

一般情報処理教育用ソフトウェア・システム (NITFOR) の開発

杉江日出澄・神藤 久・岡崎明彦・足達義則・尾崎正弘・
貝谷邦夫・川浦康成・織田昌信

情報処理教育センター
(1980年9月5日受理)

Development of a Software System (NITFOR) for General Information Processing Education

Hidezumi SUGIE, Hisashi KANDO, Akihiko OKAZAKI, Yoshinori ADACHI, Masahiro OZAKI, Kunio KAIYA, Yasunari KAWAURA and Masanobu Oda

Center of Information Processing Education
(Received September 5, 1980)

A software system (NITFOR) for the education of the general information processing has been developed. The system is made up of main task, five subtasks, and eleven tables. The tables play roles of interfaces between the tasks. The tables, the tasks, and component routines of the tasks are independent for each other, and work for each own given function. The main utility of this system is for the exercises of FORTRAN language and numerical calculation methods. The NITFOR will be further developed in the near future into NITFIV, and after that into NITSIX step by step. The microcomputer education will be included in the utilities of the NITSIX.

1. 緒言

昭和50年度に情報処理教育センターの設置が認められ、専任スタッフの陣容を整えると同時に情報処理教育専用のソフトウェアシステムの開発構想をとりまとめて、具体化に向けて作業を開始した。

専攻する学科とは原則的に無関係な共通的な教育、いわゆる一般情報処理教育を主な任務とする教育機関としての備えるべきソフトウェアのあり方を追求しつつ、ひとまず、昭和52年度開講に向けて、教育専用の一貫システムである NITFOR (Nagoya Institute of Technology FORtran education system) の基盤を完成させた。その後、実際に教育に活用した結果を踏まえて、改善すべき点などの補正を続け、ほぼ満足の状態に達したので、ここに報告する。

我国の情報処理教育センターは、ごく最近になって、情報化社会に対処してゆくために設置されたもので、設立後まだ日が浅い。そのために、教育機関に必要な教育

専用のソフトウェアを開発する機会がなく、また龐大な経費と人材を必要とするために、これまでは一般の業務処理用の汎用ソフトウェアで代用させていたのが実状である。幸いにして、本学においては、開発に必要な諸条件を整えることができたのは、関係者の協力とご支援のおかげである。

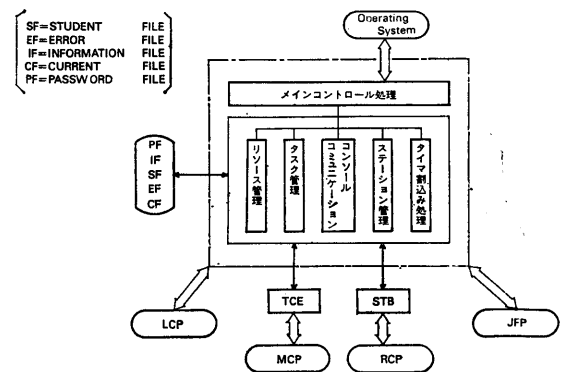


Fig. 1 Managements of main task

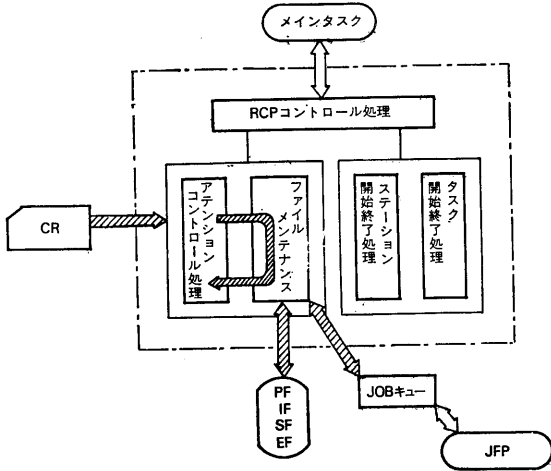


Fig. 2 Managements of reader control processor (RCP) task

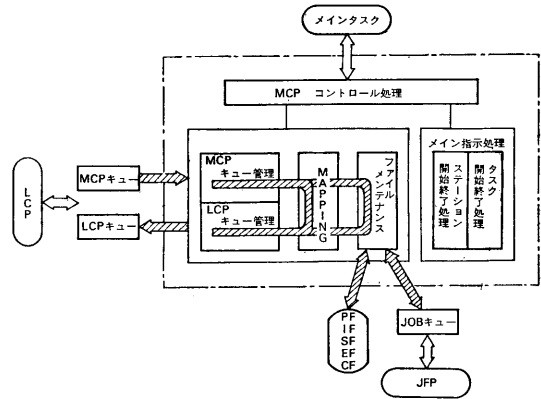


Fig. 4 Managements of message control processor (MCP) task

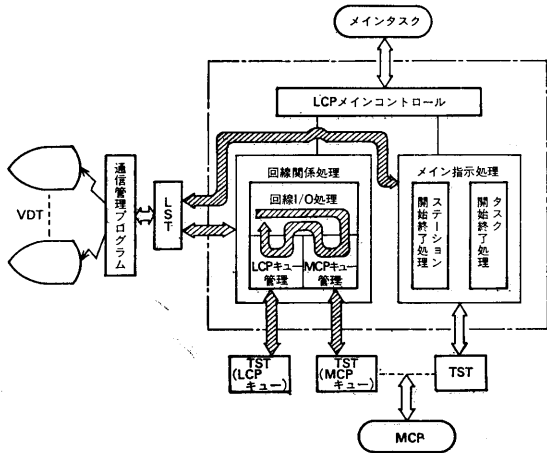


Fig. 3 Managements of line control processor (LCP) task

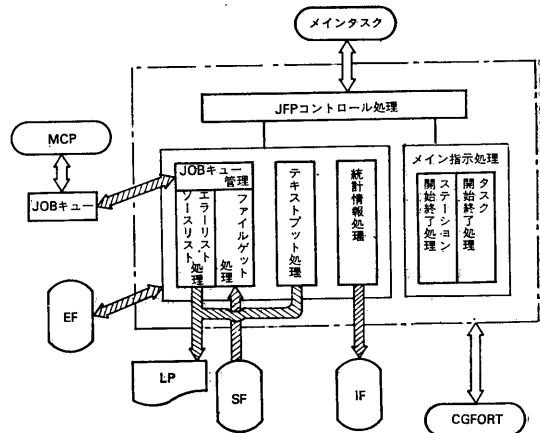


Fig. 5 Managements of job fortran processor (JFP) task

2. NITFOR システムの構成

NITFOR はメイン・モジュールと 5つのサブモジュールから構成され、マルチタスク構造となっている。各モジュールはタスク部とテーブル部より構成され、テーブル部によってタスク間の連絡を行っている。各タスクとテーブルの種類と機能をまとめて Table 1 と Table 2 に示す。

各タスクが行う処理とテーブル類との関連を図示すれば、それぞれ Fig. 1~Fig. 5 のごとくなる。また、これらの6つのタスク間の相互関係を概念図で示したのが Fig. 6である。

NITFOR を構成する各モジュールはロード・モジュール

ライブラリに格納されており、システム起動が指示された時点でコア上にローディングされ、常駐する。したがって、通常のバッチ処理で発生するジョブ毎のコンパイラローディングタイムやユーザプログラムローディングタイムは発生しない。各モジュールはプログラムが存在するタスク部と、プログラム内で使用するテーブル部が完全に分離した構造となっているために、端末の増設や変更の際にシステムのエンハンスが容易である。それぞれのモジュールは自分に与えられた一定範囲の処理を担当し、互いに独立形態をなす。また、1個のモジュールは複数個のルーチンから構成され、それらのルーチンも同様に独立形態をなしている。

3. NITFOR システム内での情報の流れ

NITFOR が外部から取り入れた情報や内部で発生した

Table 1 Functions of main task and five subtasks

タスク・モジュール	機 能	タスク・モジュール	機 能
(1) メイン処理 (Main)	<ul style="list-style-type: none"> ・ NITFOR 内モジュールのローディング制御 ・ // 各サブタスクの生成/消去制御 ・ // // 起動/停止制御 ・ 時間監視処理 ・ コンソール・入力コマンド処理 ・ ステーション開始/停止制御 ・ ファイル OPEN/CLOSE ・ サブタスクとの連絡制御 	(4) CR 制御処理 [RCP=Reader Control Processor]	<ul style="list-style-type: none"> ・ CR (card reader) 入力制御処理 ・ CR 入力メッセージ分析処理 ・ 学生ファイルに対するメンテナンス処理 ・ 言語プログラムとのインタフェース処理 (入力ジョブ・キュー接続処理, 出力ジョブ・キュー接続処理) ・ 運用コマンド処理 ・ 統計情報採取 ・ メインタスクとの連絡, JFP サブタスク, WTR サブタスクとの連絡 ・ ステーションの起動/停止処理
(2) 回線制御処理 [LCP=Line Control Processor]	<ul style="list-style-type: none"> ・ 回線/端末に対する入出力制御 ・ // ステーションの起動/停止処理 ・ メインタスクとの連絡, MCP との連絡 ・ 回線入力メッセージの MCP キュー接続, 回線出力メッセージの LCP キューよりの取り出し処理 	(5) 言語処理プログラム制御 [JFP=Job Fortran Processor]	<ul style="list-style-type: none"> ・ ステーションの起動/停止処理 ・ コンパイラとのインタフェース処理 (入力ジョブ・キュー取り出し, 出力ジョブ・キュー接続処理) ・ コンパイラ出力メッセージ編集処理 ・ 統計情報採取 ・ メインタスクとの連絡, RCP サブタスク, MCP サブタスク, WTR サブタスクとの連絡処理
(3) 入力メッセージ制御処理 [MCP=Message Control Processor]	<ul style="list-style-type: none"> ・ 回線入力メッセージの分析処理 ・ ステーションの起動/停止処理 ・ 回線に対する応答メッセージの作成処理, Mapping 処理 ・ 学生ファイルのメンテナンス処理 ・ 言語処理プログラムとのインタフェース処理 (入力ジョブ・キュー接続処理, 出力ジョブ・キュー取り出し処理) ・ 運用コマンド処理 ・ 統計情報採取 ・ メインタスクとの連絡, LCP サブタスク, JFP サブタスク, WTR サブタスクとの連絡 	(6) リスト処理 (WTR=Writer)	<ul style="list-style-type: none"> ・ ステーションの起動/停止処理 ・ コンパイラ及び MCP サブタスク, RCP サブタスク, JFP サブタスクとのインタフェース処理 (出力ジョブ・キュー取り出し処理) ・ メインタスクとの連絡, MCP サブタスク, RCP サブタスク, JFP サブタスクとの連絡処理

Table 2 Functions of tables

テ ー ブ ル	機 能	
	内 容	属 性
(1) TBL (Table)	<ul style="list-style-type: none"> ・ 各モジュールが共通使用する値 ・ NITFOR 全体を制御する情報 ・ コンスタント類 	NITFOR メインルーチン内に存在する。メインルーチンがサブタスクを起動する時に、このテーブルを知らせるので、メイン、サブの各タスクが共通に、このテーブルを利用する。
(2) TCE [Task Control Element]	<ul style="list-style-type: none"> ・ メインタスクがサブタスクを制御するのに必要な情報 ・ サブタスクがメインタスクと連絡するのに必要な情報 ・ 各サブタスクが使用すべき IFA (タスク対応のインタフェースエリア) のアドレス 	NITFOR メインルーチン内に存在する。メインタスクが、サブタスクを起動する時に、タスク対応にユニークな TCE を渡す。タスクごとに1個もつ。
(3) IFA (Interface Area)	<ul style="list-style-type: none"> ・ ワーク・エリア ・ レジスタ退避エリア ・ タスク間で引き渡す情報類 	NITFOR メインルーチン内に存在する。メインタスクがサブタスクを起動するときに、タスク対応にユニークな IFA のアドレスを TCE にセットして渡す。IFA の使い方は、各タスクごとに異なる。
(4) STB (Station Table)	<ul style="list-style-type: none"> ・ ステーションの状態, 制御を管理する情報 ・ ステーション下のデバイスのアドレス 	NITFOR メインルーチン内に存在し、各ステーションごとに1個もつ。
(5) ME (Mutual Element)	<ul style="list-style-type: none"> ・ 画面の制御情報 ・ ファイル・メンテナンスの制御情報 	NITFOR メインルーチン内に存在し、MCP, JFP が使用する。MCP が使用する場合、START キー押下時に確保し、END キー押下時に解放される。ME プールより確保、解放し、エリアの有効利用をはかっている。

(6) IOA Input Output Area	<ul style="list-style-type: none"> RCP 及び JFP で、各ファイルをアクセスする時に使用される I/O バッファ 	実体は、ダイナミックに確保する為、その都度変わり、RCP, JFP が使用する。
(7) IJQ Input Job Queue	<ul style="list-style-type: none"> 実行を行う為に必要な情報 ソース・プログラムの存在するファイルの情報 	NITFOR メインルーチン内に存在する。MCP 及び RCP が登録し、JFP が登録された順に取り出して、処理する。
(8) OJQ Output Job Queue	<ul style="list-style-type: none"> 出力先に関する制御情報 出力結果の存在するファイルの情報 	NITFOR メインルーチン内に存在する。JFP, MCP が登録し、MCP, WTR が登録されたエントリを出力先別に取り出して処理する。
(9) LST Line Status Table	<ul style="list-style-type: none"> DECBB (data event control block: 通信制御プログラムとのインタフェース制御ブロック) 回線の入出力ステータスを管理する情報 回線入出力バッファのアドレス 	NITFOR メインルーチン内に存在し、LCP が使用する。回線対応に 1 個ずつ必要。
(10) TST Terminal Status Table	<ul style="list-style-type: none"> 端末の入出力ステータスを管理する情報 LCP キュー/MCP キュー 	NITFOR メインルーチン内に存在し、LCP/MCP が使用する。LCP が端末入力メッセージの処理を MCP に依頼する時、MCP キューとなり、MCP が端末出力メッセージを LCP に依頼する時、LCP キューとなる。端末対応に 1 個ずつ必要。
(11) DCB Data Set Control Block	<ul style="list-style-type: none"> ファイル定義に関する情報 	NITFOR メインルーチン内に存在し、データ・セット対応に必要である。

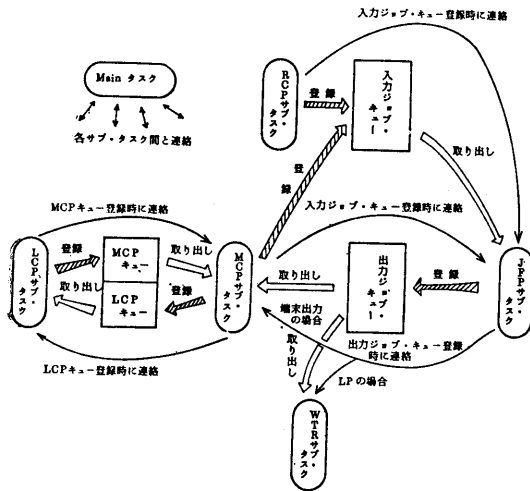


Fig. 6 Interactions between tasks

情報がどのような経路を経て処理されてゆくかを図示したものが Fig. 7 である。

(1) CR (card reader) からデータを入力した場合の情報の流れ

RCP がデータを入力しながら、入力したデータにエラーがないかを判定する。エラーがある場合、IF, EF に情報を移し、出力ジョブキューに EF の情報を渡す。WTR が出力ジョブキューから情報を得て、EF から内容を取り出して LP (line printer) に出力する。エラー

がない場合は、SF を更新し、実行指示の形態に従って処理を実行する。例えば Compile & Go の指示があれば、入力ジョブキューに登録し、JFP が入力ジョブキューから情報を得て、SF からデータを取り出して CGFORT に渡す。結果を出力 Spool に出力し、出力ジョブキューに登録する。WTR はキューを取り出し、出力 Spool の内容を LP に出力する。

(2) VDT (video data terminal) からデータを入力した場合の情報の流れ

LCP がデータを通信管理プログラム (BTAM) を経て VDT からのデータを入力する。入力したデータは MCP キューに登録される。MCP は MCP キューからデータを得てエラーかどうか判定する。

初期条件設定以前にエラーとなった場合、LCP キューにその内容に相当するエラーメッセージに登録する。LCP は LCP キューからデータを取り出し、BTAM を経て VDT に表示する。初期条件設定以後にエラーとなった場合、IF にエラー内容を書き込み、LCP キューにエラーメッセージを登録、同じく LCP はそのメッセージを VDT に表示する。VDT からの修正等の作業はすべて BTAM-LCP 経由で MCP が実行するが、この時一時的なファイルとして CF を使う。

プログラムの新規登録や修正が終了したのち、実行指示が行われると、入力ジョブキューに登録され、同時に CF の内容が SF に移され、JFP によって実行がなされる。結果は出力 Spool に出され、出力ジョブキューに

登録される。MCPがこれを取り出し、LCP-BTAMも可能である。
M経由でVDTに結果を表示する。LPに出力すること

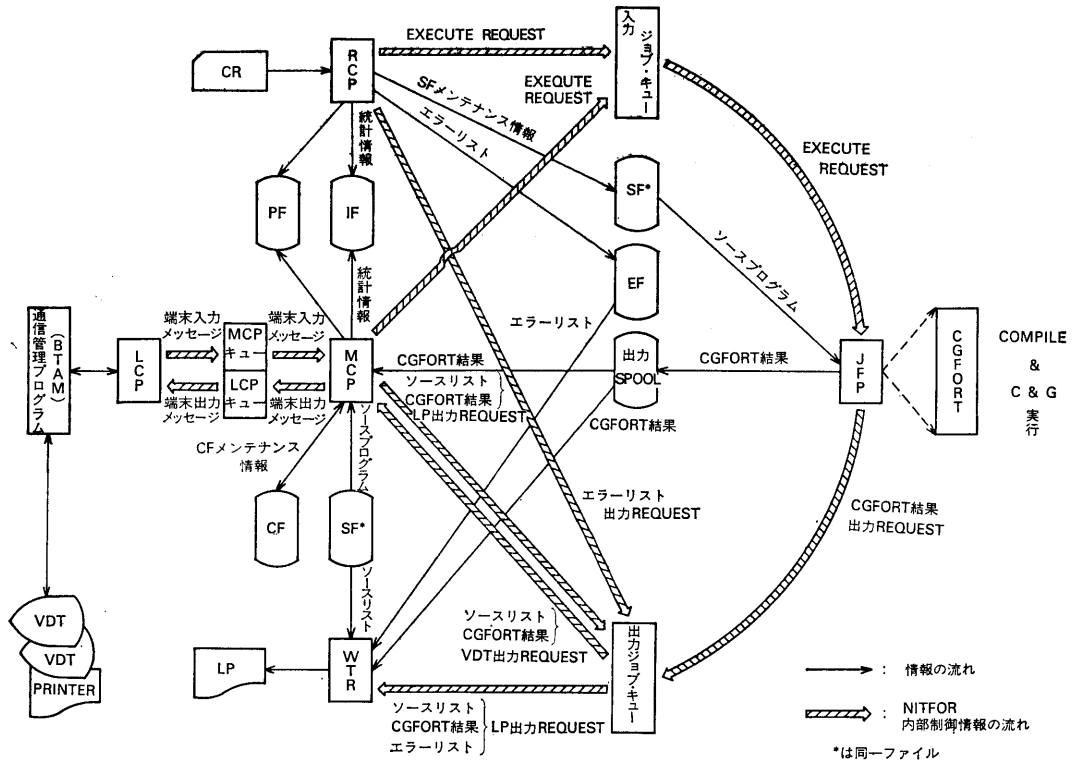


Fig. 7 Information flow chart of NITFOR system

4. NITFOR システムの特色

システム的な見地からではなく、教育の実施面から見た特色を以下に列記する。

(1) 教務統計情報の自動取得と集計処理機能

学生番号、プログラム名称、実行指示と結果の判別、入力媒体の判別、コンパイルと実行のCPUタイム、プログラムメモリサイズ、エラー統計、LP用紙の使用量、プログラムのカード換算による大きさなど約430項目を採取して、学習指導と成績管理を行っている。

(2) VDTの画面管理と操作性の体系化

画面を3区域に分け、ガイド用区域によって学生のオペレーション指導を行っている。初期設定モード、Inputモード、Updateモードの3つの作業モードに系統化し、修正用コマンド、画面移動用コマンドなどを初心者でも直感的に操作できる体系化をはかっている。

(3) 入力媒体の多様化

端末台数に較べて受講学生数は著しく多い。したがって、パンチカードでもマークカードでもVDT端末と全

く同じことができるようにしておけば、システムの柔軟性が向上する。作業モードやコマンドなどの思想はVDTと統一した体系をとっている。

(4) 学生用ファイルの開放

学生各人に一定の大きさのファイルを3本ずつ与え、極めて簡単に書き込んだり、編集、消去ができる構造となっている。授業科目ごとに指導教官の希望通りのファイルを与えることができるようになっている。

(5) カリキュラム通りの自動運転

1日の演習のスケジュール通りに、休憩もさせながら、自動運転する機能をもつ。少数のスタッフで沢山の装置類を管理しながら、多人数の教育に対処しなければならない実状対策の1つである。

5. 結言

一般情報処理教育のための専用ソフトウェアの1つの方式を実現させたNITFORについて述べた。しかし、NITFORが万全であるとは考えていない。引き続き、ハードウェア構成の増強に伴うシステムの拡張や科学技術

計算ライブラリーを常駐型 FORTRAN でも使用可能にしたり、ファイルの効率向上など、手を加えねばならないことが沢山ある (NITFIV 計画)。さらに、最近の注目の目的であるマイコンに関する教育についても考慮してゆかねばならない (NITSIX 計画)。その他に、OCR 関係、図形処理、FORTRAN 以外の言語、ビジネスゲームによる経営学の演習などさまざまな要請があるが、一歩ずつ着実な開発を積み上げて、教育用総合ソフトウェ

アシステムへと NITFOR を育成させてゆくつもりである。

謝 辞 本研究は昭和52, 53, 54年度文部省大学教育方法等改善経費の援助を受け、また、ファコム・ハイタック株式会社の協力を得て行った開発研究の一環である。関係者各位から多くのご協力をいただいたことをここに記して深く謝意を表する次第である。