

インテリジェント コンソールによる計算機間結合

COMPUTER INTERCONNECTION WITH INTELLIGENT CONSOLE

有田 五次郎

Itsujiro ARITA

北尾 正典

Masanori KITAO

堂心 一秀

Kazuhide DOUGOME

九州大学

工学部

Faculty of Engineering

Kyushu University

1. はじめに

計算機システムの機能はOSの機能によって決定される。システムに機能の追加を行おうとする場合、大型のOSではそのための手段を保持している場合が多いが、中・小型OSの場合はそのような手段がないため、機能追加、拡張のために、複雑なOSの内部構造の理解や改造に多くの労力を必要とする。やえに、ユーザが機能追加の必要性を感じても、自らの手で行うことは非常に困難である。

そこで、システムがオペレータとの通信を行なうために保持している、コンソールインタフェースにハードウェア/ソフトウェアを外付けすることにより、OSの機能を拡張するためのシステム「インテリジェント コンソール」(以下INCと称す)を用いることにより、ユーザレベルで拡張機能を用いた種々のアプリケーションが実現できる。

このようなアプリケーションの一つとして、計算機間結合があげられる。今日では、ハードウェア/ソフトウェア等の資源を集約的に利用する形態から、分散・遠隔配置された別のシステムから、センター資源にアクセスする形態への変化は一般的となっており、分散処理システムやネットワークアーキテクチャ等の概念が広まり、製品化され、実働しているシステムが日々見受けられる。こうしたシステムは、機能的には実用性の高いものではあるが、ある程度の規模と機能を備えた計算機システムでなくては

実現が難しい。

INCを用いれば、システムの拡張手段に欠ける中・小型OSの機能強化が比較的手軽に行なえるため、故障等の問題点が多いが、ある程度の機能を実現することが可能となる。本稿では、INCの概念を整理するとともに、計算機間結合の一応用例について報告する。

2. INCの概念

INCの概念を図1に示す。OSのオペレータとの通信機能は、通常はオペレータコマンドとシステムメッセージによりOSとオペレータ間、プログラマからコンソールへのメッセージ出力と入力によりジョブとオペレータ間にメッセージパスを形成している。(2)参照)

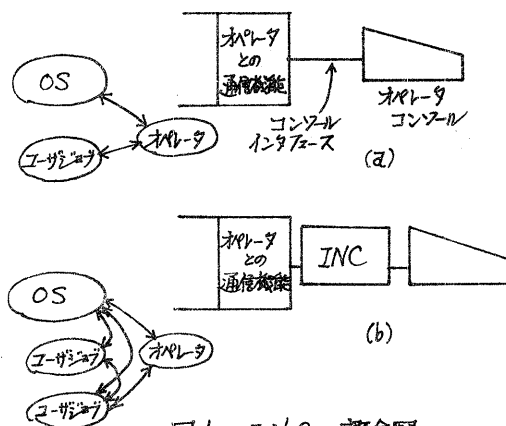


図1 INCの概念図

ミニコンソールインタフェースにINCを挿入することにより、(b)のような多くのパスを追加できる。これらのパスには、ジョブとジョブの通信やジョブと端末、ジョブとOSの通信が可能な。図2にホストとINCの通信の基本手順を示す。

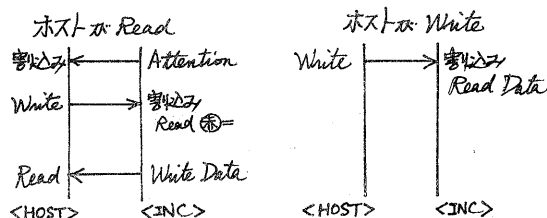


図2 ホスト-INC基本通信手順

この手順を用いてコンソールインタフェースでメッセージのやりとりを行なうわけであるが、これにはオペレータとの通信用の2つのマクロWTO (Write to Operator), WTOR (Write to Operator with Reply) を用いる。いずれもマクロ発行先ジョブId (ジョブコード) をつけた形でメッセージを送り出すが、返答を要求する場合にはWTORを用い、返答先ジョブコードをつけてメッセージを返す。メッセージ先頭に通常のオペレータとの通信と区別するためのIdコードをつけることにより、INCによる拡張機能を利用できる。この機能を用いてプログラムとINCが通信を行なう場合の手順を表1に示す。

表1 WTO, WTOR通信手順

タイプ	内 容	手 順
タイプ1	相手側に情報を送るだけで返答不要な場合 (WTO)	→
タイプ2	返答情報のみが必要な場合 (→は同期用) (WTOR)	→ ←
タイプ3	相手側に情報を送り、返答情報も必要とき (WTOR)	→ ←

ユーザジョブからオペレータコマンドを発信する場合はタイプ3の手順を用い、INCに対してコマンド発信の依頼を行なう。このときの処理手順を図3に示す。

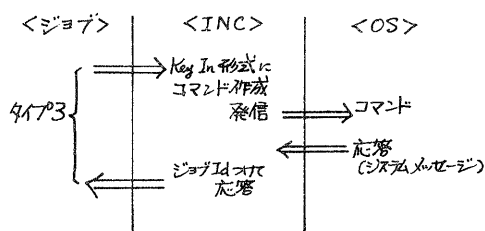


図3 オペレータコマンド発信手順

次に制御ジョブについて述べる。ジョブ間通信やリモートファイルアクセス等で、ジョブの同期をとったり、複数要求に応じたりするジョブを制御ジョブと呼ぶ。これは要求待ち部分、チェック部分、処理部分の3つに大きく分けることができる。要求待ち部分では任意の時点でジョブから送られてくる処理要求を受け取るわけであるが、どのジョブからどんな処理要求が来るかはわからない。ゆえにここはタイプ2の手順でwaitしている。要求を受け取ると、メッセージのチェックを行ない、対応した処理を行なう。その後、処理結果を要求ジョブに返す。この部分はタイプ1の手順でよい。一方、要求ジョブについて見ると、制御ジョブに対して処理要求を出し、結果を受取るのであるからタイプ3手順となる。制御ジョブの構成を図4に、要求ジョブとの通信手順を図5に示す。

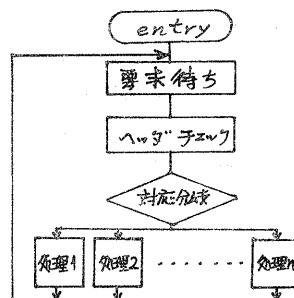


図4 制御ジョブの一般的構成

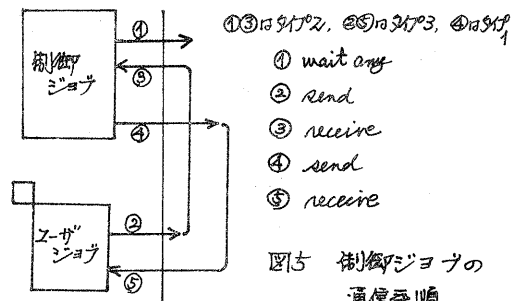


図5 制御ジョブの通信手順

制御ジョブの代表的な例として、ジョブ間通信のためのシステムがあげられる。これは、複数ジョブから同期をとるためのリンク要求や、リンク待ち要求、データの送/受信要求を受け、対応処理を行なうものである。これにより、ジョブ間の同期用にジョブに対してwait/putを行なうため、ロールイン/ロールアウトコマンドを用いたり、通信時の衝突（互いに送信要求を出している）や無視（互いに受信要求を出している）等の監視を行ない、異常状態を各ジョブに通知される。

ジョブ間通信のためにユーザジョブとして必要なのは、リンクを確立すること、リンク後データの送/受信が行えること、そしてリンクを解けること、である。このため、ユーザジョブに用意されている通信用マクロとその呼出形式を次に示す。

(1) 不特定ジョブからのリンク待合せ

CALL JWAIT (DS, DJ, SS, SJ, CC)

(2) 特定ジョブへのリンク要求

CALL JLINK (SS, SJ, DS, DJ, CC)

(3) 特定ジョブへデータ送信

CALL JSEND (DS, DJ, SLNG, SBUF, CC)

(4) 特定ジョブからデータ受信

CALL JRECV (DS, DJ, RLNG, RBUF, CC)

(5) リンク切断

CALL JLNKC (DS, DJ)

ここで、SS, SJはマクロ発信元システム名、ジョブ名、DS, DJは受信先（相手）システム名、ジョブ名、SLNG, SBUF及びRLNG, RBUFは各々、送信及び受信用バッファ長、バッファアドレスである。

ユーザジョブはこれらのマクロをコールするだけで、図5の手順やまた制御ジョブから出される図3のコマンド手順などを意識することなく、ジョブ間通信を行なえ、いくつかのシステムにまたがったネットワークジョブ等も高水準言語のみにより作成できる。

ジョブ間通信のための制御ジョブは、次の様

な制御プログラムにより、リンク制御を行なう。

ジョブコード	Ssd	Ssd システム Id
相手ジョブ名	Lst	リクエスト (確立/切断)
自ジョブ名	Rst	要求スズ
Lst	Rst	R% (ロールイン/ロールアウト)

図6 LCB (Link Control Block)

ユーザジョブがリンク待合せあるいはリンク要求を出したとき、制御ジョブはLCBを作成する。そして、あるLCBの自ジョブ名と、別のLCBの相手ジョブ名が一致し、しかもステータスの対応がWaitとLinkであった場合にリンクが確立したとする。そしてJWAIT発信ジョブにDS, DJが通知される。

以上にINCの機能と通信手順が簡単に示した。INCにより、実現可能な機能は次のようなものである。

(1) システム及び環境制御

POWER ON/OFF, AUTO IPL, AUTO STOP, ... その他

(2) モニタ

ジョブ実行監視、統計

(3) 計算機間結合

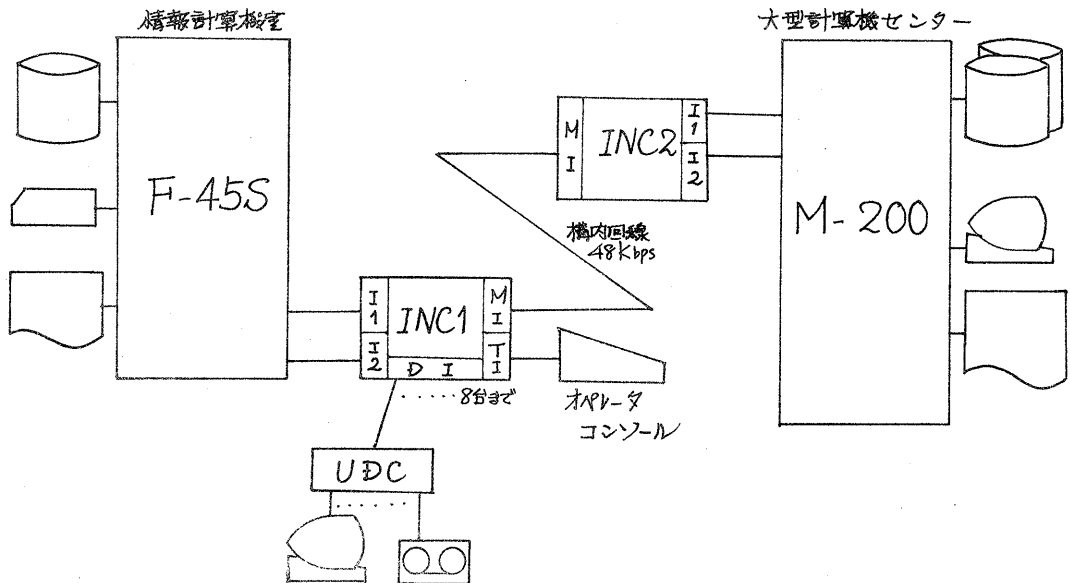
(4) 非標準I/O制御, RJE

(5) ユーザジョブレベルでの制御プログラム

この他にも多くのものが考えられる。しかしコンソールインタフェースに機能追加するわけであるから、多くの問題点が残る。インタフェース自体の速度がおそい、1度に通信できるメッセージが短い、文字しか扱えない、オペレータコマンドがシステムに危険をおよぼす、... などである。しかし各種の応用が手軽に行なえるという大きなメリットがあるわけで、そういう観点から言えば、INCはすぐれたシステムである。

次に応用例として計算機間結合のためのシステム構成、そのうちの一つの機能である、リモートファイルアクセスのユーザインタフェース、プロトコル等について紹介する。

3. システム構成



INCによる計算機間結合のためのシステムを図7に示す。M-200は九州大学大型計算機センター設置の共同利用計算機、F-45Sは情報工学科設置の教育用計算機で、2台のINCによって結合されている。INC1及びINC2は、M6800を用いたマイクロコンピュータシステムで、次の各インタフェースを保有している。

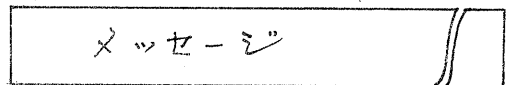
- I1 コンソールインタフェース
- I2 データ転送用IOCインタフェース
- TI コンソールタイプライタインタフェース
- MI 高速モデムインタフェース
(転送速度 48Kbps)
- DI UDCインタフェース

UDC (Universal Device Controller) はマイクロコンピュータで構成されており、複数のI/O装置の制御を行っている。

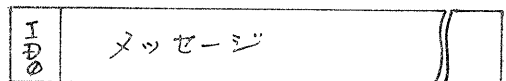
次にINC上に混在するメッセージのフォーマットを図8に示す。I2の1はマクロ発信元のジョブコードで、：ジョブコード (WTO), ④Aジョブコード (WTOR) の形をしている。I2の1はINCの拡張機能を示すコードで、通常のメッセージには未定である文字列である。

'?'を用いる。S.Iはメッセージの送り先ホストシステム (F-45S, MIM200) 用Id, U.Iは端末制御を行うためのUDC番号で、FCはINCに対する機能コードである。以上の部分もまとめて、メッセージヘッダと呼ぶ。その他に必要な機能コードや制御情報は、アプリケーション毎に定めて、転送されるデータとともにメッセージの部分に置く。

(a) オペレータコマンド/システムメッセージ



(b) 通常のWTO, WTOR (返答を含む)



(c) INC拡張メッセージ

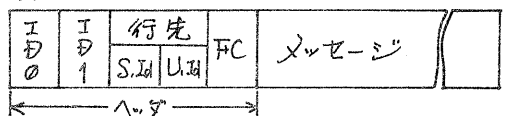


図8 メッセージフォーマット

INCは、図7の名インタフェース(エロポート)に対応したメッセージバッファをプールしている。メッセージの送信及び受信は、割込みを用いて行う。INCのメインプログラムは、バッファがフルになったか(受信完了か)、空になったか(送信完了か)をチェックしており、フルならばそのメッセージのヘッダを参照して、対応する処理を行う。指定された行先へメッセージを転送する。また空ならば、そのバッファをフリーにして、チェック部分へ送る。図9にINC内部処理フローを示す。

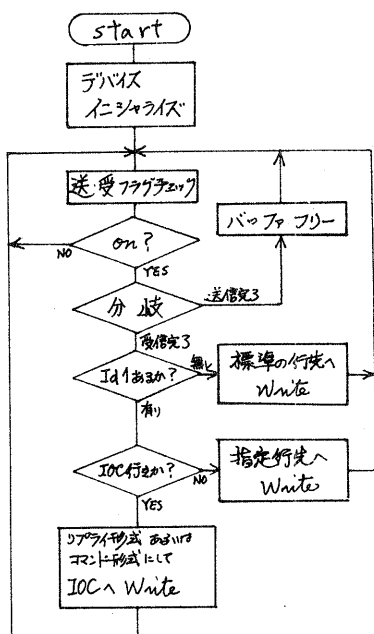


図9 INC内部フロー

以上のように、INCはOSの拡張機能全体を保持するのではなく、ホスト上の制御ジョブあるいはUDC上の端末及びアクセス要求ジョブ管理プログラム、それにユーザジョブ等の間のアクセスパスを実現する機能のみを保持し、全体として分散された機能が統合された形で各種アプリケーションを形成する。このような分散形態を採用したのは次の理由による。

- (1) INC自体に機能を持ち込みすぎるとINCのオーバヘッドが増加する。
- (2) 多くの部分がホスト上で高水準言語に

より作成され、拡張・変更が容易。

計算機間通信として実現したいアプリケーションとしては、

- (1) ファイル転送・アクセス
- (2) リモートジョブエントリ
- (3) 端末制御
- (4) ジョブ間通信

などがあるが、最初の応用例として、ファイル転送・アクセスを実現するための手段を次章で詳しく述べる。

4. リモート ファイル アクセス

計算機間とファイル操作を行いたい場合、必要とする機能としては、対象とするファイルの生成及び消滅、他システムから自システムへのファイルの一括転送、必要なファイル情報をレコード単位でアクセスできること、等がある。ここでは又台のINCにより、統合された、情報工学科のF-45Sから、大型計算機センターのM-200上のファイル(Mとはファイルのことをデータセットと呼ぶ、以下同様)に関するアクセス手段を提供するためのシステムについて、その構成、プロトコル等について述べる。図10にシステムの構成を示す。

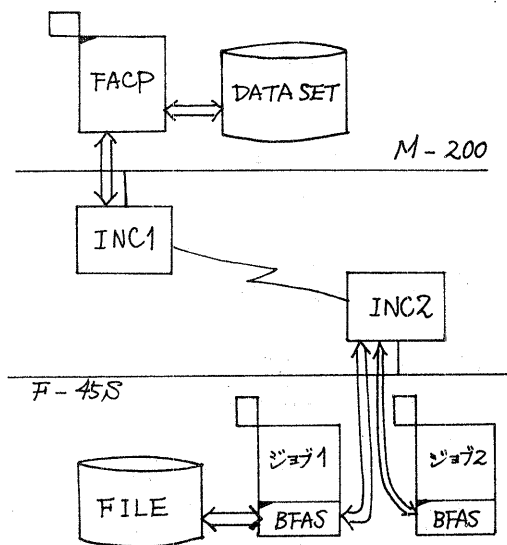


図10 システム構成

FACP (File Access Control Program) は M-200 上にあるデータセットアクセス用の制御ジョブで、INC から投入される START コマンドにより起動され、対象データセットに対する操作を行おうとともに、結果を F-45S 側に転送する。BFAS (Basic File Access Subprogram) は、F-45S 上のファイルアクセス要求ジョブに組込まれるサブルーチン群で、各操作毎に、制御情報やデータを FACP とやりとりする。アクセスには、レコードアクセスとコピーの 2 つのモードがある。

4.1 レコード アクセス モード

M-200 上のデータセットに対し、F-45S からのレコード単位でのアクセスを許すのがこのモードである。この場合、制御ジョブ FACP は、複数のユーザジョブからの複数アクセス要求を受け、並行して処理を行なう。ユーザジョブに対して用意されている基本操作用マクロとその呼出形式は次の通りである。

(1) データセットの割当て

CALL RALLOC (DD, DS, UNIT, DSP1, DSP2, SP1, SP2, REC, CC)

(2) データセットのオープン

CALL ROPEN (DD, TYP, CC)

(3) データセットの読出し

CALL RREAD (DD, BUF, BLK, CC)

(4) データセットへの書き込み

CALL RWRITE (DD, BUF, BLK, CC)

(5) データセットのクローズ

CALL RCLOSE (DD)

(6) データセットの解放

CALL RFREE (DD)

パラメータ

DD データ定義 (アクセス名)

DS データセット (ファイル名)

UNIT 'PUB' カタログデータセット

'WORK' 作業用データセット

DSP1 データセット状態 ('NEW', 'OLD'...)

DSP2 後処理 ('CATALOG', 'DELETE')

SP1, SP2 トラクタ単位の初期量, 増分量

REC 処理レコード長

CC 完了コード

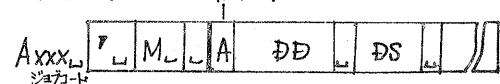
TYP 処理形態 ('INPUT', 'OUTPUT', 'UPDATE')

BUF データバッファ

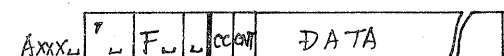
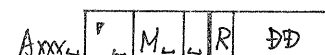
BLK データブロック長

BFAS がこれらのマクロ呼出しにより、FACP に対する依頼情報を図 8(c) のメッセージフォーマットに合わせてメッセージに組立て、WTO または WTOR マクロにより INC へ渡す。この際のメッセージフォーマットをサブルーチン毎に示す。

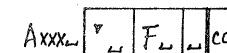
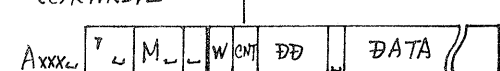
(a) RALLOC



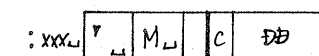
(b) RREAD



(c) RWRITE



(d) RCLOSE



(注) (a)~(c) は WTOR

(d) は WTO

図 11 Vコードアクセス メッセージフォーマット

FACP はリモート Vコードアクセス、しかも複数ユーザジョブのアクセス要求を許すため

に、各ジョブごとに、UMT (User Management Table) を、データセットごとの管理のためにDMB (Dataset Management Block) を保持している。UMTは、アクセスしたデータセットのユーザ登録名、アクセス要求を出したユーザのジョブコード、及びDMBの個数と各DMBのアドレスが記入される。DMBにはデータセット名及びDID名、データセットの状態、レコードに関するポインタ、バッファアドレス等が書かれる。図12にUMTとDMBの構成を示す。

リモートレコードアクセスのプロトコルを図13に示す。そしてプロトコル及びFACP内部処理について説明を加える。

(1)ユーザがRALLOCマクロを出すとBFASはまずMにDISPLAYコマンドを出し(①)、FACPが実行中かどうかをチェックし、実行している場合は、STARTコマンドでFACPを起動する(②)。正常に起動が行われると、アラケーションメッセージをFACPに送る(③)。FACPはデータセット管理情報をDMBに書き込み、対応ジョブコードを持つUMTがあればそのUMT、なければ新しく作成したUMTにDMBアドレスを格納し、データセットのアロケーションを行う。完了情報CCをユーザジョブへ返す(④)。

U M T	
ユーザ登録名	
ユーザジョブコード	
DMBの個数	
DMB1のアドレス	
DMBnのアドレス	

D M B		
データセット名		
DID名		
ステータス	サテライト	完了コード
DISP1	DISP2	ロード長
SPACE 1		
SPACE 2		
サテライトポインタ		
バッファ アドレス		

図12 UMTとDMBの構成

この操作は、ジョブ制御のDID文(データセット定義)をプログラム実行中に実行することに相当する。つまり、データセットの動的割当てである。

(2)次にROPENマクロを出すと、オープンメッセージはそのまゝFACPに渡され(⑤)、

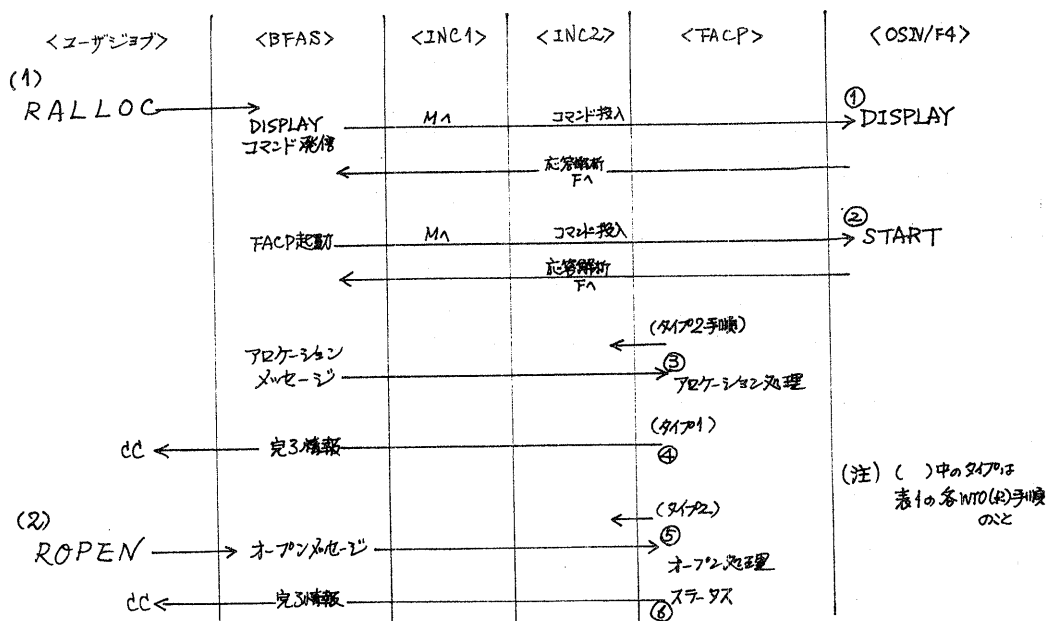


図13 リモートレコードアクセスプロトコル

(注) () 中のタイマは表1の各WTO(R)手帳のこと

オープン処理の後に宛るコードが BFAS を通してユーザジョブに送られる (④)。

ここで、データのアフセス単位について述べておく。READ/RWRITE の呼び出しにより読み書きされるデータ単位をブロックと呼び、1ブロックは複数のレコードにより構成される。WTOF をやりとり出来るメッセージ長には制限がある (120 文字以内) ので、1メッセージに乗せるデータは、レコードを 80 文字毎に分割した「サブレコード」を単位とする。図14にこれらの関係を示す。

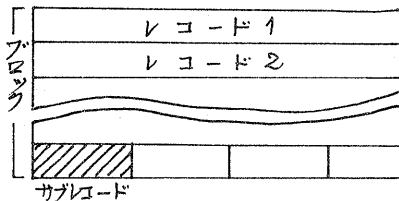


図14 ブロック・レコード・サブレコード

(3) READ マクロが送られると、図11(b)上のメッセージジョブコードとDD名とがFACPに渡される。FACPはこれより該当UMT→DMBを見出し、DMB情報にのっとりブロックリードを行なう。(⑤) そしてサブレコード毎にデータをユーザ側へ送るが、これは図11(b)下のCNTフラグにブロックエンドが表示されるまで続く(⑥)。BFASは受取ったサブレコードをユーザのバッファに格納してゆき、最後に1ブロック形成する。RWRITEの場合と同様である。

このように BFAS と FACP は互いに同期をとるが、ブロックをレコードとサブレコードの分解/組立、サブレコード単位での転送処理を行なう。例えば、1ブロックの転送中に別ブロックの転送が割込むことが可能となり、各ジョブに対して平等なサービスが行なえる。

データセットの解放は RFREE マクロの発行により行なわれるが、リポートファイルアクセスを行なっているジョブが RFREE を出さ

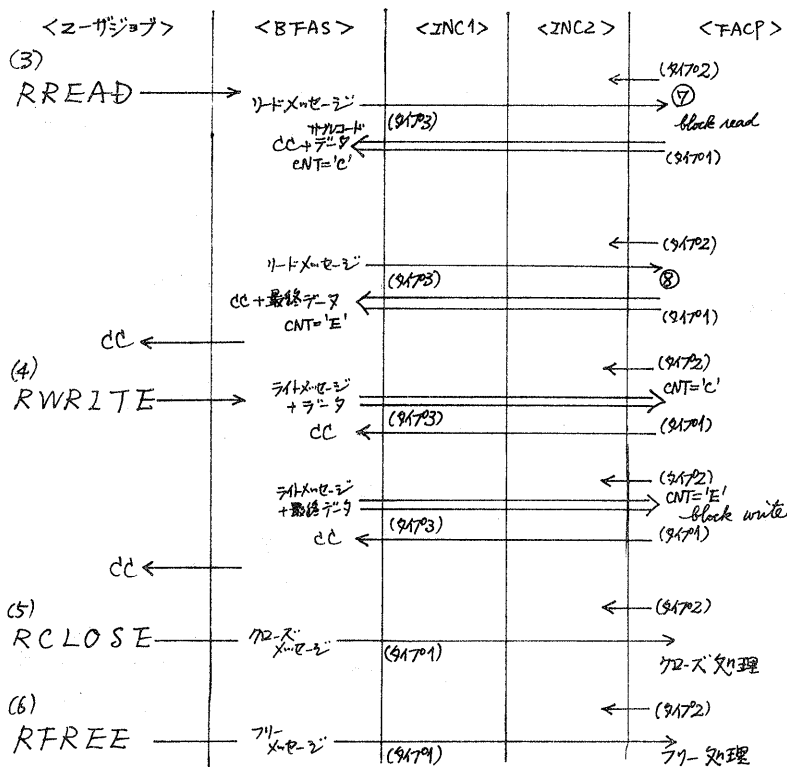


図13 リポート レコード アクセス プロトコル (続き)

ずに終るしても、INC1がジョブの終るをFACPに通知するので、解放は自動的に行われる。

4.2 コピーモード(ファイル転送)

テキストエディタをリモート使用する場合のように、INCを介してリモートレコードアクセスを頻繁にくり返すと、INCによるオーバーヘッドが大きくなってしまい効率が良くない。そこでM-200上のデータセットをF-45S上のファイルにそのままコピーし、ユーザは45Sのファイルに対してアクセスを行うようにした方が効率が良い。ここではファイルコピー機能の実現手段について述べる。

ユーザの発行できるマクロとその呼出形式は次の通り。

(1)データセットのファイルへのコピー
CALL RLOAD(AC, DD, CC)

(2)ファイルのデータセットへのコピー
CALL RSTORE(AC, DD, CC)

以上の他に RALLCC, RFREEがあるが、呼出形式は4.1の場合と同じ。ここで、ACはF-45Sのファイルのアクセス名、DDはM-200のデータセットのDD名である。

図15にファイルコピーの Protokolを示す。

(1)ユーザジョブがRLOADマクロを発行すると、BFASはまず、ジョブ制御文によってユーザが指定したファイルの属性を、F-45Sの制御表であるUAT(Unit Assign Table)から引き出すために、ACを指定してUATマクロを用いる。この時に得られた情報をもとにGETFILEマクロによりファイルを確認する。両マクロともF-45Sユーザに対するサービスマクロである。次にRLOAD中で4.1のROPENマクロを発行し、オープン正常終了後、RREADによりレコードアクセスを行い、読取したブロックをF-45Sのファイルへ書き込む。この動作を対象データセット全体が処理されるまでくり返す。

(2)RSTOREマクロの場合も同様にしてファイルを確認した後、RWRITEで指定データセットに書き込んでゆく。

コピーモードにおいてもRREAD/RWRITEを使用しているが、複数ジョブからのアクセス要求を並行して処理することは行わない。これはファイル全体をバースト転送して転送時間を短縮するためである。ゆえに、あるジョブのファイル転送中には、他ジョブからの転送要求は待たされる。

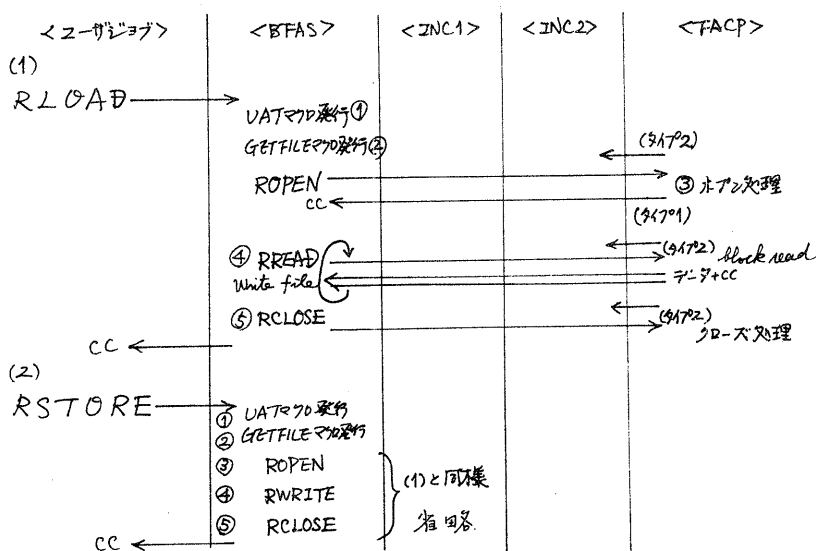


図15 ファイルコピー プロトコル

5. おわりに

以上、インテリジェントコンソールとそれを用いた制御ジョブ等の基本概念、及び応用例の一つとして、計算機間結合のリモートファイルアクセスについて述べた。

インテリジェントコンソールの利用は、計算機間結合の場合だけとしてみても有用であり、制御ジョブやユーザジョブ組込のサブプログラムにINCにおくべき処理機能を分散させることとシステム構成する場合の基本としているために、実験ラストによって得られる経験から、システムのデバッグ、変更をすることが容易である。また、OSに全く手を加えず、そのユーザやオペレータとのインタフェース部分をうまく利用することだけと各種の応用システムが作成出来ることは非常に興味深い。

INCに各ホストに特有のコンソールインタフェース部分を収録させ、各ホストが互いに相手を指定することでコンピュータネットワークの実現も可能になるとと思われる。

最後に本システム作成に当り御協力いただいた、九州大学大型計算機センター及び富士通株式会社関係各位、及び研究室で熱心に開発を担当してくれた各氏に謝意を表します。

参考文献

- (1) 有田ほか「複合計算機システムの対称型ソフトウェア」
情報処理第16回プログラミングシンポジウム報告集, 1974
- (2) 有田ほか「プログラム制御方式の利用入出力制御装置とその2,3の応用」
情報処理第16回論文集, EC-75-58, 1975
- (3) 有田ほか「システム間通信機構を用いたファイルプロセスの一構成法」
情報処理第16回論文集, 1975
- (4) 有田ほか「電話タイプライタをオンライン化するためのソフトウェア」
電算学会九州支部大会論文集, 1976
- (5) 有田ほか「インテリジェントコンソールとその応用」
電算学会九州支部大会論文集, 1978
- (6) 堀崎ほか「インテリジェントコンソールによる端末制御システム(I), (II)」同
- (7) 有田ほか「インテリジェントコンソールとその応用」
情報処理第20回プログラミングシンポジウム報告集, 1979
- (8) 北尾ほか「インテリジェントコンソールの応用、運用管理(I), (II)」
電算学会九州支部大会論文集, 1979