

報 告

マルス 105 システム (2)

—ソフトウェア設計—

尾 関 雅 則* 柴 辰 平* 桑 折 恭 一 郎*
山 形 毅 章* 名 内 泰 蔵**

1. ソフトウェアの設計方針

マルス 105 のソフトウェア開発にあたり、以下にのべる設計方針を考えた。

(1) 機能の発展性

マルス 105 システムが、それ以前のマルス 104 までと異なる最大の特徴は、中央装置に、大形コンピュータ HITAC-8700 のマルチ・プロセッサを採用したことである。HITAC-8400 を使用したマルス 104 までには、中央装置の能力に比べて、要求される処理量が大きかったため、中央装置、端末に特殊なハードウェアを開発したり、ソフトウェアも、完全に処理効率を重視した設計が行なわれた。マルス 105 では、このようなハードウェアの制約が少なくなったため、処理を、ソフトウェアですべて行なうことにした。しかし、ソフトウェアも、大形化すると機能の変更が非常に困難になるのであるが、できる限り変更しやすいソフトウェア構造とし、システムの成長を妨げないことを、第 1 の目標とした。

(2) 処理能力の向上

大形の中央装置を採用したとはいえ、約 1,400 台の端末が連続して発信し、毎秒 50 トランザクションを処理しなければならないということは、かなりきびしい条件である。とくに、空席のサーチ、出力券面 1 枚ごとの編集といった、マルス 104 までは特殊のハードウェアで行なっていた業務を、ソフトウェアで行なうことにしたため、ソフトウェアの走行ステップ数を、できる限り少なくすることにも重点をおいた。

(3) システム規模の拡大

システムの規模が大きくなるということは、それだけで新しい問題が発生する。マルス 105 では約 2,000 列車、70 万座席×80 日というほう大なデータを取り

扱っているが、これらデータの作成、変更は、非常に大きな業務量である。これらの業務を、できる限り容易に、また誤りなく行なうために、多くのサポート・プログラムの開発などの考慮が必要となる。

(4) アベイラビリティの向上

マルス 105 の障害は、全国の駅での混雑など、非常に大きな影響を及ぼす。そのため、ソフトウェア面からも、ダウン時間の短縮のため多くの考慮を払った。

2. ソフトウェアの概要

2.1 RTCS/ES

マルス 105 の OS としては、とくにオンライン処理を効率よく行なうよう開発された RTCS (Real Time Control System) を採用した。その中心をなす制御プログラムを、RTCS/ES (Executive Supervisor) と称する。RTCS/ES の特徴は、以下のとおりである。

(1) マルチ・プロセッシング

RTCS/ES では、H-8700 を、2 台の処理装置 (BPU) によるマルチ・プロセッサとして動作させる。この方式では、2 台の BPU に共有されたメモリに 1 組のプログラムをロードし、そのプログラムの処理を、2 台の BPU で行なうものである。

(2) マルチ CUP

RTCS では、オンライン処理を行なうユーザ・プログラムを、CUP (Communication User's Program) と称している。CUP は、システムが複雑になるほど多様の処理を行なわねばならず、複雑になる。このため、RTCS では複数の CUP を許し、性格の異なる処理は、独立のプログラムとすることが可能である。マルス 105 では、指定券の発売を中心に行なう RMP (Real-time Management Program) と、汎用キーボード・プリンタからの処理で、車内発売書のほか、各種営業資料の出力を行なう GMP (General Management Program) の、2 つの CUP をもっている。

* 日本国有鉄道

** (株)日立製作所

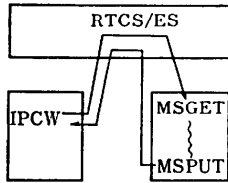


図1 プログラム間連絡機能

(3) プログラム間連絡機能

マルチ CUP 機能により、オンライン処理を複数の CUP に分けた場合、プログラム間で、情報連絡などを行ないたい場合がある。このため、IPC (Inter Program Communication) マクロを設けた。処理を受ける CUP は、回線からのインタフェースと同様に、MSGET/MSPUT マクロにより、IPC を出したプログラムとの情報の受渡しができる。マルス 105 では、CUP 間の連絡だけでなく、この IPC を使用して端末シミュレータを作成し、CUP のテストにも利用している。

2.2 通信制御プログラム

通信制御プログラム (CCP) は、端末の種類ごとに独立した複数の LCP (Line Control Program) と、共通処理を行なう単一の MCP (Message Control Program) からなっている。マルス 105 では、A・B 形端末、N 形端末、DT 形端末および時計回線の、4 つの LCP をもっている。

(1) 割込み回数の削減

RTCS では、通信制御プログラムを、ES レベルに上げてある。このため、通信制御におけるユーザのインタフェースは、MSGET/MSPUT のみで、回線の起動命令などをユーザ・レベルから出すことがないため、SVC の割込み回数が少なくなる。

トラフィックの大半を占める A・B 形および N 形端末では、入出力メッセージは、外部記憶装置に退避は行なわず、すべてコアメモリ上のバッファで、引継ぎが行なわれる。また、A・B 形端末ではブロック送信は行なわれないが、N 形端末のブロック送信は、図 2 に示す方式で行なわれる。ユーザ・プログラムは、座席位置などの出力情報を、パック形式で通信制御プログラムに送る。通信制御プログラム内には、パック・データから、券面 1 枚ずつの編集を行なうOWN・コーディング・ルーチンが組み込まれており、以後は、すべて ES レベルで、ブロック送信処理が行なわれる。このような方式により、マルス 105 で大半を占め

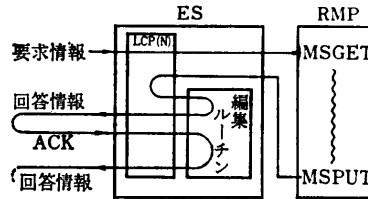


図2 N形端末のブロック送信

る指定券発売処理については、割込み回数が最小となるよう考慮した。

(2) バッファリング方式

トラフィックの高い指定券発売処理を行ない、なおかつ集線を行なっている N 形および A・B 形端末については、ラインごとに固定にもつ、デディケイテッド・バッファリング方式を採用した。また、トラフィックの低い DT 形端末では、1/2 コア・ダイナミック・バッファリング方式を採用した。

(3) メーク・ビジー機能

全ラインあるいは端末の種類ごとに、メーク・ビジー (回線の予閉塞) を行なうことができる。

2.3 運転制御プログラム

RTCS では複数の CUP をサポートするが、その内の 1 つをマスタ CUP と称し、あらかじめそのプログラム名を登録しておくことにより、自動的に起動することができる。運転制御プログラム (SOP: System Operation Program) は、マルス 105 のマスタ CUP で、システムの運転開始、終了、ダウン回復処理などを行なう。SOP の機能は、以下のとおりである。(ダウン回復処理の詳細については、4.3 参照。)

(1) システムの開始、回復処理

開始、回復処理においては、コア常駐の CUP である RMP1、GMP1 を、自動的にメモリに広げる。また、CUP が使用するファイルのオープン、MT、LP の登録、回線の起動などを行なう。

(2) 連続実行

開始、回復、終了時には、それぞれ定形化したオフライン・プログラムを走らせる処理がある。これらのオフライン・プログラム名を登録しておくことにより、SOP は自動的に順次起動してゆき、オペレーションの簡素化をはかることができる。

(3) 自動回復

オンライン・システムのダウンというものは、ハードウェアの障害など、システムの運転が続行できなくなった時に起こるものであり、その障害原因を取り除

表 1 オンライン・ユーザ・プログラムの構成

		属 性	主な処理内容
RMP	RMP 1	パーマネント・タスク コンカレント・タスク	指定券発売、取消し RMP 2 の起動
	RMP 2	トランジェント・タスク シングル・タスク	集計読出し 乗車券単独発売 残席状況表
GMP	GMP 1	パーマネント・タスク シングル・タスク	ブロック送信処理 GMP 2 の起動
	GMP 2	トランジェント・タスク シングル・タスク	車内発売通知書

かない限り、再開はできないものと考えられる傾向が強い。ところが、実際には一時的な原因で起こる場合が多く、単に回復操作を行なうだけで、運転を再開できる場合が多い。この種のダウン時間を最小にするため、自動的に回復する機能をもっている。また、手操作で回復する場合も、システム・コンソールのボタン1つで、回復処理を行なうことができる。以上の結果、自動回復でのダウン時間は約1分、手操作でも、回復処理の開始からシステムの運転の再開までは約1分30秒と、非常に短時間に運転再開ができる。

2.4 オンライン・ユーザ・プログラム

(1) プログラム構造

表1に、オンライン・プログラムの構成を示す。前にも述べたとおり、マルス105でオンライン処理を行なうプログラムは、RMPとGMPの2つがある。RMPは、指定券の発売を中心に行なうプログラムで、N形およびA・B形端末からの処理を行なう。また、GMPは、DT形端末からの処理を行なう。

これらのプログラムは、コア常駐のRMP1(GMP1)と、要求のあった時点で起動されるRMP2(GMP2)に分けられる。

RMP1(GMP1)の属性は、パーマネント・タスク・プログラムと呼ばれ、運転開始あるいは回復時に、SOPにより起動される。パーマネント・タスク・プログラムはコア常駐で、また、このプログラム中でプログラム・エラーが発生すると、システム・ダウンとなる。また、パーマネント・タスク・プログラムは、1組のプログラムで、同時に複数個のタスクの処理を行なうことができる。これは、コンカレント・タスクと呼ばれ、マルス105では、RMP1のみこの構造を採用しており、現在は、14個のタスクをもっている。

RMP2(GMP2)は、複数の独立なプログラムの総称で、端末の要求により、RMP1(GMP1)が該

当するプログラムを起動する。これらのプログラムの属性は、トランジェント・タスク・プログラムと呼ばれ、プログラム・エラーが発生した場合には、そのプログラムのみ強制終了されるのみで、システム・ダウンとはならない。

RMPでは、オーバ・ヘッドの減少、および座席ファイル(SF)を更新する呼については、処理途中で強制終了されては困るという点から、トランザクションの大半を占める指定券の発売、取消しなどの処理は、RMP1の中で行なわれる。また、乗車券単独発売、集計など、トラフィックが低く、中断してもかまわなく、また独立のプログラムとした方が、プログラムの開発、変更などが有利なものについては、RMP2レベルに落とした。また、GMPでは、GMP2起動、ブロック送信処理など共通部のみGMP1で行ない、業務の大半は、独立なGMP2レベルのプログラムとして開発した。

(2) RMP1の構造

マルス105のオンライン・ユーザ・プログラムの中心をなすRMP1の構造を、図3に示す。RMP1は命令、およびそれに付随したコンスタント類からなるアセンブル・モジュール、コア・テーブル、コンカレント・タスクのため、タスクごとの処理エリアであるTS(Temporary Storage)、およびライン・バッファから成立している。RMP1の起動時、これら全体がSYSRES(プログラム・ライブラリ)からロードされるのではなく、アセンブル・モジュール部分のみロードされる。その後、RMP1のイニシャライズ処理において、コア・テーブルはTDF(Table Dump File)より読み込み作成され、また、TSおよびライン・バッファも、それぞれTDFに書かれているRMP1のタスク数、およびCCPから教えられるライン数から初期設定がされる。このような構造とすることにより、コア・テーブルの内容変更、RMP1のコンカレント・タスク数の変更などは、プログラムを修正するこ

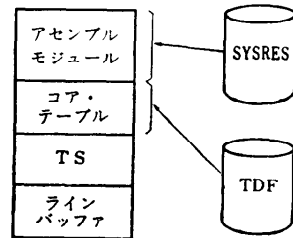


図 3 RMP 1 の構造

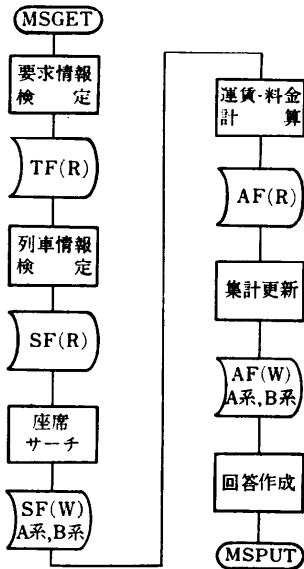


図4 発売処理の流れ

となく、TDF のデータを修正するだけで、容易に行なうことができる。

(3) 機能修正のための考慮

RMP 1 は、指定券発売処理の中心をなすものであり、非常に多くの機能をもったプログラムである。このため、システム稼働後も、たえず機能修正が行なわれるので、修正が容易に行なえるように作成することに、重点をおいている。他の担当者が、そのプログラムを理解し易いように作成することも、重要なことである。

まず、コンスタント類を、できる限り命令の中に組み込まず、独立のコンスタント、テーブルと定義するようにつとめた。とくに、重要で変更が行なわれ易いコンスタント類は、コア・テーブルとし、TDF を書き替えば変更できるよう考えた。また、TS(タスクごとの処理エリア)については、データの定義を明確にし、また、タグ名称の統一を行なった。

(4) 発売処理の流れ

図4に、指定券の発売処理の流れを示す。TF は、列車固有の情報をもつファイルで、列車の運転日、発着駅のチェックなどは、TF を読み込んだ後に行なわれる。SF は、座席の発売状況を管理するファイルであり、また、AF は、端末毎集計などを管理するファイルである。

2.5 オフライン・ユーザ・プログラム

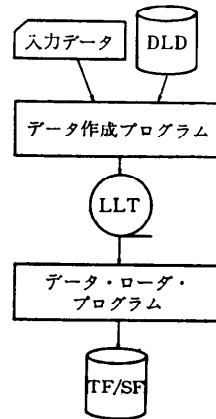


図5 列車データ作成処理

オフライン・ユーザ・プログラムは、列車データの作成を行なう FM (File Management)、統計業務に関する JM (Journal Management)、およびその他のサポート・プログラムの SS (System Support) に大別される。以下に、その特徴をのべる。

(1) DLD の採用

データ作成の中心をなす TF, SF は、非常にデータ量が多く、また、変更も頻繁に行なわれる。TF, SF には、列車ごとの停車駅、駅間のキロ程、各号車の座席形式など予約に必要な情報が格納されるが、これらのデータを直接作成すると人手もかかり、また誤りも多い。このため、列車データは、図5に示すとおり、入力データおよび DLD (Data Library Disc) から作成される。DLD には、駅間のキロ程、車両形式ごとの座席数、座席形式などが格納されており、入力データとしては、停車駅、列車編成など、できるだけ少ないデータを作成するのみでよいよう考えられている。

(2) オンライン・ローディング

列車データの 신설、変更作業は、夜間に行なうとすると、列車本数の増大にともなって、ハードの保守時間、あるいは営業時間にまで影響を及ぼす恐れがある。そこで、オンライン時間帯に、このような作業を行なうこととしている。しかし、オンライン処理中では、誤って変更した場合、非常に危険である。このため、列車データの 신설、あるいは変更(変更の場合も 3.1 にのべるとおり、変更部分の情報をつけ加えるという方法をとる)の場合、変わった部分の情報には、その日の日付をつけて列車データに書き込み、その日だけは、その部分の情報は無効にするという方法をとっている。

(3) 統計データ

オンライン処理の実績データも、有効に利用できる。マルス統計は、列車の発売実績、取扱い金額の集計、および保守用統計に大別される。列車の発売実績データは、つぎの列車設定計画などに使用される。

3. ファイル

マルス 105 でもつファイルの概要について、以下にのべる。

3.1 列車ファイル (TF)

TF (Train File) は、列車の編成、停車駅など、列車ごとにその列車固有の情報をもつファイルで、オンライン処理では参照されるだけで、更新はされない。現在、マルス 105 では、約 2,000 列車を取り扱っているが、このデータは、定期的なダイヤ改正で大修正が行なわれるほか、増結、停車駅の変更などの小変更は随時行なわれ、その回数も非常に多い。このような変更は、列車の編成、停車駅など、列車の発車日基準で変更されるものが多いため、変更前夜に書き替えるといった方式はとれない。また、作業も、できるだけバッチして行ないたいという点からも、TF の構造は、以下のようになっている。

まず、TF の情報を、編成、停車駅などデータの性格によって、情報要素という単位に分ける。1 つの情報要素は、図 6 に示すとおり、各情報要素に共通のヘッダ部とデータ部に分けられる。ヘッダ部には、その情報要素の種類、有効期間などが格納される。

1 つの TF レコードには、同一情報要素がいくつ格納されていてもよく、また、順序も、任意である。オンライン処理で TF データを参照する場合、各情報要素ごとに、レコードの先頭から有効期間を満足する最初のを有効とみなす。このため、TF データの変更を行なう場合、一般には、情報要素のデータを書き替える方式はとらず、レコードの先頭に、新たな情報要素をつけ加えることにより行なう。図 7 は、10 月 1 日から 10 月 31 日までの間、停車駅変更などで駅の情報要素が変更された場合の例である。

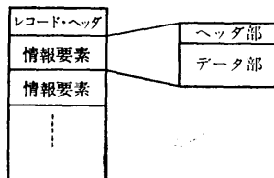
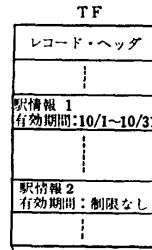


図 6 TF の構造



駅情報の有効性
10/1 ~ 10/31 → 駅情報 1
その他の期間 → 駅情報 2

図 7 TF の情報変更例

3.2 座席ファイル (SF)

SF (Seat File) は、指定券発売システムを中心をなす、座席の発売状況を管理する在庫管理ファイルである。

(1) 座席の管理

マルスでは、定数管理ではなく、座席の指定を行なうため、1 つの座席を 1 ビットで管理している。さらに停車駅ごとに乗降客があるため、理想的には、1 つの座席は (停車駅 - 1) ビットを必要とする。また、80 日分の在庫管理を行なうので、1 日あたり 70 万座席としても、そのデータ量はばく大なものになってしまう。また、駅ごとに 1 ビットで管理すると、空座席を探す部分の走行ステップ数も多くなる。そのため、実際の座席は、数駅をまとめて、1 ビットで管理している。

(2) SF データの収容

SF のレイアウトは、図 8 に示すとおり、80 日分のエリアをもつ。先頭に原始データ (座席が全然発売されていない状態のデータ) があり、残り 79 日分のエリアを、サイクリックに使用している。列車の SF レコードのアドレスを求めるには、TF から SF の原始データのアドレスを求め、さらに、コア上のテーブルから、要求の発車日のデータは、原始データより何番目にあるかを求めることにより行なう。

SF データは、列車出発の翌日の夜間に、原始デー

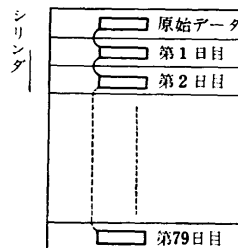


図 8 SF のレイアウト

タの内容を移し、新たに、79日後の在庫データとする。

3.3 端末ファイル (AF)

AF (Agent File) は、端末ごとにもつファイルで、つぎの2つの目的に使用される。

(1) 端末ごとの集計

その日の朝からの発売枚数、金額を加算し、端末オペレータが、取扱い金額との照合を行なう。

(2) ダウン時の座席損失対策

システム・ダウン時の座席の損失を防止するため、最新のトランザクション情報を記憶しておく。(使用法については、4の異常処理参照。)

3.4 テーブル・ダンプ・ファイル (TDF)

オンライン・ユーザ・プログラム (SOP, RMP, GMP) がもつコア・テーブルを格納するファイルである。

3.5 トランザクション・テープ (TRT)

TRT は、トランザクションの内容を記憶するテープである。これは、統計および SF データが破壊された場合の、ファイル回復に使用される。

3.6 モニタ・テープ (MTT)

MTT は、制御プログラムおよびオンライン・ユーザ・プログラムが検出した異常内容を、ロギングするテープである。これは、つぎの目的に使用される。

(1) 障害原因の追求

(2) 中央装置の予防保全

ハードウェアの障害については、再試行に成功した場合を含めて記録している。これにより、一時的障害の傾向を、把握することができる。

(3) 回線、端末の管理

伝送制御上の誤りについても、再試行に成功した場合を含めて記録している。これにより、誤りの多い回線、端末を中央で把握し、保守へのフィード・バックを行なっている。

3.7 オーディット・トレイル・ファイル (ATT)

伝送イメージの送受信情報を記録する。これにより、回線、端末、あるいはソフトウェアの障害追求を行なう。

4. 異常処理

4.1 ハードウェアの異常

(1) 中央処理装置

中央処理装置は、演算処理装置 (BPU)、主記憶装置 (MMU)、および入出力処理装置 (IOP) からなり、オ

ンライン系は、それぞれ2台で運転することを原則としている。BPU および MMU については、障害を検出した方のみ、自動回復をともなった自動切離しができるよう考えている。また、IOP については、できる限り、チャンネルなど部分的な使用停止とする。

(2) 通信制御装置

通信制御装置関係で障害が発生した場合には、システム・ダウンとはせず、回線単位、あるいは通信制御装置単位に使用停止とする。使用停止を解除するには、オペレータの手操作により行なう。

(3) ファイル装置

磁気ドラム、ディスク装置など、ファイル装置の異常については、収容されているファイルにより、つぎのように取扱いが異なる。

(a) TF/SF

TF/SF の障害については、システム・ダウンには結びつけない。また、トラック単位、レコード単位の使用停止もせず、すべてデバイス単位の管理を行なう。2重系で運転していたファイルの片系が異常になった場合は、該当する両方のデバイスを、使用停止とする。オペレータは、片系のみで使用再開するか、予備デバイスにコピーして両系にするかの処置をとる。また、両系とも異常になった場合は、ファイル回復処置をとる (4.4 参照)。

(b) AF

AF も2重ファイルであり、TF/SF と同様の処置がとられる。ただし、両系とも異常になった場合は、磁気テープより、その日の運転開始状態のデータを運転再開する。この場合、その日は、端末での集計読出し操作のみ不能となる。

(c) TDF

TDF も2重ファイルであるが、両系とも異常になった場合、システムの運転ができなくなること、またその後運転再開するまでの操作が、かなり繁雑になることが、上記のファイルとは異なっている。また、比較的信頼性の高い磁気ドラムに収容されていることから、片系でも異常になった場合は、システムダウンとしている。その後、片系のみで運転再開することもできるが、両系とも異常になった場合は、データを運転開始時の状態にもどしたのち、運転開始操作をする。

(d) SYSRES

SYSRES (System Residence) は、ロード・プログラム・ライブラリのことである。オンライン・プログ

ラムのロード時の異常は、システム・ダウンとするが、オフライン・プログラムのロード時の異常は、そのまま運転を続行する。

4.2 ソフトウェアの異常

ソフトウェアの異常の場合は、そのプログラムの属性により、つぎのいずれかの処置がとられる。

(1) 制御プログラム、およびパーマネント・タスク・プログラム (SOP, RMP, GMP のコア常駐部分) での異常は、システム・ダウンとする。

(2) トランジェント・タスク・プログラム (RMP, GMP から起動されるプログラムおよびオフライン・プログラム) での異常は、そのプログラムのみ強制終了させる。

4.3 ダウン回復処理

マルスに限らず、一般に、リアルタイム・システムというものは、要求のあった時点で処理をしなければいけないという点から、高いアベイラビリティを要求される。とくに、障害の影響は、時間が長びくにつれて大きくなるため、MTBF が長いこと以上に、ダウン時間が短いことが重要視される。

マルス 105 では、回復時間を短くするため、回復操作を、システム・コンソールのボタン操作のみで行なえるよう考えた。また、ハードウェアのインタミット・エラーの場合には、自動的に回復できる。以下に、回復操作時に行なう処理と、それらを、いかに人手介在なしに行なえるようにしたかについてのべる。

(1) コア常駐プログラムのロード

回復時には、システム・コンソールの「回復」ボタンにより、制御プログラムがロードされる。また、自動回復の際は、制御プログラムが自動的に IPL に制御を移したのち、再び制御プログラムがロードされる。その後、制御プログラムがあらかじめ登録されている、マスタ CUP (SOP) を起動する。SOP は、TDF に登録されているテーブルから、RMP 1 および GMP 1 を起動する。

(2) コア・メモリのダンプ

システム・ダウン時には、のちの障害原因の追求のために、メモリのダンプが行なわれる。この処理は、回復操作の前に、制御プログラムのエリアを含めて、破壊しないよう特別に作られたプログラムにより、LP または MT に出力するシステムもあるが、その場合には、ダウン時間が長くなってしまふ。そのため、マルス 105 では、図 9 に示す方法で、回復時に自動的に行なっている。

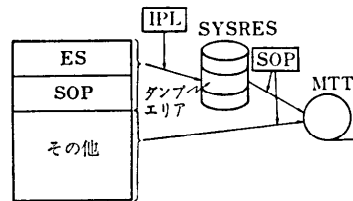


図 9 ダウン時のメモリ・ダンプ

IPL は、まず、ES および SOP のエリアを SYSRES のダンプ・エリアにダンプしたのち、ES, SOP をコアに広げる。その後、SOP が、SYSRES のダンプ・エリアおよびその他のコア・エリアを、MTT にダンプする。

(3) MT, LP などの登録

MT, LP の登録は、運転開始時には、オペレータの操作により行なわれる。しかし、回復時には、原則として、ダウン時点で使用していた MT, LP をそのまま使用するので、オペレータの操作は必要としない。これは、ダウン時処理中のトランザクションの処置のため、ダウン時使用していた磁気テープを必要としないためである。

MT, LP の使用法は、運転開始時、およびその後に使用法が変わった時点で、TDF に書き込まれる。ダウン回復時には、TDF から使用法を知ることができる。

(4) ダウン時処理中のトランザクションの処理

ダウン時には、端末より入力されたトランザクションは、種々の状態で停止している。これを放置したまま運転を再開すると、中央のファイルと端末オペレータの間で、くい違いが起こる。これをいかに解決するかが、システムの異常処理の中心をなす問題である。

マルス 105 で、回復時対象となるファイルは、SF および AF である。それらについて、以下の方針により、処置を行なっている。

(a) 座席の 2 重発売は、絶対さける。また、できる限り、座席のむだをなくす。

(b) 集計については、正確を期す。

トランザクションの中心を占める指定券の発売処理では、表 2 の方法により、処理される。すなわち、ダウン時点で、AF の A 系が書込み済みのものは、すべてファイルを更新後の状態にし、また、AF の A 系の書込み以前のものについては、すべてのファイルを更新前の状態にもどす。さらに、入力が行なわれたが出力が完了していないすべての端末を、ソフト的にロッ

表 2 ダウン時処理中トランザクションの処置

ダウン時処理ステータス	回復時	回復後 端末 操作	
		控 除	再 製
	何もせず	何もせず	何もせず
SFA W	SFAを更新前にもどす	同上	同上
SFB W	SFA, SFBを更新前にもどす	同上	同上
AFA W			
AFB W	AFBを更新	AFの情報よりSF, AFをもどす	AFの情報より発券を行なう
	何もせず	同上	同上

くし、以後、控除あるいは再製操作しか受け付けられないようにする。以上の処理を行なうために必要なトランザクションのステータス、ファイルを、更新前の状態にもどすためのデータなどは、すべてダウン時点のコアの情報を使用するため、これらの処理は、非常に短時間に行なわれる。

ダウン回復後、ロックされている端末のオペレータは、ダウン時の入力を無効としたければ控除操作、有効としたければ再製操作を行なう。これらいずれかの操作が行なわれたのちはロックを解除し、一般のファイル更新操作を許す。控除でファイルをもどしたり、再製で発券するには、AFに書き込まれているラスト・トランザクション情報を使用する。

4.4 ファイル回復

SFについては、両系のファイル破壊となりデータが失われた場合には、MTによるファイル回復を行なう。まず、ダンプMTより、運転開始時のファイルデータをもどし、その後、TRTにより、ダウン時点までのファイル更新を再現する。

この操作は、SFの全ファイルに対してのみではなく、デバイス単位に行なうこともできる。そのため、一般には、ファイル回復は、システムの稼動中に、対象のデバイスのみ使用停止して行なわれる。

一般に、2重系のものが両系とも破壊されるMTBFは、単独のMTBF、MDTより、 $MTBF^2/2MDT$ で近似される。これに実際の値を入れると、非常にまれにしか発生しないことがわかる。事実、マルス105では、MTよりのファイル回復を行なった実績はなく、また、過去10年以上のマルスの歴史においても、1回だけしか行なわれていない。

5. テスト方式

マルス105では、プログラムごとに種々の方式でテストを行なったが、以下に、オンライン・ユーザ・プ

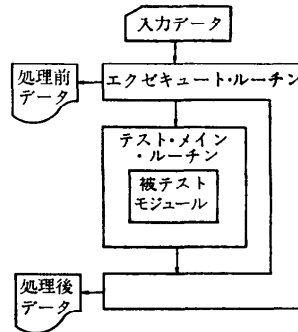


図 10 単体テスト

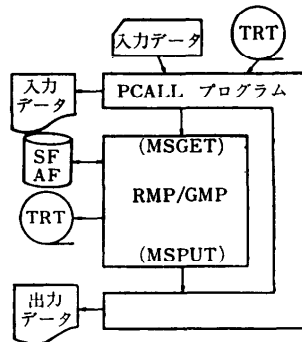


図 11 結合テスト

ログラムについて、そのテスト・ツールを中心にのべる。

5.1 単体テスト

単体テストは、図10に示す方法で行なった。エグゼキュート・ルーチンは、入力データのコアへの設定、処理前後の、コアの情報を打ち出す。テスト・メイン・ルーチンは、コア・テーブルなど共同で使用するデータを持ち、また、被テスト・モジュールへのリンク条件の設定などを行なう。単体テストにおいて、各モジュールは、原則として、全パスをテストした。

5.2 結合テスト

結合テストは、オンライン・ユーザ・プログラム全体を通すテストである。このため、前にのべたIPC機能を利用して、入力データを、端末からと同様のインタフェースでユーザ・プログラムに移す、PCALL (Pseudo-Call) プログラムを開発した。また、PCALLは、TRTを入力とすることもできるので、稼動後に、実際の端末からの入力データを流してみることも、可能である。

オンライン・ユーザ・プログラムは、システムの稼働後も、数多く機能変更が行なわれる。この場合、プログラムの修正を行なったため、在来の機能に悪影響を及ぼす恐れがある。そのため、プログラムの一応の機能確認ができるマスタ入力データを用意し、プログラム修正を行なった場合は、マスタ入力データを流し、在来の機能の確認を行なうことにしている。マスタ入力データの呼数は、使用開始時点では約1万コール、現在では約1万2,000コールとなっている。

6. むすび

以上で、マルス 105 のソフトウェアの概要をのべた。現在は、新幹線博多開業のためのシステムの増強、電話予約、団体予約システムとの結合などのため、ソフトウェアの改修を行なっている。システム稼働後のソフトウェアの発展性は、今後ますます重要な問題となるものと考えている。

(昭和49年4月3日受付)