

D2 JIPDECにおけるコンピュータ・ネットワーク・システム

日本情報処理開発センター 山本 欣子
小川 義久
伊藤 哲史

目 次

まえがき

1 ネットワークの概要

1.1 ネットワークの構成

1.2 ネットワークの機能

2 ハードウェア

2.1 IMP (TIP) に必要なハードウェア条件

2.2 インタフェース・アダプタの機能

3 プロトコル

3.1 概 要

3.2 プロトコルの設定

3.2.1 プロセスの起動・終了のプロトコル

3.2.2 プロセス間コミュニケーションのプロトコル

3.2.3 イベントに関するプロトコル

3.3 高位レベルのプロトコル

4 ネットワークのソフトウェア

4.1 概 要

4.2 NCPの概要

4.3 既製OSとのインタフェースと問題点

まえがき

昨今は西を向いても東を向いても、コンピュータ・ネットワークの花ざかりの感がある。

とくに異なった機種を連結し、ハードウェア、ソフトウェアあるいはデータベースを共有するリソース・シェアリング型のネットワークは、米国のARPAネットに刺激されてか、英国、フランス、北欧、汎ヨーロッパあるいは米欧を結ぶ国際的な拡がりて数多くの計画が相ついで打ち出されている。

多額の費用と長年の才月をかけて建設されたARPAネットに世間の目は必しも好意的なものばかりではないようであるが、ともかくも一応の成功と伝えられている。少なくともコンピュータ利用法の新しい面を開拓した功績は大きい。

JIPDECにおけるネットワーク・システムは昭和48年度より数年計画で実現すべく発足したものであるが、もとよりさほど大それた計画によるものではなく、むしろ、実用と実験の半々の要求に沿ったものである。たまたま、国産3社のそれぞれ異なった機種を持ち、各機種毎のハードウェア、ソフトウェアのリソースの限界を打破し、総合したリソース利用の可能性の追求、またファイルやプログラムの互換性の限界に対する新しいアプローチ、あるいは漢字入出力、図形の処理など特殊周辺装置の共用等の長年の希望を充足する手段としてのコンピュータ・コンプレックスの実現と、一方数年間にわたるTSSサービスの体験から、一種類のセントラル・コンピュータにのみしかアクセス出来ぬという制約を解放したいという希望、あるいは最近とみにそのニーズを云々されている分散型データベース・マネジメントの実用化研究の必要性など、諸々の要求が集積した結果としてまず自社内の3機種を結合し、内部における2、3年の実用実験を経て、適当な外部ネットワーク、またはコンピュータ・システムと連結させるとともに、TIPまたはNTS (Network Terminal System) による外からのネットワーク・アクセスも実現させたい。

現在国際的な視野から見るとコンピュータ・ネットワークに関しては日本はかなり立ち遅れているというのが一般の通説である。しかし机の上の勉強や、ネットワーク・システムに対する関心が並々ならぬものであることも自他ともに認めている。今必要なのは、机上の議論だけでなく、それぞれの立場でともかくも手を下し、実行に移す推進役がいくつか出現することであるかも知れない。

なお今回の報告は、システムのねらいを中心に、アウトラインの紹介のみにとどめた。

詳細については後日にゆずりたい。

1 ネットワークの概要

1.1 ネットワークの構成

現在、日本情報処理開発センターにはFACOM 230 / 60 (以下F60)、HITAC 8450 (H-8450)、NEAC 2200-500 (N500)、の国産3機種が設置されており、それぞれ以下のような特徴を有している。

○ F60

高速演算性能、大容量記憶装置、大容量ファイル、リモートバッチおよびTSSサービス、各種オンライン・ソフトのサポート、漢字入出力装置総合の予定。

○ H-8450

グラフィック・コンピュータの結合によるマルチプロセッサ・システムの実現と、汎用グラフィック・ソフトウェアのサポート。

○ N-500

一般バッチデータ処理、マークシート・リーダ結合の予定。

これらの異った3機種をホスト・マシンとして、コンピュータ・ネットワークを構成する為の結合方法として

は、チャネルによる直接結合、高速回線による直接結合、IMP (Interface Message Processor) を介したARPA型の結合等が考えられるが、以下のような理由からIMPを介した結合方法をとることにした。

- ① 異機種ホスト間のインタフェース調整。
- ② ホスト負荷の軽減。
- ③ ネットワーク拡張の容易さ。

また、この結合方法により、ネットワーク・システムを構成するにあたり、以下のような諸点を考慮した。

- ① コンピュータ・ネットワークとしての基本的な機能が、ほぼ全て実験可能な構成であること。例えば、HOST-IMP, IMP-IMP間のメッセージ・ハンドリング, IMP間高速回線の制御, 各種ターミナルの制御, パフォーマンス測定等。
- ② コンピュータ・コンプレックス・システムとして、当センター特有の利用法の実験に適した構成であること。
- ③ 将来他のネットワークあるいはシステムとの結合などの拡張が容易であること。
- ④ 情報ネットワーク・システムのモデルとして、集中型、分散型のデータベース・マネージメント・システムの実験が可能であること。
- ⑤ 従来、各機種で行なってきた作業が、何等変更なしに引きつづき稼働可能であること。

図1にシステム構成を示す。

1.2 ネットワークの機能

JIPDECネットワークは、以下のような利用法の実現を旨としている。

- ① CJP (Cooperated Job Processing) ネットワーク内のホスト群をコンピュータ・コンプレックスとして扱い、複数のホストにユーザの指定するジョブを発生させ、それらのジョブ間の適当な同期をとり、コントロールおよびデータの通信をおこないながら、複数ジョブの協同作業により、あるまとまった仕事を処理するものである。このような利用法はARPAにおいても、MCROSS, RSEXECの例があるが、いずれも同一機種 (PDP 10), 同一OS (TENEX) のもとのCJPである。当センターで現在計画中のものとしては、H-8450のグラフィック・システムと、F60のデータベースと高速演算能力を利用したインタラクティブ・システムがある。
- ② AJD (Automatic Job Disptaching) これは、ネットワーク・ジョブの自動的なホスト選択の機能であり、各ホストの機能、ロード等を考慮し、最適ホストを選び、利用者にホスト・マシンを意識させることなくネットワークに接続されているすべてのホストの機能を利用可能とし、かつロード・バランスを自動的に最適化するものである。

AJDの実現の為に、各ホスト・システムのハードウェア、ソフトウェアの機能を把握しておくためのARPAにおけるNIC (Network Information Center) 的な機能および各ホストの状態、リソースの現状、ロード等を把握しておくNCX (Network Control Center Extended) 的な機能が必要となり、これらの役目を果たす為のNCIC (Network Control Information Center) の機能および管理法について検討を進めている。

③ TNUS

前述のCJPおよびAJDの組合せによる拡張的な機能で、ネットワーク利用者ジョブが自動的に複数個のタスクに分割され、各ホストに分配され、Cooperated Jobとして処理される形態である。このような機能を果たすためには、同時並行処理のタスク・オリエンテッドなジョブ記述に適した新しいプログラミング・シス

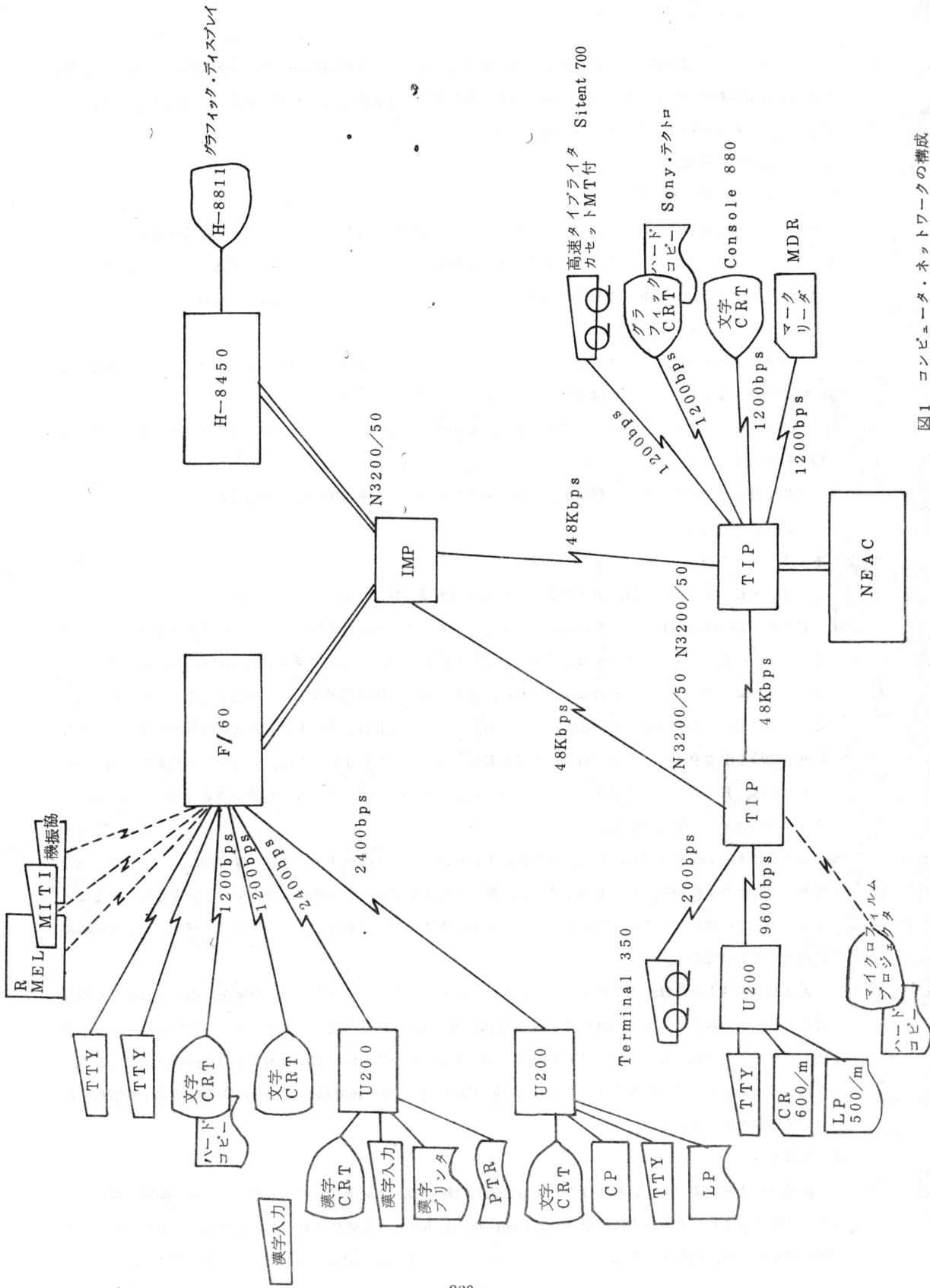


図1 コンピューター・ネットワークの構成

テムや、ネットワークの各ホストに自動分配するネットワーク・ローダ等が必要となる。

④ R D A S (Remote Data-base Accessing System) Distributed Data-base の利用をおこなうために、他のホストのファイルにアクセスする機能の実現を考えている。この機能により、各ホストのプロセスはネットワークに散在するデータ、ベースに自由にアクセスすることが可能となる。この為には、汎用のデータ管理システムの存在が必要となる。

以上のはか、A R P A ネットワークにおける T E L N E T の機能および T I P からの各種ホストへのアクセス、特殊周辺装置 (漢字入出力装置、グラフィックディスプレイ、マークシート・リーダ等) の各ホスト・ジョブからの利用等の基本機能は当然含まれる。

2 ハードウェア

2.1 I M P (T I P) に必要なハードウェア条件。

リソース・シェアリング コンピュータ・ネットワークにおけるサブネットの能力に最も影響をおよぼすのは、I M P 用コンピュータの性能と回線スピードである。当ネットワークの I M P の機種選定にあたって、ハードウェア性能としては、以下のような項目を考慮した。

- ① サイクル・タイムが速いこと。
- ② 実行命令の実行スピード (リアル・タイムミックス値) が速いこと。
- ③ メモリ容量が大きく、拡張性があること。 (バッファ確保、将来の機能拡張)
- ④ プログラム割込みをはじめ、多様の割込み機能を有すること。
- ⑤ チャンネルによるメモリー^{サイクル}・スティーリングが C P U 実行スピードに与える影響が少ないこと。
- ⑥ 高速回線、低速多種回線の制御機能を合わせ持つこと。
- ⑦ 各種のタイマーを持つこと。
- ⑧ 優先制御機能を持つこと。
- ⑨ メモリープロテクトの融通性。
- ⑩ ビット・キャラクタ処理の融通性。
- ⑪ 信頼性が高いこと。
- ⑫ 価格が安いこと。

図 2 に T I P のハードウェア構成を示す。

2.2 インタフェース・アダプタの機能

I M P と各ホストを結合する為には、特別なインタフェース アダプタが必要となる。このようなインタフェース・アダプタの機能を設定するには、ハードウェアで受持つべきか、ソフトウェアでカバーすべきなのか、両者の最適な機能分担を決定することが最も重要な事である。

また、このシステムのように異なったメーカーの機種同志を結合するには、各メーカーの協力が大きなポイントとなり、いかにしてインタフェースの完全マッチングをとるかが問題である。

このネットワークのインタフェース・アダプタは、次のコマンドを解読、実行する機能を持っている。

A R E A D D A T A (R D)

相手側の発した W D I、W D コマンドによるデータを読み取る。

相手側の発したコマンドが W D I あるいは W D のいずれであるかを知ることができる。

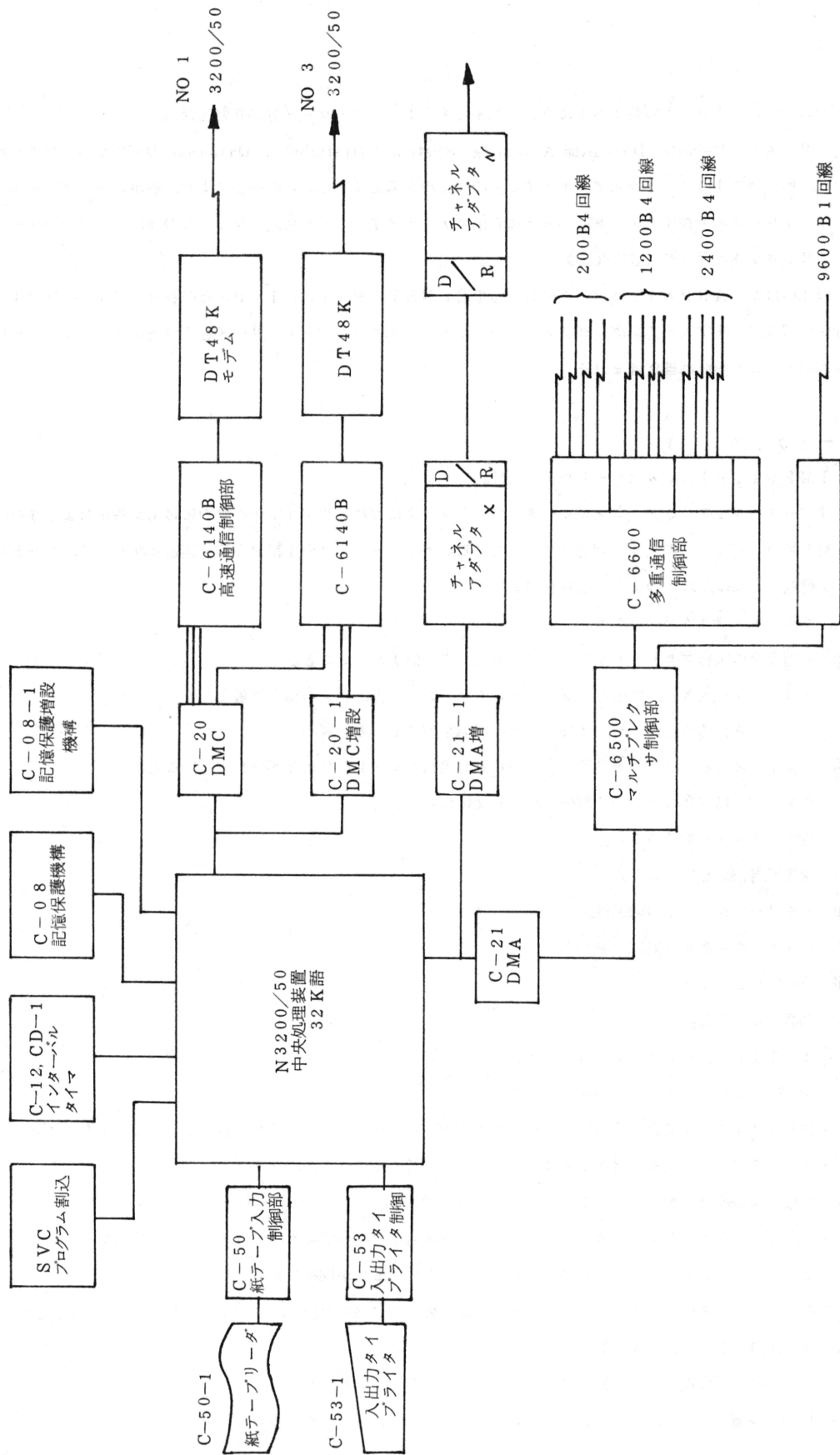


図2 TIPの構成

B WRITE DATA with INTERRUPT (WDI)

相手側CPUに対する割込をともなったWRITEコマンドである。コマンドが相手側から発せられるとデータを転送する。

WDIを発する以前に、相手側がRDを発していた場合には割込みはおこさない。

C WRITE DATA (WD)

RDコマンドが相手側で発せられると、データを転送する。このコマンドは相手CPUに割込みをおこさない。

D NO OPERATION (NOP)

何もおこなわない。

E SENSE

アダプターの状態を聞くコマンドである。

F LOCK PSW (LP)

自分側アダプターのプログラム・スイッチをlockする。アダプターにはプログラム・スイッチが2個あり、この2個がともにunlock状態の時、データを転送することができる。いずれかがlockされている時には、データ転送をせずに異常終了とする。

ロック情報に関しては、SENSEコマンドで聞くことのできるのは相手側のプログラム・スイッチの状態である。

電源が投入された初期状態では、このスイッチはlock状態となっている。

G UNLOCK PSW (ULK)

自分側アダプターのプログラム・スイッチをunlockする。

3 プロトコル

3.1 概要

プロトコル (protocol) はネットワーク上のあらゆる通信に関する規約であって、いくつかのレベルが存在する。

- (1) コミュニケーション・サブネットの効果的な運転を実現するためのIMP-IMPプロトコル。
- (2) HOSTとIMPのつなぎとして、主としてデータの授受の方式を規定するHOST-IMPプロトコル。
- (3) ユーザ・プロセスに対してサービスをするために、NCP同志が会話する方式を規定するHOST-HOSTプロトコル。
- (4) NCPがユーザ・プロセスに対してサービスするコマンドを規定したプロトコル。
- (5) ユーザに対して、機能レベル (ユーティリティ) のサービスを行なう高位のプロトコル。

ここでは、ユーザに関係の深い(4)のプロトコルを中心とし、(5)をどのようにして(4)と関連づけるかについて述べる。

3.2 プロトコルの設定

プロトコルの設定にあたって考えなければならないのは、どのような種類のものを用意する必要があるかということである。

たとえば、バッチ・ジョブの交換であれば、ジョブスタックおよびファイルの引き渡し、出力結果の引き取り

に關係するプロトコルなどが必要である。また、TSSサービスの相互利用であれば、HOSTの呼び出し、ファイルのアクセスおよび端末への出力のためのプロトコルなどが要求される。

JIPDECネットワークの基本的な性格の一面は、このネットワークをコンピュータ・コンプレックスとみなして、CJP, AJD, TNU S, RDASなど実現することにある。このうち、CJPがこのネットワークの核になる部分であり、これを中心としてプロトコルの検討を行なった。

CJPの実現は、とりもなおさず多機種間にまたがるマルチ・タスクを実現させることにほかならない。これに必要となるプロトコルとしては、次の種類のものがある。

- (1) プロセスの起動、終了に関するもの。
- (2) プロセス間のコミュニケーションに関するもの。
- (3) 事象 (event) に関するもの。

以上の3種に関して以下に述べる。

3.2.1 プロセスの起動・終了のプロトコル

プロセスとはHOSTコンピュータにおける管理の単位である。

このプロトコルは、あるプロセスがNCPに依頼して、自己HOSTあるいは他HOSTにおける他のプロセスを発生、強制終了させるもの、および各プロセスが自ずから終了を通知するためのものである。

以下のようなものがある。

- LOADP : プロセスを割り付ける。
DELP : プロセスを消去する。
EXEC P : プロセスを実行する。
RETURNP : 処理の終了を通知する。
ABORTP : プロセスの異常終了を通知する。

3.2.2 プロセス間コミュニケーションのプロトコル

プロセス間コミュニケーション・プロトコルは、ネットワーク・ジョブを形成する2つのプロセス間でデータの授受のためのコネクションの確立、コネクションを通じてのデータの授受、コネクションのつなぎ換えに関するものである。

コネクションとは、2つのプロセスの一方のデータ出力が他方の入力になる関係のことを言い、実際にはそれぞれのプロセスから出されている port - id に対して対応するものを結びつけることにより確立される。

この過程は、通常のファイルアクセスを拡張することにより、ユーザは他プロセスが入出力するデータを自己ファイルと同じレベルでアクセスすることが可能となる。

ここで port の概念を少し説明しておこう。

ネットワークを利用するユーザは、1つの入出力要求単位に対して、一般の File Control Block と類似の性格を持つ入出力情報を記述する。これを port と称する。

これに対する名称が port - id であり、これは全ネットワークにおいてユニークかつグローバルな名称となる。プロセス間のデータ授受はすべて、この port - id により関連づけられる。

図3-1は、その概念図である。

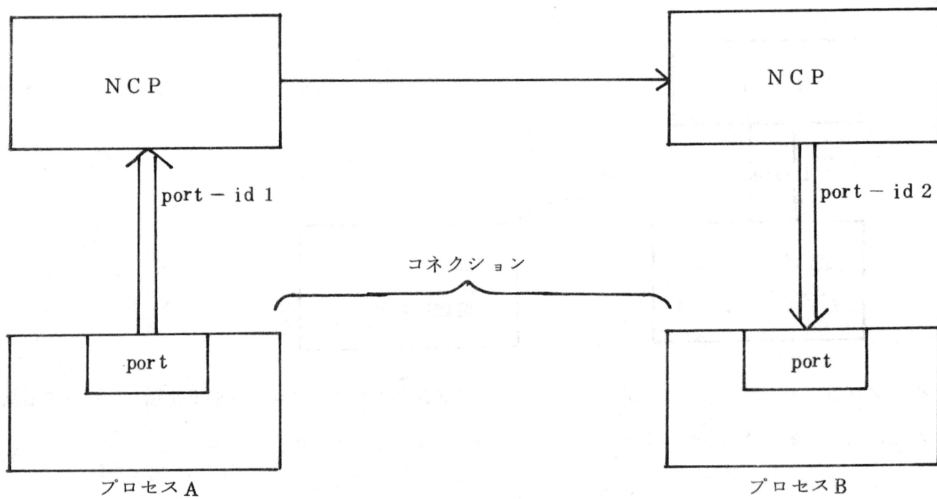


図 3-1 プロセス間コミュニケーション

プロセス間コミュニケーションのプロトコルを以下に示す。

- OPEN : コネクションの確立を要求する。
- READ/WRITE : データの読み書きをおこなう。
- CLOSE : コネクションを解放する。
- CHANGE : 自己プロセスを経由してゆく2つのコネクションを直接結合させる。
- RCON : コネクションを切り換える。
- RFF : ファイルに関する情報を要求する。

3.2.3 イベントに関するプロトコル

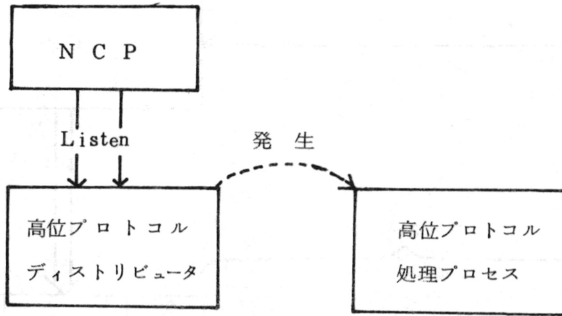
このプロトコルは、プロセスをイベント待ちにおくこと、イベント待ちを解く情報を通知するためのものである。

- NWAIT : 指定したイベントがすべて終了するのを待つ。
- AWAIT : 指定したイベントのいずれかが終了するのを待つ。
- LISTE : コネクションの確立要求を他のプロセスが出したか否かを聞く。
- POST : イベントを通知する。

3.3 高位レベルのプロトコル

今までに述べたプロトコルは、プロセスを主体としたものであるが、実際にユーザーが指定するプロトコルは、より高位である事が望ましい。たとえば、他のホストにあるデータ・ベースを自己ホスト内にあるそれと全く同一手法でアクセスできるRDAS、あるいは他ホストのTSSサービス利用のためのTELENETなどの高位のプロトコルを用意しなければならない。これらのプロトコルを実現するためには、プロセス間のコミュニケーションを使用して、それぞれの役割りを果たすプロセス(高位プロトコル処理プロセス)を予め作りあげておき、これに対してコネクションを確立することにより可能となる。

[例]



高位プロトコル・ディストリビュータは、LISTENによりコネクション要求を知り、対応する高位プロトコル処理プロセスを発生させ、これと要求元との間にコネクションを確立させる。

4 ネットワークのソフトウェア

10人年、1回に1回order

4.1 概要

ネットワークのソフトウェアは、HOST中においてネットワークに関するプロセス全般に関する管理をおこなうNCPと、IMP中においてデータ・コミュニケーションの一切を管理するMCP (Message Control Program) とがあるが、ここではNCPについて述べる。

4.2 NCPの概要

NCPは既製のOSの下、あるいは既製のOSの一部となって、ネットワーク・ジョブに対して、そのHOSTの全機能をサービスすると同時に他のNCPと連絡をとり、他HOSTの機能も使用可能とするためのプログラムである。

NCPは前章で述べたプロトコルのサービスをはじめ、次の諸機能を果たす。

- (1) プロセスの起動終了などを含むプロセスの管理。
- (2) プロセス間コミュニケーション。
- (3) プロセス間の状態情報の管理。
- (4) データ・フローのコントロール。
- (5) HOSTのリソースの使用状況の把握。

このうち(1)~(4)は、プロトコルをサービスするためのものであり、(4)はユーザ・プロセスが無制限にデータをネットワーク上に流すことを防御するための機構である。

リソースの使用状況の把握は、AJD、TNU Sなどを実現する場合に必要な機能である。

すなわち、NCICがネットワーク全体の動的情報を収集管理し、最新の情報にもとづいてJob dispatchを行なうためのものである。

現段階としては、次の情報を収集しておくこととする。

- (1) ジョブの個数に関して
 - 実行中のジョブ個数
 - 実行待ちのジョブ個数
 - NCP起動余裕ジョブ個数

- その他
- (2) メモリーの使用状況
 - 実行中のジョブが使用している領域の和
 - メモリー余裕
- (3) CPUの使用状況
 - アイドルタスクに渡っている時間を一定時間々隔ごとに求める
- (4) ファイルの使用状況
 - TSS専用ファイルの余裕
 - ワーク・ファイルの余裕

4.3 既製OSとのインタフェースと問題点

プロトコルは、プロセスの起動および監視、プロセス間コミュニケーションなどの基本的思想に統一した。これを実現するためには既製OSが、次の条件を満たせばよい。

- (1) インタフェース・アダプタに対してのアクセスが可能であること。
- (2) プロセスの発生、終了および監視が可能であること。
- (3) プロセスに必要なファイルの割り付けが可能であること。
- (4) NCP-プロセス間の連絡が可能であること。

(5) 既製のプログラムのNCPの下で動くこと

これらの項目については、基本的には前記3ホストの既製のOSにおいてはいづれも可能である。しかしながら処理手段が複雑かつ時間がかかるもの、あるいは一部標準OSの変更の必要などがおこり、やゝ問題もある。

例えば、OS自身が4.2で述べた様な、自己のシステム自身の余裕を知ることが困難な場合がある。

これらの機能が実現可能であれば、新たに作成するプログラムに関しては、CJPの目的を達することができる。

しかしながら、既製のプログラムあるいはサービスをどのような形でネットワークにあてはめていくかという問題が残る。

すなわち、TSSのファシリティを持つHOSTにおいては、擬似端末の役割りを果たすことができるプロセスが存在できること。たとえば、当センターのネットワークHOSTにおいては、FACOM230-60がこれにあたり、TCP(Terminal Control Program)をそのプロセスとする。

また、通常のプロセス、例えばランゲージ・プロセッサでは(図4-1)が通常の処理形態であるが、これをネットワークの他HOSTから使用する場合は(図4-2)となる。

ここでランゲージ・プロセッサを修正することなく、ネットワークで使用可能にするためには、プロセス間コミュニケーションを一般ファイル・アクセス方法と同一にする必要がある。そしてネットワークに対するアクセスであるか、ファイルに対するアクセスであるかの判断は、ファイル定義のコントロール・カードにより指定される方針である。

NCPの機能は、4.2で述べたように基本的にはOSの一部と見なされるものである。一般に既製のOS(通常はコンピュータ・メーカが作成する)の一部にユーザが何等かの変更、修正を行なう事は、かなり難しい。1つには、OS内の各モジュールのインタフェースが明らかにされにくいこと。もう1つは、非標準品となるため、その後のメーカのサポートが期待出来ぬことなどである。

NCPの様な機能をユーザが実現する場合に望ましいOSの設計思想としては、スーパーバイザ(Supervisor)

とユーザ・プログラムの中間段階にユーザレベルのOSがつけ加えられ、さほどオーバーヘッドが増大しない形でインタフェースが設計されている事が望ましい。

しかし完全なユーザ・モードのジョブとして作成する場合には、ジョブの起動、ジョブ間の連絡、ジョブ・テーブルの参照など、種々のマイクロ・サービスを行なわなければならない。しかしオーバーヘッドの増大、あるいはネットワーク・ジョブを一般ファイルと同一形式でアクセスする場合にやむ問題がおこる。

いずれにしろ、OS内部のコントロール受け渡しのインタフェースが明らかになっていることが必要であり、またNCPの影響により既製の各機種用のジョブに過度のオーバーヘッドの増大や、メモリスペースの圧迫を引きおこすことは極力さげねばならない。

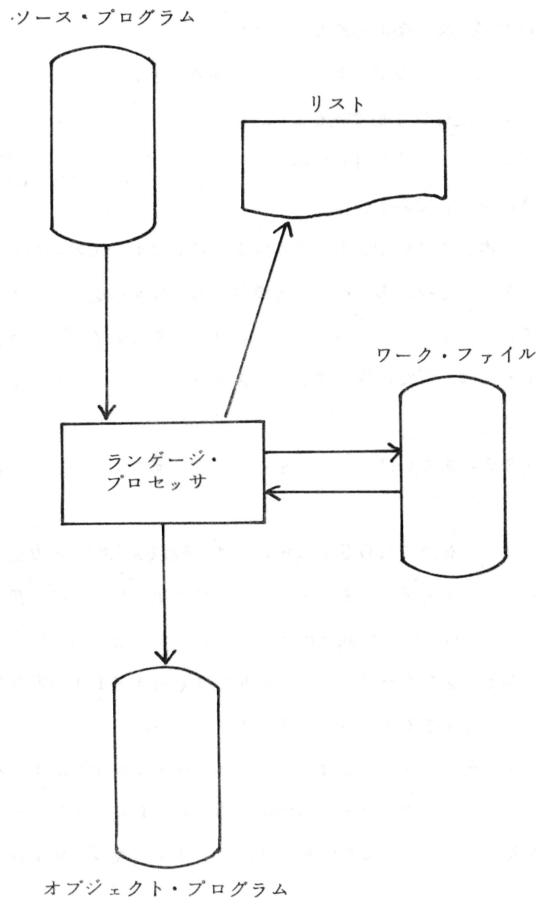


図4.1 ランゲージ・プロセッサの処理形態

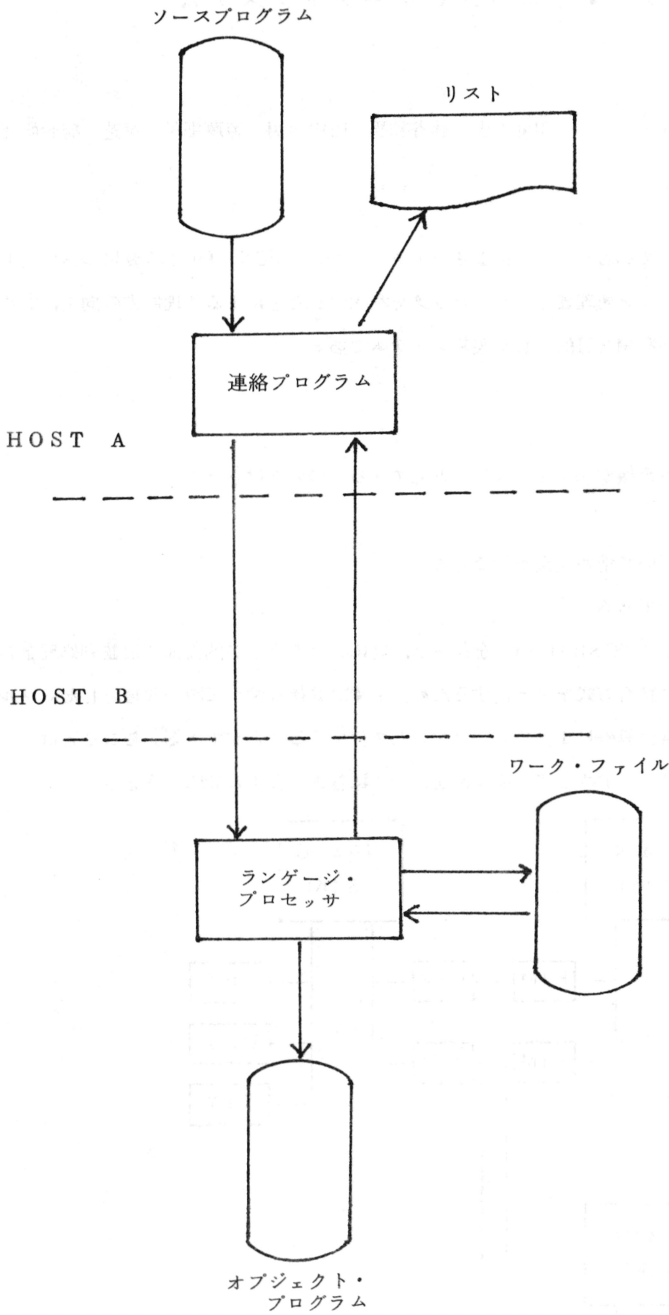


図 4.2 ランゲージ・プロセッサのネットワークにおける使用

本 PDF ファイルは 1965 年発行の「第 6 回プログラミング—シンポジウム報告集」をスキャンし、項目ごとに整理して、情報処理学会電子図書館「情報学広場」に掲載するものです。

この出版物は情報処理学会への著作権譲渡がなされていませんが、情報処理学会公式 Web サイトの https://www.ipsj.or.jp/topics/Past_reports.html に下記「過去のプログラミング・シンポジウム報告集の利用許諾について」を掲載して、権利者の検索をおこないました。そのうえで同意をいただいたもの、お申し出のなかったものを掲載しています。

過去のプログラミング・シンポジウム報告集の利用許諾について

情報処理学会発行の出版物著作権は平成 12 年から情報処理学会著作権規程に従い、学会に帰属することになっています。

プログラミング・シンポジウムの報告集は、情報処理学会と設立の事情が異なるため、この改訂がシンポジウム内部で徹底しておらず、情報処理学会の他の出版物が情報学広場 (=情報処理学会電子図書館) で公開されているにも拘らず、古い報告集には公開されていないものが少からずありました。

プログラミング・シンポジウムは昭和 59 年に情報処理学会の一部門になりましたが、それ以前の報告集も含め、この度学会の他の出版物と同様の扱いにしたいと考えます。過去のすべての報告集の論文について、著作権者（論文を執筆された故人の相続人）を探し出して利用許諾に関する同意を頂くことは困難ですので、一定期間の権利者搜索の努力をしたうえで、著作権者が見つからない場合も論文を情報学広場に掲載させていただきたいと思えます。その後、著作権者が発見され、情報学広場への掲載の継続に同意が得られなかった場合には、当該論文については、掲載を停止致します。

この措置にご意見のある方は、プログラミング・シンポジウムの辻尚史運営委員長 (tsuji@math.s.chiba-u.ac.jp) までお申し出ください。

加えて、著作権者について情報をお持ちの方は事務局まで情報をお寄せくださいますようお願い申し上げます。

期間：2020 年 12 月 18 日～2021 年 3 月 19 日

掲載日：2020 年 12 月 18 日

プログラミング・シンポジウム委員会

情報処理学会著作権規程

<https://www.ipsj.or.jp/copyright/ronbun/copyright.html>