



Title	BBSホストプログラムの開発
Author(s)	石川, 治
Citation	北海道大學工學部研究報告, 169, 21-28
Issue Date	1994-06-28
Doc URL	http://hdl.handle.net/2115/42424
Type	bulletin (article)
File Information	169_21-28.pdf



[Instructions for use](#)

BBS*ホストプログラムの開発

石川 治

(平成 5 年 12 月 24 日受理)

Development of a BBS Host Program

Osamu ISHIKAWA

(Received December 24, 1993)

Abstract

According to the presumption that personal computer communications would grow up remarkably as electronic social field, we have opened a BBS host system experimentally and examined the possibilities in collaboration with citizen.

One of our themes was development of a multi-channel BBS host program. This report describes the three fundamental gists of our program, that is to say,

- 1) Corresponding with multi-channel.
- 2) High-speed processing of transmission and reception.
- 3) Efficiency utilization of limited memories.

Our program is up-loaded to NIFTY-Serve, therefore anybody can down-load it and open a high-function BBS host easily.

1. はじめに

通産省の外郭団体である財団法人ニューメディア開発協会が、電子ネットワーク協議会の協力を得て 1993 年度に行った「全国パソコンネット局実態調査」によると、把握できたネット局数は 2210 局、延べ会員数は 196 万人に達するという¹⁾。1985 年の電気通信事業法の施行によっていわゆる「通信の自由化」がもたらされたわけであるが、その当時、ネット局数 50 局、会員数 3 万人程度といわれていたのを思うと、その発展拡大は急速であり、今後もこの傾向が続くものと予想されている。

大規模商用ネットは 1985, 6 年ころからいっせいに試験運用を開始し、今日その最大手の PC-VAN および NIFTY-Serve はそれぞれ 50 万人を越える会員を擁するほどに成長している一方、いわゆる「草の根ネット」と称せられるプライベートかつローカルなネットの開設も盛んである。

プライベートかつローカルなネットが自由に多数生成され、さらに必要に応じて互いにリンクし合えば、近い将来において、市民自身の手による情報交換システムとして社会的基盤にまで成長するのではなからうかとの予測のもとに、筆者らは実験的なネットを開設し、そこにアクセス

* Bulletin Board System (電子掲示板システム) : パソコン通信システム全体を指して用いられることが多い。
電気工学科 電気回路学講座

する多くの市民(性別, 年齢構成, 職業などさまざまな)との共同作業を通してその可能性を探ってきた。

その中で, ネットワーク活動を支えるために欠かせない要素としての, BBSホストプログラムの開発, 改良をひとつの主要なテーマとして取り組んできた。現在その成果はBMWというプログラム名で一定の評価を得ている²⁾。本稿では, パソコン上で高速高機能の多回線ホストプログラムを実現するための, 疑似マルチタスクの手法, 高速送受信, メモリの有効利用といった基本的な部分での技術的な要点について報告する。

2. 経緯ならびにホストプログラム開発のポイント

今日, 日常的にパソコンあるいはワークステーションを使いこなしているユーザで, PDSないしフリーソフトウェアと称されるプログラム(本稿では便宜上これ以降, これらを「フリーソフト」と称することにする)を使用していない人は少ないだろう。パソコン通信という土壌の上で育ってきたフリーソフトが, パソコンを使用する上での一定の基盤になりつつあるということこそ, パソコン通信, あるいはより広い意味でのコンピュータネットワークがひとつの文化として定着しつつあることを示している。

しかしパソコン通信の可能性は, こうしたフリーソフトの交換という次元にとどまるものではない。電子社会的なコミュニケーションの「場」を提供するという点にこそ, その可能性は開かれていると見るべきであり, たとえば, 研究基盤ネットワーク整備の重要性という面からも新たに着目されているところでもある³⁾。

また, ネットワーク技術の多くの部分が, 市民有志によるボランティア的な献身に支えられて発達してきたということも見逃せない事実である。

筆者らはシリアルインタフェースの研究過程でパソコン通信の存在を知り, 1986年以來積極的に関わってきた。そうした中で, 先端的なデータ通信技術が, 市民自身の手によって, 市民社会の中にかんして根付き浸透していくのかを能動的に検証することを目的として, プライベートなネットを1988年8月に実験的に開局し現在に至っている。

5年と数ヶ月が経過した現在のシステム構成を図1に示す。

電子社会的なコミュニケーションの「場」としてのネットの運用上の諸問題については, 稿を改めて報告したいが, 「場」を保証する技術として欠かせないものが, ネットのセンターにあたるホストシステムを動かすBBSホストプログラムである。

実験開始に当たっては, 当時ネットワークを通じて流通していたシェアウェアであるWWIVをホストプログラムとして採用した。このプログラムはあくまでも1回線用のものであった。実験運用が軌道に乗り, ユーザ数, アクセス数とも増加したのを見合せて, およそ3年経過後回線を増設する段階で, 複数回線に対応し得るプログラムに更新する必要性が生じた。そこまで実験局に蓄えられたデータ, アーティクルなどの貴重な資産の継続性を優先させることにした結果, データ構造の大枠は崩さずに, その上に複数回線対応システムを構築することとした。

新たに開発するシステムの主眼としては, あくまでも個人レベルでの運用が可能であるということに置いた。市民がだれでもたやすく入手可能なパソコンをホストマシンとして, 同時に複数回線にも対応するものでなければならない。

筆者らは開発のポイントを次のように設定した。

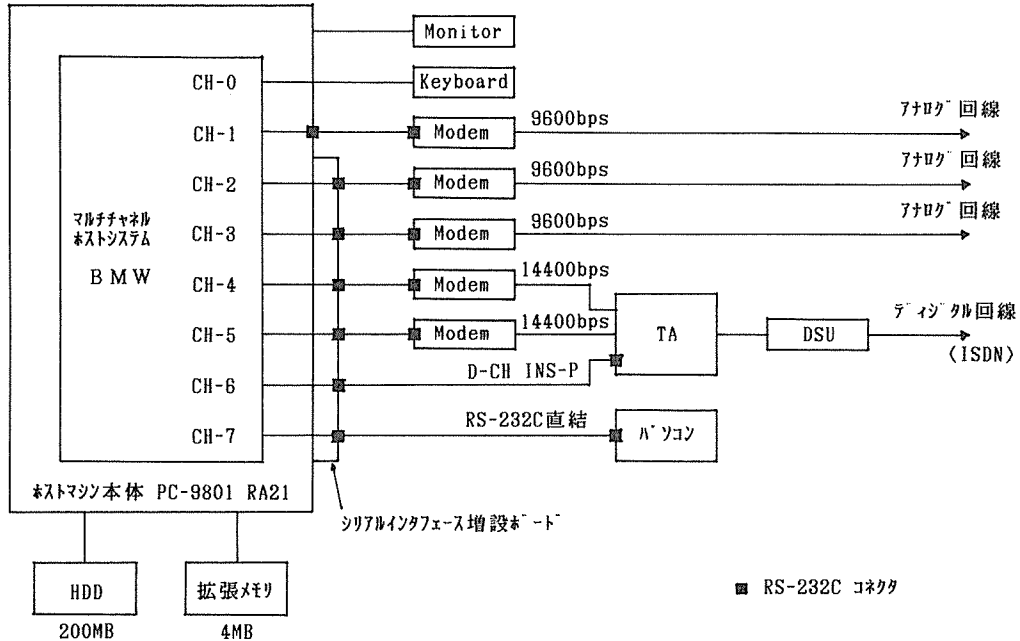


図1 実験局のシステム構成

1. システム構成について

- 1) 市民が手持ちの最小限のハードウェア構成で手軽に開設できること。

具体的には、ホストマシンはNEC PC-9801シリーズ (およびその互換機種で、メインメモリ 640 Kバイト, およびグラフィック用VRAM 192Kバイト実装であること) のパソコンをホストマシンとし、さらに最低1回線の電話回線およびモデムがあれば開設可能である。

- 2) 優れた拡張性を有すること。

拡張性の最たる部分は、複数回線に対応し得るということで、現在9回線まで拡張可能としている。その他、大容量ハードディスク、拡張メモリ (EMS, RAMディスク) を使用することにより、ホスト機能をさらにパワーアップすることができる。

2. ホストシステムとしてのサービス機能について

- 1) アクセスするユーザには、わかりやすくかつ豊富な機能を提供する。
- 2) ネットの運用者には、強力なメンテナンス機能を提供する。

本稿では、ホストプログラムのサービス機能面での細部には触れず、各種の機能を支える基本的な部分である次の3項目についてその要点を述べることにする。

1. 複数回線対応
2. 送/受信の高速処理
3. 限られたメモリの有効利用

なお、開発されたプログラムはMS-DOS上で動き、ホストプログラム本体と通信ドライバと

からなっている。プログラム本体はターボパスカル (Ver. 6)、通信ドライバはアセンブラで記述した。

3. 複数回線対応

ホストプログラム自体は単一タスクとして実行されるが、その中で回線を通して同時にログインしてくる複数のチャンネルの処理を、リアルタイムで並行して行わなければならない。各チャンネルの処理は、事実上互いに独立したタスクに相当すると見なされるが、MS-DOSは単一タスク処理を前提としたOSであるので、マルチタスクを容易に実現する仕組みは提供されていない。従ってなんらかのテクニックを用いて、疑似的なマルチタスク処理を施す必要がある。

マルチタスクの手法としては、タイムスライス方式とイベントドリブン方式とがあるが、筆者らが用いたのはイベントドリブン方式である。

MS-DOS上で現在実行中のタスク環境を保存し、別のタスクに制御を移す手法が「ジャンプコール・ユニット」⁴⁾としてPC-VANで提案されている。筆者らはホストプログラムのマルチタスク処理に、この手法を援用することとした。イベントドリブン方式の場合もハードウェア割込みによりタスクを切り替える場合が多いが、これは完全にソフトウェアに依拠したイベントドリブン方式で、プログラムの随所に記述されたトランスファー手続きがタスクスイッチャーの役割を果たす。

マルチチャンネル処理にジャンプコールユニットを援用する場合、グローバルな変数はともかく、手続きや関数の引数、あるいはその中で使用されるローカルな変数はすべて、チャンネルごとに独立した相互干渉のないスタック領域を確保してそこに保存する。タスクスイッチングが発生するつど、現在のチャンネル処理のスタックポインタとプログラムカウンタを所定の領域に退避し、かつ先に退避しておいた次に処理すべきチャンネルのスタックポインタとプログラムカウンタとをセットし直すことによってスイッチング処理を完了する。

ここで解決されるべき重要な問題は、同時に複数のチャンネルから同一の資源の使用を競合して要求された場合の処理、いわゆる排他制御である。ホストシステムの場合、通信系の資源たとえば、RS-232Cインタフェースあるいはモデムは、チャンネルの数だけ用意しなければならないので競合は生じない。競合が生じる可能性があるのはファイルのみである。

ファイル変数をチャンネルごとに独立に定義しておけば、ファイルの読み込み処理の場合には不都合は生じない。従って複数のチャンネルから同一ファイルへの書き込み処理要求が同時に発生した場合の対処を慎重にしておけばよいことになる。排他制御の一般的な手法であるセマフォを用いるまでもなく、トランスファー手続きの埋め込みに注意を払えばよいのである。これは、純然たるソフトウェア的タスクスイッチ手法であることのメリットであるといえよう。

タスクスイッチング処理は、実際の各チャンネルの処理とは無縁のオーバーヘッドであるから、それが全体の時間に占める割合が少ないほど望ましい。たとえば、通信ドライバとの中のデータ転送で、1バイト単位でタスクスイッチングするとオーバーヘッドの増大で非常に遅いシステムになるが、これを、1行(最大80バイト)単位でのブロック転送に置き換えると、見違えるほどの高速処理が可能となる、などである。

逆に、タスクスイッチング処理が少なすぎると、ひとつのチャンネルが連続してシステムを占有することになり、他のチャンネルの処理を重くする。同時にまた、使用しているシステムのCPUの性能にも大きく依存するので、ある程度の試行錯誤を必要とする。

以下に、筆者らのマルチチャンネル処理の要点をまとめる。

1. チャンネル間のタスクスイッチングにはジャンプコールユニットを用いる。
2. チャンネル間には優先度を設けず、ラウンドロビン方式で平等かつ巡回的に処理を割当てる。
3. 排他制御としては、あるチャンネルがファイルへの書き込み動作に入った場合、ファイルをオープンした後書き込み処理が完了しクローズされるまでタスクスイッチングを凍結する。

4. 送／受信の高速処理

処理の高速化を実現しようとする場合、大まかにいって次の3つのフェイズでルーチンの高速化を計らなければならない。

1. 通信ドライバにおける送受信処理
2. 通信ドライバとホストプログラム本体との間でのデータ転送処理
3. ホストプログラム本体におけるファイル入出力処理

4.1 通信ドライバにおける送受信処理の高速化

DCE（モデムないしTA）からのデータの受信に関しては、データのとりこぼしを防ぐ意味で従来からシリアルインタフェースLSIからの受信割込みが用いられていたが、データの送信に関しては送信すべきデータが発生するつど、シリアルインタフェースのステータスをポーリングし、送信可能状態を確認してからデータを送出するというポーリング制御が一般的であった。

ポーリング制御での送信の場合、送信すべきデータが発生したとき、シリアルインタフェースが送信可能状態になれば、可能状態になるまで待機せざるを得ないので処理は先に進めずループを空転することになる。通信速度が2400 bps程度で単回線のホストの場合にはそれでも十分であるが、高速多回線ホストを実現する上ではボトルネックとなる。

そこで、データの送信についても割込み制御を用いることにした。通信ドライバに、受信バッファ（割込みで受信したデータを一時的に蓄えておくバッファ）の他に送信バッファも用意し、送信すべきデータが発生した場合シリアルインタフェースのステータスのいかに関わらずデータをここに放りこんでおく。シリアルインタフェースが送信可能状態になれば自動的に割込みがかかって、このバッファ内のデータを取り出して送信する。こうすることによって本体のデータ送信処理は空転することなく直ちに次の処理に移れるわけである。

4.2 通信ドライバとホストプログラム本体との間でのデータ転送処理の高速化

4.1 で述べたように、通信ドライバは送受信とも割込み制御で駆動され、送受信用のバッファが用意されることになるが、通信ドライバ上のバッファとホストプログラム本体との間でのデータの転送も効率的に行われなければ、結果として通信速度の低下をまねくことになる。

従来は送受信とも1バイト単位での転送が一般的であったが、送信にも割込み制御を用いることにより、送受信ともデータをブロック単位でまとめて転送することが可能となった。

受信データを本体に取り込む場合には、その時点で受信バッファに溜まっている未処理データを最大128バイト単位でまとめて転送する。また送信データの場合は、テキストデータならば1行単位、バイナリーデータでは64バイトを単位として一括して本体から通信ドライバの送信バッファに送り込む。

同時に、3. でも触れたように、チャンネル間のタスクスイッチング処理をブロック転送ごとに行うように変更することによって、全体の処理が飛躍的に高速化されることとなった。

4.3 ホストプログラム本体におけるファイル入出力処理の高速化

これは電子掲示板、電子メールなどの書き込み、読みだし処理、あるいはバイナリーファイルのアップロード、ダウンロード処理と直接結びつく部分であるので、いたずらに高速化を追求するあまり逆にサービスの低下を招くことは避けなければならない。

バイナリーファイルのアップロード、ダウンロード用には、X/Y-modem、MLINKといったバイナリー転送用プロトコルを移植しサポートしているが、これらは画面制御を考慮する必要がなく、その意味でホストにかかる負担は少ない。

テキストデータの送受信の場合は行単位で処理するためデータ長が一定せず、デリミタの確認、あるいは付加など常時チェックする必要がある上、画面制御のための処理が加わる。さらに1ページ行数、1行文字数任意設定などのユーザサービスにも対応するためホストの効率は低下する。また、テキスト受信の場合には、直ちにユーザ端末にエコーバックさせなければならないので、受信データにはほぼ等しい量のデータを同時に送信もしなければならない。

それゆえこのフェーズはオプション化して、高速モードを選択した場合のみ、上記のユーザサービスの一部をカットすることと引き換えに、より高速な送受信サービスを提供することとした。

5. 限られたメモリの有効利用

ホストシステムがあらかじめ拡張メモリを使用できる環境にあれば、メモリ上での制約も緩和されるのであるが、筆者らはマシン本体に装備されている基本仕様のメモリのみで複数回線に対応させることを主眼とした。

しかし高速高機能を満足させる複数回線ホストシステムを640 Kバイトのメインメモリのみで実現するのはほとんど不可能で、なんらかの手法でメモリの制約を打開する必要に迫られた。そこでグラフィックスVRAMに着目した。文字によるコミュニケーションを主体とするホストとしては、現状ではグラフィックスを使用しない。ここに標準で装備されている192 Kバイト(96 Kバイト×2面、機種によっては128 Kバイト×2面)のメモリを有効に利用すべきである。

現時点でのコード、データおよびバッファについてのメモリ使用状況について以下に述べる。

5.1 コード

RAM上に単純に展開すると、コードのみで400 Kバイト以上を占有し、そのままでは事実上実行不能となる。それゆえ、システム運用者のみによって使用されるメンテナンスサービスのほとんどをオーバーレイーチンとしRAM上の常駐から排除、実際にコールされたときのみオーバーレイバッファ上で実行される形式とした。

5.2 データ

データセグメントは最大で64 Kバイトというコンパイラによる制約があるため、グローバルな変数でもチャンネルごとに要請されるものはそのほとんどをヒープ領域に動的に割当てることによって、チャンネル共通のデータセグメントのサイズをできるだけ圧縮した。逆に10チャンネル分の変数の割当てを可能にするため、ヒープ領域はあらかじめ十分大きく確保しておく必要がある。

スタックセグメントについても、システムの起動時、停止時に使用される共通スタックの他に、実際の運転時には3. で述べたような各チャンネルごとに独立したローカルスタック領域を確保する必要がある。しかもヒープでの動的割当てとは違って、静的に定義しておかなくてはならない。筆者らのシステムでは、第1画面用のGVRAM 96Kバイトをこれに当てた。電話回線9チャンネル+ホストコンソールからのローカルアクセスチャンネルの計10チャンネル分必要なことから、1チャンネル当たり9.5 Kバイトの割当てとしている。

5.3 通信ドライバの送受信バッファ

通信ドライバでは送受信とも割込み処理を行う。また、本体とのデータ転送をブロック単位で円滑に行うためにも、ドライバ自体にある程度余裕のある大きさのバッファを、各チャンネルごとに独立に、しかも送受信それぞれ別個に用意する必要がある。

筆者らのシステムでは、これに第2画面用のGVRAM 96Kバイトを当てた。電話回線9チャンネルの各々に、受信バッファとして6 Kバイト、送信バッファとして2 Kバイトを用意している。

以上のように、グラフィックVRAMは第1画面用をスタック、第2画面用を通信バッファとして運転中常時使用することになるため、その切り替え制御は入念になされなければいけない。

現在のメモリ構成を図2に示す。

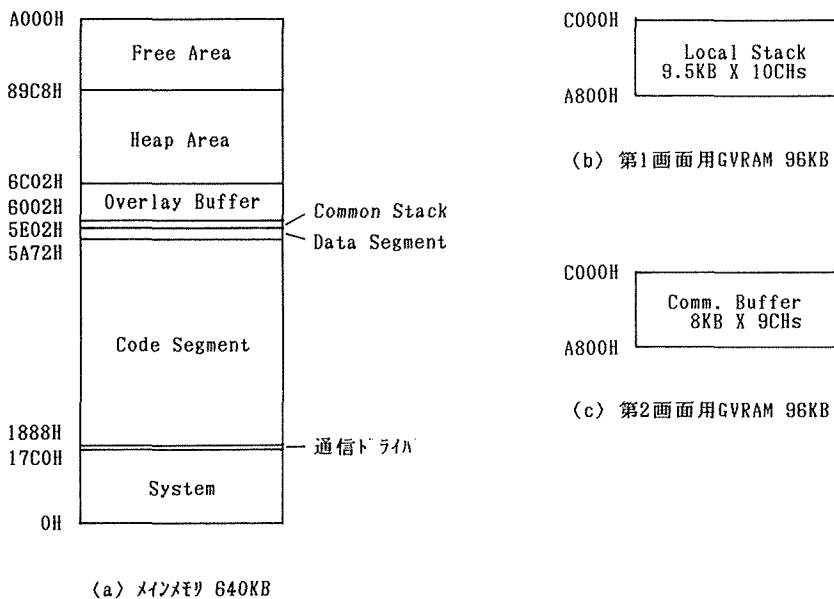


図2 実験局のメモリ構成

6. おわりに

パソコンがスタンドアロンで使用される形態から、さまざまな伝送路を用いてネットワーク化されるのは時代の趨勢である。現在パソコンネットワーク化の流れは、大きく分けて一方にLANがあり、他方にパソコン通信がある。LANは、資源の共有と分散という特性から、業務遂

行上での効率性向上の期待が大きく、企業ベースでかなりの普及を見せている。

電子掲示板システム（BBS）を中心におくパソコン通信は、電子社会的なコミュニケーションの「場」としてとらえられる。これまでところでは、企業のような階層組織とは衝突するケースが多いことが指摘されている⁵⁾が、市民が階層を意識せず1個人として参加しうるネットワークとしての発展は目覚ましい。このコミュニケーション「場」は、意見交換を通して問題解決能力を高め、新しい価値を創造する「場」としての可能性も秘めている。

個人の価値観、あるいはニーズの多様化に対応して、「たしかさ、やさしさ、しなやかさ」をキーワードとする通信のパーソナル化が大きな研究開発テーマとなりつつある^{6,7)}が、単に特定個人間の通信の利便性の向上ということにとどまらず、個人が「だれでもいつでもどこでも」、不特定多数に対して情報の発信源となりうるという意味でのパーソナルメディアの時代を考えると、パソコン通信はまさにその先駆ける技術であるといえよう。

そういう意味で、市民が入手可能なシステム構成で、しかも高速高機能のホスト局を容易に構築しうるBBSホストシステムを開発、改良することは重要であり、本報告では筆者らの開発したプログラムの技術的な要点について述べた。

また、ネットワークを通して、開発者とユーザとが意見を交換しながら開発を進めるというスタイルは、今後の研究開発のひとつの方向性をも示唆しているといえなくもない。そうした意味においても、ネットワークを通して多くの意見と激励を与えてくれた市民の皆様に感謝する。

なお、筆者らの開発したホストプログラムは、NIFTY-Serveのライブラリ(FBSS)にフリーソフトとしてアップロードしてあるので、だれでもがダウンロードして使用することが可能である。

参考文献

- 1) 財団法人ニューメディア開発協会：報道資料 平成5年度「全国パソコンネット局実態調査」の結果（1993）。
- 2) 落合正幸：最新ホストプログラム情報，パソコン通信，5-8，pp.108-109（1993）。
- 3) 渕一博：研究基盤の再整備，コンピュータソフトウェア，10-5，pp.1-2（1993）。
- 4) SBA 56403：TPV 4>コルーチンとsetjmp/longjmp MASSAN，PC-VAN，PIG，パスカル情報（1988）。
- 5) Perin, C., : Electronic Social Fields in Bureaucracies, Communications of the ACM, 34-12, pp 75-82 (1991).
- 6) 水澤純一：「通信のパーソナル化」について，信学誌，75-8，pp.845-854（1992）。
- 7) 水町守志他：A. 通信のパーソナル化と社会環境，平成5年度電気・情報関連学会連合大会講演論文集，pp.1-64（1993）。