ＣＯＢＯＬの拡張アドレスサポート方式

3D-5

三宅　立記　　花田　良平　（株）日立ソフトウェアエンジニアリング
今城　哲二　　（株）日立製作所　ソフトウェア工場

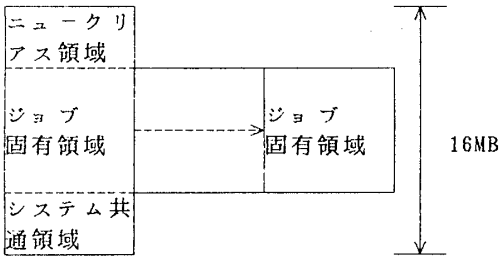
１．ＣＯＢＯＬの拡張アドレスサポートの背景

高度技術分野のめざましい進歩と、コンピュータを使ったユーザ業務の拡大から、大型汎用コンピュータのより一層の大規模化、高速化が急速に進められている。日立の汎用コンピュータは、従来２４ビットアドレシング（最大１６ＭＢまでアドレシング可能）であり、最上位ＯＳのＶＯＳ３は図－１に示すような仮想記憶構造を有していた。しかし、以下に示す２つの理由から１６ＭＢを超えるアドレシングを可能とする拡張が必要になった。

(1)　ユーザ業務の拡大により、ユーザプログラムの実行を行うジョブ固有領域が、不足してきた。

(2)　ＯＳの拡張に伴い、ニュークリアス領域、システム共通領域が増加の一途をたどっている。

以上の要求から、日立では、３１ビットアドレシング（２ＧＢまでアドレシング可能）の拡張アドレシング機構を開発すると共に、２ＧＢの仮想記憶空間をもつＯＳ（ＶＯＳ３／ＥＳ１）を開発した。これに伴って、ユーザプログラムで２ＧＢの仮想記憶空間を有効利用するため、ニーズの高いＦＯＲＴＲＡＮで拡張アドレスをサポートし、引き続き最大のユーザ数を持つＣＯＢＯＬでも拡張アドレスをサポートした。



図－１　従来のＶＯＳ３の仮想記憶構造

本論文では、ユーザプログラムの拡張アドレスへの移行を重点に置いた、ＣＯＢＯＬの拡張アドレスサポートの方針、仕様、方式について述べ、その効果をまとめる。

２．拡張アドレスサポートの方針

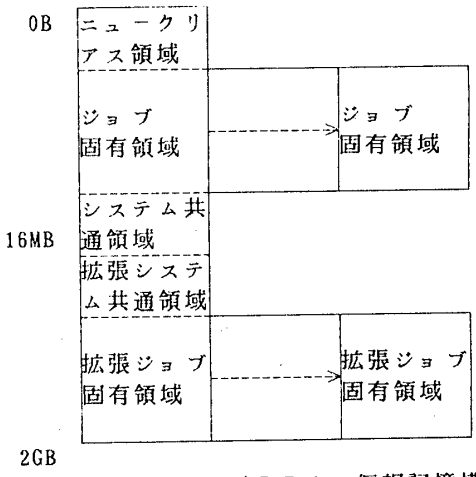
ＣＯＢＯＬの拡張アドレスサポートは、ＯＳとしてＶＯＳ３／ＥＳ１を前提としている。ＶＯＳ３／ＥＳ１の仮想記憶構造を図－２に示す。既存ユーザプログラムの移行性を確保するために、ＶＯＳ３で稼動する既存のロードモジュールは、１６ＭＢ以下のジョブ固有領域で実行される。一方、３１ビットアドレシングを用いた新規プログラムは、１６ＭＢ以上の拡張ジョブ固有領域内で実行される。

以上のことから、ＣＯＢＯＬは以下の方針で拡張アドレスサポートを行うことにした。

(1)　ＣＯＢＯＬの場合、既存プログラムが多いことから、ユーザプログラムは、拡張アドレスへ、段階的に移行可能とする。

(2)　３１ビットアドレシングのプログラムの作成は、コンパイルオプションの指定によって行う。

(3)　言語仕様上、制限を持たないものにする。



図－２　ＶＯＳ３／ＥＳ１の仮想記憶構造

The method of extended addressing support of COBOL
Tatsuki MIYAKE,Ryoohei HANADA,Tetsuji IMAJO
Hitachi Software Engineering LTD,Hitachi LTD

３．拡張アドレスサポートの仕様

２．で述べた方針に基づいて、ＣＯＢＯＬの拡張アドレスサポートの仕様は、以下のとおりとした。

(1) コンパイルオプションで、ＰＭＯＤＥ＝３１を指定したとき、拡張アドレス空間で稼動する。（ＰＭＯＤＥはｐｒｏｃｅｄｕｒｅ ｍｏｄｅの略。）

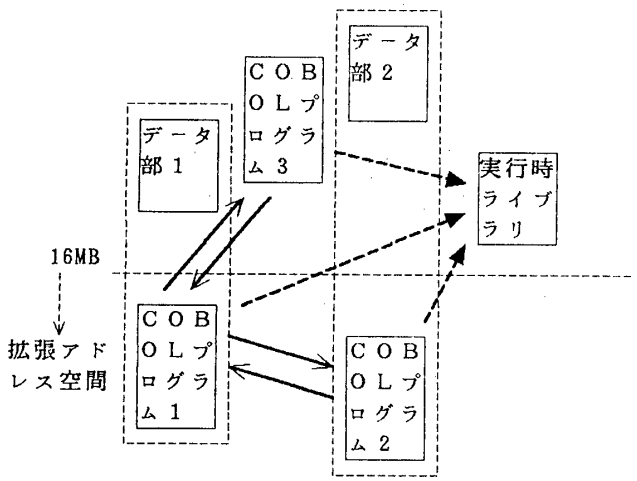
(2) ＰＭＯＤＥ＝３１でコンパイルしたロードモジュールと、従来のロードモジュールは、ダイナミックリンク（動的呼び出しによる連絡）可能とする。

４．拡張アドレスサポートの方式

拡張アドレスのプログラムを作成する上で、現行のＶＯＳ３／ＥＳ１は以下の制限を持つ。

・ ＶＳＡＭファイルを除くデータ管理は、拡張アドレスをサポートしていない。このため、データ管理インターフェースは、全て１６ＭＢ以下の空間内でとる必要がある。

この問題と、２．で述べた方針に従って以下の方式で拡張アドレスをサポートした。（図－３参照）



ＣＯＢＯＬプログラム１、２はＰＭＯＤＥ＝３１でコンパイルしたものであり、ＣＯＢＯＬプログラム３は既存のロードモジュールである。ＣＯＢＯＬプログラム１、２に対応するデータ部１、２は１６ＭＢ以下に確保される。実行時ライブラリは１６ＭＢ以下の空間にローディングされる。

図－３　ＣＯＢＯＬの拡張アドレスサポート方式

(1) ＣＯＢＯＬのオブジェクト及び実行時ライブラリでは、３バイト（２４ビット）でアドレスを保持しない。これによって、ＣＯＢＯＬは、２４ビットモードでも３１ビットモードでも、走行可能となる。

(2) データ管理が、拡張アドレスをサポートしないことから、入出力命令は、全て実行時ライブラリで処理するようにする。さらに、実行時ライブラリとオブジェクトは、ダイナミックリンクとし、実行時ライブラリは、１６ＭＢ以下の空間に、必ずローディングされるようにする。

(3) ＣＯＢＯＬプログラムのデータ部は、実行時に１６ＭＢ以下の空間にダイナミックに確保する。Ｉ／Ｏバッファ、データ管理インタフェーステーブルは、この中に含む。

(4) ダイナミックリンクの場合は、通常のＣＡＬＬ命令により、ＣＯＢＯＬが自動的に２４ビットと３１ビットのアドレシングモードを切り換えて、副プログラムに制御を渡す。（図－３参照）

５．おわりに

以上のような方式で、拡張アドレスをサポートしたことにより、次のような結果を得た。

(1) ユーザプログラムが拡張アドレス空間で実行可能となったため、ユーザアプリケーションのメモリ上の制約が緩和されると共に、より大規模なシステムの構築が可能となった。

(2) ＰＭＯＤＥ＝３１を指定してコンパイルすることにより、必要に応じて、ユーザプログラムをロードモジュール単位に、段階的に拡張アドレスへ移行することが可能となった。

(3) データ部を、実行時に動的に確保することから、リエントラントな構造となった。