

日本語 PC/VM BOND : マイクロ・メインフレーム結合の実現例

黒沢 隆, 相原 達,
平賀 瑠美, 金田 佳久, 緒方 正暢
日本アイ・ビー・エム株式会社
東京基礎研究所

日本語 PC/VM BOND は、ホスト・コンピュータの資源をユーザがパーソナル・コンピュータの環境から日本語 DOS のインタフェースでアクセスすることを可能にするソフトウェアである。我々が開発した日本語 PC/VM BOND について、マイクロ・メインフレーム結合のソフトウェアとしての位置付け、機能 (仮想ディスク機能, ホスト・コマンド実行機能, メッセージ交換機能など), 基本構成, および開発にあたり解決した問題点について述べる。また, より汎用的な機能を提供する ECF についても簡単に紹介する。

“Japanese PC/VM Bond: An Example of Micro-Mainframe Link Implementation” (in Japanese)

Takashi Kurosawa, Tohru Aihara, Rumi Hiraga, Yoshihisa Kanada
and Masanobu Ogata
Tokyo Research Laboratory
IBM Japan, Ltd.
5-19, Sanbancho, Chiyoda-ku, Tokyo 102, Japan

Japanese PC/VM Bond is a Micro-Mainframe Link software which allows personal computer users to access resources of a host computer in a single system image. In this paper we describe the functions of Japanese PC/VM Bond (Virtual Disk, Host Command Execution, Message Exchange Service, etc.), the logical structure, and some problems we solved during the implementation. We also mention on the enhanced capability of ECF.

1. はじめに

これまでパソコン（パーソナル・コンピュータ）と大型のホスト（ホスト・コンピュータ）とは、それぞれの機能的な役割の違いから、ほとんど独立にオフィスの環境に導入されてきた。しかし企業内での急速なパソコンの普及に伴い、この異なる性格を持つ2つのコンピュータを一つのシステムとして効果的に連携させるためのマイクロ・メインフレーム結合¹ [林86A], [石田86], [真田86]が、企業内の情報化に不可欠なものとして脚光を浴びている。本稿では、当社が昨年(1986年)に発表及び出荷をおこなった『日本語 PC/VM BOND』 [相原87], [平賀87], [黒沢87], [KURO87], [BOND86]を、MMLソフトウェアの一例として紹介し、その機能や基本構成、あるいはMMLとしての位置付けなどについて述べる。開発にあたり解決した問題点についても述べる。

また、日本語 PC/VM BONDと同時に発表され、より汎用的なMML製品である『システム/370 - ワークステーション拡張接続機能²』 [林86B], [関山86]についても、日本語 PC/VM BONDとの関係づけを行ないながら、簡単に紹介したいと思う。

2. 日本語 PC/VM BONDとは何か

日本語 PC/VM BONDは、パソコンとしてのIBMマルチステーション5550ファミリ³の日本語DOS⁴と、ホストとしてのIBMシステム/370のVM/SP-CMS⁵との間のMMLを実現するソフトウェアである。さらに日本語 DOS は、日本語 3270PCまたは日本語 3270PC/G⁶の環境下になければならない。

歴史的には、1984年にIBMパーソナル・コンピュータ用のPC/VM BONDが、IBMトーマス・J・ワトソン研究所の研究成果を製品化した形で発表されている。筆者らは、その米国版のPC/VM

BONDに対して、日本語処理機能の追加と端末エミュレータの変更により日本語 PC/VM BONDの作成を行なった。

ここでは、MMLの実現方法について考察し、日本語 PC/VM BONDのMMLとしての位置付けを行なう。

ホストとパソコン - 二つの異なる世界

現在のところ、一般的にホストとパソコンは、その価格、使用環境、あるいは辿ってきた歴史の違いなどによって、次のようにそれぞれ異なる性質を持った二つの世界を形成していると考えられることができる。

ホスト

- 大きなハードウェア/ソフトウェア資源を持つ。
 - 高速処理能力
 - 大容量で高速なディスク装置
 - 高速で高機能なプリンタ
 - 今までのアプリケーション・プログラムの蓄積
 - データベース
- 共用資源の集中管理に向いている。
- 大規模なネットワークに接続されている。
- バッチ処理向きである。

パソコン

- 応答性の良いフレンドリーなユーザ・インタフェースを実現できる。
- それほど大きなハードウェア/ソフトウェア資源を持たないが、個人専用である。
- 他とは独立している。
- 理解しやすく、使いやすい。システム構成の変更も容易である。

1 Micro-Mainframe Link. 本稿では「MML」と略す。

2 Enhanced Connectivity Facilities. 本稿では「ECF」と略す。

3 IBM 5540, 5550, および 5560 のことをいう。本稿では「5550」で代表させる。

4 日本語 Disk Operating System. 以後、本稿では「DOS」と呼ぶ。

5 Virtual Machine / System Product - Conversational Monitor System. 本稿では「VM」と略す。

6 それぞれ、日本語 3270 パーソナル・コンピュータおよび日本語 3270 パーソナル・コンピュータ/グラフィックスのこと。日本語 DOS のセッションと2つまでの3270 端末セッションを提供するソフトウェアである。ユーザは、特殊なキー操作でセッションを切り換えることができる。以下、総称して3270PCと呼ぶ。

日本語 PC/VM BOND: マイクロ・メインフレーム結合の実現例

この両者の比較は 必ずしも 厳密なものではない⁷が、MML は、こうした ホスト、パソコン 両者の 特長を生かす形で実現する必要がある。

MMLの実現方法

MML は、パソコン と ホスト とが異なる システム 環境 (アーキテクチャ、オペレーティング・システム、文字コードなど) を持っていることを仮定しているので、MML 実現には、両者の環境の違いを何らかの方法で解決する必要がある。最近、MML の実現方法に対して様々な議論がなされている [AUER85] [GOLD84]、[萩野85]、[PARR85]、[真田86]が、おそらく最も重要なことは、MML がパソコンのユーザやアプリケーション・プログラムに対して、どのようなインタフェースを提供するかということである。これには、次の三つのアプローチがあるように思われる。

1. パソコンとホスト両方の既存のインタフェースをそのまま提供する。ユーザはパソコンのモードとホスト端末エミュレータのモードを(特殊なキー操作によって)切り換えながら使い、ファイル転送によってデータの交換を行なう。
2. 新しい統合 OA 環境用のインタフェースを提供する。
3. パソコンの既存のインタフェースのみ提供する。

2 及び 3 のアプローチは、ユーザに単一のシステム・イメージを与えることができ、しかも先に述べたホスト、パソコン両者の特長を生かすことができる。アプローチ 2 は、統合 OA 環境のアプリケーションに対して、最適化された MML を実現することができるが、アプローチ 3 のように、既存のパソコンのアプリケーションからホストの資源にアクセスすることはできない。

日本語 PC/VM BOND (および ECF) は、アプローチ 3 によって MML を実現しており、DOS のユーザや アプリケーション・プログラムは、ホストの存在を意識することなく、ホストの資源やデータをあ

⁷ 例えば LAN (ローカル・エリア・ネットワーク) はパソコン同士を接続してホスト的な機能 (データや高価なハードウェア資源の共有) を特定のパソコン (サーバ) に持たせることが多い。

⁸ VM のミニディスクの LINK 機能を用いる。

たかもパソコンのものであるかのように使うことができる。例えば、ECF の仮想ファイル機能によって、ユーザはパソコンの既存のエディタからホストのファイルを直接編集することが可能である。

3. 日本語 PC/VM BOND の機能

日本語 PC/VM BOND は、DOS のユーザ及びアプリケーション・プログラムが、VM/CMS の資源 (ディスク、アプリケーション、ネットワークなど) を、パソコンの資源の延長として、より簡単に利用できるように、以下の機能を提供している。

仮想ディスク機能

仮想ディスク機能は、VM/CMS のディスク装置上に仮想的な DOS のディスクを作り、DOS 側から実際にパソコンの持っているディスクと同じ方法でアクセスできる様にした機能である。仮想ディスクの実体は、DOS のフロッピー・ディスクのフォーマットで書かれた CMS ファイルである。

仮想ディスクは、以下のような特長を持つ。

- DOS のブロック装置
DOS の機能呼出しを用いて仮想ディスク上のファイルの読み書きが可能である。そのため、ほとんどの DOS のコマンドを仮想ディスクのドライブに対して実行することができる。この機能は、日本語 PC/VM BOND のブロック装置駆動ルーチンによって実現されている。
- ファイルの共有
仮想ディスクは CMS ファイルなので、他のユーザと共有して⁸使うこともできる。
- 高いセキュリティ管理
仮想ディスクは VM/CMS によって管理されているため、セキュリティやメンテナンスの面で有利である。

仮想ディスクを活用するために日本語 PC/VM BOND は以下のような機能を提供する。

日本語 PC/VM BOND: マイクロ・メインフレーム結合の実現例

- 仮想ディスクの割当て
仮想ディスクを DOS から使用するには、必要な仮想ディスクをドライブ指定子に割当てればよい⁹。ユーザーは、同時に 8個まで仮想ディスクを使うことができる。
- 仮想ディスクの作成
仮想ディスクを作成する¹⁰時に、記憶容量（最小2キロバイトから最大31メガバイトまで）やディレクトリ項目数も指定することができ、目的にあった仮想ディスクを作成できる。
- CMSファイルへの変換
仮想ディスクの中にある DOS のファイルは、VM/CMS のデータ・フォーマットでないの、内容を直接 VM/CMS で利用することはできない。また、VM/CMS のファイルもそのままでは、DOSから利用できない。そこで、仮想ディスク上の DOS のファイルと CMSファイルとを変換するために、VM/CMS 側にユーティリティ¹¹が用意されており、DOS側（あるいはVM側）から呼び出すことができる。

ホスト・コマンド実行機能

日本語 PC/VM BOND のユーザーは DOS コマンド¹²を使って、DOS の環境からホストのコマンド¹³を実行し、結果を画面上に表示することができる。

このホスト・コマンド実行機能は、日本語 PC/VM BOND のもつ HOST という名前の DOS の文字装置によって実現されている。DOS プログラムによって HOST 文字装置に書き込まれた文字列は VM/

CMS 側でコマンドとして解釈・実行される。その実行結果の文字列は、ホストからパソコンへと転送され、HOST文字装置からプログラムが読み取ることによって得られる¹⁴。

メッセージ交換機能

日本語 PC/VM BOND のユーザーは、VM を通じて 漢字などの 2バイト・コードを含む日本語のメッセージを交換できる。

到着したメッセージは、いつ読みとつてもよい。メッセージを表示するために特別のプログラムを起動する必要はない。ユーザーは、都合のよいときに仕事を中断することなく、簡単なキー操作¹⁵で、ウィンドウに表示して読み取ることができる。そして、読み取ったあとは、同じキー操作をもう一度行なうことによって、ウィンドウは消えもとの画面が回復される。

メッセージ交換機能は、VM/CMS のメッセージ転送コマンド¹⁶を起動することによって実現されている。RSCS¹⁷を使ってネットワークで接続された他のノードの VM のユーザーにもメッセージを送ることができる。ただし、受信したメッセージが VM/CMSからDOSに送られるのは、何らかの PC/VM BONDの機能を使用した後となる。そのため、ユーザーがメッセージをDOS側へ取り込むためにコマンドを入力する必要が生じる場合もある。

REXX/PC¹⁸ 言語

- 9 SETDISKコマンドを使う。ただし、VM 側に仮想ディスクの割当て情報を含む PCDISK INITMAPという名前の CMSファイルを作成しておけば、VM 側のサーバ・プログラムである PC BONDの起動時に自動的に割当てが行なわれる。
- 10 PCDISKユーティリティを利用する。
- 11 PCDISKユーティリティ。このファイル変換機能は、パソコンとホスト間のファイル転送と同等の機能を提供する。
- 12 VM という名前の DOS コマンド
- 13 CMSコマンド、CPコマンド、またはEXECプロシージャ
- 14 HOST文字装置はVMコマンドのラインモードで出力された実行結果のみをDOS 側に返す。全画面モードの出力は、ホット・キーで切り換えて 3270モードの画面で見ることができ、それに対して応答することになる。
- 15 前面キー・Ctrlキー・左シフトキーの3つを同時に押す
- 16 VM/CMSのTELLコマンド
- 17 Remote Spooling Communications Subsystem

日本語 PC/VM BOND: マイクロ・メインフレーム結合の実現例

REXX/PCは、日本語PC/VM BONDと合わせて提供されるインタープリタ型のプログラム言語である。REXX/PCは、DOSのバッチ・コマンドをPascal言語のような構造化プログラミングの概念を用いて作成できるように拡張した。ユーザはREXX/PCを使って、従来の方法よりずっと簡単に、複雑な制御構造を持つ高度なバッチ・プログラムを作成することができる。

また、REXX/PCは、VM/CMSのREXXのサブセットと考えることができる。REXX/PCは、機能的には日本語PC/VM BONDとは独立しているが、日本語PC/VM BONDのユーザがDOSの環境からVM/CMSのプログラムを起動し、連携処理を行う適用業務プログラムを開発する際に大変便利である。

4. 日本語 PC/VM BONDの基本構成

3.で述べた機能を提供している日本語PC/VM BONDの基本構成、および日本語PC/VM BONDが行なっている文字コード変換について述べる。

システム構成

図1に日本語PC/VM BONDの論理構成を示す。パソコン側にはVM BONDプログラムがあり、VM/CMSに接続している。ホスト側にはPC BONDプログラムがあり、サーバとして機能している。

基本的な動作は、VM BONDがPC BONDに対してサービスの要求を出すと、PC BONDはその要求を処理して結果をVM BONDに返すという形をとる。

VM BONDの実体はDOSの装置駆動ルーチンの集まりであり、PC BONDの実体はCMSの中核拡張域に常駐するコマンド・モジュールである。VM BONDは3270PCのAPI¹⁹を通して、PC BONDとデータの送受信を行なう。また、VM側にはPCDISKプログラムがあり、先に述べたように仮想

ディスク上のDOSファイルとCMSファイルとの変換などを行なう。

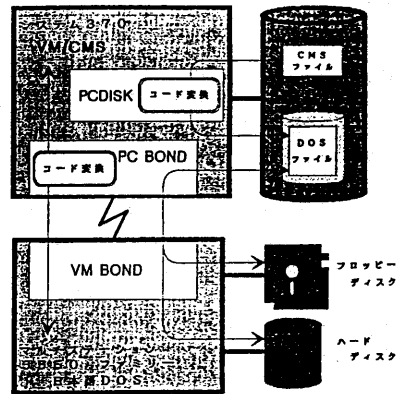


図1: 日本語 PC/VM BONDの論理構造

コード変換方式

日本語PC/VM BONDにおけるコード変換について述べる。表1に示されるように、日本語DOSとVM/CMSとは異なる文字コード・セットを用いており、日本語PC/VM BONDではPC BONDとPCDISKにおいて、両者の間のコード変換を行なっている。

表1. パソコンとホストにおける文字コード・セットの違い

	1バイト・コード	2バイト・コード
3270PCのPCモード (日本語DOS)	JIS 8ビット・コード	JISCI漢字コード
ホスト・システム (VM/CMS)	EBCDIC 英数カナ文字セット 又は英小文字セット	IBM漢字コード

以下では、1バイト文字と2バイト文字とを区別せずに、DOS側で使われる文字セットを『JISCI²⁰』、VM側のを『EBCDIC²¹』と呼ぶことにする。EBCDICの1バイト・コードには、EBCDIC英

18 Restructured Extended Executor/Personal Computer

19 Application Program Interface. (適用業務プログラム・インタフェース)

20 Japanese Industrial Standard Code for Information Interchange

日本語 PC/VM BOND: マイクロ・メインフレーム結合の実現例

小文字セットと EBCDIC 英数カナ文字セットがあり、それぞれ「英小文字」、「英数カナ」と呼ぶ²²。コード変換は、PC BOND と PCDISK において行なわれる(図1参照)。ここでのコード変換方式の特徴を以下に示す。

- コード変換はすべて VM/CMS 側で行なう。
- PC BONDプログラムとPCDISKプログラムは、コード変換のための変換テーブルを共有する。変換テーブルは、PC BOND プログラムが持つ。
- PC BONDプログラムと PCDISKプログラムのコード変換は、特別な指定がない限り 3270PCの基本文字モード²³に従う。
- JISCII と EBCDIC のコード変換のほか、ユーザ定義文字等も扱えるようにするため、ユーザ独自の交換テーブルによるコード変換もサポートする。

3270PC のファイル転送ユーティリティ では、コード変換をパソコン側で行なっている。一方、日本語 PC/VM BOND でコード変換をホスト側で行なうようにしたのは、以下の点を考慮したためである。

- 将来、パソコンどうしの結合を含めた連携処理が実現されるようになった時、JISCII と EBCDIC の間のコード変換は不要になる。
- ユーザ独自の交換テーブルを共有する場合は考えれば、交換テーブルを管理するのは、ホスト側の方がよい。

交換テーブルを PC BOND と PCDISK で共有させた理由として、以下のことが挙げられる。

- ユーザ独自の交換テーブルを使用する場合、日本語 PC/VM BOND の交換テーブルを書き替えなくてはならない。交換テーブルを共有することで、書き替えを一ヶ所にとどめておくことができる。

21 Extended Binary Coded Decimal Interchange Code

22 英小文字にはアルファベットの小文字があるが、半角カタカナ文字がない。一方、英数カナにはカタカナ文字はあるが、アルファベットの小文字がない。DOS 側ではアルファベットの小文字と半角カタカナ文字の両方を使えるが、VM/CMS 側では片方を選択しなければならない。

23 基本文字モードは、3270PC のカスタム時に英数カナ か英小文字 に決められる。

24 以後、簡単のために、日本語 PC/VM BOND を「日本語版」と呼び、基にした米国版の PC/VM BOND を「米国版」と呼ぶことにする。

日本語 PC/VM BOND: マイクロ・メインフレーム結合の実現例

- 2バイト・コードの変換テーブルのサイズが大きい。

5. 日本語機能実現における諸問題

この章では、米国版の PC/VM BOND を基に日本語 PC/VM BOND²⁴を作成するにあたって解決した主な問題点について述べる。

MMLのソフトウェアを実装する際には、ホストとパソコンで使われる文字コードの違いが重要な問題となる。さらに、日本語 PC/VM BOND を作るにあたっては、米国版に比べて以下のように文字セットが複雑になっているため扱いが難しくなった。

- VM で使われる EBCDIC では 2種類の1バイト・コードが存在する。
- 1バイト・コードと 2バイト・コードとが混在する。
- 2バイト・コードの一部は「ユーザ定義文字」として解放されている。
- 2バイト・コードの取り扱いがパソコンとホストでは異なる。EBCDIC の 2バイト・コードを 1バイト・コードと区別するために、一続きの 2バイト・コードの、前後には制御文字 SO (Shift Out) と SI (Shift In) が 入る。一方、JISCII では、2バイト・コードはコードそのもので識別できる。

コード変換

4.で述べたように、日本語 PC/VM BOND では、コード変換をホスト側で行なった。ここで解決すべき問題には、次のものがあつた。

1. ホスト側で 2種類の EBCDIC 1バイト・コードをどのようにサポートするか。
2. ユーザ独自の交換テーブルをどのようにサポートするか。

第一の問題の解決方法として、PC BONDとPCDISKの標準のコード変換をPC BONDが決定するようにした。つまり、PC BONDが起動されるときに3270PC 端末エミュレータの基本文字モードを調べるようにした。

第二の問題を解決するために、ユーザが変更した変換テーブルを仮想ディスク中のDOSファイルとしてもち、それを利用できるようなオプションを用意した。

2バイト・コードのサポート

2バイト・コードを使用するために、注意した点について述べる。

- コード変換により、データの長さが変わることがある。これは、EBCDICで2バイト文字を表すときに必要となるSO/SIが挿入されたり削除されるためである。
- PCDISK プログラムで、8バイトより長いパラメータを使用できるようにした。DOSファイル名などに対応するパラメータは、JISCIで8バイト以下の長さでも、EBCDICに変換するとSO/SIのために8バイトより長くなることもある。
- 2バイト文字を表示するために、5550 BIOS²⁵機能呼出しが常に使われる。
米国版のメッセージ・サービスは、メッセージ・ウィンドウにメッセージを表示するためにPC BIOSを使っていなかった。

日本語 3270PC の環境

- 3270PC API の使用
ホストと通信するために、米国版は独自のプログラミング・インタフェースを使っていたので、それを日本語版では、3270PCのAPIを使用するように変更した。その際、変更量を最小にするため

に、VM BONDと3270PC 端末エミュレータとの間に、米国版のプログラミング・インタフェースをシミュレートするレイヤを置いた。

- 端末インテリジェンスの使用
米国版の端末エミュレータは、CUT²⁶モードでIBM3274クラスター・コントローラとつながっている。一方、3270PCは、DFT²⁷モードでつながっており、より多くの機能を使える。日本語版では、データ交換に構造化フィールド²⁸を使い、ふたつの端末セッションを使用できるようにするなど、DFTのインテリジェンスを活用している。
- 3270PC ファイル転送プログラムとの互換性
PCDISK プログラムでは、CMS ファイルとDOSファイルの変換において、3270PCファイル転送ユーティリティと同一の機能を提供している。また、PC/VM BONDを起動した後でも、3270PCファイル転送ユーティリティを使用することができるようにした。

6. おわりに

本論文で述べた日本語 PC/VM BOND は、日本語 DOS のイメージでホストの資源にアクセスできるという意味において、ホストとパソコンの新たな接続形態を作り出すことになった。ここでは、日本語 PC/VM BOND に続いて発表された ECF²⁹について、拡張される機能を以下に述べる。実際、日本語 PC/VM BOND は、ECF の前段階の製品として発表されている。

1. VM / SP - CMS だけでなく、MVS / XA - TSO / E²⁹ のサポート
2. SRPI³⁰
ホスト側のサーバ・プログラムとパソコン側のリクエスト・プログラムの間におけるデータの交換のためのインタフェース。
3. 3つの仮想サービス
仮想ディスク機能に加えて、以下の2つの機能

25 Basic Input Output System

26 Control Unit Terminal

27 Distributed Function Terminal

28 3270 データ・ストリームの一部で、スクリーン上のデータと独立であり、より高いデータ交換効率を期待できる。

29 Multiple Virtual Storage / Extended Architecture - Time Sharing Option / Extensions

30 Server / Requester Programming Interface

日本語 PC/VM BOND: マイクロ・メインフレーム結合の実現例

が提供される。

- 仮想ファイル機能
ホストのファイルをそのまま DOS ファイルとして扱うことを可能にする機能。
 - 仮想プリンタ機能
ホストのプリンタを DOS のプリンタと同様に使える機能。
4. ホスト・データ・アクセス機能
パソコン側から ホストの データベース (DATABASE 2, SQL/DS 等) を直接アクセスすることを可能にする機能。

このように、ECFによってパソコンのユーザは、システム370のより多くの資源をDOSのイメージでアクセスすることが可能になる。特にSRPIは、ユーザ独自のMMLから簡単に実現できるという意味において、ECFの持つ機能のなかでも大変重要な機能である。

なお、この原稿の版下の作成には、当研究所研究員山内長承氏によるJANUSの漢字タグを利用した。

文献

- [相原87] 相原,黒沢,金田,平賀: “日本語 PC/VM BOND (2)”. 情報処理学会第34回全国大会(発表予定) 7M-7 (March 1987) pp. 2153-2154.
- [AUER85] Auerbach, J.: “File Request Transparency Between Heterogeneous Systems”. IBM Research Report RC11496 (November 1985).
- [BOND86] “日本語 PC/VM BOND ユーザーズ・ガイド”. 日本アイ・ビー・エム株式会社 N:SB10-7793 (1986).
- [GOLD84] Goldstein, B.C., Heller, A.R., Moss, F.H., and Wladawsky-Berger, I.: “Directions in Cooperative Processing between Workstations and Hosts”. IBM Systems Journal 23 (3) (1984) pp. 236-244.
- [萩野85] 萩野和彦: “メインフレーム・WS連携システムにおける制御構造”. 情報処理学会オペレーティング・システム研究会 OS28-1 (September 1985).
- [林86A] 林 秀幸: “ホストとワークステーションが連携するマイクロ・メインフレーム結合が急増”. 日経コミュニケーション 1986年2月10日号 (February 1986) pp. 85-91.
- [林86B] 林 秀幸: “日本IBM, 拡張接続機能でホスト-パソコン連携の基盤確立”. 日経コミュニケーション 1986年9月8日号 (September 1986) pp. 50-52.
- [平賀87] 平賀,黒沢,金田,相原: “日本語 PC/VM BOND (1)”. 情報処理学会第34回全国大会(発表予定) 7M-6 (March 1987) pp. 2151-2152.
- [石田86] 石田雅也: “本格的な普及へ向かうマイクロ・メインフレーム・リンク”. 日経コンピュータ 1987年2月16日号 (February 1987) pp. 71-84.
- [黒沢87] 黒沢隆: “日本語 PC/VM BOND”. ELAN (January 1987) pp. 54-55.
- [KURO87] Kurosawa, T.: “Japanese PC/VM Bond: An Example of Cooperative Processing between Personal Computer and Host Computer”. Denshi Tokyo 25 (1986) pp. 40-45.
- [PARR85] Parr, F.N., Auerbach, J.S., and Goldstein, B.C.: “Distributed Processing Involving Personal Computers and Mainframe Hosts”. IEEE Journal on Selected Areas in Communications SAC-3 (3) (May 1985) pp. 479-489.
- [真田86] 真田英彦: “MMLを実現するためのホストコンピュータとパソコンの機能: 4段階に分けて結合のステップをみる”. コンピュータ&ネットワークLAN (November 1986) pp. 37-40.
- [関山86] 関山茂寛: “IBMが実現するワークステーションとホスト・システムのコネクティビティ”. コンピュータ&ネットワークLAN (November 1986) pp. 41-48.