

VM/4 (ACOS-4 仮想計算機) のアーキテクチャ 難波信治*, 大野直哉*, 久保秀士*, 森末秀雄**, 磯秋義**, 大嶋隆** (* 日本電気 C&C システム研究所, ** 同 コンピュータ 技術本部, *** 同 基本ソフトウェア開発本部)

1 はじめに

最近、仮想計算機システムが広く実用化され、有効に使用されている⁴⁾。仮想計算機がサポートされていると、一つの実システム上で複数の OS と並列に実行させることが可能なため、OS の開発、システム開発、テストと通常業務の同時実行、システム移行などの場面で非常に強力なツールとなりうる⁵⁾。

ACOS-4 マシンは、OS の核機能を大幅にハード化/ファーム化したアーキテクチャをもっている^{2),3)}。VM/4 は、ACOS-4 のベアマシン (BM) の上に並列実行可能な複数の仮想 ACOS-4 マシン (VM) を実現するシステムである。

VM においては一様にオーバヘッドが問題となるが、当システムは特に高性能を狙って開発した。統合型の構成をとることにより、複数の仮想計算機と並列に通常の業務処理を VM オーバヘッドなしに実計算機上で直接に実行せしめる。また、仮想計算機においてもファームウェアの改変・追加、ACOS-450 の特別なハードウェアサポートの利用等により、オーバヘッドを極めて低くおさえている¹⁾。

ACOS-4 アーキテクチャの仮想化は、既存の仮想計算機の場合⁵⁾と異なり、ソフトウェア機能のみでは原理的に実現不可能である。たとえば、ACOS-4 では特権命令の実行のトラップは実行中のプロセスの属性に依存してなされる。OS プロセスの属性の変更は OS の改変を意味する。また、プロセスディスパッチがファームウェアで実現されているので、効率的な仮想プロセスディスパッチを行なわせるにはファームウェアの改変・追加が、言いかえると ACOS-4 アーキテクチャを拡張した VM/4 アーキテクチャが必要とされる。

我々は性能も考慮して VM/4 アーキテクチャの仕様と確立し、実際に ACOS-450 のファームウェアを改変してこのサブセットを実現し、またその上で動く VM 制御プログラムも開発して第一段階の VM/4 システムを完成させた。以下ではアーキテクチャの概要と 450 上に実現された VM/4 ファームウェアハードウェアサポートについて述べている。

2 ACOS-4 のアーキテクチャ

VM/4 が仮想化しようとする ACOS-4 のアーキテクチャ²⁾は OS の核機能を大幅に含んでいる。具体的には次のような特徴をもっている。

- プロセスの概念のサポート。具体的には、状態管理、ディスパッチング、セマフォによる同期機構など。
- セグメンテーションによるプロセスごとの多重仮想記憶空間。
- マルチオーソリタキによるプロテクション機構
- スタックを用いたコール・リターン機構
- 入出力チャンネルのスケジューリングの自動化
- 高水準言語指向の命令体系

さらに、ACOS-450 を含む上位システムのアーキテクチャは、次の集で拡張されている³⁾。

- ページングの導入

- ・密結合マルチプロセッサのサポート
- ・チャネルDAT (IOCによるアドレス変換) のサポート

以下では、これらのうち特にVM/4の実現と深い関りをもつアーキテクチャ上の特徴について簡単に説明する。特徴命令については既に述べた。

a ハードウェアとソフトウェアの接点

一般のアーキテクチャでは、ハードからは、管理対象としては各時点で実行中のタスクしか見えず、タスク切換えにはソフトが介入する。ACOS-4では、ハードが全管理対象を把握していて、プロセス切換えはハードが行う(ファームウェア化プロセスディスプレイ)。最大255個のジョブステップとそれらを構成する各々最大255個のプロセスサブ(ジョブ)番号、P(プロセス)番号により管理されている。プロセス間の同期はPVオペレーションによるためこれも必然的にハードでサポートしている。非タスクという概念は不要であり、すべてソフトウェアはプロセスとして実行される。

b アドレス空間

上記のプロセスごとに、するべきジョブステップ当り最大255個のアドレス空間をもつ。各アドレス空間は、約240 MBの広さをもつ。セグメントとこれを細分したページにより管理されている。セグメントは論理的なモジュールに対応し、プログラムの単位となり、サイズ自動的に伸縮可能である。アドレス空間の数と広さが膨大であるため、仮想計算機の実現に当っていわゆるシャドープージング方式をとることは現実的でない。

c 入出力処理

チャネルプログラム実行のスケジュール制御はCPUと独立に制御されるIOCにより行われる。チャネルプログラム内の仮想アドレスの実アドレスへの変換もIOCが行う。入出力の終了は、開始時に指定されていたセマフォに対してハードウェアからVオペレーションをかけることにより、知らされる。ソフトウェアはすべてプロセスなので強制的割り込みはありえない。

3 VM/4のアーキテクチャ

本節では、OS開発、システム移行のツールを目的とするVM/4のアーキテクチャについて、特徴、基本概念、仮想化の方式、制御方式、制御メカニズムの順で説明する。

3.1 特徴

本アーキテクチャの特徴は以下の通りである。

a 統合型仮想計算機システムを実現し得る。

システム移行の際、日常業務プログラムをBMEのOSのもとで実行できるので、これについては仮想化のオールハンドがかなりできるようになる。

b マルチプロセッサのVMを実現し得る。

VM上のOS上に加えることなくマルチプロセッサのシステムと実行できる。

c 高速なVMを実現するためにHW/FWで強力なサポートを行っている。

- ・VM上のアドレス変換とHW/FWで行う。

→同時に複数の常駐VMが実現できる。

- ・VMのチャネルプログラムとそのまわりの実行するVMでIOがある。

- ・直接実行できる特徴命令はシミュレーションせずに直接実行してしまふ。

3.2 基本概念

a 仮想計算機と仮想プロセッサ

VMを構成する各プロセッサと仮想プロセッサと呼ぶ。仮想プロセッサは、後述のVPディスパッチャによって実プロセッサにディスパッチされ、VMのCPU動作を実行する。ディスパッチの対象をVMではなく仮想プロセッサとすることにより、マルチプロセッサのVMを実現することができる。

各VMは、VM番号によって識別され、仮想プロセッサはそれぞれが属するVMのVM番号とVM内でのプロセッサ番号（VP番号）との組で識別される。VMにはそれぞれディスパッチングプライオリティが与えられ、そのVMに属する仮想プロセッサは、この値に従ってディスパッチされる。またタイムスライス制御を受ける。

b 特権プロセス、一般プロセス及びVM。

以上のプロセスをあるプライオリティで分け、そのプライオリティより低いプロセスと一般プロセス、それ以外を特権プロセスと呼ぶ。VM/4では一般プロセス全体をVM番号0のVM（VM₀と記す）として扱い、一般プロセスはVM₀の仮想プロセッサにローカルインデックスされて実行される。

3.3 仮想化の方式

a 主記憶の仮想化とアドレス変換

VMの主記憶は、BMの一つのセグメントに対応する。このセグメントを仮想主記憶セグメントと呼ぶ。VMの主記憶上の絶対アドレスは、この仮想主記憶セグメントのアドレス変換表によって、VM上の論理アドレスは、VM内のアドレス変換表と仮想主記憶セグメントのアドレス変換表とによって、それぞれHW/FWが動的に実主記憶上の絶対アドレスに変換する。

b シミュレーション事象とシミュレーションによる仮想化

仮想プロセッサ（VM₀以外）実行中にIO起動命令などその仮想プロセッサだけでは処理できない事象が発生すると、ソフトウェアによってシミュレーションを行う。この事象をシミュレーション事象と呼ぶ。シミュレーション事象が発生すると、HW/FWによってその内容を示すメッセージがBM上のセマフォを介してVM/4制御プログラムに通知される。個々の仮想プロセッサに関する事象はその仮想プロセッサのVPセマフォに、仮想プロセッサの属するVMに関する事象はそのVMのVMセマフォに通知される。このメッセージを受けとったVM/4制御プログラムはメッセージを解析して必要なシミュレーションを行う。ただし、命令のシミュレーション事象は、シミュレーションを必要とする特権命令だけに限って起す。

c チャンネルプログラムのシミュレーションとVMモードIO

VMモードIOは、VM番号、VM上でのチャンネルプログラムのアドレス、実行させる出力装置のチャンネル番号及び終了事象を受けとるBM上のセマフォのアドレスを指定してVMモードIO起動命令（VM/4で新設）で実行される。これによりIOCは指定されたVM上のアドレスからチャンネルプログラムを取り出し、チャンネルコマンドのアドレスをBMの実主記憶上のアドレスに変換しながら指定されたチャンネル上でチャンネルプログラムを実行させる。チャンネルプログラムが終了すると、指定されたセマフォに終了事象メッセージを通知する。こうすることによってアドレス変換のオーバーヘッドなしにチャンネルプログラムのシミュレーションを行うことができる。

d 割り込みのシミュレーション

ソフトウェアでシミュレーションしてVMに通知する割り込みは、

- i) 入出力割り込み
- ii) 命令のシミュレーション中に検出した例外

である。VMへの入出力割り込みは、ソフトウェア割り込みメッセージと作成して、通知先VM番号、通知先セクタのVM上のアドレス、割り込みメッセージのVM上のアドレスを指定した割り込み通知命令(VM/4で新設)で通知する。

命令のシミュレーション中に例外を検出するとソフトウェアは次のようにして通知する。すなわち例外メッセージと作成し、仮想プロセッサ制御ブロックVPCB(後述)に記録、VPCBの例外フラグをONにセットして仮想プロセッサをスタートする。これによりHWP/FWPは、仮想プロセッサのロールイン後直ちに仮想プロセッサ上のプロセスに割り込む。

e タイマの仮想化

VM上のタイマには、TOD(Time of Day)、インターバルタイマ、プロセスタイマ、タイムスライスタイマがある。後の二つは仮想プロセッサに閉じたタイマなので、これらのタイマ命令及びランアラートは直接仮想プロセッサ上で行う。TOD、インターバルタイマは、ソフトウェアでシミュレーションして仮想化する。

3.4 制御方式

a 制御構造

VM/4を制御するために、一つのセグメントを用意する。このセグメントをVM/4制御セグメントと呼び、その上にはVM/4制御構造体VM4CS、VM表、プロセッサ要求リンクPRLが存在する。VM4CS上には、VM表へのポインタ、前述の特権プロセスと一般プロセスを分けるプライオリティの値、プロセッサ待ち行列PRQへのポインタ等がある。VM表には、VM番号順にVM記述子VMDが並ぶ。

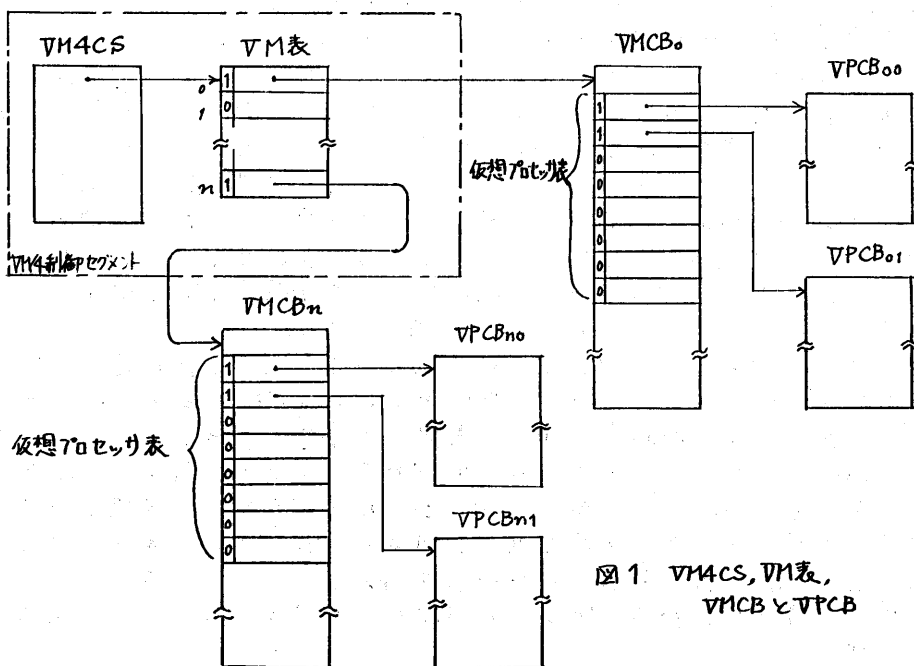


図1 VM4CS, VM表, VMCBとVPCB

であり、各VMはVM0を含めてそのVM番号をもつVMが存在するか否かのフラグと存在するならばそのVMのVM制御ブロックVMCBへのポインタから成っている。PRLは仮想プロセッサのプロセッサ要求を保持するものでPRQを作る。

VM制御ブロックVMCBは、VM0を含めた個々のVMを制御するためのもので、VMのプライオリティ、VMセマフォのBM上のアドレス、仮想主記憶セグメントのセグメント名、仮想主記憶のVM上での上限の絶対アドレス、仮想プロセッサ表等からなっている。仮想プロセッサ表は、VP番号順に並んだ仮想プロセッサ記述子VPDからなる。つまり、各VPDはその仮想プロセッサが存在するか否かのフラグと存在するならばその仮想プロセッサを制御する仮想プロセッサ制御ブロックVPCBへのポインタからなっている。

VPCBは、VPセマフォのBM上でのアドレス、仮想プロセッサのタイムスライス制御のためのタイムスライス値、汎用レジスタなどのコンラクト等からなっている。図1にVMACS、VM表、VMCB、VPCBの関係を示す。

b 制御の流れ

VM/4の機能は、VM/4制御セグメントに各種値を設定し、必要ならVM0のVMCB VPCBを作成し、VM/4活性化命令を実行することによって活性化される。7個のVMが消滅した時点でVM/4不活性化命令を実行すると不活性化される。

VMはその動作環境をとり、VMCB、VPCBを用意し、VM番号を指定してVM登録命令を実行することによってH/W/FDに登録される。VMを終了させるには、VMをサスペンドしたあとVMの登録抹消の命令を実行する。

VM及び仮想プロセッサは、それぞれスタート命令、サスペンド命令によってスタート、サスペンドされる。

VMをスタートするとH/W/FDは、そのVM内のすべての実行可能な仮想プロセッサのプロセッサ要求とそのVMのプライオリティに従ってPRQにつなぐ。仮想プロセッサをスタートすると、VMがスタートされているならば、H/W/FDによってその仮想プロセッサのプロセッサ要求がVMのプライオリティに従ってPRQにつなわれる。またアイドル状態の仮想プロセッサ上で実行可能なプロセッサが発生すると、VMがスタートされているならば後述するH/W/FDで実現されるVP活性化機構によって、その仮想プロセッサのプロセッサ要求が同様にPRQにつなわれる。PRQにつなわれたプロセッサ要求はH/W/FDで実現するVPディスパッチャによって取り出され、要求元仮想プロセッサが実行プロセッサにディスパッチされる。

ディスパッチされた仮想プロセッサは、実行を開始する。この時、例外が登録されているならば、前述のように取り出すが発生する。また、仮想プロセッサ上にプロセッサがある場合には左に、ない場合にはプロセッサディスパッチャの実行を開始されプロセッサローリングしてから、命令の実行に入る。

仮想プロセッサ実行中に与えられたタイムスライスを消費すると、H/W/FDで実現されるVPタイムスライス機構によりタイムスライス処理を受ける。シミュレーション事象が発生するとシミュレーション機構によってVMセマフォまたはVPセマフォにシミュレーション事象メッセージが通知される。

仮想プロセッサ上で実行するプロセッサがなくなるとH/W/FDで実現されるVP不活性化機構で仮想プロセッサを不活性化しVPディスパッチャにより、7次に実行する仮想プロセッサにディスパッチする。

VMのカスタントは、HW/FWからそのVM内の実行中の仮想プロセスとVPCBに
ロールアウトし、さらにVM内仮想プロセスのプロセス要求がPRQ上にあ
ればそれを取り除くことにより実行される。

3.5 制御メカニズム

a VP活性化機構, VP不活性化機構

VP活性化機構は、仮想プロセスがアイドルの時に実行可能なプロセスが
発生した場合に、その仮想プロセスを実行可能な状態にし、VMカスタート
されているならば、その仮想プロセスのプロセス要求をPRQにつなぐ。

VP不活性化機構は、仮想プロセス上で実行可能なプロセスがなくなった
場合にその仮想プロセスをアイドル状態にする。

b VPディスパッチ

VPディスパッチは、特権プロセス、VMの仮想プロセス、その他のVMの仮
想プロセスを実プロセスにディスパッチするディスパッチである。特権プ
ロセスと他の仮想プロセスよりも優先的に実行するために、VPディスパ
ッチは以上の実行可能なプロセスの待ち行列RBYQを調べる。VPディスパッチ実行
中の実プロセス上で実行可能な特権プロセスがある場合、実プロセスが
空き状態または仮想プロセスがあれば無条件に、特権プロセスがあればみ
つかった特権プロセスのプライオリティが高い場合に限って、みつかった特
権プロセスにディスパッチする。特権プロセスがRBYQ上にみつからない場合
には、その実プロセス上に特権プロセスがある場合に限ってPRQを調べる。
PRQ上にその実プロセスと要求しているプロセス要求がある場合、実プ
ロセスが空きがあれば無条件に、仮想プロセスがあれば、要求元仮想プ
ロセスの属するVMのプライオリティの方が高い場合に限って要求元仮想プ
ロセスにディスパッチする。新しく特権プロセスまたは仮想プロセスを
ディスパッチする場合に実プロセス上にプロセスまたは仮想プロセスが
あれば、これをロールアウトしプライオリティに従ってRBYQまたはPRQに
もどる。

c VPタイムスライス機構

仮想プロセス実行中に与えられたタイムスライスを消費すると、その仮
想プロセスをVPCBにロールアウトし、VPCBにあるタイムスライスカウン
トを1減算する。零になればVPセマフォにVPタイムスライス事象メッセ
ージと通知する。零でなければその仮想プロセスのプロセス要求をPRQにもど
す。この時同一のプライオリティをもつプロセス要求がある場合にはその
うちの最後尾に、なければそのプライオリティよりも低いプライオリティをも
つ最初のプロセス要求の直前につなぐ。その後、VPディスパッチに制御を
わたす。

d シミュレーション機構

シミュレーション事象が発生すると、シミュレーション機構が動作する。事象を解
析し通知メッセージを作成する。事象がVMまたは仮想プロセスと同
期して処理されるべきものなら、VMまたは仮想プロセスをサスペンドする。
次いでVMセマフォまたはVPセマフォにメッセージと通知した後VPディスパッチ
に入る。

4 ACOS-450 におけるインプリメンテーション

3で述べたアーキテクチャをもつVM/4をACOS-450上にインプリメントした。ここでは、ハードウェアでサポートしているのはアドレス変換だけであり他の部分はファームウェアによって実現した。以下に、アドレス変換のサポート、VMモードIO、ファームウェアによるサポートの順で説明する。

4.1 アドレス変換のハードウェアとVMモードIO

アドレス変換のハードウェアの概略を図2に、実主記憶のVMへの割り付け例を図3に示す。仮想主記憶セグメントは、その先頭を256KB境界に一致させ、さらに256KBの倍数として実主記憶に連続的に常駐させる。VMを登録する命令でVMCBから仮想主記憶のセグメント名を取り出し、リロケーション用RAMであるAREMの登録VM番号に対応するエントリの初期設定を行う。VMの絶対アドレスADのビット8-13とVM番号VM#とでAREMを引くと、VM上の256KBブロックに対応する実主記憶上のブロック番号が得られ、このブロック番号とブロック内アドレスとで実主記憶の絶対アドレスを得ることができる。PMの絶対アドレスはAREMによる変換を受けることはない。また、他のVMの主記憶のアクセスを防ぐためにMS2というRAMで境界チェックを行い、2113。VMの論理アドレスLAは、VM内プロセス番号JPとTLBでのVM識別子IDとでTLBによつてVM上の絶対アドレスADに変換されAREMによつてリロケーションされ実主記憶の絶対アドレスに変換される。VM#, IDは仮想プロセッサのロールイン時にセットされる。TLBでヒットしなかった時はTLBマネージャがVM上の絶対アドレスを転送してTLBにアドレス変換情報とセットする。

以上が仮想プロセッサ動作中のアドレス変換であるが、次にVMモードIO時のアドレス変換について述べる。VMモードIO時のアドレス変換はIOCによつて行われる。IOCはチャネルを動作しているVM番号、VM識別子をVM#, IDとしチャネルコマンドまたは

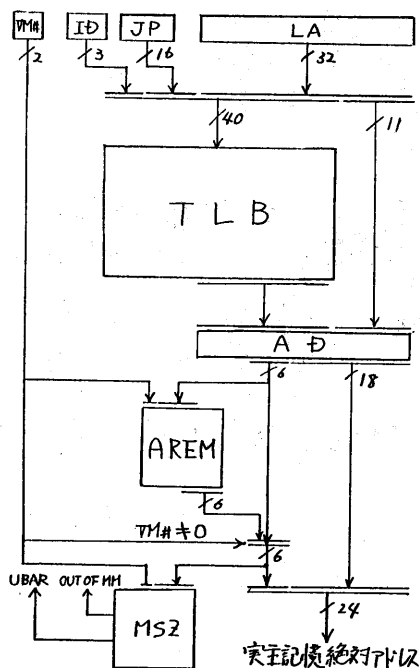


図2 アドレス変換ハードウェア

はデータバッファ等のアドレスとJP, LAとしてアドレス変換を行う。TLBは論理アドレスモードならば、前述と同様にして、絶対アドレスモード(JP=FFFF)ならばLAをそのままADにセットすることによつて、VM上の絶対アドレスをADに得る。次いでAREMにより実主記憶の絶対アドレスに変換する。この絶対アドレスをチャネルコマンドと取り出す。データバッファのアドレスを変換した場合には、IOCはその値とチャネルにあたる。チャネルは与えられた実主記憶の絶対アドレスを便つてデータ転送を行う。

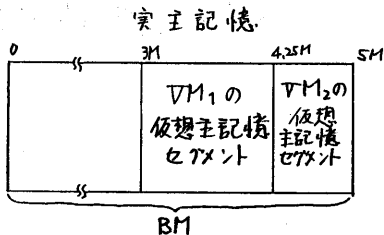


図3 実主記憶の割り付け例

本インプリメンテーションにおいては、ハードウェアとしてはAREM, MS2の追加と実主記憶の絶対アドレスを取り出すマイ

クロ命令の追加です。ファームウェアとしては、IOCのアドレス変換部で教ス
トップ; チャンネルプログラムの起動部の教ストップの変更及び約100ストップのVMモ
ID起動命令の新設です。また、TLBマネージャにはまる、たくちと入れずにす
んだ。

4.2 ファームウェアによるサポート

VPディスプレイ、VP活性化機構、VP不活性化機構及びシミュレーション機構はファ
ームウェアで実現した。特にシミュレーション命令の検出は特権命令例外ではなく、シミュ
レーションすべき命令に改造を加えてシミュレーション機構に入るようにした。よって
直接実行可能な特権命令は、仮想プロセッサ上で直接実行させている。なお現時
点ではサタインスライズ機構はインプリメントしなかったため、VPディスプレイ
で特権プロセスにディスプレイする時点で仮想プロセッサのホスティングを行
っている。

5 おわりに

OSの核機能が大幅にハード/ファームウェア化されているACOS-4マシンについて実
現された仮想計算機システムVM/4について、そのアーキテクチャの概要と実際に実
現されたVM/4のファームウェア、ハードウェアサポートについて述べる。VM/4は以下の
特徴をもつと考へられる。

- ・ファームウェア化されたOS核機能を含むアーキテクチャの仮想化の実現
- ・統合型であり、複数のVMと並列にBM上でも処理を行う(VMo)
- ・CPU, IOCにおけるアドレス変換等の強力なハードウェアサポート
- ・自由な組み合せで、BM, VM双方のマルチプロセッサ構成の考慮
- ・オーバヘッドが小さい

VM/4はOS開発用のツールとして使用され、威力を発揮している。

終りに、VM/4開発の機会を与えて頂き、また激励、尽力を頂いたC&Cシス
テム研嶺崎部長、基ソ開本辻部長、コン技本坂東課長に感謝します。

[参考文献]

- 1) 大野, 他「VM/4制御プログラムの構成と性能について」, 当研究会資料
- 2) 木野, 白井「コンピュータACOSシリーズ77の概要」, 日経エレクトロニ
クス, no. 107, 82-135(1975. 5. 5)
- 3) 日本電気「システム450/550システム概説書」, DAZ50-2, 56.8.
- 4) 桑原「機能強化が著しい仮想計算機システム」, 日経エレクトロニクス,
no. 238, 71-99(1980. 5. 12)
- 5) R.P. Goldberg "Survey of Virtual Machine Research", Computer, 7, 6,
34-45, 1974