

リモート・バッチ・システム RACS の開発

丹羽 敏行

情報工学科
(1978年 9月 25日 受理)

Implementation of RACS (Remote Access Computing System)

Toshiyuki Niwa

Department of Information Engineering

(Received September 25, 1978)

Recently, computer network linking educational and research institutions is frequently proposed as a means of making available specialized or distributed computing resources.

We have also implemented RACS, which serves mainly for RJE and makes available all resources of the terminal and the host computers.

Therefore, RACS is very useful for data conversion (e. g., from the terminal's disk to the host's MT or vice versa).

The on-line Editor are being implemented, supported by the conversational monitor.

1. まえがき

近年、コンピューターの利用はローカル・バッチからリモート・バッチ、TSS へ、さらにはデータ・ベース、特殊周辺機器プログラムの共用へと高度化しつつあり、とくに分散するリソースに対してはネットワークを介しての利用が大いに有望視されている。

すでに、大学間コンピューター・ネットワークの実験が、東京大学大型計算機センターと京都大学大型計算機センターとの間で3年程前から行われており¹⁾、名古屋地区でも名古屋大学大型計算機センターで実験²⁾が進められている。

本学の情報工学科においても教育用中型計算機の導入時より、通信制御装置と回線が用意されていたが、ホストとなるべき中型機はバッチ専用機で TSS はおろか会話形処理さえも困難な状態である。

一方、端末となるミニコンはホストとはハード、ソフト共にそのアーキテクチャが大きく異なり、既製のソフト特に通信管理プログラムは全く利用できない。

以上の2つの問題点のため、今日まで通信制御装置と回線の活用が遅れることになった。

しかし、ネットワークの基礎研究と、中型機における図形データの入出力をミニコン側のグラフィック・ディ

スプレイや X-Y プロッタを用いて可能にするため、本システム RACS (Remote Access Computing System) の開発を行った。

現時点での RACS は、端末側からのリモート・バッチ・ジョブの処理 (RJE) を中心にしているが、単なる RJE ではなくホスト及び端末計算機の全てのリソースの共用を可能としている点を大きな特徴としている。したがって、図形データのみならず一般のデータ変換にも大いに有用と思われる。

RACS の運用を始めてすでに 10ヶ月余りになり、デバッグも完了したので、その概要を報告し、将来性について考察する。

2. RACS の構成

2.1 ハードウェア構成

現時点での RACS のハードウェア構成を Fig. 1 に示す。端末 U-200 とホスト H-8450 は 4 線式の専用回線で結合されており、回線制御アダプター-ASA と通信制御装置 CCM のバッファは調歩同期式で半 2 重 1200 bps で動作する。CCM には JIS7↔JIS8 のコード変換機能、時間監視のためのタイマー、垂直パリティ・チェック、水平パリティ・チェック、CRC チェックや不当

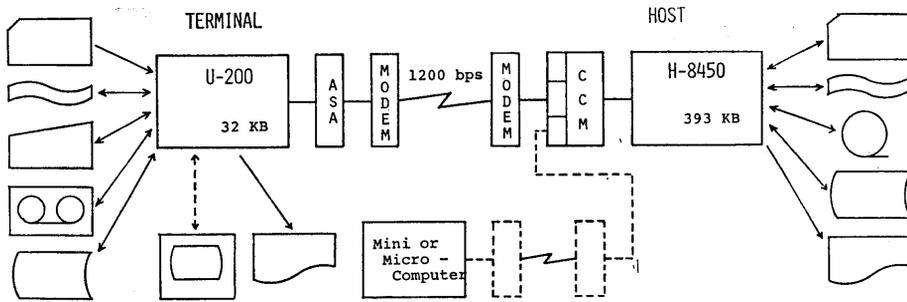


Fig. 1 Hardware configuration of RACS

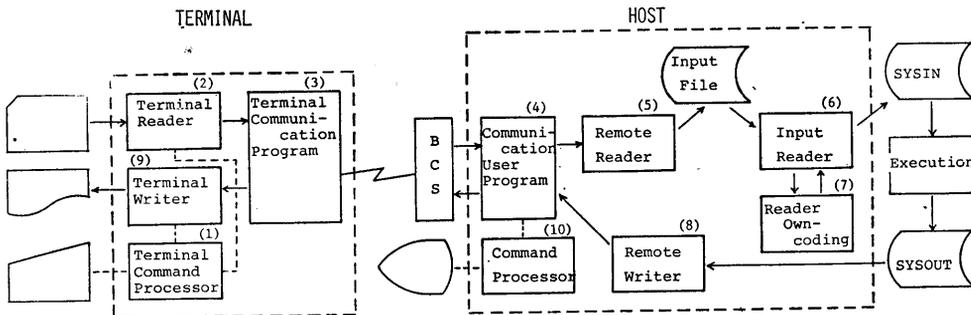


Fig. 2 Software structure and the data flows of RACS

キャラクタ・チェックが備わっているが、ASAには垂直パリティ・チェックの機能しか備わっていない。一方ASAでは伝送コードを5~8ビット(+1パリティ)の間で選択できるが、CCMでは8ビット固定となっている。

2.2 ソフトウェア構成

RACSのソフトウェア構成をFig. 2に示す。ここで破線で囲った部分が今回開発したプログラムである。

ホスト側のBCS (Basic Communication Support)とはメーカー提供の通信管理プログラムであり、コントロール・プログラムに組み込まれ、ユーザープログラムからのマクロ・コールによって特権モードで動作する。

2.3 処理形態

RACSでは次の3つの形態をサポートする。

(1) 自端末入力, 自端末出力

この場合は通常RJE (Remote Job Entry) と呼ばれ、ジョブは端末より入力し、結果を端末に出力する。

(2) 自端末入力, ホスト出力

この場合は大量の出力あるいはカセットMTからMTへのデータ変換などに利用できる。

(3) ホスト入力, 端末出力

この場合はMTあるいはディスクパックからの入力に対して有効である。

以上の3形態はソースの入力、リストの出力を中心に分類したもので、ソースとデータの入力装置が異なる場合や、リストパンチなど多種の出力がある場合は3形態の混用が可能である。

たとえば、ホスト側のMTの内容を端末側のディスクにコピーする場合に、プログラムは端末のカードリーダーから入力し、データはホストのMTから読み、リストをホストのLPに印刷すると同時に端末のディスクに書き出すといった形で実行することが容易である。

2.4 処理手順

Fig. 2を用いてRJEの処理手順を以下に説明する。

(1) '/RJIN'ジョブコンによって端末のリーダー及びホストのリモート・リーダーの両プログラムを起動する。

(2) リーダはSYSIN装置あるいは'/PARAM'コマンドあるいはジョブコンによって指定されたファイルからデータ(ジョブ・デッキ)を読み込む。

データは圧縮され、ブロッキングされ、

(3) 伝送制御プログラムを通じて回線に送出される。

この時、EBCDICからJIS 7へのコード変換及び水平パリティ・ビットの付加などがプログラムで行われる。

- (4) BCS を通じて回線よりのデータを受け取り、それがメッセージの場合は妥当性をチェックし応答メッセージを送出した後、要求メッセージに従った処理を行う。
- (5) テキストの場合は、デブロッキングし、カードイメージに復元して入力ファイルに格納する。
- (6) リモート・リーダーが終了した時点で、入力ファイルの内容を入力スタックエリア (SYSIPT) に登録する。
- (7) その際、'!' 記号で始まる簡略化ジョブコンを'//' で始まる正規のジョブコン群に展開し、同時に利用者コードその他のチェック、プライオリティと要求メモリ容量の設定、ジョブ名の付与、デバッグ用パラメータの設定、コンソールへのコメント表示などを行う。
- (8) 出力スタックエリア (SYSOUT) に格納された出力データは '//RJOUT' コマンドあるいはジョブコンで起動されたリモート・ライターによって読み出されて、圧縮され、ブロッキングされて CUP を通じて回線へ送出される。
- (9) 回線よりのデータは伝送制御プログラムでデブロッキングされもとの形に復元される。それが、メッセージの場合は、応答メッセージを送出した後、要求メッセージに従った処理を行う。
一方、テキストの場合は、ライターによりSYSLST

及び指定されたファイル領域への格納が行われる。

2.5 入力ジョブの構成

端末より入力して端末に出力するRJEの場合のカードデッキの例を Fig. 3 に示す。

入力ジョブは1まとまり毎に '//RJIN' ジョブコンの後に入力し、次のジョブコンで入力終了となる。

したがって、ホスト本来のジョブコンは端末のジョブコンと区別するため '! ...' の形の簡略化ジョブコンを使用するか '//DATA' と '//DEND' ではさんで入力する。

```
//JOB
//RJIN
!JOB J599ABCDEF, 01, G
!DTF OUT=WR, 5
!FORTRN
:
!RUN
:
!END
!JOB
:
!END
//PARAM OUT=DD00/DFILE/A, ENQ=YES
//RJOUT
//END
```

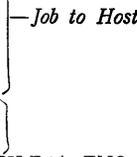


Fig. 3 An example card deck for RJE from the Terminal

2.6 コマンド及びジョブコン

端末側で現在用意されているコマンド及びジョブコン

Table 1 RACS commands for the Terminal

| Operation | Operand | Action |
|-----------|------------------------|---|
| // RJIN | [comment] | Start reading the input job |
| // RJOUT | [jobname [,jobnumber]] | Start or restart receiving the output job |
| // RJHLT | [comment] | Halt reading |
| // RJCAN | [comment] | Cancel the input job |
| // RJDLT | [jobname [,jobnumber]] | Delete the output job in SYSOUT file |
| // RJRLS | [jobname [,jobnumber]] | Release the output job from the hold status |
| // D | {JOB ...} | Display the current job status |
| | {MEM...} | Display the current memory allocation |
| // E | {SRT ...} | Start the job stage |
| | {STP ...} | Stop the job stage |
| | {HTJ ...} | Halt and delete the job stage |
| | {:} | |
| // DLT | [comment] | Delete the messages generated by 'D' or 'E' command |
| // PARAM | [IN=...] | Identify the input device and the filename |
| | [,OUT=...] | Identify the output device,the filename and the update mode |
| | [,ENQ= {NO / YES}] | Request for the instant output |
| | [,LIST= {NO / YES}] | Request for the listing of input/output data |
| // RJEDIT | | Enter the online-editor mode |

Table 2 Simplified Job Control cards and their original formats

| Simplified Job Control card for JYOHO H-8450 SYSTEM | | Original formats under EDOS-MSO | Notes |
|---|-------|--|--|
| !JOB aabbccccc, nn, l | | //aacc#nn JOB JBN=... //SYSUT1 DTF... //SYSUT2 DTF... //SYSUT3 DTF... //SYSOML DTF... //SYSLIB DTF... | Priority is variable according to the jobnumber and the jobclass l |
| !RUN | !LINK | // {LNKEDT} // {LNKEX } PROG aaccnn INCLUDE SYSUT1 | Values of TIME and LST are substituted corresponding to jobclass l |
| | !EXEC | // EXEC aaccnn, UT2 C TIME=... C LST=... | |
| !END | | // END | |
| !#xxxxxxx | | // CATLG #xxxxxxx | |
| !yyyyyy | | // yyyyyy | |

Table 3 Communication messages between the Terminal and the Host

| From Terminal to Host | | From Host to Terminal | |
|-----------------------|--|--------------------------------|---|
| Message | Meaning | Message | Meaning |
| M00 | Beginning of the input data | M00 READER START | Remote-reader is started |
| M01 | End of the input job | M01 READER STOP | Remote-reader is stopped |
| M02 | Start output | M02 WRITER START | Remote-writer is started |
| | | M03 WRITER STOP | Remote-writer is stopped |
| M04 | Halt output | M04 WRITER PAUSE | Remote-writer is in pause |
| M05 | Delete the output data | M05 OUTPUT DELETED | Output data are deleted |
| M06 | Cancel the input job | M06 INPUT CANCELED | Input job is canceled |
| M07 | Send the output data (text) | M08 | Input data (text) is accepted |
| M09 | Release the output data from the hold status | | |
| M0D | Simulate the display command | | |
| M0E | Simulate the job control command | M10 | Key in/out simulation is started |
| M11 | Send the simulated results | M11 simulated results | Simulated results |
| M12 | Delete the simulated results | M12 | Key in/out simulation is finished |
| M40 | Input data (text) | M40 blocked output data (text) | |
| | | M50 REJECT TIMING | Input command error |
| | | M51 INPUT OVER | Job stack area overflows |
| | | M52 INVALID COMMAND | Invalid command is typed in |
| | | M53 NO OUTPUT JOB | Output data is not found in the output stack area |
| | | M54 INVALID TEXT | Invalid text is found |
| | | M61 INVALID KEY-IN | Command for Key in/out simulation is not correct |
| | | M62 DSPLY DELETED | Simulated results are deleted |
| | | M63 BUFFER OVERFLOW | Buffer for simulated results overflows |
| MFF | | MFF PROGRAM ERROR | Message for debugging |

を Table 1 に示す。

ここで、'//RJIN'のみは機能上 SYSIN 装置より入力しなければならないが、他は全てコンソールより直接キー入力することによって起動することができる。

2.7 簡略化ジョブコン

端末より入力するリモート・ジョブ中の簡略化ジョブコンは、入力リーダー・OWN・コーディング・プログラム中で正規のジョブコン群に変換される。

簡略化ジョブコンとその展開形を Table 2 に示す。

3. 送受信手順

3.1 起動方式

端末側が優先、ホスト側が非優先の相互起動方式を用い、イニシャル状態ではホスト側は ENQ 待ちになっている。正常な送受信の一部を Fig 4 に示す。

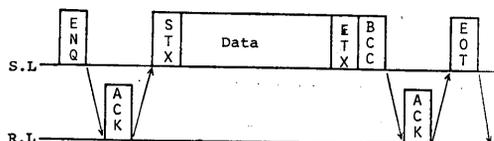


Fig. 4 A normal sequence of transmitting the data

3.2 送受信メッセージと処理

データの送受信における要求メッセージと応答メッセージを Table 3 に示す。

3.3 データ・フォーマット

送受信データのフォーマットを Fig. 5 に示す。

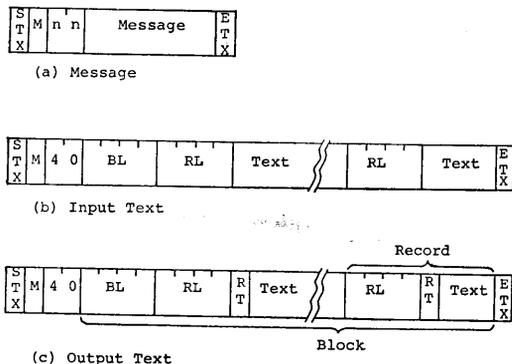


Fig. 5 Data formats

nn: Message number BL: Block length
RL: Record length RT: Record type

4. 考察

4.1 RACS の評価

単なる RJE の場合は伝送速度が 1200 bps と低速の

ため、同一棟内にある電算室へ向かう方が速いかもしれないが、実験装置からのデータを読みながら入力データを作成し処理したい場合などは便利である。

一方、ホストと端末で媒体に互換性のない場合には、RACS は最も有用となる。

たとえば、端末側での図形データの入力、ホスト側での演算処理、端末側での出力といった場合、従来は紙テープや MT を介していくつもの処理に分割して実行されていたのが、紙テープや MT の媒介及び分割処理が不要となり、結局ターン・アラウンド・タイムが短縮されることになる。

また、紙テープ以外に媒体に互換性を有しない場合は不可能であった MT などの大量のデータ変換が、端末側のディスクあるいはカセット MT などを使用して可能となる。

4.2 システムの拡張

現時点での RACS はミニコン U-200 と中型機 H-8450 の間で実現されているが、新たにマイコン 2 セットを含めての多端末化が早急に望まれている。

また、RJE でデバッグの度にソース・プログラムを転送するのは非効率であるので、現在オンライン・エディタを実現中であり、これは会話形式モニタの管理下で、あたかも TSS 風に動作する。

一方、図形データの転送の場合などバイナリ形式のままでの伝送 (トランスペアレント方式) が望ましい。RACS の場合も端末側からホスト側へは CCM の機構上不可能であるが、ホスト側から端末側へ擬似トランスペアレント方式といったものを試みている。

5. あとがき

情報工学科での多様な教育と研究を処理するには、ユーザーのソフトウェアでもって補うには限界があり、最新のハードウェアとソフトウェア・システムへのリプレイスが早急に必要であると考えられる。

しかし、それが容易でない現状では、RACS に見られるような方法でのアプローチはかなり有用と考えられる。

最後に、日頃熱心に御討論いただく名古屋大学工学部の福村先生と大型センターの吉田先生ならびにファコムハイタック (株) の楠葉氏と伊藤氏に感謝致します。

文 献

- 1) 文部省科学研究費特定研究「広域大量情報の高次処理」総括班報告, 1976-3
- 2) 吉田雄二: コンピュータ・ネットワーク, 名古屋大学計算機センターニュース, 第37号, 1978-2, pp. 49~56