

インテリジェント・コンソール†

—OS の機能拡張の一方法—

有田 五 次 郎††

計算機システムの機能は OS の機能によって決定される。システムに新しい機能を加えるためには OS の拡張、変更が必要となる。複雑、尨大な現在の OS においては、このことは計算機メーカにとっても困難なことであり、ユーザにとっては不可能に近い。これを解決する 1 つの方法としてプロセス間通信を OS の基本機能とし、OS を拡張可能に作成する方法がある。しかし現在の大部分の OS はこの機能を用意していない。ここに示すシステムは従来の OS が持っているオペレータとの通信機能を利用し、外付けのハードウェアとソフトウェアによって OS の機能を拡張するもので、OS の汎用ユーザインタフェースである。

インテリジェント・コンソール (INC) は計算機とオペレータコンソールの間にそう入されたマイクロコンピュータシステムで、ホスト計算機の拡張 OS として動作する。INC を用いることにより OS に手を加えることなくユーザジョブレベルで OS の提供していない機能を簡単に実現することができる。本論文では INC の概念と構成を示し、応用例として実験を行った、ジョブ間通信システム、端末制御システム、リモートファイルアクセスシステムについて報告する。

1. はじめに

計算機システムの機能はオペレーティングシステム (以下 OS と書く) の機能によって決定される。システムに新しい機能を追加するためには、OS の変更・改良が必要となる。十分大きな機能を持った大型の OS では、サブモニタ等の形式で機能の追加が可能になっているが、機能追加がほんとうに必要な中・小型の OS ではこのような手段が不十分で、OS を改造する以外に方法がない。OS モジュールの作成や OS の改造には制御プログラムの内部仕様に関する知識を必要とし、プログラムの作成も一般には高水準言語が使用できないため複雑で作業量も多く、デバッグ・テストは困難で、保守性・信頼性にも問題がありユーザがこれを行うことはほとんど不可能である。既存のシステムに対する非標準入出力装置の接続や、回線結合以外の異機種間の計算機間接合が困難な理由はハードウェア上の問題ではなく、ほとんどの場合はソフトウェア上の問題である。

OS を拡張可能にする 1 つの方法は、ユーザジョブレベルで制御プログラムを書く機能を OS に持たせることである。OS モジュールをユーザジョブとして作成できれば高水準言語の使用により、プログラムの作成、機能の追加・変更、システムの試験・保守等を非常に簡単に行うことができる。このためにはユーザジ

ョブと制御プログラム間及びユーザジョブ同士の同期と通信の手段が必要となる。

現在のほとんどの OS はメモリ中の表 (事象制御表) を媒介とするプロセス間の同期を基本操作とした表駆動型の機構をとっており、プロセス間の通信手段は陽に準備されていない。これに対し Brinch Hansen はプロセス間の通信機能を OS の基本操作とするメッセージ駆動型の OS の構成を提案している¹⁾。メッセージ授受の機能は同期機能を含んでおり、任意のプロセス間でメッセージの交換が可能であればこの機能だけを用いて OS を構成することが可能である。このようなシステムではインタフェースの切り分けが簡単で明確になり、プログラムの独立性、柔軟性が増す。

ここに述べるシステムは従来の OS が持っているオペレータとの通信機能を利用して、外付けのハードウェアとソフトウェアにより任意のプロセス間の通信機能を実現し、OS の機能を拡張しようとするものである。大型の OS ではこのシステムで実現できる機能のかなりの部分を標準として持っている。しかし中・小型のシステムの上で OS に手を加えずに機能拡張を行いたい場合や、大型のシステムでも非標準の機能を簡単に追加したい場合等にはこの方式はかなり有効であると思われる。

2. システムの概念と動作原理

図 1 にシステムの構成を示す。INC (INtelligent Console) はマイクロコンピュータで構成され、計算機とオペレータコンソールの間にあって拡張モニタとして動作する。CCI (Console Control Interfare) は計

† On the INTELLIGENT CONSOLE —An experiment to expand the Operating System using Attached Hardware— by ITSUJIRO ARITA Department of Computer Science and Communication Engineering, Faculty of Engineering, Kyushu University).

†† 九州大学工学部情報工学科

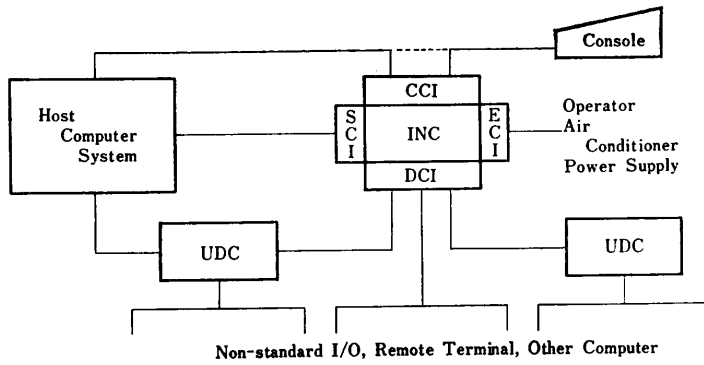


図 1 インテリジェント・コンソールの構成
Fig. 1 Configuration of INC.

算機オペレータコンソールのインタフェース、SCI (System Control Interface)、ECI (Environment Control Interface)、DCI (Device Control Interface) はそれぞれ、システム制御、環境制御、機器制御のインタフェースで必要に応じて用意する。

SCI は CPU や各入出力機器の操作パネルのスイッチの操作、表示ランプの読取り等を行い、ECI は計算機や空調装置の電源スイッチの操作、温度監視、時刻の読取り、オペレータコールのためのブザーのオン/オフ等を行う。これらはほとんど接点信号のみのインタフェースである。DCI は非標準入出力装置の接続のためのインタフェースで、接続装置が多い場合や高速で複雑な機器制御が必要な場合には、この先に UDC (Universal Device Controller) が接続される。

UDC はプログラム制御によって非標準の入出力装置を制御しようとするもので、これ自体マイクロコンピュータシステムであり、高速なデータ転送が必要な場合には I/O チャンネル経由の独立したデータ転送路をホスト計算機との間に持つことができる。

OS はその基本機能としてコンソール装置を介してオペレータと通信を行うためのコンソールサービスを持っている。計算機システムの中では、メモリ、入出力装置、ファイル等、すべてのシステム資源が排他的に管理されているがコンソール装置のみは例外である。すべての制御プログラム及びユーザジョブはコンソール装置を利用してオペレータと通信することができる。

プログラムからの通信はコンソールメッセージとしてコンソール装置に出力され、オペレータからの通信は操作コマンドまたはオペレータ

メッセージとしてコンソール装置から入力される。計算機上の任意のプログラムは、コンソールに出力する通信文の中にコンソールメッセージと区別可能な識別コードを入れることによって、INC 内の拡張モニタと通信することができ、したがって INC を通じて各インタフェースの制御、操作コマンドの発信、他のプログラムとの通信の機能を持つことができる。図 2 は INC によって実現できる通信路を示している。

このようなシステムによって実現できる機能は非常に多いが、主なものには次のよ

うなものがあげられる。

(1) システム制御

自動起動/停止、システム監視、障害自動回復

(2) 環境制御

空調装置の監視・制御、オペレータコール

(3) モニタ機能

ジョブの実行監視・制御、ジョブ統計、制御モジュールの自動発生/消滅

(4) ジョブレベルでの制御プログラムの作成

非標準入出力装置の制御、リモートバッチシステム、会話型システム、計算機間結合

一方、このシステムが OS のコンソール機能を利用していること及び INC がマイクロコンピュータであることによって、以下に示すようないくつかの問題点が存在する。

(1) 通信はすべて文字で行う必要があり 2 進データは 16 進文字にコード化しなければならない。

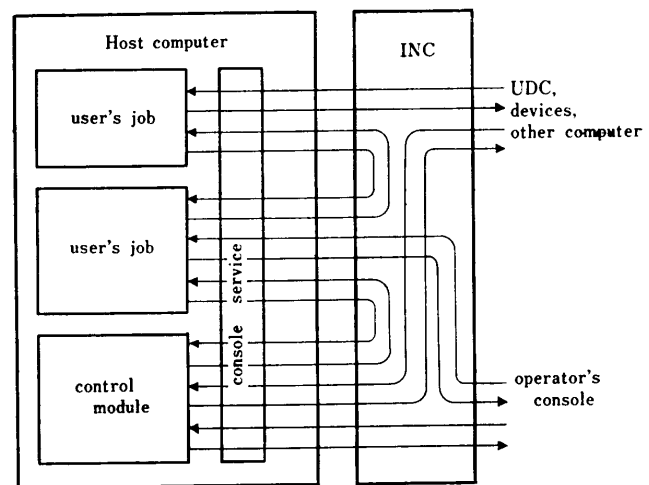


図 2 INC 上のメッセージパス
Fig. 2 Message paths on INC.

(2) コンソールインタフェースは通常低速入出力であり高速のデータ転送は行えない。

(3) 入出力メッセージの最大長が短く、多量のデータ転送のためには入出力動作の回数が多くなる。

(4) すべての操作指令をユーザジョブで使用できるので、OS の保護に十分注意してシステムを作成する必要がある。

3. システム構成と基本通信手順

3.1 ソフトウェアの構造

INC を応用したシステムのソフトウェアは、互いに関連したいくつかのプログラム群から成る。これらはホスト計算機の OS の制御モジュール、ホスト上の制御ジョブ、ホスト上の応用プログラムに組み込まれるサブルーチン群、INC 上の応用プログラム、UDC 上の応用プログラム等である。どの部分でどのような機能を分担するかは応用ごとに異なるが、応用システムの一般的構成は図 3 のようになる。なお図中で左上に印のついたモジュールは個々の応用ごとに用意するプログラム群を意味する。

このシステムは非常に柔軟性を持っており、1つの機能を実現する場合でも種々の構成法をとることができる。たとえば簡単な例として「オペレータ代行システム」を考えてみる。このシステムはコンソール上のシステムメッセージを監視しており、定型メッセージに対して決

められた応答を返したり、処理状況に応じてシステムパラメタ（ジョブの多重度、メモリ領域の配分等）を変更したり、異常検出時にオペレータコールを行ったりする。このようなシステムは、すべての処理を INC 上の応用プログラムで行うように構成することも、INC は単にメッセージの中継をするだけでホスト上のジョブですべての処理を行うように構成することも、特定の処理—たとえば障害検出とオペレータコールのみを INC で行い、他の処理はホスト上のジョブで行うように構成することもできる。どのような構成をとるかは、要求される機能、INC の処理能力、処理速度、プログラムの作りやすさ、システムの操作性・保守性等々を考慮して決定する。

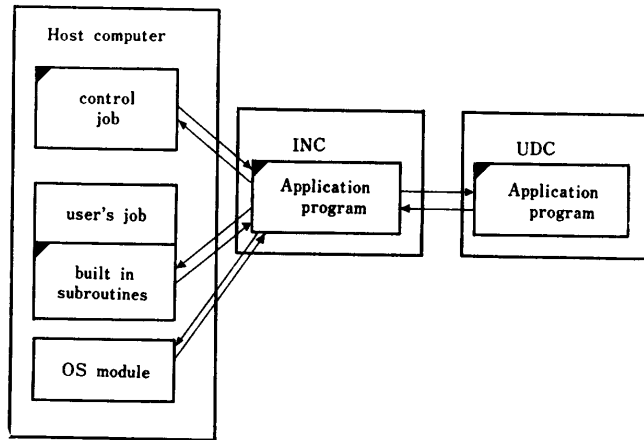


図 3 応用システムの一般的構造
Fig. 3 Structure of application software.

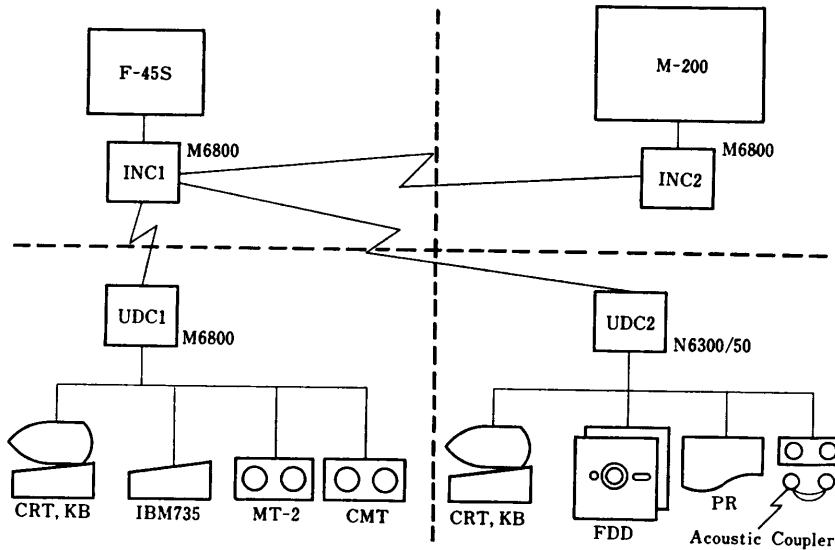


図 4 システム構成
Fig. 4 System configuration.

3.2 実験システムの構成

実験を行ったシステムの全体の構成を図4に示す。F-45 S は情報工学科設置の教育用計算機 FACOM 230-45 S, M-200 は大型計算機センタの共同利用計算機 FACOM M-200 であり, UDC 1 及び UDC 2 は情報工学科に置かれた端末システムである。INC 1, INC 2 及び UDC 1 は 6800 を使用したマイクロコンピュータ, UDC 2 は NEAC N-6300/50 である。F-45 S と M-200 間は構内回線で結ばれており, 48 Kbps の速度でデータ転送ができる。INC 1 はこの他に 8 回線のモデムインタフェースを持っており, 300, 1, 200, 9, 600 bps のいずれかの速度でデータ転送ができる。したがってディスプレイ, プリンタ, X-Y プロッタ等モデムインタフェースを持った市販の端末を自由に接続することができる。

3.3 基本通信手順

F-45 S 及び M-200 はジョブとオペレータとの通信用に WTO (Write To Operator), WTOR (Write to Operator with Reply) マクロを持っている。WTO はメッセージにジョブ識別記号を付けてオペレータコンソールに出力するマクロで, WTOR はメッセージを出力した後にオペレータからの応答を受け取るマクロである。ジョブから INC に対する通信には, 応答を必要としない場合は WTO, 応答を必要とする場合には WTOR を使用する。

INC の上にはシステムメッセージ, オペレータと通信をするための WTO, WTOR によるメッセージ, INC との通信用の WTO, WTOR によるメッセージ, オペレータからの応答メッセージ, UDC からの応答メッセージ等が混在する。これらのメッセージの形式を図5に示す。F-45 S では ID0 は WTO,

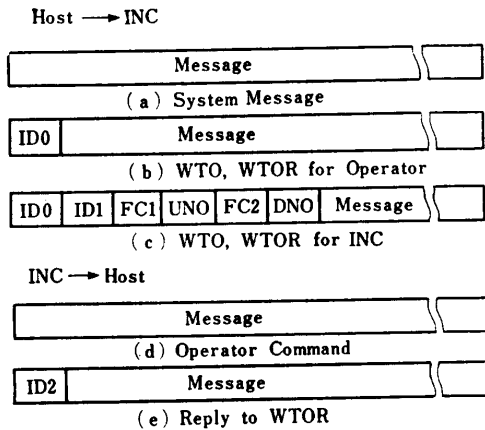
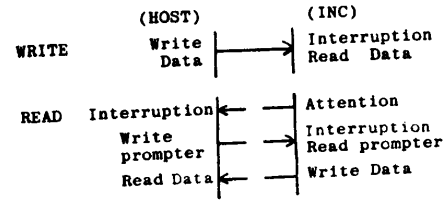
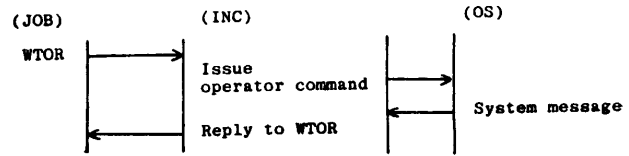


図5 メッセージフォーマット
Fig. 5 Message format.



(a) Basic protocol



(b) Issuing operator command from job

図6 Host-INC 通信手順
Fig. 6 Protocol of Host-INC.

WTOR の各々に対して: job id, ⓐAⓑ job id の形をしている。ID1 はシステムメッセージやジョブからオペレータに対する通常のメッセージに現れない特殊な文字列で, INC 内の拡張モニタに対するメッセージであることを示す。FC1 は INC に対するファンクションコードで, UNO, FC2, DNO はそのパラメタである。たとえば FC1 が 0 であればこのメッセージは UNO で指定された UDC に送られ, DNO で示される装置に対して FC2 で指示された動作を行う。ID2 は A job id の形をしており, OS のコンソールサービスは job id で指定されたジョブにこのメッセージを引き渡す。

ホスト-INC 間の基本通信手順を図6に示す。(a) は OS 中のコンソールサービスの基本手順で, 出力側は WTO または WTOR によるメッセージの出力時に, 入力側は WTOR の応答及び制御モジュールへのコマンド入力時に実行される。ホスト-INC 間の WTO, WTOR による通信では常にホスト側に主導権があり, WTOR が出ていないかぎり INC 側からホスト上のプログラムにメッセージを送ることはできないが, 制御モジュールに対する操作コマンドは常時発信可能である。図6(b)はホスト上のジョブが操作コマンドを発信する場合の手順を示している。

ホスト上のジョブ間, あるいはホスト上のジョブと INC, UDC 上のプログラム間の通信手順は応用ごとに決められる。

4. INC の応用

INC は非常に簡単なシステムであるが, 種々の興味ある応用が考えられる。この章では INC のいくつ

かの応用例を示す。なおここでは実装の詳細については述べない。実装は対象とするホスト計算機の OS に大きく依存し、また INC として使用するマイクロコンピュータの性能によっても異なるが、コンソールインタフェースのハードウェアさえ用意されていれば、どの機種に対しても実装はそれほど困難でない。

4.1 ジョブ間通信

OS の制御モジュールは他の制御モジュールや不特定のユーザジョブとの通信機能を持たなければならない。したがってジョブ間通信の機能はユーザジョブレベルで制御プログラムを作成するための必須の機能である。従来の OS ではジョブを論理的に独立した処理単位として扱っているため、ジョブ間通信の機能を用意していない。しかし INC を利用すると従来の OS に簡単にジョブ間通信の機能を付加することができる。

ジョブが制御モジュールとして動作するために必要なジョブ間通信の機能は次の通りである³⁾。

- (1) 不特定ジョブからのメッセージの受信
- (2) 特定ジョブからのメッセージの受信
- (3) 特定ジョブへのメッセージの送信

これらの機能に加えて、システムマクロ、操作コマンドを使用して OS 中の制御モジュールと通信ができればジョブは制御モジュールとして動作可能となる。

ジョブ間通信システムは IJAM (Inter-Job Access Method) 及び IJCCP (Inter-Job Communication Control Program) から構成される。IJAM はホスト上のユーザプログラムに組み込まれる機械語サブルーチン群でマクロ処理の入口に相当し、IJCCP は INC 上の制御プログラムでマクロ処理の本体に相当する。IJAM は上記の3つの機能に対応する次のサブルーチンから構成されている。

- RCIVA (SJID, RVMSG)
- RCIV (WJID, RVMSG)
- SEND (DJID, SDMSG)

ここで SJID はメッセージを送ってきたジョブのジョブ Id, WJID はメッセージを受け取りたい相手のジョブ Id, DJID はメッセージを送りたい相手のジョブ Id であり, RVMSG, SDMSG はそれぞれ、メッセージ受取り及び送出のためのメッセージバッファである。

これらのサブルーチンは適当なヘッダを作成して WTOR を発信し、IJCCP に処理要求を出してその応答を待つ。

IJCCP は次の機能を持つ。

- (1) 同期

各要求間の同期はランデブ表 (以後 R 表という) によってとられる。R 表にはマクロコード、要求元ジョブコード、相手ジョブコード、到着時間が登録されている。IJCCP は要求が到着するごとにこの表を検索し、対応するものがなければその要求を登録する。

(2) メッセージ転送

対応する要求が R 表にあれば送信側のジョブに回答メッセージを返すことによって送信データを受け取り、受信側のジョブにこれを送出する。このとき対応がとれた要求は R 表から除去される。

(3) 時間監視

IJCCP は一定時間間隔で R 表を走査して、決められた時間を超過した SEND 及び RCIV をこの表から除去し、要求元にタイムアウトのメッセージを返す。これにより相手がないジョブに対する同期が保持される。

ジョブ間通信機能を用いて不特定ジョブにサービスする制御プログラムを構成することができる。制御ジョブは RCIVA を出して不特定ジョブからのサービス要求を待つ。サービスを受けたいジョブは制御ジョブのジョブコードを指定して SEND を発信し、サービス要求の発生を通知すると共に必要なパラメタを送る。サービスの完了は、制御ジョブが処理を完了した後に発信する SEND によって通知され、要求ジョブは RCIV を発信してこれを知ることができる。図 7

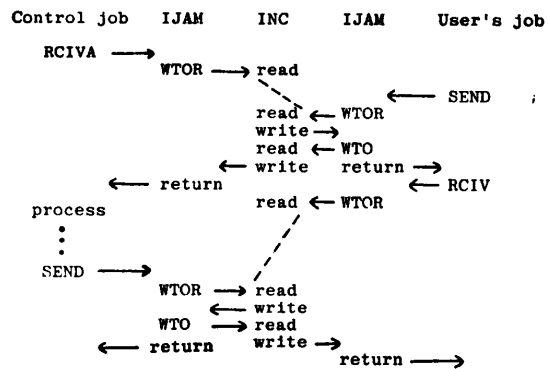


図 7 ジョブ間通信の伝送手順
Fig. 7 Inter-job communication protocol.

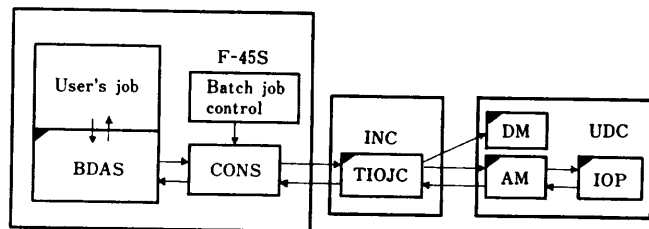


図 8 端末入出力制御システム TIOCS の構成
Fig. 8 Structure of TIOCS.

にジョブ間通信機能を用いて制御ジョブを構成する場合の通信手順の例を示す。

4.2 端末制御システム

UDC に接続された非標準の入出力装置を F-45 S 上から使用するシステム TIOCS (Terminal I/O Control System) の構成を図 8 に示す。TIOCS は BDAS (Basic Device Access Subroutine), TIOJC (Terminal I/O Job Control), TIOCP (Terminal I/O Control program) から構成される。BDAS はホスト上のジョブに組み込まれる機械語サブルーチンでマクロ処理の入口に相当し、TIOJC は INC 上の制御プログラムで、OS のバッチ制御モジュールの拡張部分に相当する。TIOCP は UDC 上の制御プログラムで、マクロ処理の本体に相当し、DM (Device Management), AM (Access Management), IOP (I/O Program) から構成されている。

BDAS に含まれるサブルーチンを以下に示す。

TOPEN (TN, FNAME, BSIZE, DIRC CC)
 TCLOSE (TN)
 TREAD (TN, BADR, LNG, CC)
 TWRITE (TN, BADR, LNG, CC)
 TCONT (TN, CONT, CC)

ここで、TN は UDC 番号と装置番号から成る端末番号、FNAME はファイル装置に対するファイル名、BSIZE はバッファサイズ、DIRC はデータ転送の方向、CC は終了状態であり、BADR, LNG はそれぞれ入出力バッファのアドレス、入出力データの長さである。サブルーチン TCONT は磁気テープ装置の巻戻しやプリンタ装置のホームフィード等機器ごとに異なった制御を行うもので、その内容は CONT で指定される。

BDAS の各サブルーチンは OS 中の制御モジュール CONS (CONsole Servise) の機能である WTO, WTOR を使用して INC 上の TIOJC 経由で DM, AM と通信しながら処理を進める。

TIOJC は TIOCS を使用しているジョブのジョブコードを保持するジョブ表 (以後 J 表という) を持つ。ジョブが TOPEN を発信すると TIOJC は J 表にそのジョブコードを登録し、TCLOSE を発信するとジョブコードを消去する。TIOJC はまたバッチジョブ制御の出すメッセージを監視しており、J 表に登録されているジョブに対してジョブ完了メッセージが出力されると TCLOSE を代行し、J 表からジョブコードを消去する。これにより異常終了等でジョブが入出力装置を解放しないで終了する場合にもシステム側で対

処することができる。

DM は UDC に接続されている装置を登録したデバイス制御表を持ち、これにより装置管理を行う。装置は TOPEN でジョブに割り当てられ TCLOSE で解放される。AM は入出力処理の共通部分で TREAD, TWRITE, TCONT に対応した処理を行い、IOP は各装置に対する機器制御プログラム群である。

4.3 リモートファイル・アクセスシステム

複数の計算機を結合して負荷分散や資源の共用を行ったり、既設の計算機ネットワークに参加したりすることは従来の計算機システムにとってそれほど容易なことではない。しかし INC を利用すると計算機間結合は容易に実現でき、また UDC にハイレベルデータリンク制御手順 (HDLC) を実装することによってパケット交換網を介して計算機ネットワークに接続することも可能となる。ここでは INC を用いた計算機間結合の例を示そう。

異なる計算機上のジョブ間通信の応用である RFAS (Remote File Access System) の構成を図 9 に示す。RFAS は F-45 S のジョブから M-200 のファイルを使用するためのシステムで、M-200 上の制御ジョブ FACP (File Access Control Program) と F-45 S のジョブに組み込まれるサブルーチン BFAS (Basic File Access Subprogram) から成る。FACP F-45 S はこのジョブが発信する M-200 の操作コマンド、START FACP によって起動され、WTO, WTOR で BFAS と通信しながらファイルアクセスに必要な M-200 のデータ管理マクロを発行する。

RFAS は直接アクセスモードと転送モードの 2 つ

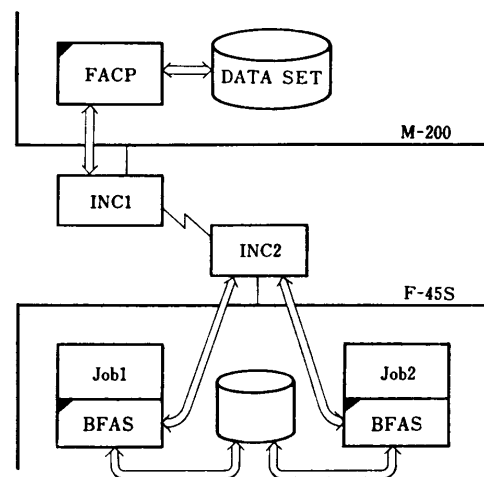


図 9 リモートファイル・アクセスシステムの構造
 Fig. 9 Remote file access system.

の動作モードを持つ。直接アクセスモードではジョブは M-200 のファイルに対して直接 Read/Write ができ、転送モードでは M-200 上のファイルを一旦 F-45 S 側に転送し、Read/Write は F-45 S 上で行われる。

直接アクセスモードで使用可能な BFAS のサブルーチンを以下に示す。

RALLOC (DD, DS, UNIT, DSP 1, DSP 2,
REC, CC)
ROPEN (DD, TYP, CC)
RREAD (DD, BUF, BLK, CC)
RWRITE (DD, BUF, BLK, CC)
RCLOSE (DD)
RFREE (DD)

ここで DD (データセット定義名), DS (データセット名) 等は M-200 の対応データ管理マクロが必要とするパラメタであり、各サブルーチンはこれらを FACP に送り処理を依頼する。

転送モードのためのサブルーチンは次の 2 つである。

RLOAD (AC, DD, CC)
RSTORE AC, DD, CC)

ここで AC, DD はそれぞれ F-45 S, M-200 のファイルのアクセス名で CC は完了状況である。AC に対するファイルはこのジョブのジョブ制御文で、DD に対するファイルは上記の RALLOC で割り当てられていなければならない。RLOAD は M-200 上のファイル DD を F-45 S 上のファイル AC に移し、RSTORE はその逆を行う。

5. おわりに

以上 INC の概念といくつかの応用例について述べた。INC は OS が用意していない非標準のハードウェア、ソフトウェアのインタフェースと考えることができる。INC の応用は多岐にわたり、システム構成の自由度は非常に大きい。ここで述べたいいくつかの応用例について実験した結果の結論は次の通りである。

(1) OS が非標準としている処理に関して INC の利用は非常に有用である。

(2) 機能をかぎれば INC を応用したシステムを構成することはハードウェア、ソフトウェア共に簡単である。しかし多くの機能を 1 つの INC で実現したり汎用性を持ったシステムを構成したりすることは、システム設計、実装の両面でかなりの労力を要し INC の簡便さをそこなう。

(3) メッセージレベルのプロセス間通信機能を持

つことは OS の柔軟性を非常に高める。

(4) OS の標準機能として INC の機能がほしい。

(3), (4) に関しては、自動運転のためコンソールメッセージを OS 内で受け取る手段を持った OS も開発されている⁶⁾。この機能を用いることにより、ここで述べた INC の応用のかなりの部分を実現することができる。このような機能が OS の標準機能となり、利用者に解放されることが望まれる。

なお INC の着想は九大大型計算機センターで開発した夜間無人運転のための警報装置にある²⁾。計算機の自動運転システムや依頼ジョブの状態の問合せに対する自動応答、電話によるジョブの完了通知等、計算機システムの運用管理に関して INC は種々の応用を持っている。これらについては別稿で報告したい。

謝辞 本システムは 1978~80 年にかけて実験を行ったシステムで⁵⁾、この間多くの方々のご協力を得た。ハードウェア、ソフトウェアの製作に関しては、九大工学部末吉敏則助手、当時学生であった塚崎悦生、北尾正則(現 TDA)、兼田彦三(現三菱プレシジョン)、堂込一秀(現富士通)、村上健一郎(現電電公社)の諸君に担当していただいた。また実験期間を通じて大型計算機センターの景川次長、松尾研究開発部長ならびに関係各位、横田耕一氏をはじめ富士通(株)の関係各位には多大の便宜を計っていただいた。なお本論文をまとめるに当り九大工学部牛島和夫教授、東工大木村泉教授に多くのご助言とご鞭撻をいただいた。ここに記して謝意を表したい。

参 考 文 献

- 1) Brinch Hansen P.: The Nucleus of a Multi-programming System., CACM, 13, 4 (1970).
- 2) 有田他: 計算機システム障害警報装置の開発, 九大大型計算機センター広報, 5, 3 (1972).
- 3) 有田他: 複合計算機システムの対称形ソフトウェア, 情報処理学会, プログラミングシンポジウム報告集, 16 (1974).
- 4) 有田他: プログラム制御方式の汎用入出力制御装置とその 2・3 の応用, 信学技報, EC-75-58 (1975).
- 5) 有田他: インテリジェント・コンソールとその応用, 情報処理学会, プログラミングシンポジウム報告集, 20 (1979).
- 6) 藤本他: 無人化を指向した自動運転システム, 情報処理学会, 第 21 回全国大会予稿集, 4 D-5 (1980).

(昭和 56 年 8 月 31 日受付)

(昭和 56 年 11 月 18 日採録)