

実計算機モードと仮想計算機モード間の 動的切替え制御方式について†

田 口 敏 夫^{††} 堀 越 彌^{††} 栗 原 潤 一^{††}

仮想計算機システム (VMS: Virtual Machine System) を用いると、1台の実計算機のもとで複数個の OS を同時に走行させることができ、かつ、システム開発のための豊富なテスト機能が使えるので、システム開発に従事する者にとってきわめて魅力的なシステムである。すなわち、VMS を利用することによって、計算センタ・サービスとシステム開発作業とが並行して行えるわけである。

しかし、問題点としては、断続的に行われるシステム開発作業のために、実計算機システム (BMS: Bare Machine System) から VMS へ切り替える必要が生じ、このために30分以上のロス・タイムが生じることがあった。VMS から BMS へ移る逆の場合にも同一のロス・タイムが生じ、これらの時間が無視できないため、いわゆるオープン使用でシステム開発を行うのが従来の姿であった。

そこで、筆者らはこの改善策としてシステム開発作業が必要になったときに、計算センタ・サービスを停止することなく VMS の制御プログラムが忍び込み (クリープ・イン動作)、動的に VMS の環境を創り出し、システム開発作業の終了後は逆に BMS へ戻る (クリープ・アウト動作) 機能を実験的に開発した。

本機能は VOS 3 (Virtual-storage Operating System 3) を対象として、(1) VOS 3 下で走行する新たな制御プログラムと、(2) VMS との通信機能を設けて実現している。現在、BMS から VMS への切替えは 17 秒、逆への切替えは 2 秒で実現できている。

本機能によって、最近の 24 時間運転サービスとシステム開発作業のための時間確保という 2 つの相反する要求を満たすことが期待される。

1. ま え が き

大規模計算センタでは、システム開発や新システムのテストなどが常に必要であり、その都度、センタ・サービスを停止させねばならないのが実状である。これに対して、仮想計算機システム (VMS: Virtual Machine System^{1)~3)} のもとでは、

(1) 複数個の OS が同時に走行でき、

(2) システム開発のための豊富なテスト機能が使えるので、

これを用いることにより、システム開発のためにセンタ・サービスが中断するという問題は一応解決できる。

しかし、通常の用い方では、VMS の制御プログラム (VMM: Virtual Machine Monitor) が消費する CPU 時間、すなわち CPU オーバヘッドが 300%~500% (システムの性能が 1/4~1/6 に低下する) と大きく、計算センタ業務の処理能力に重大な影響を及ぼすことになる。このオーバヘッドは、最近一般化して

いる VMS の高速化機能⁴⁾ を使用しても 100% 程度までしか低減できず、計算センタで一般的に用いるにはいぜんとして壁がある。これに対して、筆者らは CPU オーバヘッドを 30% 以下に抑える方式を実験的に実現⁵⁾ して、これを 1 歩進めた。

しかし、VMS のオーバヘッドはいぜんとして無視できないので、システム開発やシステム・テストが必要なきにのみ VMS を利用し、システム開発作業が終了した後は、再び実計算機システム (BMS: Bare Machine System) のもとで計算センタ業務を処理したいという要求が生じることになる。この要求を満足する方式は、最近の 24 時間運転サービスの中で、システム開発作業のための時間を確保するという問題にも解決を与えてくれることになる。

そこで、筆者らは計算センタ・サービスを停止することなく、BMS の中に VMM が忍び込む (クリープ・イン動作) ことによって VMS の環境を創り出し、システム開発作業を行った後に再び BMS へ戻る (クリープ・アウト動作) 方式^{6), 9)} を検討し、その実験システムを開発した。

本稿では、現在までに開発した実験システムの方式と、得られた結果について報告する。なお、実験は、(1) HITAC M-180⁷⁾ 6 MB システムのもとで、

† A Method to Change Operating Modes between Bare Machine and Virtual Machine by TOSHIO TAGUCHI, HISASHI HORIKOSHI and JUNICHI KURIHARA (Central Research Laboratories, Hitachi, Ltd.).

†† (株)日立製作所中央研究所

(2) VOS 3[®] (Virtual-storage Operating System 3) を対象として行った。

2. 動作モードを切り替えるときの問題点

BMS の動作モードから、図1の下段に示すようなVMS の動作モードへ切り替えるには、通常、

- (1) バッチ・ジョブのイニシュエータを終了させ、
- (2) TSS, ネットワーク・ジョブなどの業務を終了させた後、
- (3) OS を終結 (シャットダウン) することになり、さらに、図1に示した、
- (4) VMS の VMM を IPL (Initial Program Loading) して、VMS の環境を創り出し、再び、
- (5) OS を IPL して、
- (6) TSS, ネットワーク・ジョブのサービスを開始し、
- (7) バッチ・ジョブのイニシュエータを再起動、

させなければならない。また、逆に VMS から図1の上段に示した BMS の動作モードへ戻するには、再び逆の手順が必要であり、これら一連の処理のために約30分程度のロス・タイムが生じる。なお、長時間のバッチ・ジョブが走行している場合には、動作モードの切替えに30分以上要することもある。

これに対して、筆者らが検討した方式は、上記の手順を踏むことなく計算センタ・サービスを続行させながら、30秒以内で動作モードの切替えを可能にするものである。

3. 動的切替え制御方式

この章では、動作モードを動的に切り替える制御方式を実現するための基本概念、および実現のために検討すべき事項と解決方法を明らかにする。

3.1 基本概念

計算センタ・サービスを停止することなく、BMS の動作モードから VMS の動作モードへ動的に切り替えるためには、CPU (Central Processing Unit), 主メモリ, 入出力装置などのハードウェア資源の制御を、瞬間的に VOS 3 から VMM へ移す必要がある。逆に、VMS から BMS へ戻る場合も同一のことが言える。これらの操作をハードウェア資源の切替え制御ということにする。

したがって、ハードウェア資源の切替え制御を円滑に行うためには、VOS 3 のサービスを瞬間的に凍結させ、その凍結時間内に VMS の制御プログラムである VMM が**忍び込み** (クリープ・イン: Creep-In), 図1に示した VMS の環境を創り出し、VMM がハードウェア資源を制御すれば良いことになる。逆に、VMS から BMS へ戻るときも VOS 3 のサービスが瞬間的に凍結している間に VMM が**抜け出す** (クリープ・アウト: Creep-Out) ことになる。そこで、筆者らは BMS の動作モードから VMS の動作モードへ移る機能をクリープ・イン VM 機能[®] (Creep-In VM Feature) とよび、逆の機能をクリープ・アウト VM 機能[®] (Creep-Out VM Feature) とよんでおり、両者を合せてクリープ・イン/アウト VM 機能ということにする。

したがって、上記機能を実現することは、OS (筆者らの例では VOS 3[®]) が動作するマシンの CPU 利用率が、図2に示すように変化することになる。すなわち、クリープ・イン動作、クリープ・アウト動作が開始されると VOS 3 のサービスは徐々に低下し、ある時点 (t_1, t_2) では VOS 3 のサービスが完全に凍結することになり、この凍結時間 (Freezing Time: T_{in}/T_{out}) 内に動作モードを切り替えるための処理がなされる。

以上をまとめると、動作モード

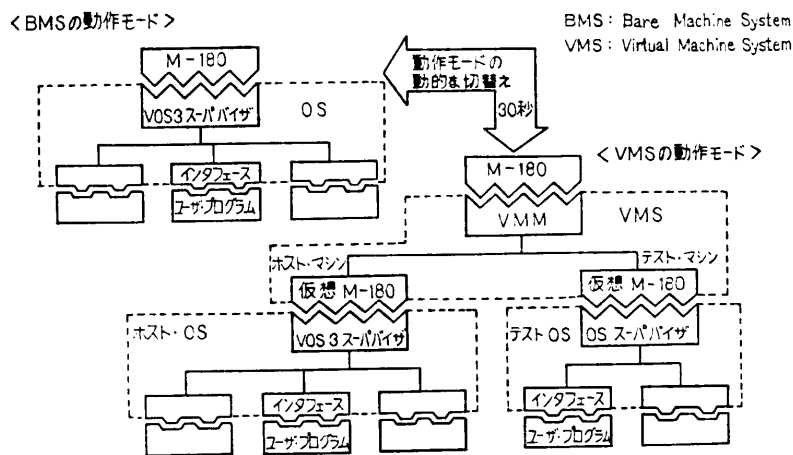


図1 実計算機システムと仮想計算機システムの関係

Fig. 1 Relationships of bare machine system and virtual machine system.

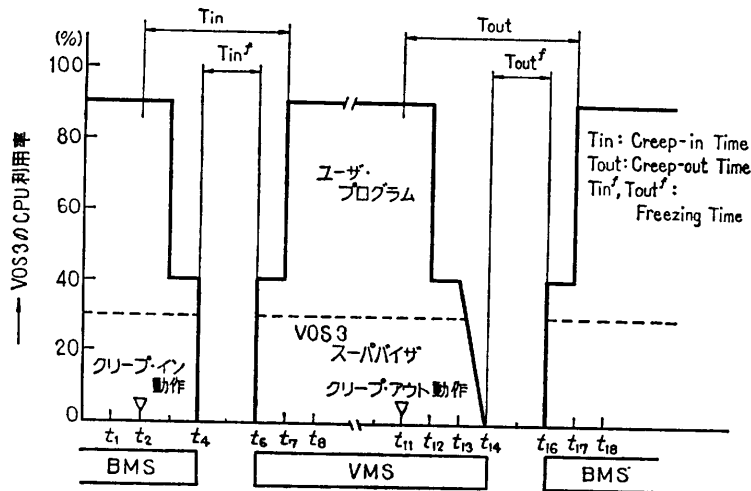


図2 動作モードの切替え時における CPU 利用率の変化
Fig. 2 CPU utilization during exchanging of operating mode.

を動的に切り替えるための機能、すなわちクリープ・イン/アウト VM 機能の実現上、次の3つの指標が定義できる。

(1) クリープ・イン・タイム (Creep-In Time: T_{in})

BMS の環境から VMS の環境へ移ることが指令されてから、VMS の環境が創り出されるまでの時間。

(2) クリープ・アウト・タイム (Creep-Out Time: T_{out})

(1)とは逆に、VMS から BMS の環境へ戻るまでの時間。

(3) 凍結時間 (Freezing Time: T_{in}' , T_{out}')

VOS 3 の動作が完全に停止する時間であり、TSS などの端末ユーザへの応答が中断する時間。

以上3つの指標のうち、(1)(2)は図2で示したように、VOS 3 がほかのサービスを並行して行える区間も存在するので、 T_{in} , T_{out} が30秒前後、 T_{in}' , T_{out}' は15秒前後を目標としてシステムを開発すれば、システムのユーザにはほとんどサービスが中断したようには見えないと考えられる。すなわち、 T_{in}' , T_{out}' は TSS 端末ユーザへの応答が完全に停止する時間であり、この時間が15秒以内であれば端末ユーザへの影響も少ないと考えた。

3.2 検討事項と解決方法

3.1 節で述べたように、クリープ・イン/アウト VM 機能を実現するためには VOS 3 サービスを瞬間的に凍結させる方式を検討しなければならない。これは、すなわち

(1) CPU サービスと、

(2) 入出力サービスの停止・再開方法を検討することになる。さらに、VMM が動作するために、

(3) 主メモリを確保する方法、

(4) VMM のローディング方法も検討する必要がある。(1)(2)はクリープ・イン動作時、クリープ・アウト動作時に必要なものであり、(3)(4)はクリープ・イン動作時に必要である。

表1は、検討すべき事項と解決方法を対比して示したものである。表1において、○印は採用したことを表わし、×印は VOS 3 の拡張が必要となる

ため不採用としたものである。表1より、クリープ・イン/アウト VM 機能を実現するためには、VOS 3 の制御下で走行する制御プログラムを設ければ良いことが分かる。この制御プログラムを CIOP (Creep-In/Out Program) ということにする。

この CIOP は、

- (1) VOS 3 の V=R 領域*で動作し、
- (2) 特権モードで走行する

制御プログラムとなり、VMS を創り出すための主メモリを確保し、VMM を主メモリ内にロードした後、クリープ・イン/アウト動作を制御することになる。

このように CIOP を設けたのは、本機能を実現する上で VOS 3 の拡張を一切行わずに済むようにしたためである。

以下に CIOP による VOS 3 サービスの凍結、すなわち CPU サービス、入出力サービスの停止・再開方法について述べる。

3.3 VOS 3 サービスの凍結方法

表2は、CPU サービスと入出力サービスの停止・再開方法を、クリープ・イン動作、クリープ・アウト動作について示したものである。

3.3.1 CPU サービス

CPU サービスの凍結は、表2に示したように CIOP が、VOS 3 の制御⁹⁾のもとに、

- (1) V=R 領域、
- (2) 特権モード、
- (3) 最高優先順位

* V=R 領域: 仮想アドレスと実アドレスとが1対1に対応している領域のこと。

表 1 検討事項と解決方法
Table 1 Problems vs solutions.

検討事項	説明	解決方法	効果・問題点等	評価
1	主メモリの確保と解放 VMS が動作するために最小限 1 MB が必要	(a) VOS 3 にスーパーバイザ・マクロを新設する.	OS に依存する.	×
		(b) VOS 3 のもとで V=R ジョブを走行させる.	OS の機能を利用	○
2	VMM モジュールのローディング VMM モジュールはファイル形式で格納されていない.	(a) VOS 3 の凍結時間内にロードする. (SIO 命令)	凍結時間の増加	×
		(b) VMM をファイルとみなして前処理中にロードする.		○
3	CPU サービスの停止・再開 クリープ・イン/アウト動作のための凍結時間を作り出す.	(a) 要求元以外のタスクのディスパッチを禁止するスーパーバイザ・マクロ	OS に依存する.	×
		(b) VOS 3 のもとで動作するプログラムが制御する.	特権モードで動作する.	○
4	入出力サービスの停止・再開 同 上	(a) 入出力サービスを禁止するスーパーバイザ・マクロ	OS に依存する.	×
		(b) クリープ・イン動作時には VOS 3 下で動作するプログラムが割込みを管理	VMS の環境下でシミュレートする.	○
		(c) VOS 3 と VMM との間で交信する. (クリープ・アウト)	入出力サービスの終了を待つ.	○

表 2 VOS 3 サービスの凍結方法
Table 2 Freezing methods of VOS 3 service.

項目	CPU サービス	入出力サービス
条件 クリープ・イン動作	1. 最高優先順位のプログラム 2. 外部割込み禁止で動作 3. V=R リージョンで動作 4. 特権モードで動作 5. VOS 3 の PSA を書き替える.	1. 入出力割込み処理アドレスを CIOP 内とする. 2. 割込み情報をスタックし、後に VMS のもとでシミュレートする.
クリープ・アウト動作	1. 最高優先順位のプログラム 2. 外部割込み禁止で動作 3. V=R リージョンで動作 4. 特権モードで動作	1. 動作中の入出力処理が完了するまで待つ. 2. 完了確認のために、VMM との交信機能を設ける. (ハイパバイザ・コール機能)

で動作し、CIOP に制御が移ったときに外部割込み⁷⁾ (External Interruption) を禁止状態とすることによって、CPU 資源を自由に制御できる。これは図 2 に示した Tin' 、 $Tout'$ の凍結時間を作り出すことになり、VOS 3 のスーパーバイザは一切関知しない状態である。

したがって、この Tin' 内に動作モードを BMS から VMS へ切り替えることになり、逆に $Tout'$ 内に戻るための処理を行うことになる。

CIOP は動作モードが切り替わると、外部割込みを許可状態とした後、VOS 3 へ制御を返すことによって、VOS 3 の CPU サービスが再開できる。

3.3.2 入出力サービス

入出力サービスの凍結方法は、CPU サービスの場合よりも複雑になる。すなわち、動作モードを瞬間的に切り替える前に、

- (1) 新たな入出力動作を実行させないこと、

- (2) 現在実行中のすべての入出力動作が完了すること、

を保証しなければならないためである。上記において、(1)は 3.3.1 項で述べた CPU サービスを凍結させることで保証できるが、(2)はクリープ・イン処理、クリープ・アウト処理で異なった方法を採用している。

(1) クリープ・イン処理

CIOP は PSA (Prefixed Storage Area)⁷⁾ 内の IONPSW (I/O Interruption New PSW)⁷⁾ の値を CIOP 内の入出力割込み処理アドレスに書き替えることによって、図 2 の t_4 時点以降の入出力割込みは CIOP が処理できることになる。したがって、CIOP は“割込み情報ブロック”を順次作成し、VMS の環境が創り出されてから VMM がその“割込み情報ブロック”に基づいて割込みのシミュレーション処理を

行えば良い。以上によって、入出力サービスの凍結が可能となる。

この処理方法については第4章で述べることにする。なお、“割込み情報ブロック”内に格納する情報は次のものである。

- (a) 割込みを起したデバイス・アドレス
- (b) CSW (Channel Status Words)⁷⁾
- (c) センス情報

上記のうち、(c)はCSWにエラー報告(ユニット・チェック: Unit Check)がなされているときに必要である。

(2) クリープ・アウト処理

CIOPはクリープ・イン処理のときのように、VOS3のPSAの書替えを行わずに入出力動作が完了するまで待つ方法を採用する。これは、(1)のクリープ・イン処理と同一の方法を採用したならば、動作モードがVMSからBMSへ戻ったときに、入出力割込みのシミュレーション処理をCIOPが行う必要が生じるためである。

したがって、CIOPはVOS3スーパーバイザを介さずに直接VMMに入出力装置の動作状態を調べてもらうようにしており、VOS3の改造を必要としない。

このインタフェース、すなわち交信機能はVMMのハイパバイザ・コール⁴⁾(HVC: Hypervisor Call, ダ

イアグノーズ命令⁷⁾を用いて、クリープ・アウト処理用のコードを新設してある。

VMMから“動作中”の応答があると、CIOPは1秒間待った後に再びHVCを発行する。CIOPが1秒間待っている期間はVOS3の入出力割込みマスクが許可状態になっており、VOS3の以下の処理のみ動作する。

- (a) 入出力割込みにともなう非同期ルーチンの処理。
- (b) 入出力スーパーバイザの待ち行列に登録されている入出力要求の処理。

したがって、CPU利用率は図2に示したように除除に低下してゆき、*t14*で完全に凍結することになり、クリープ・イン処理との違いが生じる。

4. システム構成と処理方式

第3章で述べた方式によって、BMSとVMS間で動作モードを動的に切り替えることができる。この章では、筆者らが開発した実験システムの主な処理方式について述べる。

4.1 システム構成

図3は、クリープ・イン/アウトVM機能のブロック図を表したものであり、従来のVMMを拡張した個所のみ示してある。図3より、本機能はアセンブ

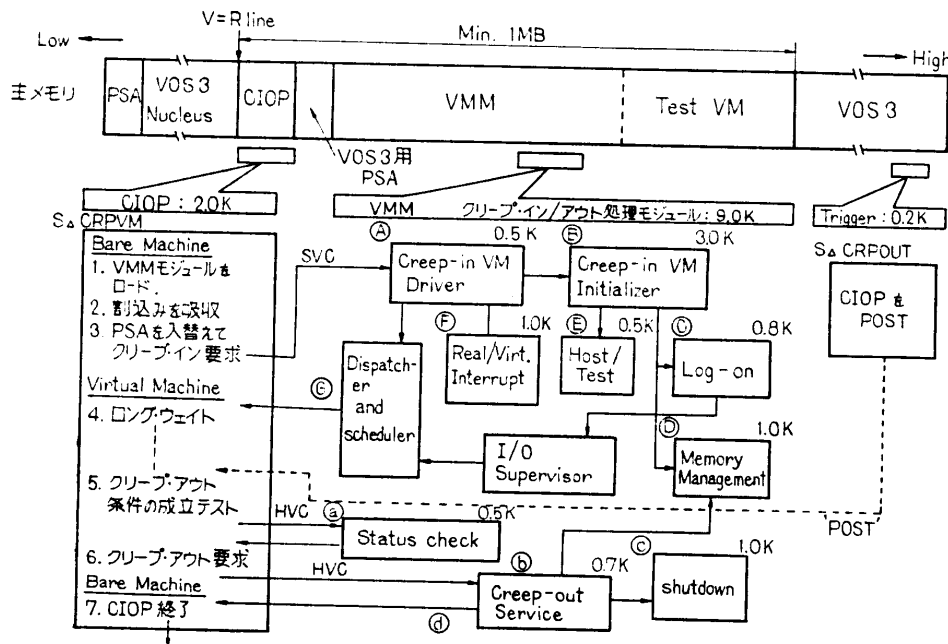


図3 クリープ・イン/アウトVM機能のブロック図
Fig. 3 Block diagram of Creep-In/Out VM feature.

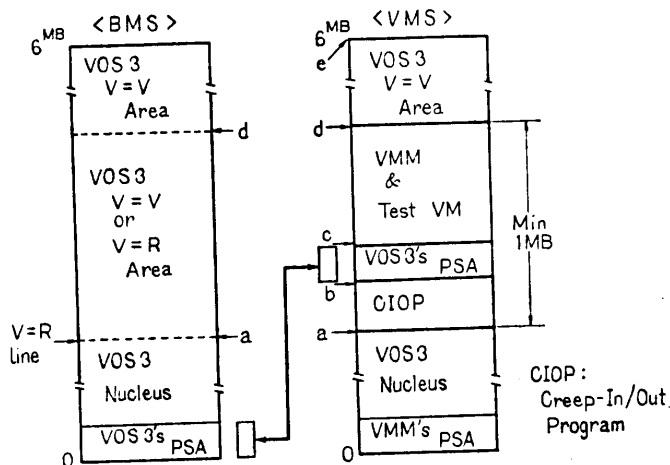


図4 主メモリの割当て方法
Fig. 4 Method of memory allocations.

ラ言語で約 11k ステップで実現できており、各々、

- (1) VOS 3 側の制御プログラムが 2.2k ステップ、
- (2) VMM 側の処理プログラムが 9.0k ステップである。

図3において、④～⑥がクリープ・イン処理に関する VMM のモジュール群であり、①～③がクリープ・アウト処理に関するモジュール群である。

4.2 主メモリの割当て方法

図4は、BMS と VMS とでの主メモリの割り当て方法を表わしたものであり、図の左側は BMS、右側は VMS の場合である。クリープ・イン動作のために CIOP が走行すると BMS のもとで a～d の領域が確保されることになる。この領域は最小限 1MB 必要であり、CIOP が消滅するとほかのジョブに割り当てられる。

クリープ・イン動作によって、動作モードが BMS から VMS へ切り替わったとき、VMM は図4の c～d で走行する。また、実の PSA は VMM が使用し、VOS 3 の PSA は b～c の領域に移される。

なお、VMM によってクリープ・インされた VOS 3 の主メモリ領域は、BMS のときと同一であり、この例では 6MB が与えられる。したがって、a～d の領域は VMM と VOS 3 間で共用することになるが、

- (1) VOS 3 は CIOP に割り当てたものと思っており、ほかのジョブに割り当ててことはない、
- (2) VMM はこの領域に対して VMM のページング領域をあらかじめ割り当てており、VOS 3 が誤って参照/更新しても VMS が破壊されないため、VMS は保護されていることになる。

VMS 下でのシステム開発用の仮想計算機 (テスト・マシン, Test-VM) は、VMM と同様に c～d の領域を使用する。

クリープ・アウト動作によって BMS へ戻ったときには、実の PSA を VOS 3 が再び使用する。

4.3 クリープ・イン処理

クリープ・イン動作の起動は、BMS のもとで CIOP を走行させることによってなされる。CIOP は図3に示したように、以下の処理を行う。

- (1) VMM モジュールを主メモリ内にロードする。
- (2) 外部割込みを禁止し、以後に発生した入出力割込みに対して“割込み情報ブロック”を作成する。

(3) VOS 3 用の PSA と VMM 用の PSA を入れ替えた後に、VMM にクリープ・イン要求のためのスーパーバイザ・コール (SVC: Supervisor Call) を発行する。

以上によって、図3に示したように制御が VMM へ移ることになる。

VMM のクリープ・イン処理モジュール群によって図5に示す一連の処理 (①～⑥) がなされ、VMS が割り出される。VMS が割り出されるとクリープ・イン前の VOS 3 用の仮想計算機はホスト・マシン⁵⁾となり、自動的に高性能化⁵⁾のサービスを受ける。

VMM は CIOP をディスパッチすると、CIOP は外部割込みを許可状態にした後、VMS を形成するために必要な主メモリを確保し続けるためにロング・ウェイト状態となる。これによって、VOS 3 は動作モードが BMS から VMS へ切り替わったことを全く知らされずに動作するのが特徴である。

なお、ハードウェア資源の切替え、すなわちレジスタ、タイマ、主メモリなどの仮想化処理は、図5のログ・オン処理中に行われる。

4.4 入出力割込みの吸収と反映処理

クリープ・イン処理を正常に遂行するためには、クリープ・イン動作のために VOS 3 のサービスが凍結されてから動作モードが VMS に切り替わるまでに発生する入出力割込みに、完全に追従できることである。すなわち、図2で示した Tin' 内に発生する入出力割込みを後にホスト・マシンへ反映できるように憶えておく手段と、後にシミュレートできる手段が必要であ

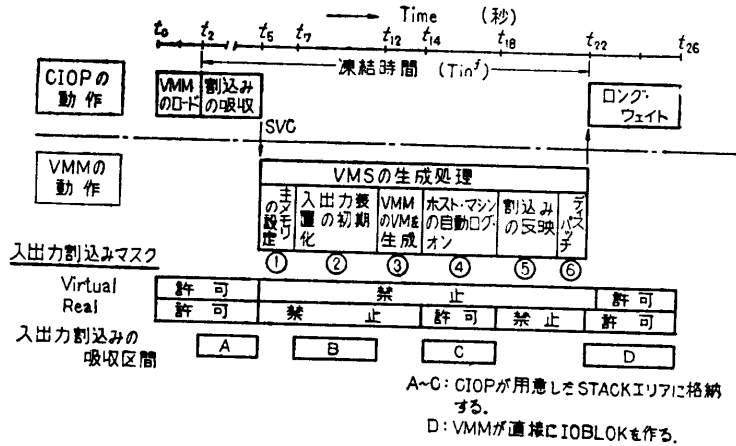


図5 クリープ・イン処理順序と割込みの吸収
Fig. 5 Procedure of Creep-In operations.

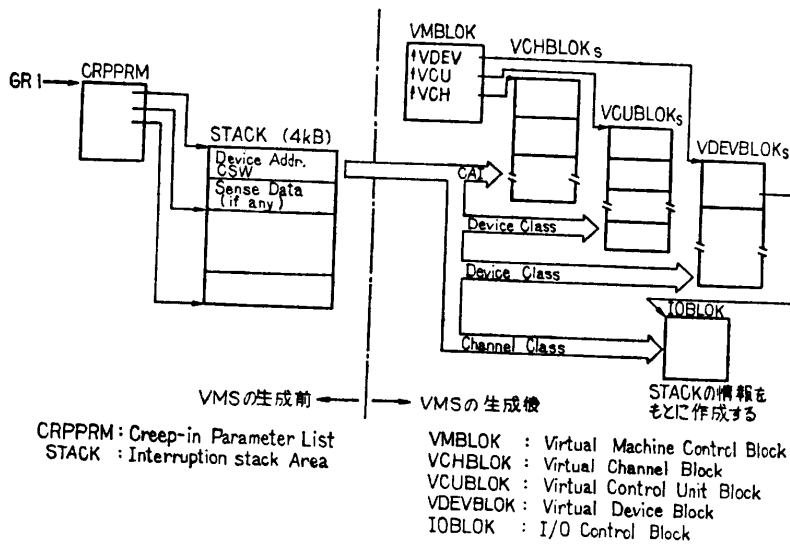


図6 入出力割込みの吸収と反映方法
Fig. 6 Stack and reflect mechanism of I/O Interruptions.

る。これを入出力割込みの吸収と反映処理という。
図5では、入出力割込みの吸収期間として以下の4つが示されている。

- (1) CIOP が吸収する期間 (A)
- (2) 入出力装置の初期化処理中 (B)
- (3) ホスト・マシンのログ・オン処理中 (C)
- (4) VMS が創り出されてから吸収 (D)

上記において、(4)はホスト・マシンが生成されているので通常の割込み処理がなされる。また、(1)はCIOP が VMM へ制御を移す前に行うものであり、人間の動作(端末オペレーション、カード・リーダーの起動など)をとまなわない入出力動作の完了を期待し

ている。(2)は入出力割込み禁止期間であるが、TIO(Test I/O 命令)を用いて割込み保留状態⁷⁾を吸収している。(3)は VMM の入出力スーパーバイザが吸収するものである。

なお、(1)~(3)はホスト・マシンが VMS 内にログ・オンしていないため、“割込み情報ブロック”は図6に示すような専用の領域内(これを STACK という)に格納される。

一方、割込みの反映処理は STACK 内の“割込み情報ブロック”を順次取出して、図6に示すように、ホスト・マシンのデバイス制御ブロック⁴⁾(VCHBLOK, VCUBLOK, VDEVBLOK)に反映することになる。

表3は、割込み情報の反映方法を表したものである。表3に示すように、入出力割込みを4種類に分類することによって、すべての割込みの反映処理が可能となる。

なお、デバイス制御ブロック内に反映された割込み情報は、後にホスト・マシンが入出力割込みを許可状態にしたとき、すなわち図5に示したようにCIOP がロング・ウェイト状態になったときに、VMMのディスパッチャによって

取り出され、割込み動作としてシミュレートされる。

4.5 クリープ・アウト処理

クリープ・アウト動作の起動は、VMMのコマンドによって行われる。コマンド形式は図7の通りであり、nn分後にクリープ・アウト動作を行うことを表す。

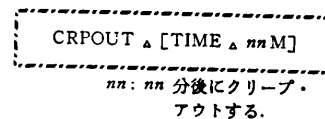


図7 クリープ・アウトコマンド形式
Fig. 7 Command format of Creep-Out operation.

表 3 割り込み情報の反映方法

Table 3 Method of reflection for I/O Interruptions.

項番	分類	CSW の 2 語目		処理方法	備考
		CSW の 1 語目 01.....31	(DSB) 01..... 7		
1	Channel Available Int. (CAI)	0—0	0—0	0—0	VCHBLOK に CAI ビットをオンにする。最初処理する。
2	Program Controlled Int. (PCI)	××—×	0—0	1×0—0	VDEVBLOK 内の CSW 格納エリアの PCI ビットをオン。チャンネル・エンド時にオアする。
3	Device Class Interruption (DCI)	0—0	××—×	0—0	VDEVBLOK の CSW エリアに、CSW を格納する。アテンションデバイス・エンドコントロール・ユニット・エンドなど
4	Channel Class Interruption (CCI)	××—M—×	××—×	××—×	VMS 下で発行された入出力動作の完了とみなして IOBLOK を作成する。

×: 0/1 であっても良いことを表わす。
M: 1 でなければならない。

このコマンドが投入されると、すべてのテスト・マシンに対して、
“VMS の環境が終了した。”
のメッセージを送出する。nm 分経過するとテスト・マシンを強制的にログ・オフさせ、ホスト・マシンに対して図 3 の右側に示した、

S ▲ CRPOUT
を生成する。これは、VOS 3 のオペレータ操作の簡略化を図ったものである。

この CRPOUT プログラムは VOS 3 のもとで実行され、先にクリープ・イン動作後ウェイト状態となっている CIOP を再起動するためのものである。したがって、この CRPOUT プログラムをトリガ・プログラムと呼んでおり、CIOP に対して仮想空間連絡機能⁸⁾ (Cross memory POST) を使用している。

再起動された CIOP は 3.3.2 項で述べた方法、すなわち VMM との通信手段によってクリープ・アウト動作の条件が成立すると、VMM にクリープ・アウト処理のための HVC (ハイパバイザ・コール: Hypervisor Call) を発行する。

図 8 は、クリープ・アウト処理の手順を示したものであり、図 8 の t₆~t₁₅ が凍結時間 (T_{out}) となる。

クリープ・アウト処理は、以下の事項を行うことによって完了する。

- (1) VMS 内の統計情報の収集

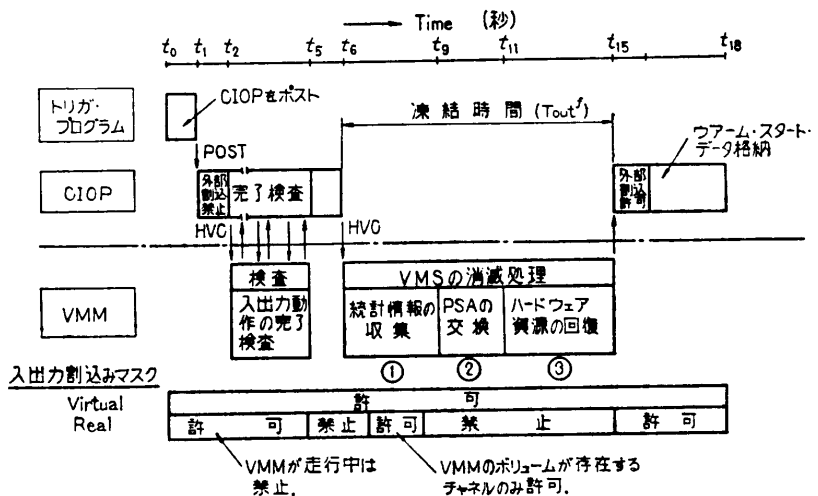


図 8 クリープ・アウト処理順序

Fig. 8 Procedure of Creep-Out operations.

- (2) VMM と VOS 3 間で PSA の交換 (図 4)
- (3) ハードウェア資源の回復
 - (a) レジスタ
 - (b) タイマ
 - (c) TLB (Translation Look-aside Buffer)⁷⁾

なお、VMS のウアーム・スタート・データ (Warm start data) は、BMS に戻ってから CIOP が格納している。これは図 8 に示したように、クリープ・アウト処理のための凍結時間を短縮させるためである。

動作モードが VMS から BMS へ切り替わると、CIOP が消滅することによって VMS が動作するための主メモリ領域が VOS 3 に返され、ほかのジョブのために使用されることになる。


```

0000 19.27.38 STC 18 $ CREWU-NBC ←-----CIOPの起動
0200 19.27.38 STC 18 *JSS300 CRPVM ON STCINRDR
4000 19.27.39 STC 18 *JSS463 CRPVM STARTED
4000 19.27.39 STC 18 JDJ275I CRPVM STARTED TIME=19.27.39
4000 19.27.39 STC 18 JDJ093I CRPVM ,CRPVM1 ,NBC ,WAITING FOR 02048K V=R STORAGE
8000 19.27.40 STC 18 *08 CREEP-IN READY, REPLY GO OR CANCEL
0000 19.27.46 R 08,GO ←-----クリープ・イン動作指令
8000 19.27.46 JDE033I REPLY 08 IS 'GO'
0000 19.27.46 STC 18 *09 *** WARNING *** VM CONSOLE POWER ON ??
0000 19.28.00 R 09,YES
8000 19.28.00 JDE033I REPLY 09 IS 'YES'
-----
0000 19.28.15 STC 18 VARY CO9,ONLINE
0000 19.28.15 JDE260I CO9 ONLINE
0000 19.28.15 STC 18 VARY CO9,CONSOLE,ROUT=NONE
0000 19.28.15 JDE354I CONSOLES
331 CONSOLE/ALT COND AUTH ID ROUTCB
331 CO9/CO1 A ALL 04 NONE
4000 19.28.15 STC 18 *** IN-LINE VMF ***
332 CREEP-IN VMF STARTED ... 19:28:00.132
332 INTERRUPTION STACK STARTED ... 19:28:02.183
332 INTERRUPTION STACK ENDED ... 19:28:07.698
332 VM/CP INITIALIZE STARTED ... 19:28:08.317
332 VIRTUAL MACHINE READY ... 19:28:15.191
332 USED CPU TIME (M-SEC) ... 00:00:00.000
-----
C000 21.25.11 STC 9 01 CH2 BUSY
C000 21.25.11 STC 9 02 CH3 BUSY 79,866,643
C000 21.25.11 STC 9 03 CH4 BUSY
C000 21.25.11 STC 9 04 CH5 BUSY 44,883,840
C000 21.25.11 STC 9 05 CH7 BUSY
C000 21.25.11 STC 9 06 CH8 BUSY
C000 21.25.11 STC 9 07 CHC BUSY
C000 21.25.11 STC 9 08 P.NIB 20,383,990
C000 21.25.11 STC 9 09 P.NITLB 3,720,032
C000 21.25.11 STC 9 10 NOT WAIT 861,904,633
C000 21.25.11 STC 9 11 EI 2,705,280,902
C000 21.25.11 STC 9 12 P.N-WAIT 795,237,792
C000 21.25.11 STC 9 13 NIB 26,072,774
C000 21.25.11 STC 9 14 NITLB 6,191,687
C000 21.25.11 STC 9 15 P.EI 2,558,215,197
4000 21.25.11 JOB 24 *JSS463 M0905604 STARTED - INIT 2 - CLASS C
0000 21.25.11 *JSS016 *DI7-9
0000 21.25.11 *JSS003 SPECIFIED LOCATION L= 2A UNAVAILABLE
4000 21.25.11 JOB 24 JDJ275I M0905604 STARTED TIME=21.25.11
4000 21.25.22 JOB 17 JDJ490I M0905503 ENDED TIME=21.25.22
4000 21.25.23 JOB 17 *JSS467 M0905503 ENDED
0000 21.25.23 JOB 23 *DI7-9,L=2A
4000 21.25.23 JOB 23 *JSS463 M0905504 STARTED - INIT 4 - CLASS C
0000 21.26.35 $ CRPOUT ←-----トリガプログラムのスタート
0200 21.26.36 STC 27 *JSS300 CRPOUT ON STCINRDR
4000 21.26.37 STC 27 *JSS463 CRPOUT STARTED
4000 21.26.37 STC 27 JDJ275I CRPOUT STARTED TIME=21.26.37
0000 21.26.37 STC 24 VARY CO9,OFFLINE
-----
0000 21.26.37 JDE261I CO9 OFFLINE
4000 21.26.39 STC 24 *** IN-LINE VMF ***
784 CREEP-OUT VMF STARTED ... 21:26:37.539
784 VMS SHUTDOWN STARTED ... 21:26:39.735
784 BARE MACHINE READY ... 21:26:39.809
8000 21.26.39 STC 24 *12 CREEP-IN READY, REPLY GO OR CANCEL
4000 21.26.40 STC 27 JDJ490I CRPOUT ENDED TIME=21.26.40
4000 21.26.40 STC 27 *JSS467 CRPOUT ENDED
0200 21.26.40 STC 27 *JSS350 CRPOUT IS PURGED

```

性能データ
 Tin -----15.0秒
 Tinf -----13.0

VMSでの動作

性能データ
 Tout -----2.9秒
 Toutf -----0.1

- 条件
1. Hardware -----M-180 6MB
 2. OS -----VOS3
 3. Batch Initiator数-----6
 4. TSS端末数(アクティブ)----20

図9 クリープ・イン/アウト VM 機能の実験例
 Fig. 9 Examples of Creep-In/Out VM feature.

5. 実験結果

筆者らは、BMS と VMS 間で動作モードを動的に切り替える方式、すなわちクリープ・イン/アウト VM

機能の実験システムを開発した。この章では、本機能の実験例について述べる。

図9はクリープ・イン/アウト VM 機能の動作例を表したものである。この例では ①～⑧ の期間が

VMS の動作となっており、このときの VOS 3 の負荷状況は次の通りである。

- (1) バッチ・ジョブのイニシエータ数：6本
- (2) アクティブな TSS の端末数：20台

図9より、以下のことが分かる。

(1) CIOP が処理を開始してからクリープ・イン動作が完了するまでの時間、すなわち、クリープ・イン・タイム (Tin) は 17 秒であり、目標値の 30 秒以下を達成している。

(2) クリープ・イン動作における VOS 3 の凍結時間 (Tin') は、目標値の 15 秒に対して 14 秒であり一応達成しているが、このうち CIOP が割込みを吸収する時間に 5 秒間要している。今後、この時間を 1~2 秒間とすることによって、凍結時間をさらに短縮させることも検討している。

(3) クリープ・アウト動作に関しては、

- (a) クリープ・アウト・タイム (Tout) が 2 秒、
- (b) VOS 3 の凍結時間 (Tout') が 0.2 秒

となっており、目標値を十分に満たしている。

6. む す び

以上、計算センタ業務を停止することなくシステム開発作業が必要なときに、動作モードを実計算機システム (BMS: Bare Machine System) から仮想計算機システム (VMS: Virtual Machine System)^{1)~3)} へ動的に切り替える機能—クリープ・イン VM 機能、逆に VMS から BMS へ動的に戻れる機能—クリープ・アウト VM 機能について、その実現方式と実験システムの結果を述べた。

本方式は VOS 3⁸⁾ を対象にして実現しており、

(1) VOS 3 のサービスを瞬間的に凍結させ、

(2) その凍結時間内に VMS の制御プログラムが 忍び込んだり (Creep-In)、抜け出す (Creep-Out) ものである。主な特徴は、

(3) 動的切替え制御のために VOS 3 下で走行する制御プログラム (CIOP: Creep-In/Out Program) を設けたこと、

(4) VOS 3 の拡張が一切不要であること、

(5) CIOP と VMM 間の交信機能を設けたことにある。さらに、実験システムの性能は、

(6) クリープ・イン・タイムが 17 秒、クリープ・アウト・タイムが 2 秒であり、

(7) VOS 3 の凍結時間は各々 14 秒、0.2 秒で実現できている。

本機能は TSS を含む一般のバッチ・センタの用途に対しては十分効果を発揮できるものであり、現在、当社中央研究所の計算センタにおいて試験的な使用を開始している。

最後に、本研究の遂行にあたって、開発の機会を与えていただいた(株)日立製作所中央研究所提副所長、ならびにご協力いただいた日立製作所ソフトウェア工場野口主任技師、井上主任、および実験の便宜を図っていただいた中央研究所桑原センタ長、本林主任研究員に感謝いたします。また、システム開発研究所の久保主任研究員、中央研究所の長島主任研究員、および吉住研究員の直接ご指導に対して、ここに感謝の意を表します。

参 考 文 献

- 1) Goldberg, R. P.: Survey of Virtual Machine Research, COMPUTER, pp. 34-45 (Jun. 1974).
- 2) 日立製作所: VMS 解説, 日立マニュアル, 8040-3-001.
- 3) IBM 社: VM/370 Introduction, IBM 社マニュアル, GC-20-1800.
- 4) IBM 社: VM/370 System Programmer's Guide, IBM 社マニュアル, GC-20-1807.
- 5) 田口, 堀越, 栗原: 仮想計算機システムの制御効率を向上するための方式と実験結果, 情報処理学会論文誌, Vol. 20, No. 4 (1979-7).
- 6) 田口, 堀越, 栗原: 実計算機モードと仮想計算機モード間の動的切替え制御方式について, 情報処理学会第 20 回全国大会, 論文集 (1979-7).
- 7) 日立製作所: M 170/M 180 処理装置解説, 日立マニュアル, 8080-2-001.
- 8) 日立製作所: VOS 3 概説, 日立マニュアル, 8090-3-001.
- 9) 栗原, 田口, 堀越: 実計算機モードと仮想計算機モード間の動的切替え制御方式について—実計算機モードへ戻る方式, 情報処理学会第 20 回全国大会, 論文集 (1979-7).

(昭和 54 年 8 月 1 日受付)

(昭和 55 年 11 月 20 日採録)