

ネットワーク志向型オペレーティングシステムの マイクロコンピュータへの適用と評価

斎藤梅朗，山田耕一，峯岸厚司（電気通信大学）

1. はじめに

D-net（仮称）は昭和50年から電通大で始められたコンピュータ・コミュニケーション・ネットワークプロジェクトの名称である。電通大には研究用共通計算機システム（リモートステーションを兼ねる）のほか各研究室には研究用個別システムが実験、制御などに使われているが、近年研究内容の質的变化に伴い、画像処理、波形解析、パターン処理を含む研究、数値データベースの建設など計算機システムの資源に対する要求は大きく多様化している。一方、電々公社の新データ網を利用した東大-京大間のネットワーク（N1プロジェクト）¹⁾の研究も進み1~2年後には実用化されすべての学術分野の研究の広域化、共同化が進むことが期待される。

D-netはこのような広域情報網への参加と学内の諸計算機システムの有機的結合による効果的利用をめざすものである。しかし、ネットワークの建設は一朝一夕には実現されず多くの向題点をかかえているが、特にネットワーク内で主に資源を保有するホスト計算機（以下HOSTと呼ぶ）は異なる種類の場合が多く、そのすべてに同一のアーキテクチャーのネットワーク・コントロール・プログラム（以下NCPと呼ぶ）と接続装置を求めることは不可能である。またこれらを共同開発する場合でもネットワークに参加する研究室自身が手持ちの計算機システムをHOSTとして機能させるために必要な技術を養うには相当の時間を費し、ネットワークの発展が事実上困難となる。

そこで、我々のプロジェクトではこれらの状況を考慮して次のような基本方針を決めた。

- (1) 異種間のパケット交換方式のネットワークである。
- (2) ホスト計算機の費用が極めて安い → マイクロコンピュータの利用。
- (3) 異種間の共通オペレーティングシステムの開発。
- (4) 各種目的のために共通仕様マイクロコンピュータの開発。
- (5) HDLC手順をネットワーク標準接続インターフェースに採用。²⁾

本報告では、D-netプロジェクトの一部として開発したネットワーク志向型オペレーティングシステム（NOOS）のマイクロコンピュータへ適用例と性能評価について述べ、ネットワーク処理を志向する場合のシステムの諸条件、特にマイクロプロセッサの命令セット、割込み機能などを検討する。

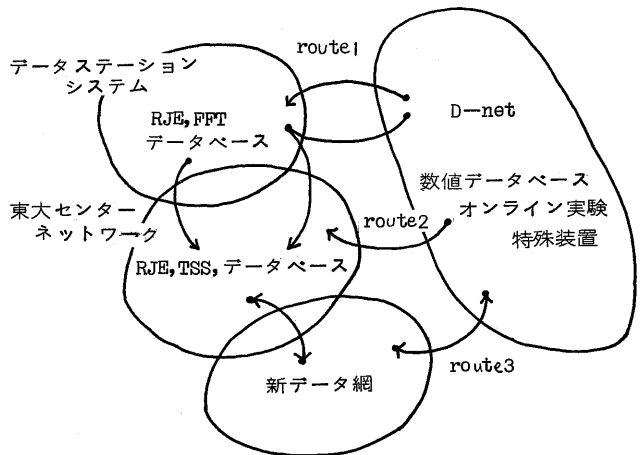


図.1 D-netと外部ネットの関連

2. D-netの概要

D-netと外部ネットワークとの関連を図1に示す。D-netでは大きな処理能力をもったHOSTをもたないため大きな資源を保有する外部ネットワークと密接に通信し利用する反面、外部ネットワークでは得られない資源を提供することが可能となる。構成は図2に示すような分散型ネットワークで星形構造である。D-netは1つのメッセージ交換処理装置（以下IMPと呼ぶ）と複数のHOSTがHDLCの分岐回線上に接続されている。HOSTにはミニコンピュータ、マイクロコンピュータが使われ、その一部にNOOSが適用されている。

CCPはHDLC分岐回線を制御するマイクロコンピュータである。IMPはCCPを通してHOST向のメッセージ蓄積交換をするがHOST1としての機能も持ち外部ネットワークとインターフェースする。HOST2～5は研究室の実験設備の計測制御とネットワーク処理をおこなう。HOST6はネットワーク接続の機能を持たない簡易端末を制御するための専用HOSTである。HOST7は網内の複数台のTSS端末を1200bps加入電話回線を通して多重利用するメッセージ交換専用HOSTである。

網内の各HOSTは互に通信することができると同様にHOST1から外部ネットワークに対してRJEとFFT（ファイル転送）のサービスが受けられる。外部ネットワークとの3つのルート（予定）を接続でき、ルート1は4800bps SYN同期方式の専用線で、ルート2は1200bps加入電話回線およびルート3は新データ網に直接接続する。

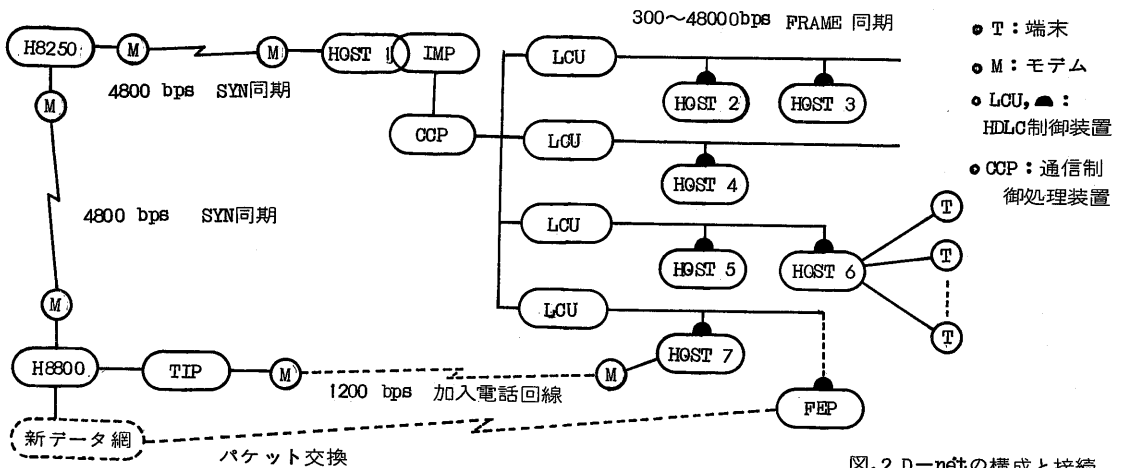


図.2 D-netの構成と接続

3. ネットワーク志向型オペレーティングシステムの条件

コンピュータ・コミュニケーション・ネットワークでは物理的に張りめぐらされた通信路上にメッセージを通すための論理的パスを形成する。パスはいくつかのパイプからなりD-netではHDLC分岐回線を通してIMPに接続するためのパイプ1と特定のHOSTとIMPがセグメント単位でメッセージを送るためのパイプ2がある。

ネットワークは、HOST、IMP、端末および回線等の異なる要素からなる複合体であるから諸要素間の相互関係を明らかにするために階層構造をもったプロトコルを適用する。D-netのプロトコルとプログラムの関係を図3に示す。低位プロトコルとしてIMP-HOST、HOST-HOSTがあり高位プロトコルにはリモートバッチプロトコル（RBP）、ファイル転送プロトコル（FTP）がある。

NCPIは計算機システムがネットワークに参加するために必要な低位プロトコル

を制御する。NCPはオペレーティングシステムの援助で複数のユーザプロセスと通信しメッセージの送受信をおこなうことができる。更にユーザプロセスに対して他のHOSTと通信する手段としてサービスコマンドを提供する。たとえば、ジョブの起

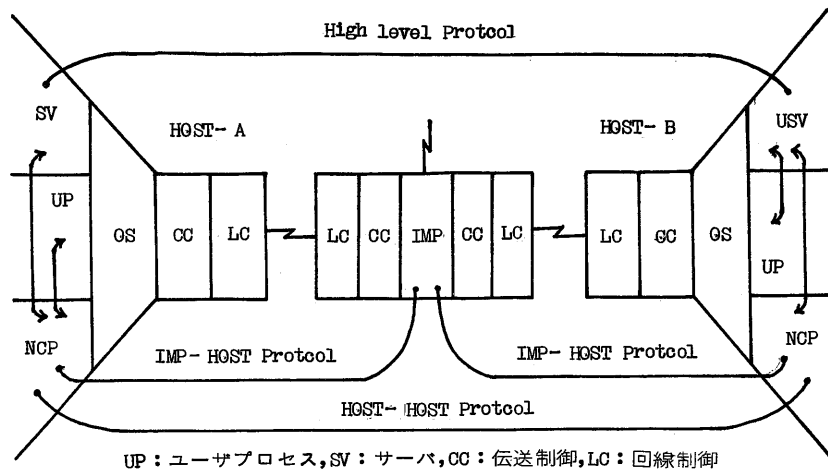


図.3 D-netのプロトコルとプログラム構成

動、資源の確保、データの転送などがある。このようにNCPはネットワークを構成する上で最も重要なプログラムであり、計算機システムがNCPを実現できるようなシステム環境をもつことが絶対条件となる³⁾。したがってNOSは次に掲げる条件を満たさなければならない。

- (1) 多重処理機能があること。
- (2) NCPとユーザプロセスの間に通信手段が確立されていること。
- (3) NCPからプロセスの起動、停止、強制終了などおこなう制御手段が確立されていること。
- (4) 資源が動的に割当て可能なこと。
- (5) ネットワーク接続装置およびその制御プログラムがビットストリングの透過性をもっていること。

4. NOSの方式

NOSの機能を列記すると次のようになる。

- (1) 4レベルのシステムタスク、4レベルのユーザタスクをもつ多重処理機能。
- (2) プロセスの起動と停止、抑止、強制終了の機能。
- (3) プロセスの優先処理機能。
- (4) 入出力装置の並行処理機能。
- (5) 資源割当ての機能。
- (6) 操作員との会話機能。
- (7) プロセス間通信機能。

4.1 タスク構造と管理

表1にしめす8種類のタスクをNOSでは定義している。T₀~T₃はシステムタスク、T₄~T₇はユーザタスクと呼び、T₀~T₂は割込みによって起動し、T₃~T₇はプロセスとして起動されたのちタスクスケジューラによって優先処理される。T₃はシステムタスクであるが、プロセスとして起動される特異なタスクで操作員がコンソール入出力装置

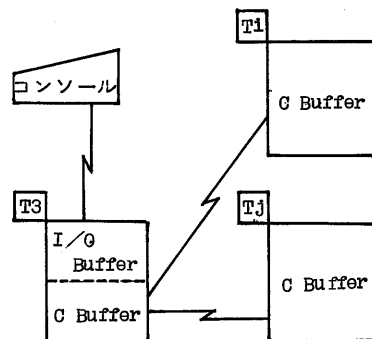


図.4 コンソールタスク T3 の動作

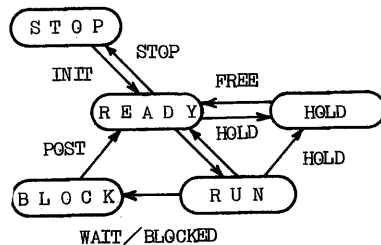
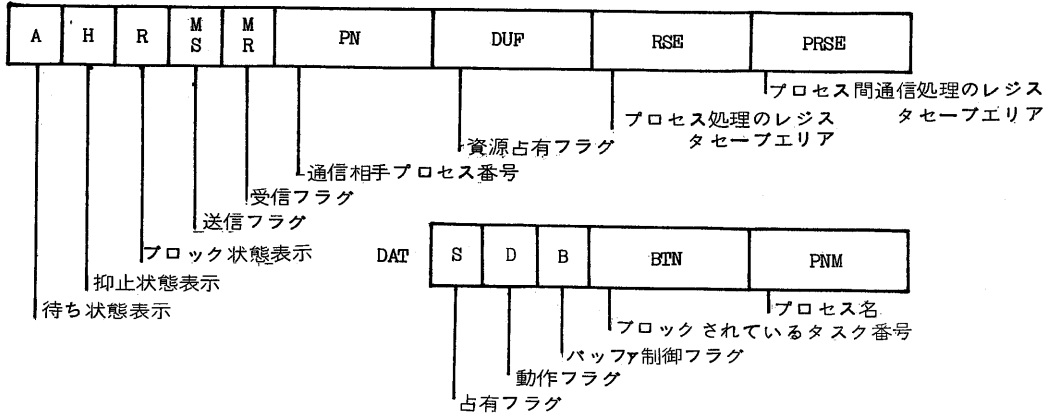


図.5 タスクの状態遷移

TCB (task control block)



TASK CONTROL BLOCK の構造

DEVICE ALLOCATION TABLE の構造

図. 6 タスク管理テーブルの構造

で複数のプロセスと会話するために常にコンソール入出力装置を占有し、入出力メッセージをプロセス間通信を用いて他のプロセスに渡す(図4参照)。

NCPの適用方法には①OSの一部として組込む。②ユーザプログラムとして実行。があり適用する計算機システムの仕様によって異なるがNOOSでは仕様変更が比較的容易な後者を採用しT₄で実行することにした。

タスクの実行はタスクスケジューラによってプロセッサの割当てをうけるが、資源の競合、入出力動作完了待ちなどのために図5のような状態遷移をする。停止状態(STOP)のプロセスはINT-SVCによって実行待ち(READY)となりスケジューラされて実行権を得る(RUN)。実行中に優先度の高いタスクが起動(再起動)されると再び実行待ちとなる。あるいは競合、同期待ちが生じると実行権を放棄する(BLOCK)。低レベルのタスクを一時的に優先処理するために高レベルのタスクをHOLD-SVCで実行抑止状態(HOLD)におくことができる。HOLDはタスクがBLOCK状態でなければ任意に設定し、解放はFRR-SVCでおこなう。

タスクの管理はT₀~T₇に図6に示すTCBを各1つ割当てタスクスケジューラがおこなう。A, R, Hの各ビットは図5の状態遷移をあらわしINT-SVCでA=1, H=R=0となる。タスク

タスクレベル	タスク名称	種類	起動条件	終了条件
T ₀	ハードウェアタスク	ST	内部割込み	} 割込み処理の終了
T ₁	SVCタスク	ST	SVC割込み	
T ₂	I/Oタスク	ST	I/O割込み	
T ₃	コンソールタスク	ST	} プロセスの起動	} STOP-SVC
T ₄	ユーザタスク 0	NCP		
T ₅	ユーザタスク 1	} UP		
T ₆	ユーザタスク 2			
T ₇	ユーザタスク 3			

表. 1 NOOSのタスク構成

がスケジューラされ資源待ち、バッファ待ち、入出力待ちなどでブロックされるとR=1, HOLD-SVCでH=1となる。スケジューラアルゴリズムを次に示す。

a. n = 1.

b. if A_n = 0 then go to f.

SP : 実行開始ポインタ。

c. if H_n = 0 then go to f.

NSE : RSEが指す実行開始ポインタ。

d. if R_n = 0 then go to g.

CSE : PRSEが指す実行開始ポインタ。

- e. run T_n ($SP \leftarrow NSE$)
- f. $n = n + 1$, if $n = 8$
then go to a.
else go to b.
- g. if $DUF_{no} = 0$
then go to f.
else run T_n
($SP \leftarrow CSE$)

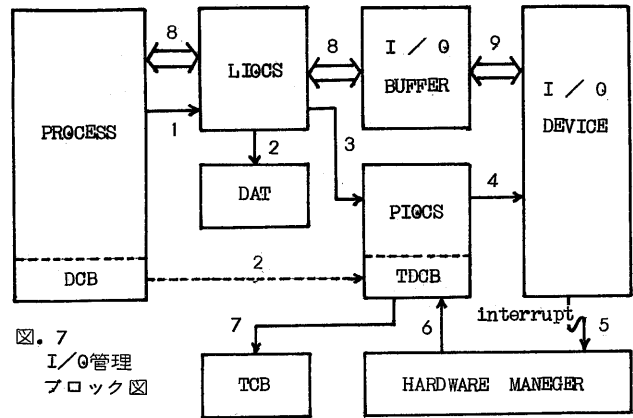


図. 7
I/O管理
ブロック図

4.2 資源管理

NOOSでは常駐タスクのみを扱っているので資源管理では主に周辺装置の割当てに関する制御を図6に示すDATでおこなう。TCBのDUFには各タスクが占有している装置を表示し、DATは装置を占有しているプロセス名、資源待ちのタスク番号(BNT)を保持している。S, D, Bは資源の状態をあらわし、割当てはタスクの優先順におこなう。

プロセス名X, $TN = 4$ のプロセスがLDV(論理装置番号)=1の入力装置に対してOPENマクロを発行し $S=0$ ならばDATの $S=1, D=B=0, BNT=0, PNM=X$ としTCBの $DUF_n=1$ として占有したことを宣言する。資源の解放はCLOSEマクロで、 $DUF=0, DAT$ の $S=0$ としておこなうがそのとき $D \neq B \neq 0$ ならば $D=B=0$ となるまで待つ。また、 $D=B=0$ で $BNT \neq 0$ のときは直ちに最も高いレベルのタスクをポストする。

4.3 入出力管理

入出力装置の種類方式に依存しない統一的な入出力アクセス法をプロセスに提供するために入出力マクロ、OPEN, PUT, GET, READ, WRITE, CLOSEと各装置を直接制御する入出力ハンドラをインターフェースする入出力制御プログラムがある。入出力処理は図7に示すような制御の流れでおこなわれるが処理の詳細はプロセスがOPENマクロの発行時に指示するDCBに制御文字(CC), 制御字数(CW)など指定する。入出力の例を図8に示す。

(1) 入力処理: プロセスからOPENが発行されるとLIQCSはDATを更新、DCBを展開をおこなったのちPIOCSに対して入力起動を要求する。PIOCSは入力装置を起動する。プロセスは任意にGETできるが入力処理が完了してないためブロックされる。制御条件を満足したならばポストし、CLOSEで資源を解放する(1)

(2) 出力処理: DCBの制御条件を満足すデータがPUTされるとPIOCSが出力起動してプロセスと並行処理する。

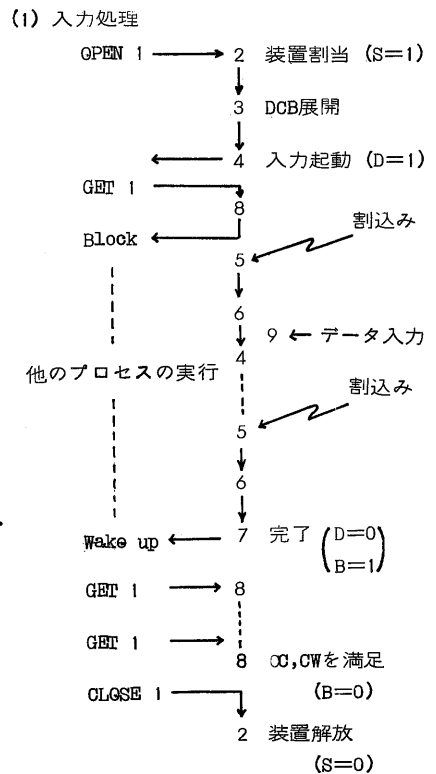


図. 8 入力処理の例

4.4 プロセス間通信

プロセス間通信管理はプロセスが1対1あるいは1対複数のプロセスと通信するための通信路の確保とメッセージの転送および同期のための手段を提供する。プロセス間通信は各プロセスの通信バッファ間に仮想的に作られるプロセス間通信チャネルを用いておこなない、メッセージはこのチャネルを通して他の通信バッファに転送される。従ってプロセス間にどのようなチャネルを作るかが問題となる。NGGSでは次に示すようなプロセス間通信を可能にしている。

- a. プロセスから他の1つのプロセスに対するメッセージの送信。
- b. 1つのプロセスからのメッセージの受信。
- c. 複数のプロセスの1つからのメッセージの受信。
- d. 1回の通信量は最高256バイト。

チャネルの管理は入出力装置のチャネルと同等とみなしTCBのDUFOでおこなないメッセージの転送制御は、MS、MR、PNを用いる。チャネルの確保とメッセージ転送のアクションをおこなすのはプロセス間通信SVCで次のようなものがある。

(1) 送信要求：RTS

送信側プロセスが相手プロセス番号を指定してメッセージの送信をシステムに要求する。

(2) 受信準備完了通知：PRR

受信側プロセスが受信準備完了をシステムに通知する。この時PN=0ならばCタイプの受信となる。

(3) 送信終了：ESM 送信が終了したことをシステムに通知する。

(4) 受信終了：ERM 受信が終了したことをシステムに通知する。

プロセス間通信をおこなうプロセスのプログラムは通常の処理プログラム部分(NPM)の他にプロセス間通信用プログラム部分(CPM)を持たなければならない。プロセス間通信の例を図9に示す。

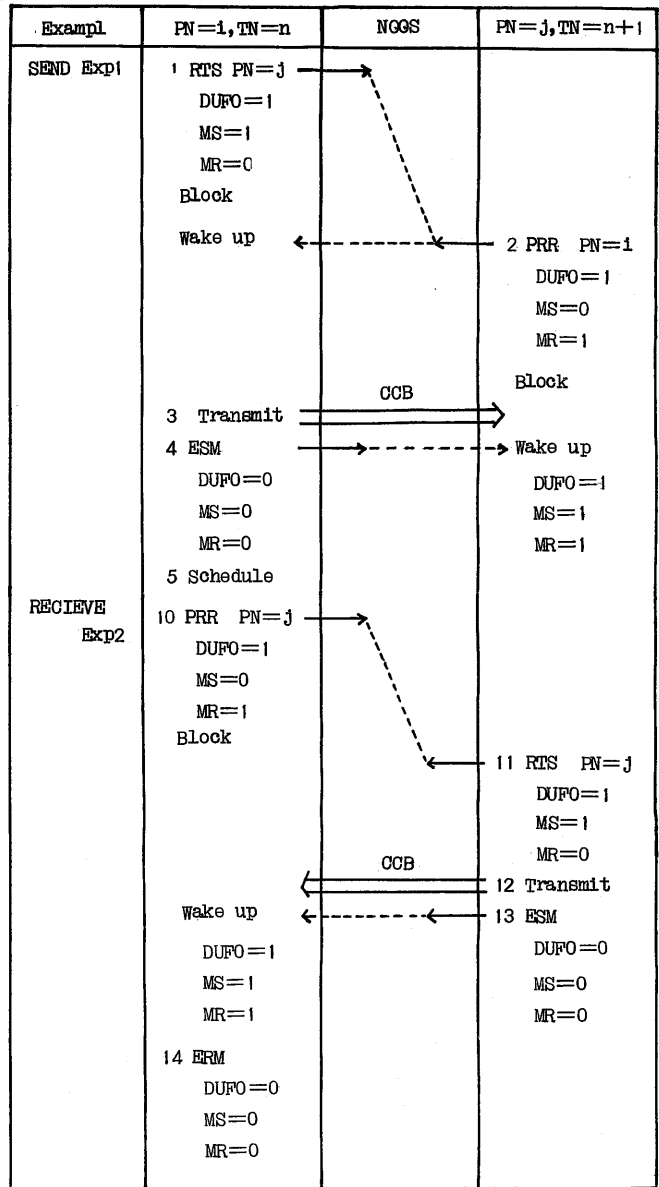


図. 9 プロセス間通信の例

5. NOOSの適用例と評価

NOOSの設計思想をHITAC10(以下H-10と呼ぶ)とSMP/80に適用した場合について述べる。なお、プログラム作成はプログラム歴半年程度の学生がおこなった。

H-10は1命令16ビットのミニコンピュータで平均命令実行時間(t_m)は約4.4 μ sec*でバイト、ビットおよび表操作命令を持たずデータレジスタは16 \times 2ビットである。一方SMP/80はintel 8080系のマイクロプロセッサを使ったマイクロコンピュータで $t_m = 6.9$ *とクロック500msecの場合の4.6*に比較して50%程度遅い。

H-10	4,488 w
共通分	2,097 w
個有分	1,844 w
データ分	547w
SMP/80	5.2KB
中核分	1.8KB
T3分	2.4KB
データ分	1.2KB

5.1 メモリの利用効率

NOOSをH-10, SMP/80に適用した場合のメモリの占有量を表2に示す。両者の占有量の比較はメモリの語構成が異なるため一様に論じることはできないが、プログラム部分についてH-10は1語1命令、SMP/80は1命令平均1.8*バイトと仮定すれば静的ステップ数でH-10が約67%程度大きい。これは8080プロセッサが命令数、レジスタ数でH-10より上まわり、更に処理内容が16ビット演算をほとんど含まないためと思われる。

テーブル、バッファ等のデータ部分はほぼ語長の比に準じたスキエを占めている。

表. 2
メモリの占有量

テーブル、バッファ等のデータ部分はほぼ語長の比に準じたスキエを占めている。

5.2 処理速度

NOOSを構成する各プログラムレベルにおける処理速度を図10, 11, 12に示す。SVCはプロセスがオペレーティングシステムに対してサービスを要求する最も重要な機能であるが両者ともにハード的にこの機能を持たないので不正割り出しあるいはソフト的に実現した。図10に主なSVCの処理所用時間を示す。SMP/80はH-10に比較して1.8~4.5程度遅い。

入出力処理のLIOCS, PIOCSの主なものを図12に示す。入出力データ転送の転送速度を決定するのはPIOCSの応答速度である。図11の結果によるとエコーバックを含まない転送の応答速度はH-10で約800 μ secの10Kビット, SMP/80で約1.6~2.0 msecの4~5 Kビットとなる。従って同じ方式でLCUを制御した場合前者は9,600 bps, 後者は4,800 bpsの通信をプログラムモードで実現できることになる。

また、PIOCSのCPU占有率は低速では2.6~3.2倍、高速ではほぼ同程度でありその値も最大44.4%
*いくつかのプログラムの実行結果から試算。

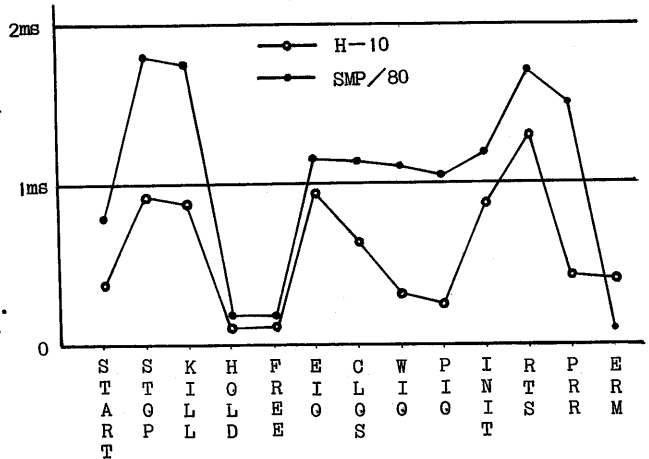


図. 10 スーパーバイザー。コールの処理時間

CPU占有率	H-10	SMP/80
TTY IN	1.9%	5.7%
TTY OUT	1.0%	3.2%
PTR	39.9%	44.4%

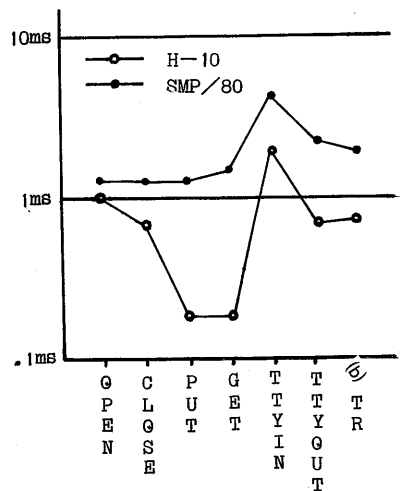


図. 11 入出力管理の処理時間

でマルチプログラミングの環境も保障されている。

操作員コマンドの処理時間を図12に示す。この例では全体的にH-10よりSMP/80が速い。この原因はコマンド処理のかなりの部分が文字処理であるためと思われる。LOADコマンドの差は両者のローダ形式の違いによるものである。

6. まとめ

本報告ではコンピュータ・コミュニケーション・ネットワークに接続するために必要なオペレーティングシステムの諸条件を明らかにした。ネットワーク志向型オペレーティングシステムNOOSをミニコンピュータ、マイクロコンピュータに適用し価格において数倍〜十数倍安いマイクロコンピュータがネットワークのホスト計算機として充分使用可能な性能を持っていることを実例で示した。

しかし、NOOS適用の過程でプロセッサのハードウェア機能が不适当であったり不備な点が指摘された。マイクロプロセッサはLSI技術の発達により多くの機能を持ち汎用化される反面専用化も推進される。そこでオペレーティングシステム専用マイクロプロセッサは次の諸機能を持つことが望ましい。

- (1) 各種内部割り込み機能。
 - (2) 多重レベルの外部割り込み機能。
 - (3) プログラム割り出し機能。
 - (4) ベースレジスタ, 多レジスタ, マルチプログラムカウンタ等の機能。
 - (5) ビット, 表操作機能
 - (6) 多重間接命令
- } レジスタの自動たひひおよび復帰機能。

謝 辞

本研究の基本構想は斎藤が提案しD-netゼミで検討したものであり、討論の過程で有益な助言をいただいた熊本芳郎教授, 岩倉博講師および卒論テーマとして作成に参加した宮野善世君、更に研究推進に大きな御理解と御援助をいただいた土方克法教授に深謝致します。

引用文献

- 1) 島内ほか, "広域大量情報の高次処理" 総合報告, 1975.
- 2) 伊藤, 平塚, "簡易なHDLC回線制御装置", 電子通信学会総合全国大会, No. 1989.
- 3) 情報センター, "コンピュータネットワークJIPNETの研究開発", 44-S100, 1975.

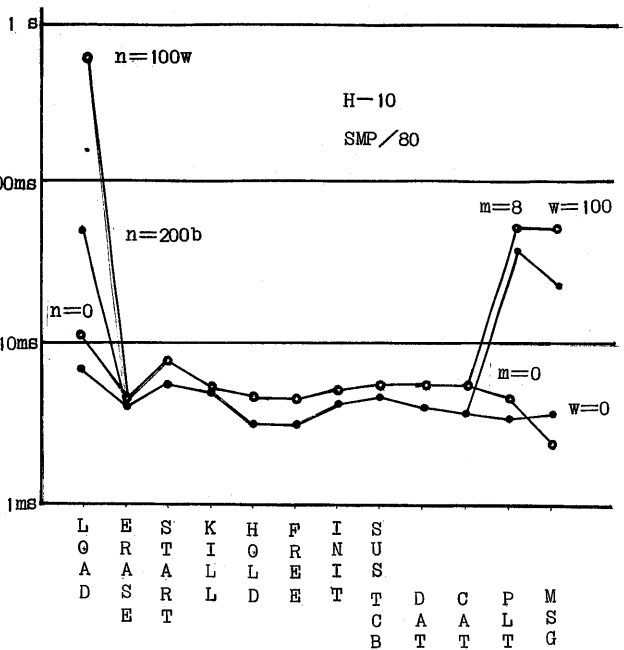


図. 12 操作員コマンドの処理時間