

割り込みサービスルーチン(ISR)

ISR は、IRQL と呼ばれる割り込み優先レベルを持ち、ハードウェア割り込み要求レベルに対応付けられます。Windows NT オペレーティングシステムでは、32 段階の IRQL(0 ~ 31)が使用されます。このうち、12 から 27 までの IRQL は、ハードウェア・アブストラクション・レイヤ(HAL)によって、16 段階の PC ハードウェア割り込み要求レベルにマッピングされます(表 1 を参照)。ハードウェア割り込み要求が発生し、その IRQL が現在実行中のオブジェクトの IRQL より低い場合は、HAL 内の仮想プログラマブル割り込みコントローラ(VPIC)によって、割り込みのディスパッチが保留されます。

割り込みがディスパッチされると、IRQL がその割り込みに対応した値に高められます。ただし、IRQL の値が増減されるのは、該当する ISR が実行されている間だけに限られます。ISR から制御が返されたときには、IRQL はもとの値に戻されています。現在の IRQL が ISR の IRQL レベルよりも低くなり、その時点で保留されている割り込みがある場合は、現在保留されている割り込みのうち、優先順位が最も高いものが即座にディスパッチされます。

ISR では、(KeRaiseIrql 関数で)IRQL レベルをいったん上昇させ、その後(KeLowerIrql 関数で)IRQL レベルを下降させることによって、ISR の実行が終了するまでプリエンブションを禁止することができます。さらに、(リング 0 で実行される)特権ルーチンは、(x86 の CLI 命令によって)割り込みを無効にし、IRQL レベルを操作することなく迅速さが要求される処理を実行することができます。(x86 の STI 命令によって)割り込みを再び有効にすると、割り込み処理が再開されます。この手法は、Windows NT オペレーティングシステムのデバイスドライバの規則に反していますが、ドライバのプロデューサにとって他の手段がないときに(厳密なタイミングが要求されるハードウェアの問題に直面していて、IRQL を操作することでは対処できない場合など)、コードパス長を大幅に短縮するための実践的な手法として使用が認められています。

表 1 : Windows NT オペレーティングシステムにおける割り込みレベルの割り当て

割り込みレベル(IRQL)	割り当て
レベル 31	ハードウェアエラー割り込み(NM)
レベル 30	電源障害割り込み
レベル 29	プロセッサ間の相互呼び出し
レベル 28	クロック割り込み
レベル 12-27	PC ハードウェアの IRQL(0 ~ 15)にマッピング
レベル 4-11	未使用
レベル 3	ソフトウェアデバッガ割り込み
レベル 0-2	システム同期化のためのソフトウェア割り込み

遅延手続き呼び出し(DPC)

ISR は、自分自身が実行されている間に DPC の実行を要求することができます。これらの DPC は、ISR に代わって後ほど実行されます。同様に、ドライバもカーネルや I/O のタイマを使用して、DPC のキューへの格納を要求することができます。カーネルの API は、0.10 μ sec 単位のタイマ分解能をサポートしています。ただし、DPC が実行用のキューに格納されることは保証されていますが、カーネルのタイマサービスがこの分解能で機能することは保証されていません。

Microsoft では、ISR をできるだけコンパクトにし、その DPC のディスパッチが少々遅れても支障がないようにドライバを設計することを推奨しています。これは、同一マシン上のドライバの相互干渉を防止するためです。あるドライバのために長期間にわたって割り込み処理が禁止されると、他のドライバの動作に支障が生じるおそれがあります。

通常、DPC は先入れ先出し(FIFO)方式のキューに格納されます。キュー中の DPC は、ISR の実行が終了したときにディスパッチされます。DPC には優先順位はなく、キューに格納された順に実行されます。マルチプロセッサシステムでは、それぞれのプロセッサで別々の DPC を同時に実行することができます。さらに、マルチプロセッサシステムで、いずれかのプロセッサの IRQL レベルが引き上げられていない場合は、キューに格納された DPC を即座に実行することが可能です。

PROCESS PRIORITY CLASSES					BASE PRIORITY
REALTIME	HIGH	NORMAL_FOREGROUND	NORMAL_BACKGROUND	IDLE	
TIME CRITICAL					31
					30
					29
					28
					27
					26
HIGHEST					25
ABOVE NORMAL					24
NORMAL					23
BELOW NORMAL					22
LOWEST					21
					20
					19
					18
					17
					16
IDLE					15
	TIME CRITICAL	TIME CRITICAL	TIME CRITICAL	TIME CRITICAL	14
	ABOVE NORMAL				13
	NORMAL				12
	BELOW NORMAL				11
	LOWEST	HIGHEST			10
		ABOVE NORMAL			9
		NORMAL	HIGHEST		8
		BELOW NORMAL	ABOVE NORMAL		7
		LOWEST	NORMAL		6
			BELOW NORMAL	HIGHEST	5
			LOWEST	ABOVE NORMAL	4
				NORMAL	3
				BELOW NORMAL	2
				LOWEST	1
	IDLE	IDLE	IDLE	IDLE	0
RESERVED FOR IDLE PROCESS					

図 2 : Windows NT オペレーティングシステムのプロセスクラスとスレッド優先順位

スレッドは、ディスパッチされている ISR や、保留されている DPC が一つもないときにディスパッチされます。Windows NT オペレーティングシステムでは、内部的に 32 段階のスレッド優先順位レベルが使用され、優先順位が最も高いスレッドがディスパッチされます。ただし、優先順位が 0~15 のスレッドと、16~31 のスレッドでは、処理の方式が異なります。後者のスレッドでは優先順位が固定されていて、完了、破棄、または阻止されるまでスレッドの実行が継続されます。同じ優先順位のスレッドが複数存在する場合は、それらが巡回式に実行されます。Win32 API では、これらの優先順位は Real-time 優先順位クラスとして表現されます。

優先順位が 0~15 のスレッドにおいても、優先順位が最高のスレッドがディスパッチされます。ただし、これらのスレッドは単純に完了、破棄、阻止されるまで実行されるのではなく、同じ優先順位のその他のスレッドとともに、10msec 単位のタイムスライス方式でマルチタスク処理されます。さらに、これらのスレッドでは、Windows NT オペレーティングシステム独自の探索的な手法により、スレッドの優先順位が自動的に変動します。Win32 API では、これらの優先順位は Normal, High, Idle の 3 つの優先順位クラスとして表現されます(図 2 を参照)。デスクトップアプリケーションは、Normal 優先順位クラスで実行されるのが普通です。

特権レベル

ISR と DPC は常に特権レベル(リング 0)で実行され、スレッドは特権レベルとアプリケーションレベル(リング 3)のどちらかで実行されます。リング 0 のコードでは、特権命令(x86 の CLI, STI, HLT 命令など)を実行することができます。リング 3 のコードでも特権命令を実行できますが、特権命令違反の例外が発生してしまいます。この例外が発生した場合は、Windows NT オペレーティングシステムの例外ハンドラにより、該当するスレッドの実行が中止されるか、または代用命令が実行されます。

対称型マルチプロセッシング

Windows NT オペレーティングシステムをマルチプロセッサシステムで使用した場合は、対称型のディスパッチ方式が適用されます。このシステムでは、優先順位が最高のオブジェクトが複数存在する場合に、そのうちの N 個のオブジェクトが常時実行されます(N はプロセッサ数)。実行可能なスレッドがあり、複数のプロセッサが同等に利用できる場合は、キャッシュを有効に活用するために、前回実行されたときと同じプロセッサにスレッドが割り当てられます。初期の一部の SMP HAL では、バスアクセスの同期処理を単純にするために、1 つのプロセッサにしか ISR がディスパッチされませんが、最近のほとんどの HAL は、ISR のデュアルディスパッチ機能をサポートしています。2 プロセッサ構成のシステムはすでに広く普及しており、現在はクアドプロセッサ(4 プロセッサ)構成のシステムも登場し、SMP 処理能力も強化されています。

まとめ

Windows NT オペレーティングシステムでは、ISR、DPC、スレッドの 3 種類のオブジェクトがディスパッチされます。ISR は DPC より常に優先して処理され、DPC は先入れ先出し(FIFO)順に実行されます。また、スレッドは 4 種類のクラスに分類され、優先順位に従ってスケジューリングされます。スレッドの 4 種類のクラスのうち、最高優先順位のクラスには優先順位の動的変更機能が適用されず、該当するスレッドは完了、破棄、または阻止されるまで実行が継続されます。残りの 3 種類のクラスでは、デスクトップ機能を強化するために開発された優先順位の動的変更アルゴリズムが適用され、10msec 単位のタイムスライス方式のマルチタスク処理が実行されます。

Windows NT オペレーティングシステムのパフォーマンス

ソフトウェア・リアルタイム・システムでは、ISR のパフォーマンスと、Real-time 優先順位クラスの最高優先順位(優先順位 31)のスレッドのパフォーマンスが最も重要になります。ISR は、標準の Windows NT オペレーティングシステムで実現可能な最高のソフトウェア・リアルタイム・パフォーマンスを提供します。ただし、他のデバイスドライバの ISR の動作を妨げないために、制御アルゴリズムのコードパス長が制限されてしまいます。さらに、Windows NT オペレーティングシステムの標準ドライバでは、このレベルから I/O スキャナアダプタにアクセスできないため、ISR で制御を行なう場合は、専用の I/O アクセス機能を用意する必要があります。

Real-time 優先順位クラスの優先順位 31 のスレッドは、ハードウェアの動作を妨げることなく、Windows NT オペレーティングシステムで実現可能な最高レベルの確定性を提供します。さらに、これらのスレッドは標準機能だけでデバイスドライバにアクセスでき、I/O スキャナも Windows NT オペレーティングシステムの標準のデバイス・ドライバ・モデルを使って作成することができます。そのため、ソフトウェアの標準化がより高いレベルで実現され、Microsoft や OEM メーカーのプラットフォームテストでは保証されていないハードウェア上の問題が発生するリスクを大幅に低減することができます。

スレッドを最高優先順位に設定することに加えて、スレッドのコードとデータをすべて物理メモリに読み込み、これらをその場所にロックする必要があります。これは、ページフォルトによってスレッドの実行に遅延が生じないようにするためです。Windows NT オペレーティングシステムには、そのための API が用意されています。スレッドを最高優先順位で実行すると、制御スレッドのコードとデータだけをメモリ上にロックするだけですむという利点も生まれます。コントローラに対するプログラミングアクセスやスーパーバイザアクセスで使用する、重要度の低いルーチンでは、通常どおりページフォルトが認められます。これらのページフォルトが発生しても、制御スレッドの動作には影響は及びません。

ISRのパフォーマンス

200MHz の Pentium® Pro システムでは、ISR の実行時間は最大で 5 μ sec 程度になります。このルーチンの命令あたりの平均クロック数を 6 クロックとすると、この時間は約 165 の命令に相当し、Microsoft から提供されているコードパス長のデータと一致します。ISR の実行時間はコードパス長に依存するため、チップのアーキテクチャが同一だとすると、実行時間はクロック速度に反比例すると見なすことができます。ただし、この値は割り込み遅延時間の最小値にすぎません。

優先順位の低いハードウェア割り込みにおける割り込み遅延時間の最大値は、200MHz の Pentium Pro システムで 75 ~ 80 μ sec 程度、33MHz の 486 システムでは 450 ~ 480 μ sec 程度になります。先ほどと同様に、命令あたり 6 クロックを消費すると仮定すると、ISR の最大パス長は 2,500 ~ 2,700 命令程度になります。このようにパス長が増大するのは、より優先順位の高い割り込みのためにプリエンプションが発生することに一因があります。

このことから、ISR でソフトウェア制御を行なう場合は、制御コードのパス長を 2,500 命令以下に抑えれば、トラブルフリーな動作が保証されるはずですが、これはアプリケーションのキャパシティをかなり控えめに見積もった場合です。通常、高速な Pentium Pro 用の周辺機器は、より低速な 486 システムで一般的な割り込みディスパッチ遅延時間を許容できるように設計されているため、15,000 命令程度のパス長でも問題は生じないはずですが、しかしながら、一般には、制御アプリケーションの最も緊急性の高い処理のためだけに ISR を使用した方が無難です。

ISR の確定的制御パフォーマンスに関しては、その反復性能は 200MHz の Pentium Pro システムで 70 ~ 80 μ sec 程度となります。ソフトウェア制御アプリケーションを ISR のコンテキストで実行する場合は、そのパス長を控えめに見積もって 2,500 命令(多めに見積もっても 15,000 命令)程度に抑えないと、他のデバイスの正常な動作が阻害され、システムの動作が予測不能になるおそれがあります。ただし、対称型マルチプロセッシングシステムでは、ISR のディスパッチ中に他のプロセッサで残りの処理を実行できるため、より長期間にわたって ISR をアクティブにすることができます。

リアルタイムスレッドのパフォーマンス

ISR では、他の ISR のパフォーマンスを考慮するだけですみますが、Real-time 優先順位クラスのスレッドでは、ISR だけでなく、その DPC によるプリエンブションも考慮しなければなりません。DPC のパフォーマンスは、リアルタイムスレッドの主要な変動要因となります。当社のテストでは、ディスクドライバとネットワークドライバが、DPC の実行時間が増大する主な原因であることが判明しています。これらに比べると影響は少ないものの、ビデオドライバもその原因の一つとなります。これらのドライバでは、大規模なバッファコピーなどの処理のために、コードパス長が必然的に長くなるためです。

当社では、各種のプラットフォーム上でリアルタイムスレッドのパフォーマンスを検証しています。その結果、定期的スキャンサイクルの反復性は、ネットワークデバイスやディスク・ドライバ・デバイスの影響を受けることが明らかになっています。高性能な周辺機器を使用した場合は、パフォーマンスはプロセッサの速度に比例し、対称型マルチプロセッシング(SMP)システムを採用すると、パフォーマンスの大幅な向上が実現します。

高性能な周辺機器を使用してディスクとネットワークの負荷テストを行ったところ、100MHz の Pentium による単一プロセッサシステムでは約 16msec だった変動幅が、200MHz の Pentium Pro による単一プロセッサシステムでは約 8msec に短縮されました。さらに、デュアル・プロセッサ・システムでは、同じテストの結果が大幅に改善されることが確認されています。133MHz のデュアル・プロセッサ・システムでさえ、変動幅は 3~4msec にまで短縮され、200MHz のデュアル・プロセッサ・システムでは、その値がわずか 2~3msec になります。このことから、2 台目のプロセッサの存在がパフォーマンスの向上に大きく寄与することがわかります。クアド・プロセッサ・システムでは、プロセッサの同期処理のために ISR の実行効率が若干低下することを差し引いても、パフォーマンスのさらに大幅な向上が見込めるものと予想されます。

ハードウェアパフォーマンスのより綿密な認定の必要性

当社のテストでは、一部のネットワークカードの性能に問題があり、ネットワーク負荷テストで 30~40msec の遅延が生じることが判明しています。さらに、単一プロセッサシステムで高解像度の多色表示を行なう場合は、ビデオドライバの性能もボトルネックになります。これらの問題は、デバイスドライバ、ドライバ構成、およびデバイスに原因があると考えられます。したがって、標準の Windows NT オペレーティングシステムを使用してソフトウェア制御を行なう場合は、ハードウェアとドライバに関して、Microsoft や OEM メーカーが実施している基本テストよりも厳格な認定プログラムが必要となります。そのため、当社では Windows NT オペレーティングシステムによるソフトウェア制御を対象に、ハードウェアパフォーマンスのさらに高レベルな認定を目的とした標準テストの開発を進めています。

制御の必要条件の観点から見たシステムパフォーマンス

ISRの実行タイミングは、非常に高速な制御アプリケーションでも十分すぎるほど高速です。ただし、他のデバイスドライバの動作を妨げる可能性があるため、ISRのコンテキストではあまり大規模な制御アプリケーションは実行できません。また、Windows NTオペレーティングシステムの標準のデバイスドライバアクセスモデルも使用できないため、I/O スキャナカード用のインターフェイスを特別に作成する必要があります。

十分に高性能な機器を使用した場合のリアルタイム制御スレッドの変動幅は、200MHzのPentium Proによる単一プロセッサシステムでは最大8msecに、100MHzのPentiumによる単一プロセッサシステムでは最大16msecになります。これらの変動幅は、最新のPLCと比べるとわずかに及ばないものの、基本的な制御アプリケーションとしては十分なレベルです。この程度の性能が実現されていれば、プロセス時定数が250msecを超える調節制御や、秒単位未満のタイミングが要求されない連動ロジックに十分に対応することができます。また、対称型マルチプロセッシング(SMP)システムでは、リアルタイム制御スレッドのパフォーマンスが劇的に向上します。その場合のスキャンの変動性は、PLCの水準と比較しても遜色のないものとなります。

まとめ

Windows NTオペレーティングシステムの確定的制御パフォーマンスは、一部の制御アプリケーションを除けば十分すぎるものです。高速な単一プロセッサシステムにおけるISR、およびほとんどのデュアル・プロセッサ・システムにおけるリアルタイムスレッドは、最新のPLCプロセッサと比べても遜色のないパフォーマンスを発揮します。単一プロセッサシステムにおけるリアルタイムスレッドは、パフォーマンスの点で普及型のPLCにわずかに及ばないものの、時間の重要性が低いプロセスや、ディスクリット制御アプリケーションには十分といえます。

ビデオ表示、周辺機器、デバイス用のドライバの中には、他のシステムと比較して性能が劣るものがあります。これらの性能に問題があるシステムを使用すると、必要な制御パフォーマンスが得られないおそれがあります。当社では、使用するオペレーティングシステムに関係なく、ソフトウェア制御性能を保証できるようにするために、各種装置のより高度な認定規格が必要だと考えています。そのため、当社は各種装置の制御用としての性能を確認できるよう、システム負荷テストの開発を推し進めています。現時点では、限定的ながらテスト済みのハードウェアリストを作成している段階です。

Windows NT オペレーティングシステムのデザイン

当社では、ソフトウェア制御アプリケーションのプラットフォームとしての Windows NT オペレーティングシステムを評価する際に、まずこのオペレーティングシステムの確定性に加えて、いくつかの主要なシステムデザインに着目しました。最も重要なことは安全性ですから、迅速な障害検出、カーネルの適切な例外処理といったシステムの完全性に関する機能が主要な検討事項となりました。さらに、PC の標準的な周辺機器を、I/O スキャナや産業用ネットワークアダプタと組み合わせ使用したときに、動作干渉のリスクを最小限に抑えられるかどうかも確認する必要性がありました。そのほか、スレッドの同期化プリミティブの効率や、同期化オブジェクトの優先度が逆転する可能性を簡単に回避できるかどうか、検討事項の一つとなりました。

ここでは、これらのシステムデザインに関する調査結果を報告するとともに、ソフトウェア制御にとって非常に望ましく、その他の制御システムではまだ実現されていない、Windows NT オペレーティングシステムならではの先進的なシステムデザインについて解説します。まず、予期しないカーネル例外処理とブルー画面について説明することにします。

Windows NT オペレーティングシステムのブルー画面

Windows NT オペレーティングシステムの安定性と「ブルー画面」の意味は、かなり誤解されているようです。ブルー画面の詳細に移る前に、まず予期しないカーネル例外とその原因について検討することにします。

Pentium プロセッサは、実行中にいくつかの自己診断用のフォルト例外とアボート例外を発生します。これらの例外は例外ハンドラによってチェックされ、再起動可能と判断されたフォルトは再起動することができます。ただし、アボート例外については必ず実行が中止されます。制御システムで発生するフォルト例外やアボート例外は、ハードウェアやソフトウェアの障害に起因しているものがほとんどです。最も一般的な例外を以下の表にまとめます。

0	除算エラー-IDIV または DIV でのゼロ除算	フォルト
2	NMI 例外-ハードウェアエラー	アボート
5	境界チェック配列アクセス-境界チェックエラー	フォルト
6	無効なオペコード-不正なコード境界または破壊されたコード	フォルト
8	例外ディスパッチ時の二重フォルト例外	フォルト
10	無効な TSS-外部割り込みによる無効な TSS を介したタスク切り換え	フォルト
12	スタック例外-スタックセグメントのオーバーフロー/アンダフロー	フォルト
13	一般保護違反-保護されたメモリや無効なメモリへのアクセス	フォルト
14	ページフォルト-まだ読み込まれていない仮想ページへのアクセス	フォルト
16	浮動小数点エラー-浮動小数点演算の例外	フォルト

フォルト例外がリング 0 で発生した場合は、Windows NT オペレーティングシステムの例外ディスパッチャによって特殊なチェックが実行され、プロセッサが割り込みスタック上で動作していたかどうか、その IRQL レベルが DPC 処理用のレベル以上になっていたかどうかを確認されます。この 2 つの条件に当てはまる場合は、システムコードまたはデバイスドライバが実行されていたため、回復不能な例外と判断されます。その場合、システムはスレッドの実行をすべて中止し、バグチェックを実行し、その後ホールド状態に移行するか、再起動します。バグチェックでは、設定に応じて以下の処理を行なわせることができます。

- 何もしない

- ブルー画面シンドロームダンプを作成する
- メモリダンプファイルを作成する
- カーネルデバッグを起動する

先ほどの表から分かるように、これらの例外は特権コードによるカーネルメモリの破壊、または致命的なバグの実行によって予期しない結果が生じた場合や、ハードウェア障害(メモリパリティエラーなど)が発生した場合に発生します。そのため、エラーコードが破棄されたときには、すでにリング 0 の書き込み可能なメモリ領域(I/O デバイスや制御デバイスの制御レジスタやステータスレジスタを含む)が破壊されている可能性があります。確実なことは、コードの実行が失敗したことだけなので、プロセッサをできるだけ早くフォルト状態に切り換え、I/O をフェールセーフ状態に移行させる以外に、安全な対処法はありません。

予期しないカーネル例外の発生後に処理を続行することは、制御処理とデータ処理のどちらでも、必ず支障が生じるとは言えないまでも、潜在的な危険が付きまといまいます。このことは、Windows NT オペレーティングシステムに限らず、その他のオペレーティングシステム、リアルタイム・オペレーティング・システム、コントローラシステムにも当てはまります。Windows NT オペレーティングシステムでは、プロセッサを即座にホールド状態に切り換え、I/O スキャナのウォッチドッグをタイムアウトにすることで、カーネル例外に適切に対処します。Windows NT オペレーティングシステムのブルー画面は、単に PLC-5 プロセッサの赤ランプフォルトに相当します。

Windows NT オペレーティングシステム用の拡張機能を提供している一部のベンダーは、メモリを区別化することにより、予期しないカーネル例外が発生した場合にもリアルタイムシステムの動作を安全に継続できると主張しています。当社は、この手法には見るべきところはなく、例外に対する対処法として潜在的な危険を伴うと考えています。その理由を以下に示します。

- リアルタイムシステムのコンテキストでも、予期しないカーネル例外が同等以上の可能性で発生する(以下の項を参照)。
- 適切なフェールセーフ動作を実現するには、Windows NT オペレーティングシステムと同様に、リアルタイムシステムもホールド状態に移行させる必要がある。
- この状態からは、シャットダウンハンドラを安全に呼び出すことができない。たとえ呼び出せたとしても、破壊、割り込みスタックのオーバフロー、NMI からの回復には使用できない。そのため、ウォッチドッグのタイムアウトが発生させ、I/O をフェールセーフ状態に切り換えるには、I/O スキャナアダプタを使用する必要がある。
- 複数のシステム障害モードを使用すると、システムの複雑さが増すだけで、システムの完全性テストがはるかに困難になる。

システムが複雑になるほど、そして徹底したテストが行なわれないほど、オペレーティングシステムやデバイスドライバのバグが発生しやすくなります。Windows NT オペレーティングシステムは、提案されているリアルタイムシステムよりも複雑ですが、徹底的なテストによってその機能が検証されています。ソフトウェア制御システムで最もテストされていないコンポーネント、すなわち最もカーネル例外の原因になりやすいコンポーネントは、実際には I/O アダプタ・スキャナ・ドライバです。上記の区別化されたシステムでは、これらのドライバがリアルタイムシステムに組み込まれているため、Windows NT オペレーティングシステムよりも予期しないカーネル例外が発生しがちになります。

Compaq®, DEC®, Dell®, Hewlett-Packard®, IBM®, ロックウェル・オートメーションといった最高品質のハードウェア OEM メーカーから提供されている Pentium システムを使用した場合は、Windows NT オペレーティングシステムは極めて優れた信頼性を発揮します。これらの主要な OEM メーカーによって検証されたカーネルやドライバには、事実上バグがありません。実際のケーススタディでも、信頼性の問題と、完成度の低いコンポーネントやデバイスドライバを使用することは表裏一体の関係にあります。システムの BIOS を修正しないと、Windows NT オペレーティングシステムを実行できないような場合などは、メーカーによる十分な検証が行なわれていないものと考えた方がよいでしょう。

当社では、リアルタイムシステムと Windows NT オペレーティングシステムを同じ PC 上で実行するよりも、OEM の Pentium プラットフォームを使用して、Windows NT オペレーティングシステムの通常のアップグレードを行った方が信頼性が高まると確信しています。PC ベースのどの制御システムでも、徹底的にテストされた高品質なハードウェアを使用する必要があることは言うまでもありません。

Windows NT オペレーティングシステム内のカーネルオブジェクトは、メモリを区分化するかどうかに関係なく、デュアルポート RAM のメモリマッピング用の物理メモリのほとんどに常時アクセスし、さらにハードウェア I/O ポートにもアクセスします。カーネル例外が発生した場合は、バグを含んでいるコードが実行されたか、ハードウェア障害のために予期しない状態が検出されています。バグを含んだコードは即座に破棄されるとは限らず、すでにデバイスやシステムの制御構造体が破壊されている可能性があります。そのため、たとえ区分化されたリアルタイムシステムを使用したとしても、処理の安全な続行は不可能となります。

相互運用性/チューニングの容易さとリアルタイムパフォーマンス機能

サードパーティ製のハードウェアの簡単かつトラブルフリーな使用は、オープンな制御を実現するうえで欠かせない、システムデザイン上のもう一つの主要な問題です。同時に、リアルタイムパフォーマンスの向上に寄与する高度な機能がサポートされていれば好都合です。すでに説明したように、Windows NT オペレーティングシステムでは、相互運用性とパフォーマンスを保証する目的から、標準のドライバ規則が定められています。すなわち、ISR のコード長をできるだけ短くし、DPC の処理が少々遅れても支障がないようにドライバを設計することが推奨されています。

これらの規則や、PnP(プラグ・アンド・プレイ)構成に関するその他の数多くの規則に従って作成されたドライバは、お互いに問題なく共存することができます。このことは、「オープン」な制御、すなわちハードウェア機器のトラブルフリーな混在使用を実現するうえで基本となる条件です。ただし、リアルタイムパフォーマンスを限界まで高めようとした場合には、この条件が足かせになることも事実です。

Windows NT オペレーティングシステムのリアルタイムパフォーマンスに批判的な人々は、いくつかの点で繰り返し不満を表明し、その解決策をリアルタイム拡張機能に求めています。ただし、これらの拡張機能のほとんどは、リアルタイムパフォーマンスと引き換えに、チューニングの容易さ、トラブルフリーなドライバ動作といった、Windows NT オペレーティングシステムのその他の設計思想を大幅に犠牲にしています。

よく聞かれる批判と要望をいくつか取り上げてみましょう。

スレッドがISR やDPC より常に低い優先順位で実行される。 Windows NT オペレーティングシステムの API の多くは、同期化用のカーネルオブジェクトを使用できないため、IRQL レベルが引き上げられた状態では呼び出せません。IRQL レベルが引き上げられた状態でスレッドを実行できるようにすると、それらにも ISR やDPC と同じ制約が適用されることとなります。タイムクリティカルなイベントは、システム割り込みやタイマからしか発生しないので、ISR やDPC でも同等の機能を実現できます。スレッドは、ISR やタイマDPC よりも使いやすいプログラミングモデルかもしれませんが、パフォーマンスの点では大きな違いはありません。

DPC が優先順位のないFIFO キューで処理される。 この場合、だれが優先順位を割り当てるかが問題になります。また、優先順位の高いDPC が繰り返し実行されると、優先順位の低いDPC の実行が、その内容に関係なく大幅に遅れることも考慮しなければなりません。このような機能を適切に使用するには、エンドユーザが優先順位を調節する必要があり、チューニングが複雑になってしまいます。他にどのようなドライバが使われるか分からないため、ドライバ自身に優先順位を決定させることはできません。Windows NT オペレーティングシステムでは、FIFO を採用することで、それぞれのドライバがキューを確実に使用できるようにしています。

DPC が割り込みより常に低い優先順位で実行される。 DPC の目的は、割り込み処理を完了させたり、タイマルーチンを実行することです。IRQL レベルが引き上げられた状態でDPC を実行する必要がある場合は、その処理をISR の内部で実行してください。割り込みより優先されるDPC を使用することと、その処理をISR の内部で実行することには、実質的な違いはありません。

Windows NT オペレーティングシステムでは、DPC のタイマルーチンよりも割り込みの方が緊急性が高いと見なされます。タイマルーチンにDPC 用とは異なる IRQL レベルを割り当てると、ポーリング方式のドライバの確実性が大幅に向上します。しかし、Windows NT オペレーティングシステムの観点から見ると、タイマルーチンはI/O 要求のタイムアウト機能を実現し、デバイスをポーリングして処理の完了を確認するために使用されるのが普通です。タイマ割り込みルーチンを指定し、専用の IRQL レベルを割り当てることができれば便利かもしれませんが、この機能が絶対に必要というわけでもありません。

WaitForObject が優先順位付きではないFIFOで実現されている。また、同期化オブジェクトが優先順位の自動調節に対応していないため、優先度が逆転する可能性がある。これらの問題により、相互排他セマフォ(ミューテックスオブジェクト)、クリティカルセクションなどの単純な同期化オブジェクトを使用したときに、優先順位の高いスレッドが優先順位の低いスレッドによって妨害される可能性があります。

WaitForObject に優先順位を付けると、優先順位の高いスレッドがキューの前方に挿入され、最小の待機時間で同期化オブジェクトにアクセスすることができます。さらに、同期化オブジェクトを所有しているスレッドを、待機スレッド用の最高レベルの優先順位まで自動的に引き上げると、単純な同期化オブジェクトの優先度の逆転は問題にはなりません。

しかし、Windows NT オペレーティングシステムには、探索的な手法による優先順位の動的変更機能があるため、これらの概念は当てはまりません。同期化オブジェクトは、スレッドの優先順位クラスとは無関係に使用されるため、上記の機能を取り入れるとスレッドディスパッチャが混乱し、その処理が複雑になります。スレッドディスパッチャの動作にもかなりのオーバーヘッドがかかります。Windows NT オペレーティングシステムでは、リアルタイムスレッドの優先順位でさえ実行中に変更できるので、上記の機能をサポートすると、プロセスサービス API が大幅に複雑になってしまいます。

かわりに、プリミティブを使用してやや複雑な同期化オブジェクトを作成すると、ロックパフォーマンスにあまり影響を与えずに、これらの問題に対処することができます。優先度の逆転を防止し、優先順位付きの待機を実現するカプセル化クラスを使用、作成することが可能です。ただし、ランタイムライブラリの内部から同期化を行なう際に、意識されないロックが発生する可能性は依然として残されています。

システムのスケラビリティと移植性

Windows NT オペレーティングシステムのデザインのうち、ソフトウェア制御にとって最適といえるのが、システムのスケラビリティと移植性です。Windows NT オペレーティングシステムは、ローエンドの Pentium システムから Pentium Pro システム、さらにはデュアル(2)またはクアッド(4)のマルチプロセッサシステムまで、広範なプロセッサパワーをサポートしています。そのため、ソフトウェアに手を加えることなく、CPU パフォーマンスを約 8 倍に向上させることが可能です。Pentium II プロセッサの投入により、今後 1, 2 年の間にハイエンドパフォーマンスはさらに倍増することが予想されます。対称型マルチプロセッシング (SMP) を導入すると、アプリケーションスレッドが効率的に実行され、パフォーマンスが一層高まります。また、Windows NT オペレーティングシステムは複数の CPU アーキテクチャに対応しており、x86 プロセッサと Alpha プロセッサの間でリング 3 のアプリケーションコードを簡単に移植することができます。ドライバの移植はアプリケーションコードほど簡単ではありませんが、C 言語で記述することができるので、高レベルな移植性が実現されます。

標準ドライバと管理のしやすさ

標準ドライバとプラグ・アンド・プレイ規格を直接利用できることも、Windows NT オペレーティングシステムの優れた設計思想の利点の一つです。ソフトウェア制御では、I/O スキャナ、産業用ネットワーク、ソフトウェアコントローラが、TCP/IP ネットワーク、シリアルポート、SCSI ポート(さらに将来的には USB や FireWire デバイス)にアクセスする必要があります。スレッドは Microsoft の標準ドライバを自由に使用でき、Windows NT オペレーティングシステムのドライバ間アクセス、クラスドライバ、ミニポートドライバの機能により、各種デバイスへの対応も簡略化されています。ただし、これらは標準の Windows NT オペレーティングシステムを使用した場合だけ得られるメリットです。

Windows NT 5.0 オペレーティングシステムで追加される新機能を使用すると、デバイスの管理が一層容易になります。これは、Microsoft の第 2 世代のプラグ・アンド・プレイ機能によるものです。Microsoft が提供しているデバイスドライバの自動構成システムと、ドライバの振る舞いに関する規則は、保守性と相互運用性に優れたオープンな制御を実現する優れた基盤となります。

まとめ

ブルー画面に代表される Windows NT オペレーティングシステムの予期しない例外処理は、ソフトウェア制御アプリケーションでも許容されます。例外処理ハンドラによって多くのシステム障害モードが統一的に処理され、制御スレッドの処理がただちに中止されるので、I/O スキャナのウォッチドック機能を使用して、システムの障害を速やかに検出し、I/O をフェールセーフ状態に切り換えることができます。

Windows NT オペレーティングシステムのデバイスドライバには、リアルタイムパフォーマンスを向上させる先進の機能が若干欠けていますが、これらの機能のほとんどは、ユーザによる高度なチューニングと管理を必要とし、さまざまなベンダーから提供されるハードウェアやソフトウェアの統合性を大幅に損います。Windows NT オペレーティングシステムは、デバイスの相性問題を回避する厳格なルールを保ちながら、高レベルのパフォーマンスを実現しています。

Windows NT オペレーティングシステムは、その先進の機能により、卓越したスケーラビリティ、移植性、管理性を実現しています。ネットワーク処理や通信のための主要なドライバも用意されており、アプリケーションやドライバから利用することができます。また、綿密に計画されたプラグ・アンド・プレイ機能により、セットアップやハードウェア設定もかつてないほど簡単になっています。

Windows NT の信頼性とフォルトトレランス

ターゲットとする市場

Windows NT オペレーティングシステムは、ソフトウェア制御アプリケーションを直接視野に入れた製品ではなく、基幹業務サーバや高性能ワークステーションをメインターゲットとしています。そのため、Windows NT オペレーティングシステムではラバスタ性、信頼性、パフォーマンスが重視されており、同時にユーザによる偶然または故意の攻撃に対する防御や、個々のアプリケーションの分離性が考慮されています。ハードウェアコンポーネントやシステムの品質を保証するため、Microsoft は先進の検証プログラムも開発しています。

Microsoft の検証/認定戦略

Windows NT オペレーティングシステムは、200 種類以上の一般的なプラットフォームでの互換性テストや負荷テストを経て市場に投入されています。Microsoft 社内でのアルファテストやベータテストも広範に実施されており、品質に対する評価は PC 用のオペレーティングシステムの中では群を抜いています。Windows NT オペレーティングシステムの出荷実績は、1997 年の時点でインストールベースで約 500 万本に達しており、Windows NT Advanced Server (NTAS) は約 80 万のシステムで採用されています。その普及度は、NTAS 単独でもすべての UNIX システムの合計出荷実績をしのぐほどです。

Microsoft は、社内でのアルファテストやベータテストに加えて、Windows Hardware Quality Labs (WHQL) によるプラットフォーム検証テストを実施しています。WHQL は、ハードウェアベンダーに各種のシステムハードウェアに関する仕様書を提供します。さらに、ハードウェア OEM メーカーには、ハードウェア検証用の標準化された自動テスト機能が提供されます。これらのテストに合格したハードウェアは、Windows のロゴを付けることが認められます。この検証プロセスが、最高水準の自己認定プログラムの基盤となります。

当社をはじめとする OEM メーカーは、自動テストのログを WHQL に提出します。WHQL はこのログをチェックし、テストレポートを作成し、基準を満たしている製品をハードウェア互換性リスト (HCL) に追加します。OEM メーカーは、ログに加えてシステムのサンプルを Microsoft に提供します。Microsoft は、これらのハードウェアを無作為にスポットテストする権利を持ちます。また、エンドユーザから Microsoft ナレッジベースに繰り返し障害が報告された場合にも、これらのハードウェアがテストされることとなります。このテストに不合格となった場合は、該当するハードウェアの製造メーカーは自己テストを行なう権利を剥奪され、正規の認定メーカーとしての地位を取り消されます。

Microsoft は、このような徹底的な社内/社外テストを通じて、業界の歴史の中でも最高品質のソフトウェアプラットフォームを、巨大な互換ハードウェア市場に適応させる体制を確立しています。ハードウェア互換性テストには、Microsoft と OEM メーカーの多大な努力が注がれているため、当社は標準的なシステムに適用される広範なテストを無駄にしないよう、極めて慎重に Windows NT オペレーティングシステムに取り組んでいます。

フォルト・トレランス・テクノロジー : RAID と Wolfpack

Windows NT オペレーティングシステムがターゲットとしている基幹業務アプリケーションの分野では、信頼性に対する徹底的な検証に加えて、フォルトトレランス機能も必要となります。Windows NT オペレーティングシステムは、ソフトウェア制御に適した数々の先進機能を備えています。これらの機能の中でも特に重要となるのが、RAID (Redundant Arrays of Independent Disks)ファイルシステムのサポート機能と、ファイルサーバ/データベース・サーバ・ソフトウェア用の Wolfpack クラスタリングソフトウェアです。

標準の Windows NT オペレーティングシステムでは、レベル 0 (DSA : パリティなしデータ・ストライピング・アレイ)、レベル 1 (MDA : ミラーリング・ディスク・アレイ)、レベル 5 (PDA : 分散パリティ付き並列ディスクアレイ)の 3 種類の RAID がサポートされます。MDA では、すべてのハードディスクアクセスが 2 台のディスクドライブに重複して適用されます。どちらか一方のディスクドライブに障害が発生しても、システムはフォルトを引き起こさずに処理を続行します。MDA は、ディスクシステムの活線挿抜(RIUP)を可能にすることを目的としています。障害が発生したドライブを交換すると、サービスを中断することなく、残りのドライブとのディスクイメージの同期処理が自動的に実行されます。

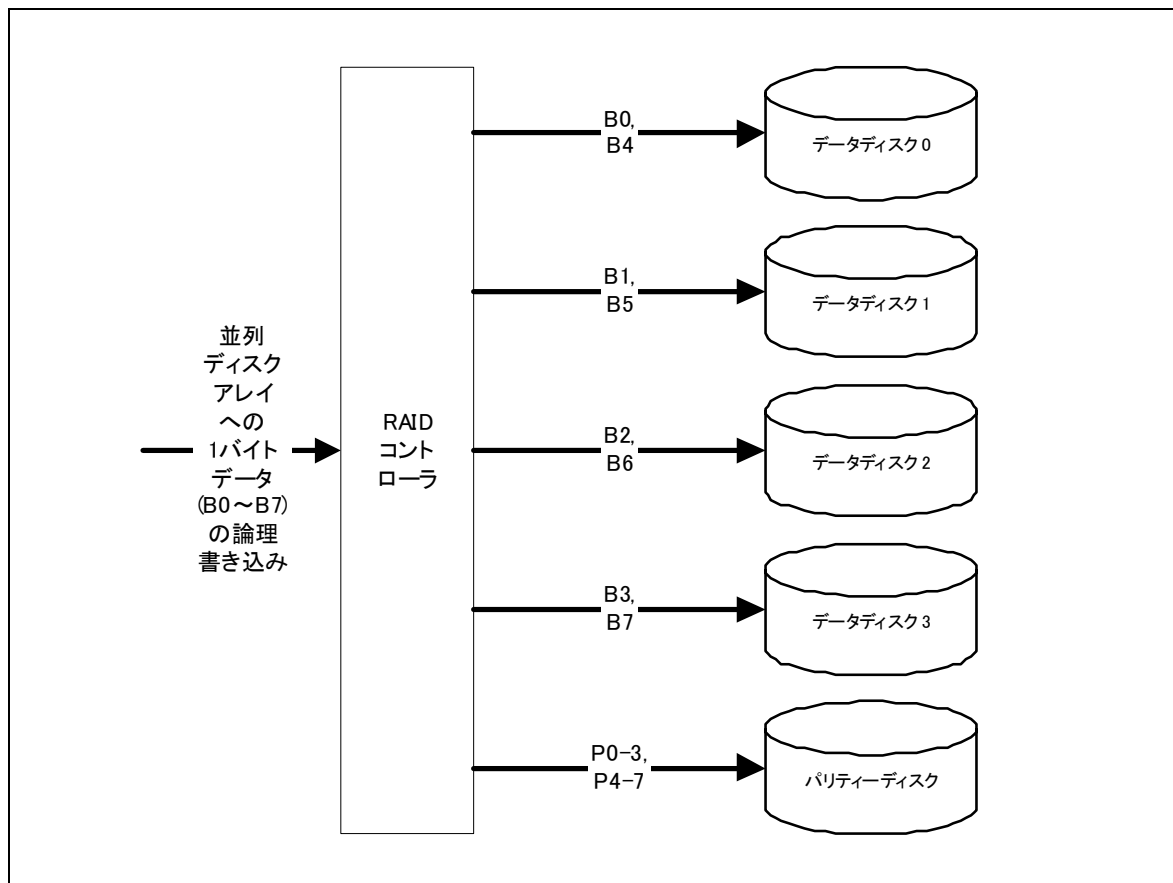


図 3 : RAID レベル 3 の動作(RAID レベル 5 も同様)

レベル3のRAID (PDA)では、ディスクアレイ中のN台のデータディスクにデータが分散して転送され、さらに冗長チェックディスクに各ビットのパリティビットが記録されます。いずれか1台のディスクが故障しても、残りのドライブのデータとパリティデータを組み合わせることで、故障したディスクの状態を復元することができます。Windows NT オペレーティングシステムでサポートされているレベル5のRAIDでは、処理方式がこれよりも多少複雑になります。PDAは、フォルトトレランス機能だけでなく、ディスクのバースト転送速度の向上にも効果的です。たとえば、7台構成のPDAを使用すると、単一のドライブを使用した場合と比べてバースト転送速度が6倍になります。MDAと同様に、PDAでもドライブを交換したときの自動同期処理がサポートされています。さらに、高度なRAIDシステムでは、ドライブスピンドルの同期化もサポートされているため、大規模な順次アクセスの性能を最大限に高めることができます。

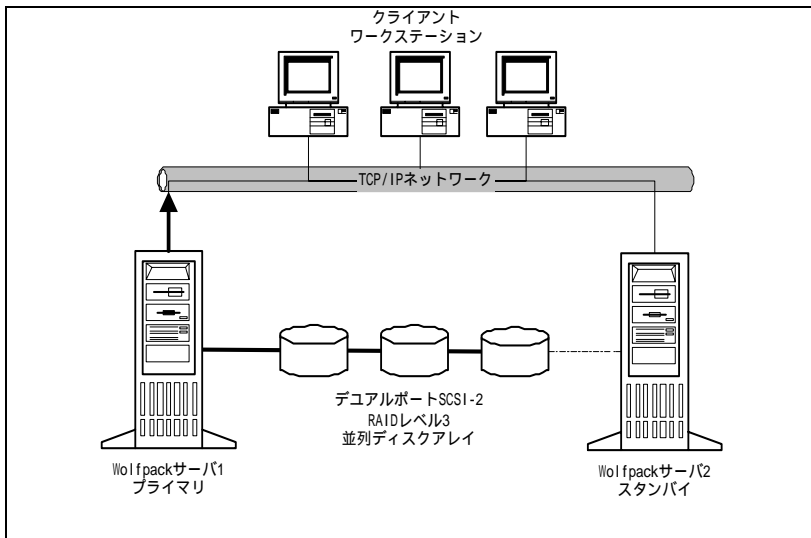


図 4 : Microsoft の Wolfpack システムのブロック図

Microsoft は RAID 機能に加え、Wolfpack クラスタリングプロジェクトを通じて、ファイルサーバやデータベースサーバ用のフォルトトレランス機能の開発にも取り組んでいます。フェーズ 1 の Wolfpack では、デュアルポート RAID ドライブシステムを共有する 2 台のサーバを、フォルトトレラントなファイルサーバや SQL Server データベースサーバとして機能させることができます。これらのサーバは IP アドレスを共有するため、ネットワークからは単一のシステムとして認識され、クライアントシステムに冗長性を透過的に提供します。Wolfpack には、プラットフォーム上でフォルトトレラントなアプリケーションを作成するためのシステム API も用意されています。将来的には、より大規模な汎用クラスタやフォルト・トレラント・クラスタもサポートされる予定です。

まとめ

Windows NT オペレーティングシステムは、基幹業務サーバと高性能ワークステーションの両方をターゲットとしています。Microsoft は、Windows Hardware Quality Laboratory (WHQL) を通じて、比類のない信頼性を実現する OEM ハードウェア検証プログラムを実施し、極めて広範囲の互換ハードウェアを提供しています。Windows NT オペレーティングシステムには、標準でレベル 0, 1, 5 の RAID をサポートする高性能なフォルト・トレラント・ディスク・アレイ機能が搭載されています。Wolfpack クラスタリングソフトウェアは、現時点ではまだベータテストの段階ですが、冗長性を備えたファイルサーバ/データベースサーバ機能を Windows NT オペレーティングシステムに提供してくれるはずです。

拡張アプローチの主な問題点

ここまでは、確定的制御に関する一般的な背景情報を説明し、Windows NT オペレーティングシステムのパフォーマンス、デザイン、信頼性、フォルトトレランス性を詳細に検討しました。ここからは、標準の Windows NT オペレーティングシステムに取ってかわる代替ソリューションの問題点をいくつか取り上げることになります。これらのソリューションでは、ハードウェア・アブストラクション・レイヤ(HAL)にリアルタイム機能を追加し、Windows NT オペレーティングシステムとリアルタイムカーネルを組み合わせる手法などが採用されています。

その位置付けに混乱が見られるにもかかわらず、VenturCom, Imagination Systems, Radisys といった当社が検証した代替ソリューションは非常に似通っており、HAL の拡張、および独自のリアルタイムカーネルによるリアルタイム処理のスケジューリングを共通して採用しています。これらのソリューションは、時間の経過とともに相互に影響を及ぼし合い、類似性はますます高まっています。各社のリアルタイムカーネルは、過去の製品をベースにしたものだと言われていますが、実際にはコードが一新されています。リアルタイム拡張機能の点では、これらのソリューションはどれも似たようなものです。

VenturCom, Imagination Systems, Radisys のリアルタイム拡張機能の概要

Radisys INtime では、Intel プロセッサに搭載されているハードウェア・タスキング・モデルを使用して、リアルタイムカーネルと Windows NT オペレーティング・システム・カーネル間のメモリ保護機構を実現しています。これが、各社から提供されているソリューションの最も大きな違いです。この理由から、INtime は Intel プロセッサにしか適用できず、対称型マルチプロセッシング(SMP)にも対応していません。デュアル・プロセッサ・システムがすでに広く普及しており、クアッド・プロセッサ・システムも登場しはじめていることを考えると、この制約は大きなマイナス要因となります。

VenturCom, Imagination Systems, Radisys が提供しているソリューションは、どれも Windows NT オペレーティングシステムの確定的制御パフォーマンスを確実に向上させる手段となりますが、デザイン、複雑さ、サポート、そして長期的な有効性の点で大きな問題を抱えています。

デザインに関する問題

各社のソリューションでは、リアルタイムシステムの応答性を維持するために、ハードウェア用の最高の IRQL より高い割り込みレベルでリアルタイムカーネルが実行されます。これらのカーネルは、あたかも Windows NT オペレーティングシステムの最高優先順位の割り込みであるかのように機能します。オペレーティングシステムから見ると、これらのカーネルは最高優先順位の割り込みサービスルーチン(ISR)と見なすことができます。Windows NT オペレーティングシステムの典型的なドライバでは、ISR のコードパス長が 2,500 命令以内に、さらに遅延時間が 80 μ sec 以下に制限されます(200MHz の Pentium Pro の場合)。

これらのソリューションでは、IRQL を引き上げた状態で大規模な制御アプリケーションが実行されるため、Windows で定められているドライバ作法が侵害されるおそれがあります。多くのデバイスでは、割り込みがあまり長期間阻止されるとデータが消失してしまいます。たとえば、16650 UART は 16 バイトの FIFO を使用します。19.2Kbps の転送では、このキューは約 8msec しかデータを保持できません。同様に、ネットワークアダプタも受信パケット用のキューとして 8~64KB の RAM を使用します。これらのキューのオーバフローが発生すると、新しく着信したフレームが破棄され、タイムアウトロジックによる再受信が必要になります。

一部のベンダーからは、Windows NT オペレーティングシステムのスターベーションを防止する機能が提供されています。これらの機能は、頻繁な(250~500msec 間隔の)タイムスライス処理、もしくは SMP システムを必要とします。タイムスライス処理を行なうと、ISR のクリティカルセクションのプリエンブションが発生する可能性が大幅に高まり、好ましくありません。標準ドライバでは、IRQL を引き上げてプリエンブションを防止していますが、この手法はリアルタイム拡張システムには適用できません。リアルタイムシステムのドライバは、検証されていないモード(クリティカルセクションのプリエンブションが発生する可能性がある状態)で動作することになり、その振る舞いはガイドラインに反したものとなります。さらに、この状態ではドライバの再生と分離が極めて困難になり、無作為なシステムクラッシュが発生します。また、すでに説明したように、SMP システムは拡張機能を使用しなくても優れたパフォーマンスを発揮するため、SNP システムを常時使用する場合は、Windows NT オペレーティングシステムをあえて修正する必要はありません。

また、IRQL が引き上げられた状態では、カーネルとの同期処理を行ない、カーネルサービスを直接呼び出すことができません。同様に、デバイスドライバやシステムライブラリも呼び出せません。リアルタイムシステムのメモリを区分化しない場合は、リアルタイムスレッドを起動する前にカーネルサービスを呼び出し、ディスクイメージや DLL(ダイナミック・リンク・ライブラリ)をロードしておくことができます。しかし、区分化メモリを採用しているシステムでは、この手法も使用できません。その結果、これらのシステムでは専用のプロセスローダとイメージローダが必要になります。

既存のデバイスドライバを使用できないことも、これらのソリューションの大きな欠点となります。ハードディスク、シリアルポート、パラレルポート、TCP ネットワーク用のドライバは、ソフトウェア制御アプリケーションを作成するうえで欠かせません。さらに、リアルタイムシステムではデバイスの区分化も必要になりますが、Windows NT オペレーティングシステムの検出機能や、今後搭載される新しいプラグ・アンド・プレイ機能は、区分化されたデバイスを認識できません。

今後は、周辺機器用の USB (Universal Serial Bus)規格や FireWire (IEEE 1394)規格が、ソフトウェア制御の分野で重要な役割を果たすようになると予想されます。これらのサブシステム用の Microsoft のクラスドライバは、リアルタイムシステムでは使用できません。USB は現在の PC のキーボード用、プリンタ用、通信用の各ポートに取ってかわるもので、ポート当たり最大 126 のデバイスを接続可能、4 ~ 12Mbps の転送レート、活線挿抜(RIUP)のサポートといった特長があります。FireWire は、デバイス間的高速転送(200Mbps 以上)を実現し、マルチドロップ機能や RIUP 機能を提供します。

リアルタイムシステムでは、Microsoft 標準の静的ランタイムライブラリに対するリンクが認められません。これらのランタイムライブラリは、オペレーティングシステムの同期と DLL に依存しているためです。したがって、リアルタイムカーネルのスレッドや割り込みルーチンで浮動小数点演算を実行できるようにするには、浮動小数点関数と超越関数をサポートした、マルチスレッド型の完全に再入可能なランタイムライブラリをベンダー自身が作成する必要があります。このようなベンダー独自のランタイムライブラリを使用すると、制御製品に Microsoft のキーテクノロジーを直接取り込むことが不可能になります。当社では、DLL、COM(Component Object Model)コンポーネント、ActiveX コントロールといった Microsoft のテクノロジーと制御システムとの統合が、オープンな制御が普及する鍵となると考えています。また、Windows ベースの MMI/SQL データベースシステムとの直接的な統合も、オープンな制御に欠かせない条件となります。

複雑さに関する問題

リアルタイムシステムでは、開発面での制約があり、予測不能な動作が発生する可能性があることに加えて、システムが複雑になることも問題となります。汎用モーション制御(GMC)、コンピュータ数値制御(CNC)、協調駆動制御といった、浮動小数点演算を多用する高速アプリケーションでは、リアルタイムシステムによる確定性の向上が望ましく、場合によっては必要となります。しかし、Windows NT オペレーティングシステムをあえて修正しなくても、はるかに単純な方法でこの問題を解決する手段があるはずです。

PCIバス・マスタリング・コプロセッサ・ボードを使用すると、Windows NT オペレーティングシステムの動作を妨げることなく、PCIバス上で非常に高速な制御を実現することができます。この手法を使用すれば、Windows NT オペレーティングシステム上で動作する制御アプリケーションに制約を与えずに、GMC/CNCアプリケーションに適したリアルタイムパフォーマンスを実現することができます。これらのコプロセッサボードは、メインプロセッサがハードウェアストップ(CLI; HLT;)を実行したり、クリティカルなNMIを検出した場合にも処理を続行できるので、どのリアルタイム拡張システムよりもファイアウォール性に優れています。

デバイスやドライバを、リアルタイムセットやWindows NT オペレーティング用のセットに区分化する必要があることも、システムを複雑にする要因の一つです。このことは、一見すると深刻な問題ではないように思えます。しかし、リアルタイム環境とWindows NT オペレーティングシステム環境の間で、共通クラスのデバイス(ディスクドライブ、シリアルポート、パラレルポート、ネットワークアダプタ)を共有する必要があることを忘れてはなりません。さらに、産業用インターフェイスカードのベンダーは、Windows NT オペレーティングシステム用とリアルタイムシステム用の2種類のデバイス・ドライバ・セットを用意しなければなりません。

リアルタイムシステムとWindows NT オペレーティングシステム間のローカル・プロシージャ・コール(LPC)機能を用意すると、これらのシステム間での情報の受け渡しが可能となります。しかし、この機能も複雑さが増す原因となります。LPCのプロキシやスタブは、Windows NT オペレーティングシステムのサービスにアクセスします。ベンダーから提供されているルーチンの中には、Windows NT オペレーティングシステムのレジストリ、ディスクドライブ、汎用ドライバにアクセスするものもあります。当然のことながら、リアルタイムスレッドがWindows NT オペレーティングシステムのサービスを使用すると、そのスレッドはもはやリアルタイムスレッドとは言えず、リアルタイムシステムのオペレーティングシステムからの独立性も失われてしまいます。

信頼性に関する問題

拡張された Windows NT オペレーティングシステムは、信頼性の点で標準の Windows NT オペレーティングシステムにどう見ても及びません。500 万人を超えるユーザ、開発時の徹底的なテスト、OEM メーカーを対象とした万全なハードウェア検証体制を考えると、Windows NT オペレーティングシステムの信頼性は極めて高いと考えられます。発生する障害の大半は、ハードウェア互換性リストに登録されていないか、リストに登録されている資格のないハードウェアやデバイスドライバに起因しています。

ただし、リアルタイム拡張システムはまだ開発の段階か、せいぜいバージョン 1.0 程度です。各ベンダーは過去の製品の実績からユーザの信頼を得ようとしています。製品デザインが一新されているため、多大な信頼は寄せられません。これらのシステムのほとんどはベータテストの段階のため、今後の信頼性はどうか分かりません。しかし、各社の規模と膨大なソリューション業者数から見て、Microsoft に匹敵するプラットフォーム検証体制を確立できる可能性はほとんどないと思われます。

各ベンダーのうち、Radisys INtime だけは、カーネルやそのデバイスドライバで予期しないカーネル例外が発生しても、動作の継続が可能だと主張しています。しかし、この状態で動作を安全に継続できるとは思えません(前述の「Windows NT オペレーティングシステムのブルー画面」を参照)。INtime は、当社の制御アプリケーション全体に対応できるほどの環境ではありません。仮に、INtime をはじめとするリアルタイムシステムをベースとして当社の制御システムを構築したとすると、その動作は Windows NT オペレーティングシステムとリアルタイムシステムの両方の正常動作に依存することになります。

サポートに関する問題

リアルタイム拡張システムが直面している最大の問題は、おそらく長期間にわたるサポートでしょう。一般に、Microsoft はサービスパックを配布し、その機能と目的に関する情報を提供することで目覚ましい成果を上げています。しかし、拡張環境をテストし、ソリューション業者を通じて要望を報告させ、ソリューション業者を通じて顧客にサービスパックを提供する作業を、サードパーティベンダーに頼って実施することは悪夢です。Windows NT オペレーティングシステムのメジャーリビジョンでも問題は深刻になるばかりです。サポートのレベルは次第に悪くなりがちです。なぜなら、システムはインストール状態から徐々に変貌していき、ベンダーの興味は新しい顧客に移るからです。

リアルタイム拡張システムでは修正された HAL が使われるため、これらのベンダーは OEM メーカーの HAL との競合にも対処しなければなりません。Microsoft は、拡張プラットフォーム用のハードウェアサポート機能を追加できるように、OEM メーカーに HAL のコードをライセンス供与しています。対称型マルチプロセッシング、電源管理、マルチバスなどの機能をサポートするには、HAL のカスタマイズが必要になる場合があります。残念なことに、HAL はモノリシックな DLL なので、カスタマイズされた各種の HAL をマージすることができません。そのため、一部の OEM ハードウェアはリアルタイム拡張システムではサポートされません。特に、高性能サーバプラットフォームでは互換性の問題が発生しやすくなります。

無責任なリアルタイム拡張システムのために、Microsoft と OEM メーカーのプラットフォーム認定テストが台無しになるおそれもあります。リアルタイム拡張システムでは、標準のデバイスドライバが実行されるときに、テストでは検証されていないストールやプリエンプションが発生するため、これらのドライバで障害が発生しやすくなります。また、この環境では障害の分離や報告も非常に困難です。

Windows NT 5.0 オペレーティングシステムでは、プラグ・アンド・プレイ機能の大幅な強化が予定されています。このことは、エンドユーザにとっては朗報となりますが、リアルタイム拡張システムのベンダーにとっては、サポートの面で深刻な問題をもたらす可能性があります。Windows NT 5.0 オペレーティングシステムでは、デバイス・ドライバ・モデルが変更されるため、新しいプラットフォームへの移行が急速に進むものと思われます。プラグ・アンド・プレイ機能は区分化されたデバイスを認識できないため、この機能を無効にするか、巧妙な回避策を考えないと、リアルタイムシステムの設定に支障が生じてしまいます。

長期的な問題

デザイン面での制約、システムの複雑さ、信頼性、サポートの問題を解決したとしても、まだテクノロジーの進歩による長期的な問題が残されています。Moore の法則、(Intel およびその他各社の)プロセッサの高速化、手頃な価格の SMP システムの普及などにより、Windows NT オペレーティングシステムのシステムパフォーマンスは今後さらに上昇し、リアルタイムパフォーマンスは自然と改善されていくことが予想されます。さらに、64 ビットアーキテクチャのサポートなど、Microsoft の継続的な開発努力を考慮すると、パフォーマンスの向上速度はさらに高まるはずで、その結果、Windows NT オペレーティングシステムを修正する手法は、徐々に姿を消していくものと予想されます。

ソフトウェアの成熟曲線も、ソフトウェア制御用のプラットフォームを選択するうえで重要な検討事項の一つとなります。ソフトウェア制御はまだ幼児期にあり、最も高速で、最もタイムクリティカルなアプリケーションにいきなり適用することは無謀といえます。現在のソフトウェア制御では、パフォーマンスのボトルネックに関心の大半が向けられています。しかし、製品が成熟した段階では、これらのボトルネックは意味を持ちません。これらの製品は、まず各種のアプリケーションでフィールドテストと検証を行なう必要があります。そうしないと、賢明なエンドユーザは、最もミッションクリティカルなアプリケーションにこれらの製品を適用する気にならないでしょう。

Windows NT オペレーティングシステムの限られたデータをコンテキストから切り離して提示して、制御には向かないと主張しているベンダーがあります。これらのベンダーは独自の拡張機能の開発に着手し、Microsoft の標準テクノロジーを利用する機会を取り逃がしています。

1998 年の上半期には、Windows NT 5.0 オペレーティングシステムに Wolfpack 冗長/クラスタリング機能と OnNow テクノロジーが導入されました。OnNow テクノロジーは迅速な再開/再起動を実現しますが、Windows NT オペレーティングシステムによるメモリアクセスが制約されるシステムでは、機能する可能性は低いと思われます。また、Windows NT 5.0 オペレーティングシステムは次世代のプラグ・アンド・プレイ (PnP) 機能を搭載し、ドライバの動的なロードと除去、PCMCIA, USB, FireWire デバイスの活線挿抜 (RIUP) などを実現します。

小規模なベンダーが、オペレーティングシステムの市場で Microsoft の開発ペースに遅れずについていけるかどうかは大いに疑問があります。最大規模の拡張システムベンダーでさえ、従業員数は 300 人程度にすぎません。しかも、その収益の大半はハードウェア製品から生み出されています。このベンダーでリアルタイム拡張システムに直接従事しているのは、最大でも 20 人程度だと予想されます。このようなベンダーが、Windows NT オペレーティングシステムを直接基盤とするソリューションを提供することは困難です。

当社では、Microsoft のオペレーティングシステム開発に追従する戦略をとっています。これにより、完全なオープンシステムを実現すると同時に、十分にテストされた実証済みのテクノロジーを提供しています。

オープンシステム戦略に加えて、当社では Windows CE に関するプロジェクトにも着手しています。Windows CE バージョン 3.0 では、50 μ sec の割り込み遅延時間が実現され、ワーストケースパフォーマンスも理論的に実証されており(「ハードウェアリアルタイム性」)、小型システム用のプラットフォームとして最適です。

当社は、Windows NT と Windows CE の両方で SoftLogix コントローラを提供することを目指しています。

改善が望まれるその他の問題

Windows NT オペレーティングシステム用の拡張機能に関する数々の問題に加えて、見過ごされている問題がまだ数多く残されています。残念ながら、リアルタイム拡張システムのベンダーは、確定的制御パフォーマンスに目を向けすぎているようです。Windows NT オペレーティングシステムによるソフトウェア制御で改善が望まれる点を以下に示します。

- オペレーティングシステムおよびソフトウェア制御システムの保守性の改善
- 起動時間の短縮 (Microsoft の OnNow 機能の活用)
- 自己診断機能の改善
- システムを稼働させた状態での修復-電源投入状態での取り外しと装着、RIUP (Microsoft の PnP 機能による USB, FireWire, PCMCIA の活線挿抜)
- プラットフォームのスワップ性を強化することによる、システムの MTTR(平均修復時間)の短縮 (Microsoft の ZAW (Zero Administration Windows)もこの問題に取り組んでいます)

まとめ

リアルタイム拡張システム、およびこのシステムをベースとする制御アプリケーションを提供しているベンダーは、Windows NT オペレーティングシステムがソフトウェア制御アプリケーションには向いていないとの印象を植え付けようとしています。当社は、独自のテストと経験から、標準の Windows NT オペレーティングシステムは広範なソフトウェア制御アプリケーションに十分に対応しており、互換性のレベルや選択肢の豊富さの点では、リアルタイム拡張システムをはるかに凌駕していると考えています。

アプリケーションによっては、現行レベルを超えるパフォーマンスが望まれることは事実ですが、Windows NT オペレーティングシステムを強化する拡張システムは以下の問題を抱えています。

- 未熟さ
- Windows NT 製品との統合性の低さ
- システムの複雑さ
- サポートと信頼性に関する長期的な懸念

これらの拡張システムは、高レベルの確定的制御パフォーマンスを必要とする特殊なアプリケーションで有効な手段となることでしょう。ただし、そのためには Windows NT オペレーティングシステムとの互換性を高め、オペレーティングシステムがバージョンアップしても互換性を維持できるようにすることが条件となります。

結論

当社が Windows NT オペレーティングシステムを選択した理由の再確認

当社は、オープンな PC アーキテクチャベースのソフトウェア制御用として、どのオペレーティングシステムを採用するかという点で重大な決断に迫られていました。市場での状況から、Windows との互換性がソフトウェア制御を成功に導く鍵であり、Windows NT オペレーティングシステムが基幹アプリケーションに適していることは明らかでした。選択肢は Windows NT 互換システムに絞られましたが、市場には標準の Windows NT オペレーティングシステムに加えて、いくつかの拡張システム(Radisys INtime, VenturCom RTX, Imagination System Hyperkernel など)が投入されていました。

Windows NT オペレーティングシステムの機能、および拡張システムの機能とリスクについて詳細な調査を行った結果、当社は標準の Windows NT オペレーティングシステムをソフトウェア制御の基本的なプラットフォームとして採用しました。高速な Pentium プロセッサシステムや対称型マルチプロセッシングシステム上での Windows NT オペレーティングシステムは、広範なソフトウェア制御アプリケーションで十分すぎるほどのパフォーマンスを発揮します。また、Windows NT オペレーティングシステムのデザインや設計思想も、ソフトウェア制御の目標と完全に適合しています。

Windows NT オペレーティングシステムの RAID 機能(ミラーリング・ディスク・アレイと並列ディスクアレイ)は、制御アプリケーションにとって非常に好都合です。さらに、OnNow, Zero Administration Windows, Wolfpack クラスタリング、強化されたプラグ・アンド・プレイ、USB, IEEE 1394, FireWire, PCMCIA デバイスの活線挿抜とドライバの動的ロードといった Microsoft の新しいキーテクノロジーにより、近い将来数多くの革新的な新機能が実現されるはずで

リアルタイム拡張システムは、Windows NT オペレーティングシステムを上回るパフォーマンスを実現する可能性を秘めています。しかし、これらのシステムには以下の問題があります。

- リアルタイム拡張システムは必要なく、真のオープン制御の妨げとなる。
- 技術が成熟していない。
- システムが複雑になる。
- ハードウェアの互換性、サポート、将来性の点で長期的なリスクがある。
- PCI バス・マスタリング・コプロセッサ・ボードや専用コントローラでも、ほぼ同等のパフォーマンス改善効果が見込める。これらのハードウェアを使用すれば、Windows NT の修正、阻止、プリエンプションに関するリスクを回避できる。



Rockwell Automation helps its customers receive a superior return on their investment by bringing together leading brands in industrial automation, creating a broad spectrum of easy-to-integrate products. These are supported by local technical resources available worldwide, a global network of system solutions providers, and the advanced technology resources of Rockwell.

Worldwide representation.



Argentina • Australia • Austria • Bahrain • Belgium • Bolivia • Brazil • Bulgaria • Canada • Chile • China, People's Republic of • Colombia • Costa Rica • Croatia • Cyprus • Czech Republic • Denmark • Dominican Republic • Ecuador • Egypt • El Salvador • Finland • France • Germany • Ghana • Greece • Guatemala • Honduras • Hong Kong • Hungary • Iceland • India • Indonesia • Iran • Ireland • Israel • Italy • Jamaica • Japan • Jordan • Korea • Kuwait • Lebanon • Macau • Malaysia • Malta • Mexico • Morocco • The Netherlands • New Zealand • Nigeria • Norway • Oman • Pakistan • Panama • Peru • Philippines • Poland • Portugal • Puerto Rico • Qatar • Romania • Russia • Saudi Arabia • Singapore • Slovakia • Slovenia • South Africa, Republic of • Spain • Sweden • Switzerland • Taiwan • Thailand • Trinidad • Tunisia • Turkey • United Arab Emirates • United Kingdom • United States • Uruguay • Venezuela

Rockwell Automation Headquarters, 1201 South Second Street, Milwaukee, WI 53204 USA, Tel: (1) 414 382-2000, Fax: (1) 414 382-4444

Rockwell Automation European Headquarters SA/NV, avenue Herrmann Debrouxlaan, 46, 1160 Brussels, Belgium, Tel: (32) 2 663 06 00, Fax: (32) 2 663 06 40

Rockwell Automation Asia Pacific Headquarters, 27/F Citicorp Centre, 18 Whitfield Road, Causeway Bay, Hong Kong, Tel: (852) 2887 4788, Fax: (852) 2508 1846