

Mach3.0マイクロカーネル上のマルチプルDOSサーバー

1 P-10

本田 良司

日本アイ・ビー・エム (株) 東京基礎研究所

1. はじめに

OSをモノシリック、すなわち1つのOSがすべての機能を担当するから、マイクロカーネルベースのOSサーバーへ作りかえていくことは、実用の点でモジュラリティの向上、生産性・保守性の向上という点で重要になってきている。ここでは、CMUで作られたMach 3.0マイクロカーネルをベースにして、DOSサーバーを実現した方法を述べる。このDOSサーバーは、マイクロカーネル上で単独で動作可能である一方、CMUで作られたBSD Unix¹⁾サーバーとも共存できる。

2. マイクロカーネルとその上のOSサーバー

Mach3.0マイクロカーネルは、CMUにおいて、Mach2.5 (4.3 BSD Unix と互換) をベースに、Unix依存部分を取り除き作られたもので、タスク、スレッド管理、メモリー管理、ポートとメッセージ機能そしてデバイス・ドライバー部分より構成される[1]。同時に、BSD 4.3 Unixサーバーが、CMUにおいて作られ、そのマイクロカーネル上で動作する。OSサーバーは、専用のSystem Programming Interface(SPI)を通してカーネル・タスクであるマイクロカーネルとIPC通信して、そのOSにおけるプロセス/スレッド、メモリー空間、デバイス・ドライバーをエミュレーションする。(図.1)

Mach3.0マイクロカーネルそのものは、Unix依存部分が全く無くなったとはいえず、BSD 4.3以外の種々のOSを乗せることは、Mach3.0をより、マイクロカーネルとして洗練させたものにするために重要である。Mach3.0の中には、まだ外部(ユーザーレベル)で実現できるものが残されており、実際にいろいろなOSサーバーを動作させて、新しく必要なSPIを追加し、またいらぬものを取り除いて、本当に必要なものだけをマイクロカーネルの中に残せば、SPIの種類とコードサイズも減っていくであろうと考えている。

今、現存するOSサーバーは、BSD Unixの他に、DOS, Macintosh²⁾ OSそしてOSF/1[2]がある。CMUで作られたDOSサーバーは、現在のところBSDサーバーの機能を利用しており、マイクロカーネルの上で単独で動作しない[3]。

3. マイクロカーネル上の単独DOSサーバー

筆者は、Mach3.0(MK58)マイクロカーネル上に単

¹⁾ Unix is a trademark of Unix System Laboratories, Inc.

²⁾ Macintosh is a trademark of Apple Computer, Inc.

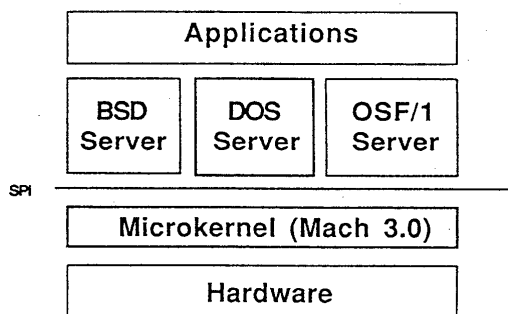


図.1

独に動作可能で、しかも複数DOSセッションを開くことが可能なDOSサーバーを実現した。これは、CMUのBSDサーバーとも並列に動く。動作するDOSのバージョンは、基本的にどのバージョンでも良く、DOS 3.x, DOS 4.0, DOS 5.0, DR DOS 6.0³⁾ が動くことが確かめられている。また、DOS/VなどのDBCS(Double Byte Character Set)を表示する日本語DOSも動作できる。現在、使用している機械は、PS/2⁴⁾ であるが、このDOS用にわずかに拡張されたマイクロカーネルをPC AT⁵⁾ 互換マシンなどにポートすれば、同様のことができる。

以下、実現方法の概要を述べる。

CMUのMach3.0(MK58)マイクロカーネルのPC ATリファレンス・ポートをベースに、PS/2へポートした後、DOSを動かすための拡張を行なった。

DOS及び、その上のDOSアプリケーションは、合わせてマイクロカーネルのシングル・スレッド、シングル・タスクとして実現され、i386/i486の仮想86モードで動く。

DOSタスクが必要としているメモリー空間は、マイクロカーネル・タスクのとは違い、ユーザー・レベルのアドレスが0番地から始まる必要があるため、タスク生成時にマイクロカーネル内部でメモリー空間の貼りなおしを行なっている。また、ROM BIOSのコードも、その時その仮想メモリー空間へマップしている。仮想86モードのイニシャル・コンテキストは、INT 19H(ブート・プログラム)BIOS機能の先頭アドレスから実行するように設定し、利用者の望むDOSがディスクケットまたはハードディスクからブートされるようにした。

DOSは、DOSカーネルとBIOS(基本ハードウェアの依存部分)からなる。従って、DOSをそのまま仮想86モードで動かし、BIOSをマイクロカーネル上でBIOSサーバーを作ってBIOSを全てエミュレーション

³⁾ DR DOS is a registered trade mark of Digital Research Inc.

⁴⁾ PS/2 is a registered trade mark of IBM Co.

⁵⁾ PC AT is a registered trade mark of IBM Co.

ンすれば良いように思うが、これだけではうまくいかない。DOSアプリケーションやデバイス・ドライバの中には、キーボードなどのハードウェア・インタラプト・ベクターなどを上書きして、アプリケーションのハードウェア・インタラプト・ルーチンが呼び出される必要があるものがあるからである。また、BIOSには、そのハードウェア特有のコードが含まれていて、そのまま実行したほうが良いものがある。

一方、DOSアプリケーションのなかには、DOS/BIOSコールをするだけでなく、直接ビデオ・ハードウェアやキーボード・ハードウェアなどをさわるものがあり、ハードウェア・レベルでのエミュレーションが不可欠なものがある。

これらのことから、BIOSのいくつかの機能を除いて、BIOSもなるだけそのまま(一旦、マイクロカーネル内部でエクセプションを拾うが)実行させるようにし、タイマー、キーボード、マウスそしてProgrammable Interrupt Controllerなどのハードウェア・レベルのエミュレーションをするようにした。

このハードウェアおよびBIOSのエミュレーション・ライブラリーは、マイクロカーネルの中に置くこともできるし、仮想86モード・タスクのシェアード・ライブラリーとしても置ける。現在のところは、ハードウェア及びROM BIOSのエミュレーションは、その機械(PS/2, PC-ATなど)に対するリファレンス・ポートと考え、マイクロカーネルの中に置いてある。

以下、各機能のエミュレーションの方法について簡潔に述べる。

(A) DOS /BIOS ソフトウェア・インタラプト

DOSアプリケーションは、INT命令を使用して、DOS/BIOSを呼び出す。このときソフトウェア・インタラプトが引き起こされ、マイクロカーネルのエクセプション・ハンドラーに制御が移る。そこで、仮想86モード・タスクであれば、エクセプションを引き起こした原因を調べて、然るべき機能のエミュレーション・ライブラリーを呼び出す。ここで、BIOSのオリジナル・コードを実行してもよいものは、そのまま実行させる。DOSの場合は、全てそのまま実行させた。エミュレーションしているBIOS機能は、現在のところ以下のものだけである。

- INT 11H (メモリー・サイズ照会)
- INT 13H (ファイル入出力)
- INT 14H (Com.ポート)
- INT 15H (EMS/ハイメモリー・ムーブ他)

(B) BIOS ハードウェア・インタラプト

前に述べたように、以下のハードウェア・インタラプトは、そのまま実行させた。

- INT 8H (タイマー・インタラプト)
- INT 9H (キーボード・インタラプト)

ハードウェア・レベルのエミュレーションをするのでマイクロカーネルの信頼性は、損なわれない。

(C) ハードウェア・レベルのエミュレーション

以下のハードウェアを IN/OUT 命令実行のエクセプションを使ってエミュレートする。

タイマー、キーボード、マウス、PIC、ビデオのハードウェアのエクセプションを拾うかは、

i386/i486の I/O Permission Bit MapをMachのシステム・コールである thread_set_state() を使っておこなっている。

(D) cli, sti, pushf, popf, retf, int 命令

DOSアプリケーションの完全性のためには、これらの命令をエミュレーションする必要がある。

4. マルチプル・サーバー間の資源の衝突の回避

DOSサーバーは他サーバーと共存できるように設計されている。複数のDOSサーバーを動かしたり、BSDサーバーと共存させれば、当然資源の衝突が起きる。基本的には、各ハードウェア毎にlockを取れば良い。しかし、ビデオ、ディスク/ディスクット、キーボードのようなハードウェア資源については、特別な考察が必要だった。

5. パフォーマンス

DOSサーバーとそのアプリケーションのパフォーマンスは、そのオリジナルに比較しても遜色ない。ファイル入出力に関しては、マイクロカーネル内のデバイス・ドライバを利用して分遅くなっている。フォアグラウンドのスクリーンにあるDOSアプリケーションがビデオバッファにアクセスしているときは、2.の(C)で述べたエクセプションを一時的に起きないようにして、グラフィックス描画速度を向上させている。2.の(D)で述べたエクセプションもグラフィックス描画中に起きないようにするとより速くなる(よりオリジナルに近くなる)ことが確認されたが、これはシステムの信頼性を損なうことになるのでしていない。

6. おわりに

互換性に関し、DOS/V上のLotus 1-2-3 R2.3J¹⁶ や SimCity¹⁷ などのゲーム・ソフトウェアが問題なく動作している。Windows 3.0¹⁸ もリアル・モードで動作可能である。

今回作成したDOSサーバーは、バイナリ互換を最大限に得られるように作られている。従って、このアプローチではオリジナルより速くなることは決してないが、エミュレーション・ライブラリーに機能を追加して、段階的にDOS/BIOSを置き換えて、マイクロカーネル上のマルチタスク/マルチスレッド・サーバーにしていくことが可能である。このDOSサーバーを作成して学んだことは、マイクロカーネルの外側で、ハードウェアを制御できることである。(もちろん、できないハードウェアもある。)

7. 参考文献

- [1] Microkernel Operating System Architecture and Mach, D.L.Black他, Journal of Information Processing.
- [2] Architecture and Benefits of a Multithreaded OSF/1 Server, OSF Research Institute Memorandum.
- [3] DOS as a mach 3.0 Application, Gerald Malan, 他, School of Computer Science, CMU.

¹⁶ Lotus 1-2-3 is a registered trademark of Lotus Development Co.

¹⁷ SimCity is a trademark of SimBusiness, Inc.

¹⁸ Windows 3.0 is a trademark of Microsoft Co.