

## OS/omicon 第4版におけるOSエミュレーション機構の設計と実現

1F-10

石坂淳、早川栄一、並木美太郎、高橋延匡  
(東京農工大学 工学部)

## 1. はじめに

パーソナルコンピュータの普及によるユーザ数の拡大に伴い、ユーザのコンピュータに対するスタイルが多様化している。ユーザごと、またそのユーザのコンピュータ利用局面においても必要とするOSインタフェースは異なる。1台のコンピュータに複数のOSを搭載しユーザが明示的にOSを切替え解決する方法もあるが、OSの切替えの煩雑さや高いコストを伴う。そのため、各ユーザの目的に応じて複数のOSインタフェースを提供できるマルチパーソナリティOSの必要性が認識されている。本稿では、当研究室で開発されているOS/omicon 第4版において他のOSをエミュレートする機構の設計と実現について述べる。

## 2. V4マイクロカーネルインタフェース

OSエミュレーション機構を実現する基盤であるV4マイクロカーネル（以下、V4MK）[1]の割込みの処理、タスク間通信について述べる。V4MKには、割込みやタスク間の通信にthrowという機能が用意されている。throwには現在、割込みのthrow（以下、i-throw）とタスク間のthrow（以下、t-throw）がある。t-throwは、呼び出された関数が別のタスクのコンテキストで実行される機能であり、i-throwは割込みを割込まれたタスクとは別のタスクで処理させる機能である。主にこれらの機能を用いてOSエミュレーション機構を実現する。

## 3. 対象OSとエミュレーション方法

対象OSはMS-DOSとし、MS-DOSのシステムコール、BIOSコール、ハードウェアをエミュレートする。

MS-DOS用アプリケーションプログラム（以下、AP）はV4MKの一つのタスクとして稼動するため、タイムシェアリングによって、CPUや周辺機器を共有でき、また、V4MKのマルチタスク環境下でAPを利用できる。

## 4. MS-DOSサーバの全体構成

MS-DOSサーバは大きく分けてシステムコール管理部と例外管理部からなり、V4MKの1サーバとしてプロテクトモードで動作する。APは仮想86モードで動作し、APでint、in・out、cli・stiなどの命令を実行しようとした場合、一般保護例外が発生する。その際、V4MKの例外処理部からMS-DOSサーバに各命令に応じた通知がなされ、処理が移る。

システムコール管理部、例外管理部について4.1、4.2に示す。また、その動作について図1、図2に示す。

## 4.1 システムコール管理部

int命令中でもシステムコールに関しては、先に述べたthrowを用いて実現する。APでソフトウェア割込みが発生するとi-throwにより、MS-DOSサーバのタスクで割込みの処理が行われる。システムコール管理部は、システムコールフック部（以下フック部）とシステムコール処理部からなる。

## (1) システムコールフック部

フック部はV4MKにフックしたい番号のint命令を登録する機能とシステムコール処理部の各ファンクションへのテーブルを持つ。

## (2) システムコール処理部

システムコール処理部は、割込み番号20h以降のシステムコールと同等の機能を持つ。

次にシステムコール管理部の動作について述べる。まず、フック部によってV4MKにフックしたい番号のint命令が登録される。登録されたint命令が実行されるとV4MKからMS-DOSサーバにi-throwがかかり、フック部はint命令の発生したタスクのコンテキストによりシステムコール処理部の各ファンクションへ処理を移す。そしてシステムコール処理部での処理結果がint命令の発生したタスクのコンテキストへ書き戻される。

## 4.2 例外管理部

V4MKはプロテクトモードで動作しているため、BIOSコールの番号がプロセッサ予約と重なっている。また、共有資源へのアクセスは保護の面からみて監視する必要がある。よって、V4MKの例外管理

部を一般保護例外が発生したタスクが仮想 86 モードである場合は MS-DOS サーバに通知されるように拡張することにした。例外管理部は、例外内容チェック部（以下、チェック部）、BIOS 処理部、in・out 命令処理部からなる。cli・sti 等のフラグ操作命令は実行可能にした場合、仮想 86 モードのタスクが割り込みを不許可にしたまま無限ループに入れるなど保護の面からみて問題があるので、例外内容チェック部で nop として処理することにする。

(1) 例外内容チェック部

現在の V4MK の例外処理部を一般保護例外が発生したタスクが仮想 86 モードで動作しているかどうか判別し、仮想 86 モードの場合はそのタスクのコンテキストを MS-DOS サーバの方へ通知し、MS-DOS サーバの方で例外処理を行うように拡張する。チェック部は BIOS コールか in・out 命令かを判別し、それぞれに処理を移す機能と BIOS 処理部の各ファンクションのテーブルを持つ。

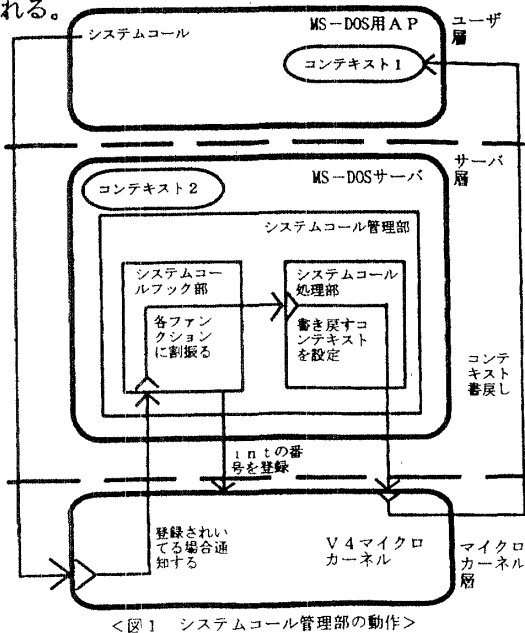
(2) BIOS 処理部

BIOS も MS-DOS のシステムコールと同じく int 命令によって呼び出されるが、システムコールとは異なり、V4MK や MS-DOS サーバの動作モードであるプロテクトモードのプロセッサ予約と割り込み番号が重なっているという問題があるので、MS-DOS システムコールとは別に処理部を用意する。

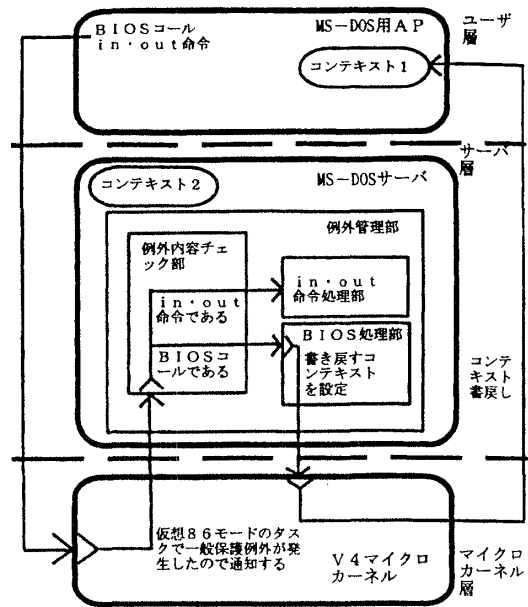
チェック部から BIOS 処理部へ処理が移った後はシステムコールと同様に処理結果を BIOS コールが発生したタスクのコンテキストに書き戻す。

(3) in・out 命令処理部

チェック部から in・out 命令処理部へ処理が移り、in・out 命令は仮想化されたハードウェアに対して実行される。



<図1 システムコール管理部の動作>



<図2. 例外管理部の動作>

5. OS エミュレーション機構の評価

設計を行った中のシステムコールの部分について実現し、評価を PC/AT 互換機 (Pentium 166MHz, メモリ 32M バイト) で int21h のファンクション 58h を 80 万回実行するのに要する時間の比較を行った。結果を表1に示す。

<表1. 実行時間>

MS-DOS	3秒
エミュレーション機構	63秒

表1からわかるように本機構の実行速度は約 20 分の 1 であった。V4MK による int 命令のフックから MS-DOS サーバへの通知と MS-DOS サーバから AP のコンテキストへの書き戻しの部分が速度低下の原因で低下分のほぼすべてを占めている。

6. おわりに

本稿では、OS/omicon 第 4 版において他の OS をエミュレートする機構の設計と実装について述べた。今後は BIOS やハードウェアのエミュレート部分の実現を行う予定である。

参考文献

[1] 森永, 他: "OS/omiconV4 のためのマイクロカーネル設計", 情報処理学会オペレーティングシステム研究会報告 OS-69-5, 東京農工大学 工学研究科, 1995