



Green Hills Software

INTEGRITY™概要 説明資料

October 3, 2002

Advanced Data Controls Corp.



INTEGRITY登場の背景

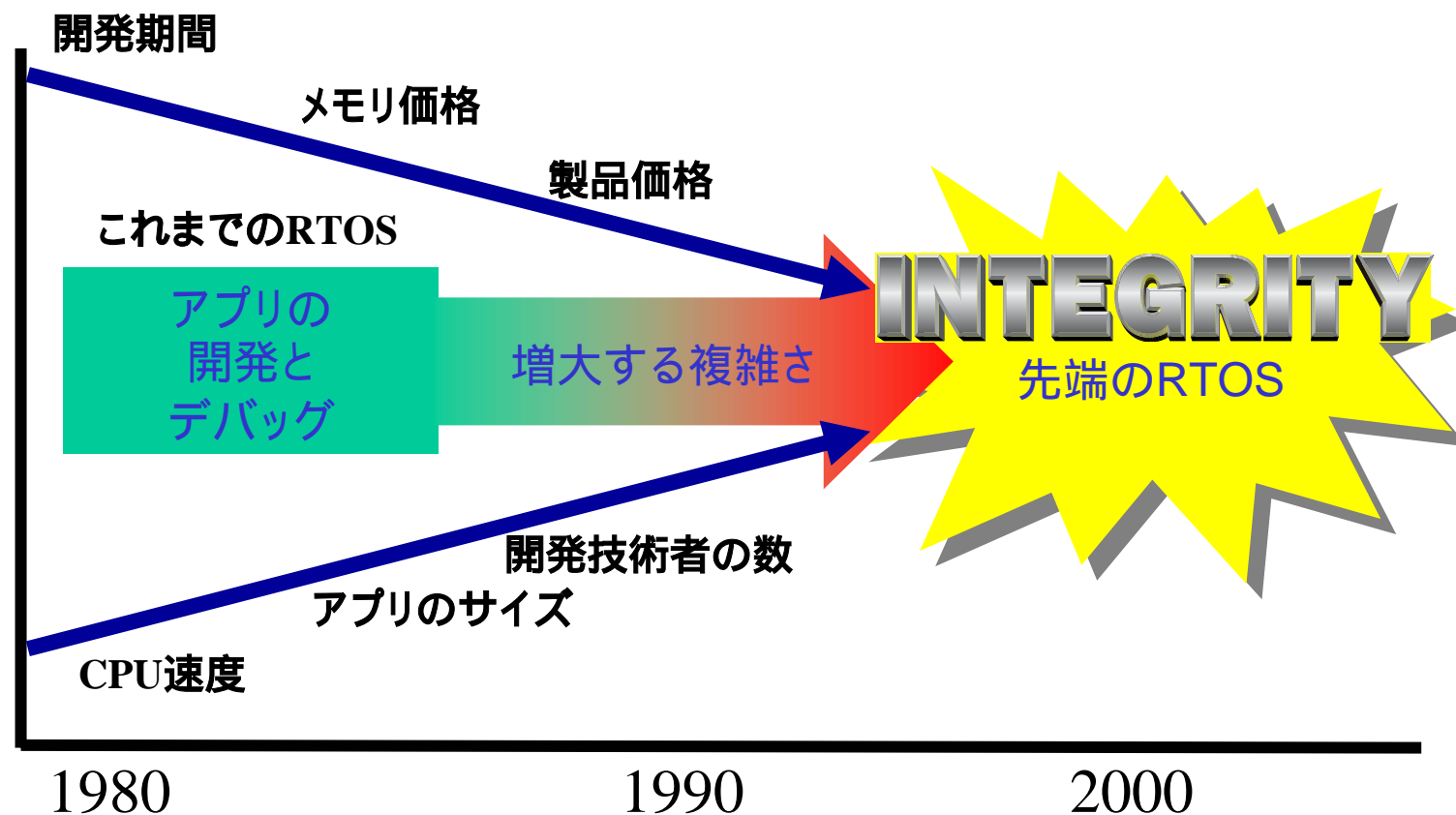
組み込みシステム開発を取り巻く状況の改善を
目指したRTOS



現状のプログラム開発状況

- **基本テクノロジー**
 - マイクロプロセッサ価格の急激な低下
 - マイクロプロセッサテクノロジーの微細化による複雑化 (SOC)
 - メモリ価格の急激な低下
 - 多くの商用RTOSには変化なし(テクノロジー・ロイヤリティともに)
- **大量生産によるコストダウン**
 - 組み込まれるマイクロプロセッサの価格や最終製品価格は低下
 - 最終価格に占めるソフトウェアコストの割合が増大
- **複雑化するアプリケーション**
 - より多くのメモリを使用し、より多くの技術者が開発に従事し、複雑さが増すため、バグの数も増大し、バグ修正に多くの時間が必要
 - 比較的短い開発期間

組み込みシステムの傾向



他のOSは

■ これまでのRTOS

- マーケットシェアの減少
- 過去のリリースとの互換性維持のために技術的な限界
- メジャーなRTOSではロイヤリティが必要
- ソースライセンスは非常に高価

■ Linux

- 興味を持つ人が増加
- 昔からのUNIXをベースとした設計上の技術的な限界
- ロイヤリティ不要
- 簡単なソースコード入手

顧客の要求は

- **高信頼性** これまでのRTOSの信頼性では不十分
- **開発効率** これまでの開発環境では本来の開発能力が十分に発揮できない
- **ライセンス形態と価格** 最善の製品を最善の価格で入手したい

INTEGRITYは

- マーケットが増大中(成長率でトップ)
- 先端RTOSテクノロジーによる高信頼性の達成
 - 完全に保護されたアドレス空間
 - CPU時間とメモリの保証
 - 安全でスケーラブルなタスク間通信
 - ハードリアルタイムスケジューリング
- 開発期間短縮に役立つ開発ツール群
- ロイヤリティ不要・GPL制限無し
- 比較的低価格のソースライセンス



高安全性アプリ向けの INTEGRITY導入例

■ 航空・宇宙・軍事 – 安全性が重要な市場

- Boeing
 - **B1B**, B-52, F-22, C-17のアビオニクス装置
- Lockheed Martin
 - **Joint Strike Fighter**, F-16, X-35, C130-Jのミッションコンピュータ
- BAE Systems
 - **Eurofighter** のコックピット表示装置
- L3 Communications
 - 民間航空機用衝突防止装置(TCAS)
- Goodrich
 - 一般航空機用の多目的コックピット表示装置
 - ヘリコプタ用タービンエンジン制御装置
- TRW
 - Comancheヘリコプタのアビオニクス装置
- Rockwell Collins
 - S-92 Sikorskyヘリコプタ
- Kaiser Electronics
 - F/A-18 Fighter Jetのコックピット表示装置



高信頼度アプリ向けの INTEGRITY導入例

- 通信、医療、プロセス制御
 - Lucent
 - 光ファイバ交換機
 - Nokia
 - ブリッジルータ
 - RFL
 - イーサネットプロトコルコンバータ
 - Outerlink
 - 無線SATCOMデータシステム
 - Bitrage
 - 無線データシステム
 - Kodak
 - 医療用画像処理装置
 - DAN Controls
 - Electronic Gas Volume Converters
 - Cymer
 - 印刷装置





INTEGRITYの技術的な 概要

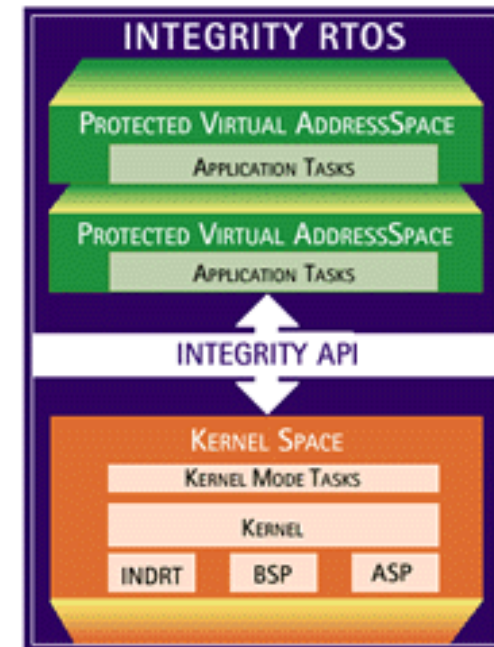
**組み込みシステムにおける究極の安全性と検証
可能な信頼性を追求した最先端のRTOS**



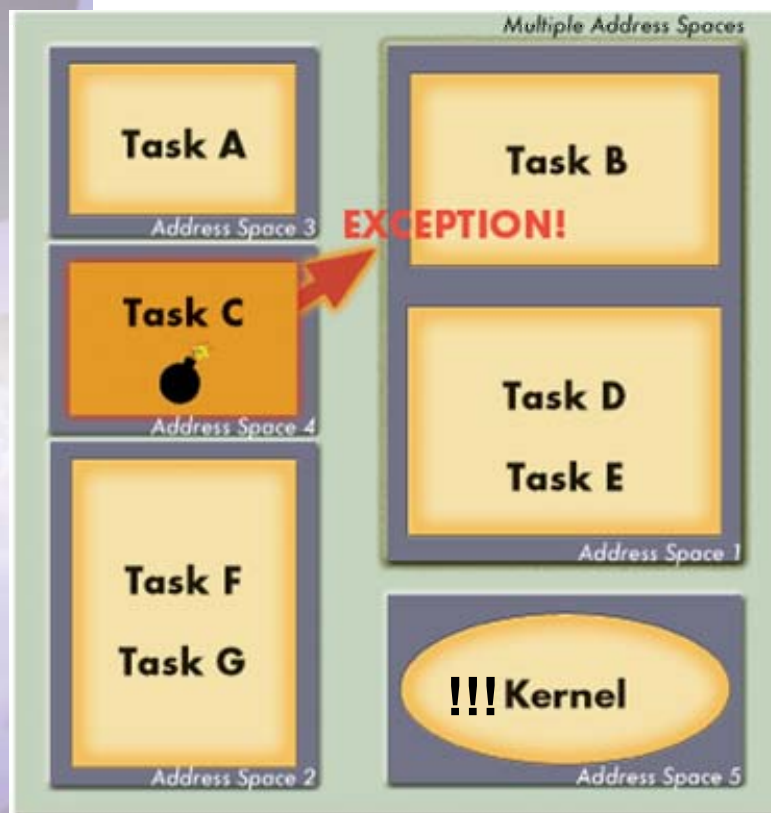
INTEGRITYの 基本的な優位性

- タスクは保護されたメモリ内で実行
- メモリとCPU時間の保証
- DO178B Level Aで認証可能
- 安全なタスク間通信
- 極小の割り込み遅延時間
- 高速でデターミニスティックなカーネル動作
- 最小70KBからのスケーラビリティ
- 分散マルチCPUへの対応
- 高品質開発ツールとの統合
- オブジェクト指向
- ロイヤリティ無し
- 比較的安価なソース

Reliable • Safe • Secure



完全なメモリ保護

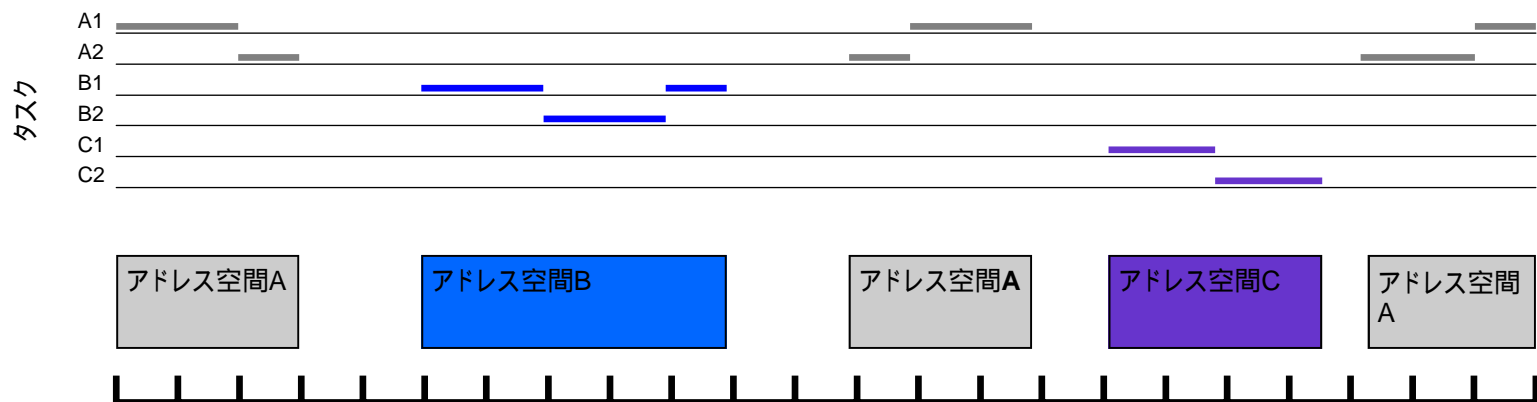


INTEGRITYでは、チップ内蔵のMMUにより保護された仮想アドレス空間を複数利用することが可能:

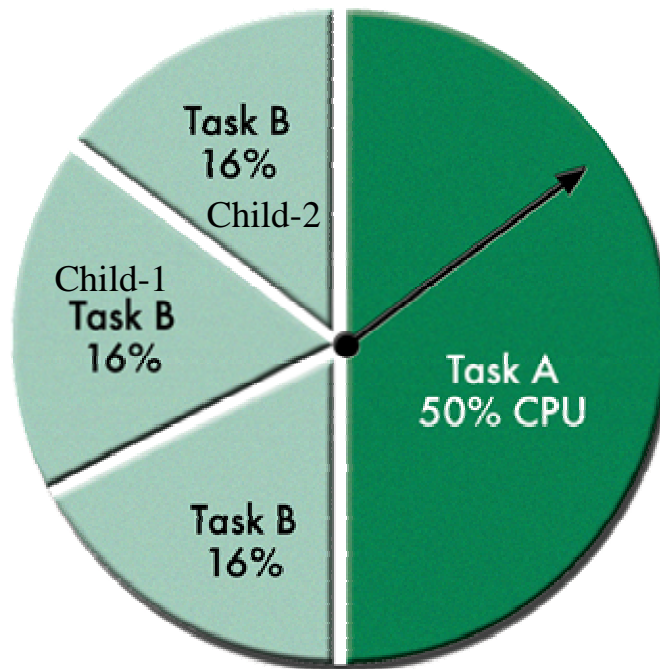
- タスクが自身のメモリ空間を越えるアクセスをしようとした場合
- カーネルは直ちにそれを検出し、当該タスクをサスペンド
- その後の処理はプログラマブル:
 - ✓ 当該タスクのKill・再スタート
 - ✓ エラーの通知やロギング

パーティションスケジューリングによるCPU時間の保証

- 各パーティション(アドレス空間)ごとの実行スケジュールが定義可能
- 特定のパーティションがアクティブなとき, そのパーティションに対するCPU時間を完全に保証
- 明示的な指定がない場合、バックグラウンドパーティションを実行
- ARINC 653 – Avionics Application Software Standard Interface
 - ARINC 653で規定されたパーティションスケジューリングに従ったアドレス空間スケジューリングを実現

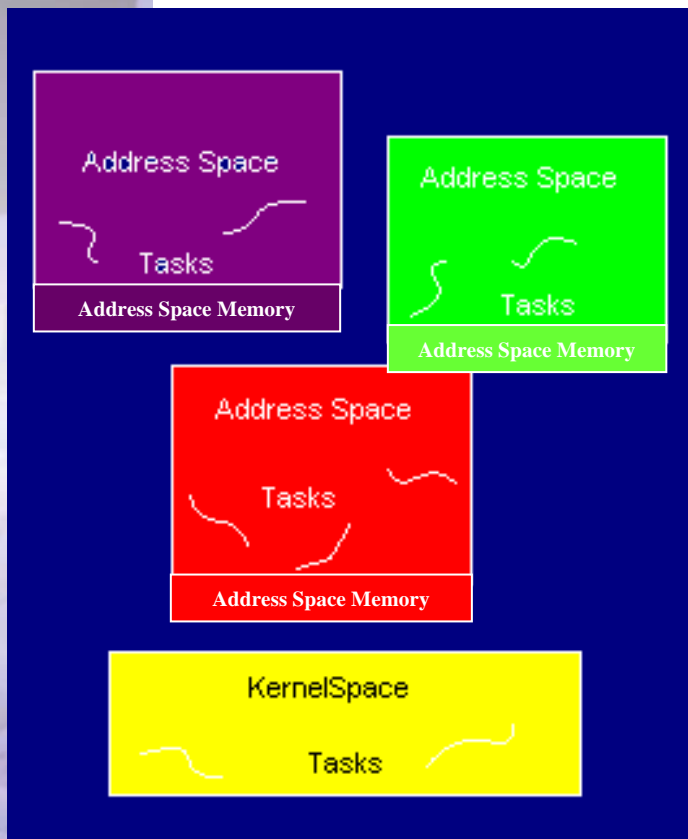


標準スケジューラによる CPU時間の保証



- 同一優先順位内の各タスクに重み (weight) によるCPU時間の割当て
- タスクが他のタスクを生成しても親子タスクを合わせたCPU時間を増加させることは不可能
- 他タスクから生成されたタスクには、本来指定されているweightではなく生成元をベースとする新しいweightが割り当てられる

メモリの保証



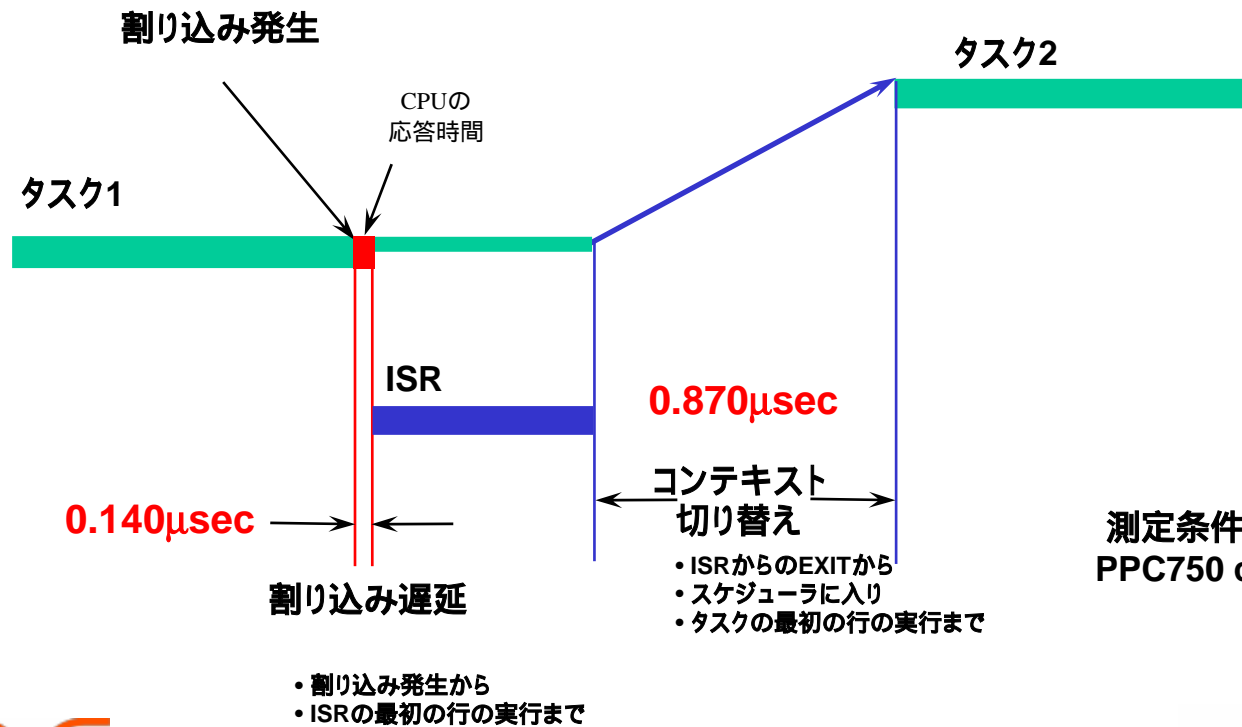
- システムを複数の仮想アドレス空間に分割
- 各アドレス空間ごとに物理メモリの割り当て
- リンク時と静的ブートテーブルでの割り当て指定
- 割り当ては固定的に行われ、後での変更・追加は不可能
- カーネル内部にメモリプールを置かず、カーネル管理対象のオブジェクト作成時にはタスク空間のメモリを使用

割り込み遅延 (カーネルでの確定的な動作)

- 割り込みを一切マスクしないINTEGRITYのサービスコール(常に割り込み受付可能)
 - 最高優先順位の割り込みに対するサービスは常に極小の遅延で実行
 - 割り込みサービス終了からスケジューラへの移行は、スケジュール用のタイマ割り込みにより通知
- ほとんどのサービスコール処理は、それぞれ決められた短い時間ですぐに完了
- それ以外のコールは、すべて、プリエンプティブか制限時間以内に再スタート可能
- カーネルではミューテックスを不使用

リアルタイムパフォーマンス

INTEGRITYは必要な応答時間と実行時間を保証



測定条件: 233 MHz
PPC750 cPCI ボード

セマフォ

- INTEGRITYでは3種類のセマフォが利用可能
 - バイナリセマフォ
 - 優先度継承プロトコル
 - カウンティング(計数)セマフォ
 - Highest Locker セマフォ(優先度上限プロトコル)
 - 優先度逆転やチェーンブロッキングを避ける最善の方法
 - セマフォ生成時の優先度は、リソースを取り合うタスクの中での最高の優先度
 - タスクがセマフォを獲得すると、その優先度をセマフォの優先度に直ちに変更
 - Rate-Monotonic Analysis (RMA)を利用可能

その他の保護

- タスクごとのスタックサイズの決定
 - gstackユーティリティの利用
 - デバッガタスクウィンドとリソースアナライザの利用
- スタックオーバフローからの保護
 - MMUによるスタック保護、あるいはコンパイラが生成するチェック用関数呼び出しによる保護を利用
 - 必要なスタックサイズをリソースアナライザを使って決定
 - 最大利用量の記録
- アプリケーションの読み出し専用セクションを保護
- 各種保護によりエラーが早期に発見でき、開発期間も短縮可能
- タスクの監視と再スタート機能も利用可能

INTEGRITYでの高信頼度

- バグ無しのアプリを書くことは現実的には不可能
- アプリでの小さなエラーでもシステム全体の致命的な問題を引き起こす可能性のあるこれまでのRTOS
- プログラムのエラーがあっても十分な保護がサポートされているINTEGRITY
 - カーネル空間内部での問題からの保護
 - アプリ空間での問題からの保護
 - Denial of Service (DoS)からの保護

トータルソリューションとしての INTEGRITY

開発ツール
群

DO178B-A
認証可能

解析ツール
群

高可用性
(HA)

IPネットワーク
スタック

新しい
ネットワーキング

INTEGRITY

組み込み用
Webサーバ

マルチCPU
対応

グラフィックス

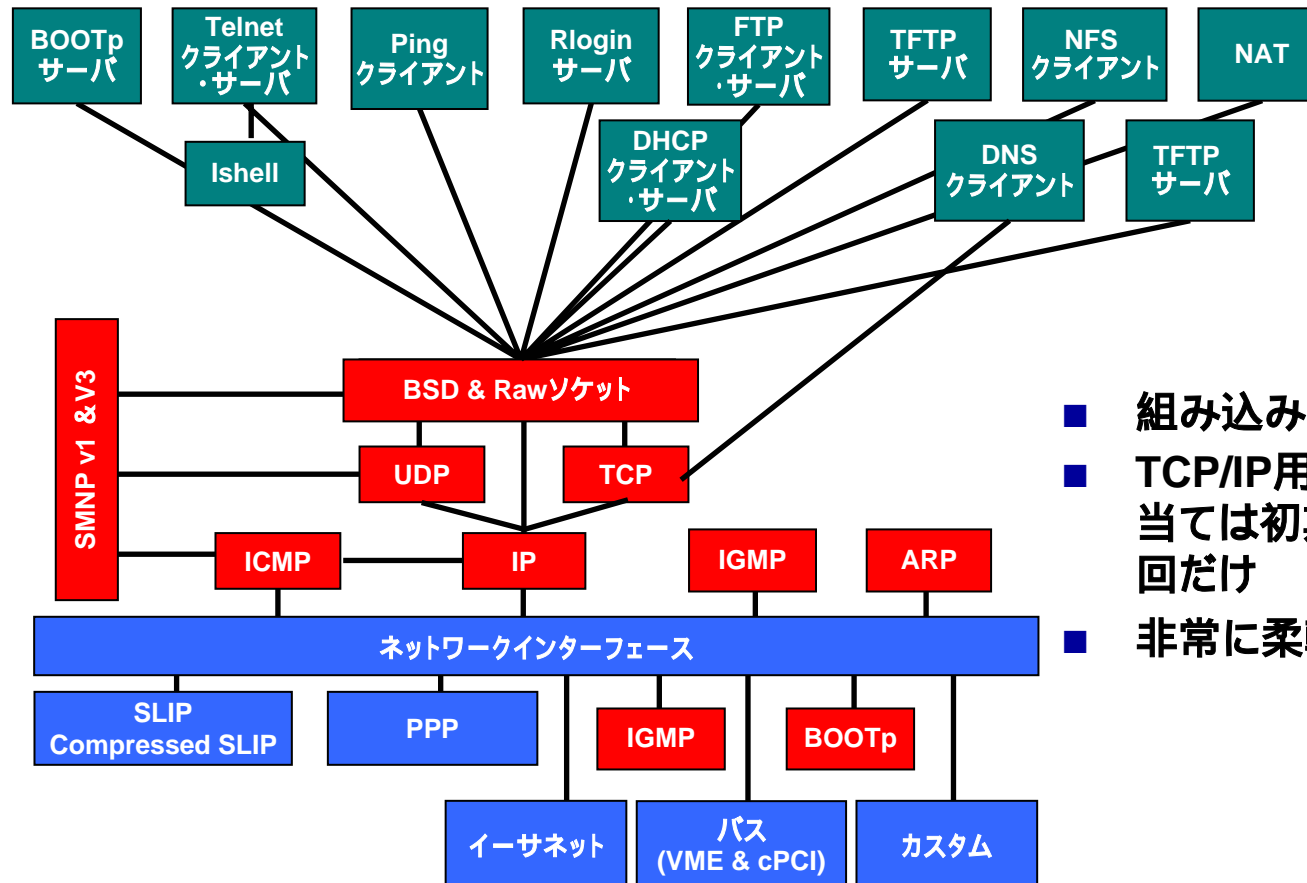
コンパイラ

VME, cPCI,
カスタムボード

分散処理



IPネットワークキング



- 組み込み向けに最適化
- TCP/IP用のスタック割当ては初期化の際の1回だけ
- 非常に柔軟な構成

グラフィックス

■ いくつかのソリューションを利用可能:

- PEG (Portable Embedded GUI)
- 標準OpenGLクライアント・サーバ
- 標準X11クライアント・サーバ
- 標準Motif
- 組み込み可能でDO178B認証可能な2D・3Dグラフィックス用OpenGL



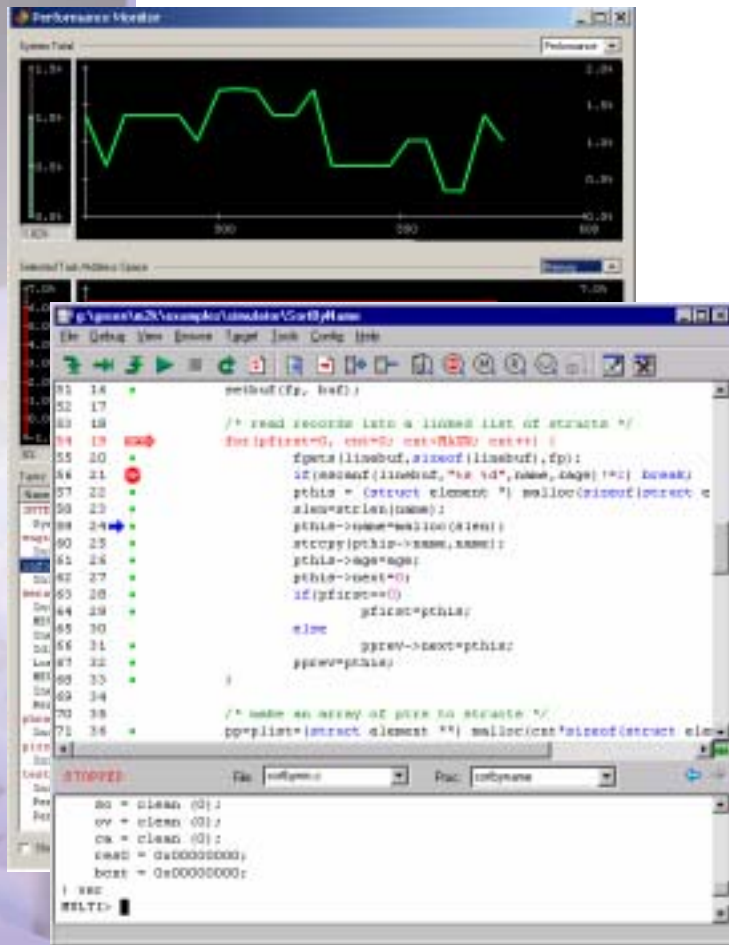


生産性向上を目指した 開発ツール

*先進のテクノロジーにより
プログラマの生産性を向上*



最良の開発環境



INTEGRITYと業界随一の開発環境を統合

MULTI開発ツール群

- C/C++, Ada, Fortranコンパイラ
- プロジェクトビルダ
- GUIベースのFlashメモリプログラミング
- INTEGRITYプロジェクトウィザード

MULTI解析ツール群

- グラフィカルなイベントアナライザ
- グラフィカルなリソースアナライザ
- システムとタスクレベルのプロファイリング
- スタック使用量表示

MULTIフィールドデバッグ

Green Hillsプローブでのデバッグ



Integrate GUIを使った 静的タスク生成

The screenshot displays the Integrate GUI interface. The main window, titled "D:\Integrity30\integrate.exe", shows a project workspace with a canvas containing a task graph. The graph includes an "Initial" task, a "sender1" task, and a "receiver1" task, all connected to a "message2_3" object. A red console window in the center shows the following output:

```
Object List (9 rsvd):  
10) Task: sender1  
11) Task: receiver1  
12) Connection: mess...  
13) Connection: mess...  
  
Object List (20 rsvd):
```

A "Configuration options for Task: receiver1" dialog box is open, showing the following settings:

- Name: receiver1
- Entry Point: receiver
- Priority: 1
- Weight: 1
- Stack Size: 0x1000
- Start at kernel startup
- TimeSlice: Maximum (non-preemptible) Minimum Other
- Seconds: 0
- Fraction: 1

The bottom of the main window shows a log window with the following text:

```
Target file 'default.bsp' not found, using de  
Clearing canvas  
Target file 'default.bsp' not found, using de  
Clearing canvas  
Task Initial added to Address Space message2_  
Task receiver1 added to Address Space message  
Task sender1 added to Address Space message2_
```

統合C/C++コンパイラ

- 1つのコンパイラで5種類の言語をサポート
 - C++ (ISO/IEC 14882:1998)
 - EC++ & ETC++
 - C99 (ISO/IEC 9899:1999 – 新ISO)
 - C89 (ISO/IEC 9899:1989 – 旧ISO)
 - K&R C
- 64ビットデータのサポート (C99での新機能の抜き出し)
- GNU C互換のための拡張
- *Green Hills*のコンパイラから生成されたコードはサイズ・実行時間の両方で業界トップクラス
 - 業界標準の第三者ベンチマーク機関であるEEMBCで公表



INTEGRITYプロジェクトウィザード

■ INTEGRITYプロジェクトウィザード

- 容易な条件選択:
 - ターゲットとするBSP
 - アドレス空間の数
 - 動的モジュールロードか単一イメージか
 - 必要モジュールの自動指定:
 - TCP/IP
 - コネクションマネージャ
 - リソースマネージャ
 - デバッグエージェント
 - 共有ライブラリかそれ以外か

■ INTEGRITYアプリの作成が数クリックで可能!



INTEGRITYシミュレータ

ISIM

- ハードウェアが利用可能になる前やハードウェアの数が少ないときにもデバッグが可能
- WindowとSolarisのどちらでも利用可能
- ターゲット向けのGHSツールをそのままISIM用にも利用
- ターゲットハードウェアにロードするアプリケーションイメージと完全に同じものをISIMにもロード
- イーサネット経由のマルチタスクデバッグ, 動的ダウンロード, TCP/IP, コネクションマネージャを利用できる完全なデバッグ環境
- ホスト上で動作する1つのMULTIで、すべてのINTEGRITYターゲット(ISIMでも実ハードでも)のデバッグが可能
- ソケットを使ったプロセス間の通信もサポート
- メモリ保護を含むすべてのAPIが利用可能
- 周辺カスタムデバイスのシミュレーションを可能にするAPI
- MULTIからもコマンドプロンプト/シェルからも起動可能



INTEGRITYリソースアナライザ



システム全体用ウィンドウ

選択したタスクまたは
アドレス空間用ウィンドウ

メモリとスタックの使用範囲

使用したメモリの%

使用したCPU時間の%

INTEGRITYカーネルで 管理される各種オブジェクトの表示

INTEGRITY Task Explorer (MULTI target ID 0x3c8)

Options

Address Space	Task	Connection	Activity	Semaphore	Memory Region	Link	Clock	I/O Device	Obj
Connection ID	Address Space	Other End	AS of Other End	Object Index					
0x752900	pizzahut	0x7fa940	phonecompany	pizzahut: 10	pl				
0x789900	information	0x7fa900	phonecompany	information: 10	pl				
0x7c1900	engineer	0x7fa980	phonecompany	engineer: 10	pl				
0x7fa900	phonecompany	0x789900							
0x7fa940	phonecompany	0x752900							
0x7fa980	phonecompany	0x7c1900							
0x159a80	kernel	0x159ac0							
0x159ac0	kernel	0x159a80							
0x159fc0	kernel	0x7fb040							
0x7fb040	kernel	0x159fc0							

Tasks for "C:\demo_i40\rtos\rtssrv -loaddir C:\tftpboot -..."

Options Group Target Tools Windows Help

All Pizza

Tid	Name	Stack	Status	Priority
0x71b000	Initial	0x01f0/0x1000	SysHalt	127
0x753000	Initial	0x0170/0x1000	SysHalt	127
			SysHalt	127
			Pended	127
			SysHalt	1
			SysHalt	254
			SysHalt	254

INTEGRITY Task Explorer (MULTI target ID 0x3c8)

Options

Address Space	Task	Connection	Activity	Semaphore	Memory Region	Link	Clock	I/O Device	Object
Semaphore ID	Object Index	Owner	Binary	Address Space					
0x159dc0	kernel: 29	N/A	No	kernel					
0x159e00	kernel: 30	N/A	No	kernel					
0x159e40	kernel: 31	N/A	No	kernel					
0x159e80	kernel: 32	N/A	No	kernel					

INTEGRITY Task Explorer (MULTI target ID 0x3c8)

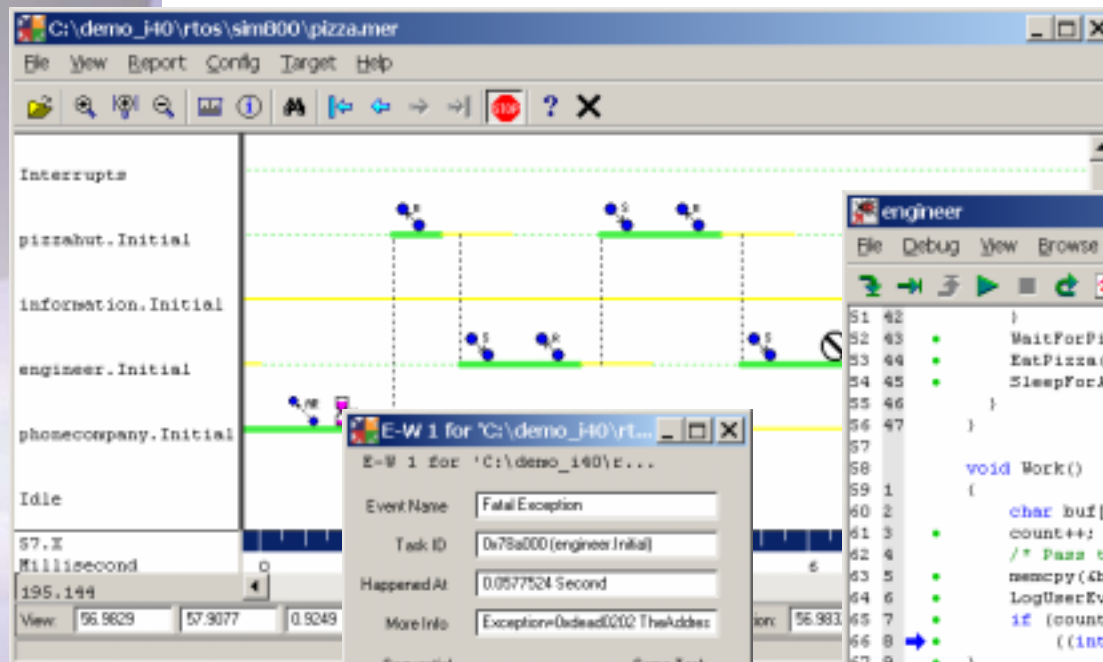
Options

DomainID	Name	Vital	Objects
0x7fa000	phonecompany	Yes	35
0x7c1000	engineer	Yes	33
0x789000	information	Yes	33
0x753000	pizzahut	Yes	33
0x159000	kernel	No	354

Name	TaskID	Status	Priority	Stack Start	Stack End	Stack HWM/Size	Object Index
Initial	0x15a000	NoProcess	255	0x1421e0	0x1461e0	0x3250/0x4000	kernel: 9
ResourceManager	0x7fe000	Pending	254	0xe2278	0xe2878	0x0208/0x0600	kernel: 16
NEVLogTask	0x7fe000	NoProcess	254	0xe3878	0xe4078	0x00a0/0x0900	kernel: 27
LoaderTask	0x7fd000	Pending	254	0xe4878	0xe5078	0x02d8/0x0800	kernel: 33

Number of Tasks in the reference list: 4

INTEGRITY イベントアナライザ



```
51 42     }
52 43     WaitForPizza();
53 44     EatPizza();
54 45     SleepForAWhile();
55 46     }
56 47     }
57
58     void Work()
59 1     {
60 2         char buf[24];
61 3         count++;
62 4         /* Pass the value of "count" as extra data for LogUserEvent */
63 5         memcpy(buf, &count, sizeof(count));
64 6         LogUserEvent(CurrentTask(), 0 /* subtype */, (Address) buf, 24);
65 7         if (count > 10)
66 8             ((int *)0x40000)[count] = 0;
67 9     }
68
```



Green Hills プローブ

- 高速ダウンロード: 400 Kbytes/sec以上
- ホスト: Windows, UNIX, Linux
- 接続: USB, RS-232, イーサネット
- 対応CPU: PowerPC, ARM, MIPS, その他
- 単一JTAGチェーンまたはBDMで複数コア向けのデバッグを実現
- ホストのWebブラウザで設定・診断などが可能



結論

- 新しいRTOSテクノロジーをGreen Hills Softwareがリード
- 2000年代に求められるビジネスモデルをGreen Hills Softwareが提供
- 「ロイヤリティ時代」の終焉
- 最強の組み合わせであるINTEGRITYとMULTIをGreen Hills Softwareが提供

組み込みソフトウェア開発者にとって
「正しい」パートナー

