

MKng プロジェクトにおけるマイクロカーネル移植技法：

3 Z - 6

SPARC 用リアルタイムマイクロカーネルの実装

船渡大地[†] 斉藤鉄也[†] 徳田英幸^{†‡}[†] 慶應義塾大学 政策・メディア研究科[‡] 慶應義塾大学 環境情報学部

1 はじめに

慶応大学を中心に行なわれている次世代マイクロカーネル (MKng) 研究プロジェクト [1] において、複数のプラットフォームへの Real-Time Mach (RT-Mach) マイクロカーネルの移植を行なっている。本稿では SPARC プラットフォームへの移植について述べ、その中でも特にカーネルのリアルタイム処理部、I/O ドライバ部、及び UNIX サーバである 4.4 BSD Lites サーバの移植に関して報告する。

2 プラットフォーム/開発環境

MKng プロジェクトではエンジニアリングタスクとして、先進的な RISC アーキテクチャ上への RT-Mach リアルタイムマイクロカーネルの移植をすすめている。複数のプラットフォーム上での動作を実現することで、OS プラットフォームの共通化を計り、同時にマイクロカーネルと計算機アーキテクチャとの整合性及び問題点の洗い出しを狙っている。

開発のターゲットの一つとして、Sun Microsystems 社の SPARC Station 20 (SS20) を用いた。SS20 では SPARC version 8 に準拠した Super SPARC チップが採用され、IU、FPU、MMU 及び、インストラクション/データの区別のない 64K バイトのキャッシュメモリにより構成される [4]。SPARC アーキテクチャへの Mach の実装は本プロジェクト以前にも株式会社リコーと慶應大学を中心に断続的に開発が行なわれてきたが [2]、本プロジェクトにより RT-Mach マイクロカーネル、フリーなサーバである 4.4 BSD Lites サーバ、各種 I/O ドライバの移植が行なわれた [3]。

開発環境として、SunOS4.1.4 上で GCC を使い ODE-MAKE と呼ばれる MAKE 環境を用いカーネル及びサーバのビルドを行なっている。ターゲットマシンと開発環境はシリアルラインと Ethernet によって接続され、ターゲットマシンのコンソールはシリアルラインへ、カーネルは tftpboot によって Ethernet 経由で開発環境から取得される。デバッグは Mach の提供する DDB 及び SPARC OBPRM の機能を利用することにより行なっている。

Daichi FUNATO¹, Tetsuya SAITO¹, Hideyuki TOKUDA^{1,2},
¹ Keio University Graduate school of Media and Governance,
² Keio University Faculty of Environmental Information, 5322
 Endo, Fujisawa, Kanagawa, 252 Japan
 Special thanks to Hiroyuki Kinbara from RICOH

3 リアルタイムスレッドの移植

RT-Mach のアーキテクチャに依存する部分には、主に割り込み処理部、メモリ管理部、コンテキスト切替え処理部、デバイスドライバ部がある。本プロジェクトではカーネルのリアルタイム処理についての研究に重点をおいている。以下ではリアルタイム処理部のアーキテクチャ依存部に関して述べる。

3.1 周期スレッドのアーキテクチャ依存部

RT-Mach における通常のスレッドのアーキテクチャ依存部は、コンテキストの初期化、保存、ロード、PCB の初期化、スレッドの起動、スレッド状態の取得、設定が挙げられる。これらに加え、リアルタイムスレッドのアーキテクチャ依存部は以下のものが挙げられる。

- 周期スレッドのリセット
- リアルタイムクロック処理

3.2 周期スレッドのリセット

図 1 は周期スレッドの実行の流れを示している。

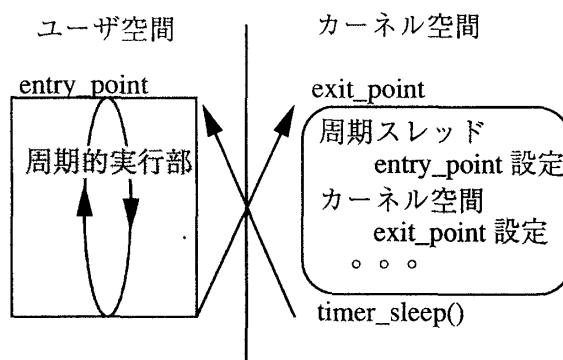


図 1: 周期スレッドの流れ

周期スレッドは周期的に起動されるため、ユーザ空間にある周期的実行部が終了する毎に、次にスケジューラされた時に関数の最初から実行を行なえるようにプログラムカウンタ及びカーネル空間内の終了ポイントを設定しなおす必要がある。その他にも引数の設定や、スタックポインタの設定も必要になる。SPARC アーキテクチャでは引数を渡すのにレジスタを使用しており、さらに RISC アーキテクチャのため、遅延命令が周期スレッドの実行に関わることを注意して実装しなくてはならない。

3.3 リアルタイムクロック処理

RT-Mach では、動的にクロックのレゾリューションを変更する操作が提供されているため、通常仮定される 10ms

に1回のクロック割り込みを動的に変更する仕組みを提供する必要がある。

SPARC アーキテクチャにはプロセッサ毎に Processor Counter Register(PCR) と呼ばれるクロックが存在し、500ns インターバル (2MHz) の精度をもつ。プロセッサ毎のクロックの他に System Counter Register(SCR) が存在し、任意の割り込みレベルでインタラプトがかけられる。

リアルタイムスレッドはクロックオブジェクトにバインドされて起動される。RT-Machは複数のクロックオブジェクトをサポートし、Realtime ClockとUpdate Clockに分けられる。PC-AT アーキテクチャ上ではclockpitと呼ばれるデバイスを Realtime Clock、およびUpdate Clockに使用している。今回の実装ではPCRをRealtime Clock及びUpdate Clockに使用した。

4 I/Oドライバ

SS20では標準のI/OモジュールとしてSEC(Sbus to Ebus Controller)[5]があり、EPROM, RTC(Real Time Counter), SCC(Serial Communication Controller)を持ち、システムの割り込みを管理している。SECのデバイス内、シリアルコンソールドライバ、RTCドライバを実装した。

MSI(Mbus to Sbus Interface)[6]では、SBus空間をMbus空間上にマップすることによりSBusモジュールをサポートしている。MSIではEthernetをサポートするLANCEとSCSIコントローラが用意されている。ここではEthernetとSCSIドライバを実装し、さらにSBusに接続するFore社のSBA-100, SBA-200ATMアダプタ用のデバイスドライバを作成した。

5 Litesサーバの移植

UNIXサーバはRT-Mach上のOSパーソナリティであり、サーバとエミュレータからなる。LitesサーバはUCBよりリリースされた4.4BSD-liteのコードを基にヘルシンキ工大で開発されたUNIXサーバである。エミュレータはRT-Mach上で実行されるユーザプログラムが実行するUNIXのシステムコールをLitesサーバへ転送する。エミュレータの機能は、UNIXのシステムコールのエミュレーションとRT-Mach上でプロセス間通信を行なうためのメッセージの作成である。

SPARCアーキテクチャ上にLitesサーバを移植するためには、異なるアーキテクチャそれぞれに適したデバイス名の指定や、ホストOSとのバイナリ互換性、移植で利用する開発ツールを利用するためのコマンド名の指定などを行なう必要がある。例えば、エミュレータはUNIXのシステムコールをエミュレーションするためアーキテクチャ依存の実装が必要になる。他にもMIG(Mach Interface Generator)は任意にダブルワード境界からワード境界へ

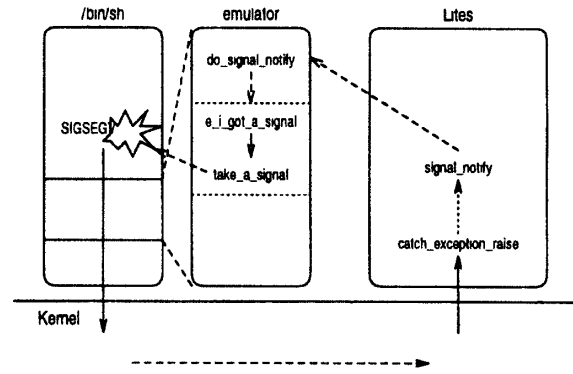


図 2: Litesサーバのシグナル配送

データ長を変更する。そのためダブルワードでアクセスをするSPARCアーキテクチャの命令を実行した場合はエラーが発生する原因となる。これに対応するためにワード境界からダブルワード境界に変更する必要がある。さらに、Litesではシグナル処理が未実装であったためその実装を行なった。図2はLitesサーバ上でのシグナル配送の流れと必要な機能を示す。

現在はLitesサーバが起動し、ネットワークが利用できることを確認している。今後は安定化を進め、UXサーバ、モノリシックカーネルとの性能評価を順次行なって行く予定である。

6 まとめと今後の方針

MKngプロジェクトのエンジニアリングタスクでは、複数のプラットフォームにリアルタイムマイクロカーネルの移植を進めている。本稿ではSPARCアーキテクチャに関して述べ、マシン依存部の少ないことからRT-Machのリアルタイム処理部、及びLitesサーバの移植性の高さが確認された。MKngプロジェクトではSPARCプラットフォームを高速ネットワークアーキテクチャ、及びマルチコンピュータアーキテクチャの基盤として実装を進めている。特に、ATMインターフェースを用いた帯域予約をサポートが可能なST2+の実験、さらに分散共有メモリによるマルチコンピュータの実験を計画している。

参考文献

- [1] 徳田 他: “MKng:次世代マイクロカーネル研究プロジェクトの概要”, 第55回情処全大論文集, (1997).
- [2] 金原 他: Mach3.0実装ボードにおけるシステムバスの測定情報処理学会 計算機アーキテクチャ研究会 94-ARC&OS
- [3] D. Funato. “Porting Real-Time Mach to SPARC Architecture” In *Proceedings of RT-Mach Workshop'96*, Pittsburg, August, 1996
- [4] Texas Instruments “SuperSPARC User’s Guide” *Open Systems Products*, 1993
- [5] LSI LOGIC “L64861 SBus to Ebus Controller (SEC)” *Technical Manual*, 1992
- [6] LSI LOGIC “L64862 Mbus to Sbus Interface (MSI)” *Technical Manual*, 1992