

## プログラミング教育環境支援システム開発に関する研究

塚本邦昭

芦屋大学教育学部産業教育学科

オフィス・コンピュータを利用した情報処理教育の一環としての、プログラミング言語の基礎的な教育で、学生が汎用機オペレーティング・システムやエディタ、コンパイラのコマンドを直接入力してコンピュータを利用している。そのため学生が複雑な操作を習得しなくてもよいような単純なシステムが必要である。そこで、学生が自然な形で十分な教育効果が可能な、プログラミング教育環境を支援をする為のシステムについてその設計・試作開発を行った。これは「CAI環境」を網羅したシステム開発である。

**Study on the development of supporting system for  
programming educational environment**

Ashiya University, Faculty of Education, Department of Industrial education

13-22 Rokurokuso-Cho, Ashiya City, Hyogo 659, Japan

Students often use office-computers in computer education including fundamental instruction of programming language with directly inputting commands of operating system, editors or compilers. It is need to develop the comfortable system for students without learning difficult operations. Then, the plan and prototypal development of supporting system, being able to give sufficient educational effect for students with natural use, for programming educational environment was done. This is the development of system throughout computer assisted instruction environment.

## 1. 本研究の意義と目的

今日の我が国では、高度情報化への要求が高まり、社会の様々な分野でのコンピュータ利用が推進されている。文科系大学といえども教育・研究などの情報化への取り組みが必要である。コンピュータの普及に伴わない、コンピュータを使える人を必要としている。ため、大学などの各教育機関における情報関係学部や学科でのコンピュータ教育に大きな関心が増加している。

教育におけるコンピュータの利用方法が、今後の教育の在り方にも重要な影響を与えている。

オフィス・コンピュータを利用した情報処理教育の一環としての、プログラミング言語の基礎的な教育では、初心者向けのコンピュータ利用のための対話型プログラミング環境には、プログラミング支援と呼べるものはあまりなく、支援用ソフトウェアの研究開発が、ほとんど行われずにコンピュータメーカーが、汎用機で提供しているオペレーティング・システムやエディタ、コンパイラを直接学生がコマンドを入力し、使用しているのが現状である。しかし、多くのオペレーティング・システムは多くの機能をもっているが、その資源を有効に使うためには、学生の理解を妨げる。多くの詳細を知らなければならない。それらは使いがってが悪く、教育を行う上で大きな障害となる。そこで、初心者にも自然な形で十分な教育効果が可能な、プログラミング教育環境を支援をするシステムについてその設計・試作開発を行った。

学校教育において行行情報処理教育は、単に操作の練習をするのではなく、ある仕事をさせるプログラムを作ろうとするとき、どの部分にどのような問題点があるかということ进行分析し、プログラムを作る時の背後にひそむ問題点を把握させるための問題解決能力を育てる事が必要で、各人の思考力を高め、問題解決の能力を育てる事が何よりも重要な課題である。

そして、種々の部分的問題点を解析することの積み重ねの過程を通じて、より大きな問題に対応する知識技能が習得される。プログラムが完成するということを実感をもって体得し、コンピュータで表されたものを読み取り、コンピュータによって自分の考えを表現できる能力を養成することが大切である。

プログラミング教育環境支援システムの目的は、複雑です。面倒なオペレーティング・システムやエディタ、コンパイラの操作手続きを経ず様々な水準の学生が、直ちにコンピュータを利用出来るような効率良く便利な環境に改善するものである。したがって、学生は教育に最適な環境で効率的にプログラミング言語教育を進めることが出来た。

コンピュータを利用する場合に学生は、まず、機器の電源を投入すると、次に、それを如何に稼働させるかを考えなければならない。すなわち、学生はまずオペレーティング・システムの操作の方法を習得しなければならない。しかしながら、オペレーティング・システムの操作は非常に複雑で、この過程を経ずして直ちにプログラミングができないとすると、プログラミング教育を受ける学生は、オペレーティング・システムの操作方法を習得した人だけに限定せざるを得ない。

このプログラミング教育環境支援システムは、オペレーティング・システムを知りつくしていない学生が、簡単な方法で操作出来るような環境を開発することが必要と考え開発した。学生との対話を通じて段階的に進めることによって学生が操作方法を習得する時間の節約が図られた。

今回研究・開発し教育に使用してきたこのシステムが、他の教育機関でコンピュータを利用し、プログラミング教育を行っている方に参考になればと思い報告いたします。

最後の頁にプログラミング教育環境支援システムのプログラムリストを公開いたします。

## 2. 教育環境支援システムの概要

### 1) システムの条件

このプログラミング教育環境支援システムを利用する上で、学生が複雑な操作を習得しなくてもよいように、可能な限り単純なシステムの開発を行った。

学生が必ずしも、すべての情報を指定しなくても、いくつか可能な経路のうちの一つを一意に決めるだけの情報があればよいということである。そもそも、プログラミング教育環境支援システムの利用者は、将来

プログラム作成を目的としたコンピュータ技術者ではないことを忘れてはならない。

プログラミング教育環境支援システムの開発に際して、下記の諸条件を満たすことを重要課題とした。

#### (1) 簡便性

プログラミング教育環境支援システムの利用の仕方は、可能な限り単純にする。

#### (2) 非手続き性

出来る限り高水準にすべきであり、学生は何が欲しいかを指定するだけで、それをどのようにやればよいかはプログラミング教育環境支援システムにまかせる。

### 2) システムの構造

開発を行ったシステムは、オフィス・コンピュータでのオペレーティング・システム (UNIOS/F5) 上でCOBOL言語によって開発を行った。

このオペレーティング・システムは、ワークステーション環境下でバッチ処理からインクワイアリ処理までの広範囲な利用形態をサポートすることが出来、利用形態をサポートすることが出来る。オペレーティング・システムが持つ多くの機能は、コマンドという形で提供される。

コマンドは、ファイルの作成、プログラムの翻訳や他のプログラムを呼出すなどの機能を実行出来る。

開発を行ったプログラミング教育環境支援システムは、モジュールと呼ばれる小プログラムの集まりである。このモジュールの中には、COBOLプログラムよりオペレーション・システムのコマンドを直接操作する機能がある。

図1. でも明らかなように、プログラミング教育環境支援システムは、あたかもシェルのようにオペレーティング・システムやエディタ、コンパイラを包み込みユーザとの間に介在して、オペレーティング・システムやエディタ、コンパイラの操作手続きが簡単に効率良く行われるように、プログラミング教育環境支援システムが自動的に教育の支援を行う。

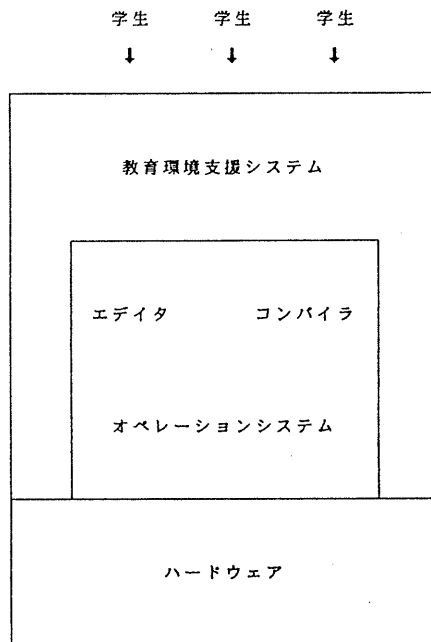


図1. プログラミング教育環境支援システム

### 3) システムの機能

このプログラミング教育環境支援システムは、原始テキストの作成及び、編集を行い尚且つ、そのプログラムのテキストを行う時にオペレーティング・システムと学生との間のインターフェース機能を提供する。

#### (1) 編集の操作の支援

原始テキストの作成及び、編集エディタを起動させ原始テキストの作成及び、編集の操作の支援を行い、プログラムファイルの管理を行う。

#### (2) 原始テキストの翻訳と結合

入力したプログラムをコンパイラにより翻訳し、実行時に必要な他のプログラムとリンクし、すぐ実行出来るオブジェクトに変換してディスク・ファイルに登録する。

### (3) オブジェクトのテスト

オブジェクト・プログラムの実行テストとテスト時に、オペレーティング・システムとテストプログラムの間に介在し、学生とのデータの受け渡しを行う。

## 4. 支援システムの操作方法

プログラミング教育環境支援システムと学生との実際の対話の概略を示すと、まず、端末よりプログラミング教育環境支援システムの操作の方法は、学生がモニター上のシステムの日本語化されたメニューやメッセージに対して、YES、NOなどの簡単な選択方法で対話を行う。

モニタの画面は、二つの領域に分かれている。(フレーム1参照) 上部領域は、学生が入力する項目とそれに対する注意事項を示し、下部領域は、本システムによって用いられ、操作に対する説明が表示されている。特に赤字で表示される時は警告時である。

最初のフレーム(フレーム1)で、学生はプログラミング言語のCOBOLとFORTRANのいずれかを選択し言語種別を定義する。

COBOLを作成するときは「1」、FORTRANのときは「2」を入力する。もし、その選択方法に誤りがあれば、システムより警告メッセージが表示され、また処理の選択が正しければフレーム2に進みます。

次のフレーム(フレーム2)では、学生はテキスト名を入力すると、システムで自動的に学生の名前とテキスト名を合成したプログラムの名称が新しく定義される。学生ごとに異なる名称になるように、プログラムを一元的に管理を行う事が出来る。

フレーム3では、学生がプログラムを新しく作るのか、すでに、登録してあるプログラムの変更かのどちらかの選択を行う。

言語は何ですか?  
1 COBOL  
2 FORTRAN  
3 終了  
番号を入力して下さい—  
ENTER...次に進む

フレーム1

原語テキスト名?—   
ENTER...次に進む  
PF10...前画面に戻る

フレーム2

何をしますか?  
1 プログラムの作成  
2 プログラムの修正  
3 プログラムのみの実行  
番号を入力して下さい—  
ENTER...次に進む

フレーム3

コンパイルしますか?—  
(Y/N)  
ENTER...次に進む  
PF10...前画面に戻る

フレーム4

FORTRAN でプログラムを  
使用していますか?—  
(Y/N)  
ENTER...次に進む  
PF10...前画面に戻る

フレーム5

コンパイラリスト いますか?—  
(Y/N)  
ENTER...次に進む  
PF10...前画面に戻る

フレーム6

プログラムを新しく作成するときには「1」、修正をするときは「2」を入力する。もし、その選択方法に誤りがあればシステムより警告メッセージが表示され、また処理の選択が正しければエディタが起動されてプログラムの作成および修正が出来る。

プログラムの入力および修正が終了すればフレーム4の、プログラムの翻訳の有無の選択を行う画面に切り替わる。翻訳を行うときには「Y」、翻訳を行わ

いときは「N」、と入力します。

特にFORTRAN言語の時には、フレーム5の、原始テキストの中でサブルーチンを使用しているかどうかの指示の画面になり、サブルーチンを使用していれば「Y」、使用していなければ「N」、と入力する。

次にフレーム6の、翻訳リストの印刷を行うかどうかの選択を行う画面になる。

リストを作成するのであれば「Y」、作成しないときは「N」、と入力する。

翻訳時には、モニタ画面に「コンパイル中」と表示され、翻訳が終了するが、もし翻訳時にエラーが発生すると、モニタ画面にそのむね表示される。

以上がプログラミング教育環境支援システムの操作方法についての簡単な説明である。

	新方式	従来の方式
システムの立ち上げ	学生No テキスト名 3桁	EDITOR (EDITORプログラムの起動) プログラム名 6桁 ライブラリ名 6桁 言語種別@CBL又は@FTN
プログラムの翻訳	COBOL 言語種別を"1"入力	COBOL STORE-@YES, ERRL-@YES, REP-@YES, SRCL-@YES, PUBAUT-@YES, SRC- 原始テキスト名、ライブラリ名
	FORTRAN 言語種別を"2"入力	FORTRAN STORE-@YES, ERRL-@YES, REP-@YES, SRCL-@YES, PUBAUT-@YES, SRC- 原始テキスト名、ライブラリ名
	FORTRAN でリンク編集を使用している時 "Y"入力 (リンク済)	LINKED MAIN-FTMAIN, NAME-ACC, REP-@YES, PUBAUT-@ALL
システムの終了	"Y"入力	SIGNOFF

表1 従来の方式と新方式の相違点

## 5. システムの相違点

従来のコンピュータでのプログラミング教育の行い方は、プログラミング言語の教育を行いながら、実習用の汎用コンピュータのオペレーティング・システムそのものの操作方法を学生が習得してから、プログラムを作成しコンピュータで実行することを行っていた。

旧方式と新方式との相違点を表1に基づき表します。

### 1) エディタの起動方法

旧方式は、EDITORコマンドを入力し、プログラム名6桁とライブラリ名6桁、そしてプログラム種別としてCOBOLは@CBL、FORTRANは@FTNと入力。

新方式は、プログラム名3桁と学生NO、を入力すると、EDITORが起動されたのちにプログラム名が自動的に編集され、学生が使用するライブラリは、システムより自動的に割り当てが行われる。

### 2) プログラムの翻訳方法

旧方式は、COBOLの翻訳を行うには、COBOL STORE-@YES、ERRL-@YES、REP-@YES、SRCL-@YES、PUBAUT-@YES、SRC-原始テキスト名。

FORTRANの翻訳を行うには、FORTRAN STORE-@YES、ERRL-@YES、REP-@YES、SRCL-@YES、PUBAUT-@YES、SRC-原始テキスト名。

新方式は、言語の種別の選択時にCOBOL=1と、FORTRAN=2の選択を行い、次に翻訳に有無の問い合わせに対し"Y"、"N"と入力する。

### 3) 翻訳時のサブルーチン使用法

旧方式は、LINKEDIT MAIN-FTMAIN、NAME-ACC、REP-@YES、PUBAUT-@ALLと入力。

新方式は、FORTRANのサブルーチンの使用の有無の問い合わせに対し、“Y”、“N”と入力する。

#### 4) システムの終了方法

旧方式は、“SIGNOFF”と入力する。

新方式は、プログラム終了しますかの問い合わせに対し“Y”と入力するだけでシステムが終了する。

以上のように、従来のシステムでの欠点は、コマンドという形の命令を学生が記憶し、操作のつど学生が記憶をたどりながらコマンドを入力しなければいけなかったが、開発したシステムでは、このような不便を解消し、各フレームでの日本語による質問に対し、1、2または、Y、Nと入力するだけでよく、操作を習得する必要がなく、初心者の学生にも簡単な操作でプログラミング教育を効率的に行える環境にした。

#### 5. 最後に

プログラミング言語教育支援システムは、コンピュータと学生の間を介し、学生と会話をしながら自然かつ簡単にコンピュータの制御が行える。いつでも学生が利用でき、学生は、より多くの時間を問題そのものの解決にあてることができ、プログラミング教育環境支援システムのおかげで学生のプログラミング言語教育の効果は格段に上がり、効率的に授業を進めることが可能になった。

現在はオフィス・コンピュータでの情報教育環境から、パソコンによるイーサネット高速LANでの教育環境が整えられて、新しい教育方法を研究し教育を行っていくのが今後の課題である。

#### 参考文献

デイヴィッド・R・バーストウ：ソフトウェア開発支援環境  
TBS出版会

FACOM UNIOS/F5プログラム開発手引書  
富士通株式会社

FACOM UNIOS/F5 JIS COBOL  
E使用手引書  
富士通株式会社

FACOM UNIOS/F5 システムコマンド集  
富士通株式会社

プログラミング教育環境支援システムのプログラムリスト

```

003510 PROCEDURE ..... DIVISION,
003520 ..... 7 0 2 1 5 (1.0)
003530 ..... SECTION,
003540 ..... SECTION,
003550 PROCESS-RUN ..... SECTION,
003560 PROC-ENT. ....
003570 PERFORM ..... MAIN-RUN,
003580 PERFORM ..... END-RUN,
003590 STOP .....
003600 PROC-EXIT. ....
003610 EXIT. ....
003620 .....
003630 .....
003640 ..... 7 0 2 1 5 4 - 1 5 (3.0)
003650 ..... SECTION,
003670 MAIN-ENT. ....
003680 ..... 5715 4725 / 531 177
003690 MOVE SPACE, TO CDATE
003700 .....
003710 ..... CHED-512
003720 ..... CHED-513
003730 ..... CHED-514
003740 ..... CHED-515
003750 ..... CHED-516
003760 ..... CHED-517
003770 ..... CHED-518
003780 ..... CHED-519
003790 ..... CHED-5110
003800 ..... CHED-5111
003810 ..... CHED-5112
003820 MOVE SPACE, TO WHEO-S12
003830 ..... WHEO-S13
003840 ..... WHEO-S14
003850 ..... WHEO-S15
003860 ..... WHEO-S16
003870 ..... WHEO-S17
003880 ..... WHEO-S18
003890 ..... WHEO-S19
003900 ..... WHEO-S110
003910 ..... WHEO-S111
003920 ..... WHEO-S112
003930 ..... WHEO-S113
003940 ..... PSW-AREA,
003940 MAIN-10,
003950 OPEN INPUT IHSO-FL,
003960 OPEN OUTPUT OHSO-FL,
003970 MOVE SPACE TO WSW-ERR,
003980 PERFORM INITIAL-RUN,
003990 MOVE WKS1H-NO TO NUS-DATA,
004000 .....
004010 .....
004020 ..... VS 7-7 AND IIF 7-7
004030 .....
004040 OPEN I-O WS-FILE,
004050 CALL 'IDLEN' USING NUS-KEY NUS-LIMIT,
004060 MOVE NUS-BLCK TO NUS-SAVE,
004070 MAIN-12,
004080 READ WS-FILE INVALID KEY GO TO MAIN-13,
004090 GO TO MAIN-13,
004100 MAIN-13,
004110 IF NUS-BLCK = 99999 GO TO MAIN-14,
004120 ADD 1 TO NUS-BLCK,
004130 IF NUS-BLCK = NUS-SAVE GO TO MAIN-14,
004140 IF NUS-BLCK NOT = NUS-LIMIT GO TO MAIN-12,
004150 IF NUS-SAVE = ZERO GO TO MAIN-14,
004160 MOVE 0 TO NUS-BLCK,
004170 GO TO MAIN-12,
004180 MAIN-14,
004190 MOVE 'ERR' TO WSW-FLAG,
004200 CLOSE WS-FILE,
004210 GO TO MAIN-17,
004220 MAIN-15,
004230 CLOSE WS-FILE,
004240 .....
004250 OPEN I-O IIF-INST,
004260 MOVE WS-SYDZOKU TO IIF-INST,
004270 MOVE WS-HEN TO IIF-HEN,
004280 MOVE WS-HST TO IIF-HST,
004290 READ IIF-INST INVALID KEY MOVE 'ERR' TO WSW-FLAG
004300 ..... CLOSE IIF-HST
004310 GO TO MAIN-17,
004320 ..... MOVE IIF-HAIE TO WH2-HAIE CH2-HAIE,
004330 IF IIF-SYDZOKU = 3
004340 MOVE 'X' TO WH2-KUHI CH2-KUHI
004350 ELSE
004360 IF IIF-SYDZOKU = 4
004370 MOVE 'Z' TO WH2-KUHI CH2-KUHI
004380 ELSE
004390 MOVE IIF-CLASS TO WH2-KUHI CH2-KUHI,
004400 MOVE IIF-NO TO WH2-NO CH2-NO,
004410 MOVE IIF-SYDZOKU TO WH2-GAKKO CH2-GAKKO,
004420 MOVE WDATE-H TO WCHECK1-H,
004430 MOVE WDATE-D TO WCHECK1-D,
004440 IF IIF-SYDZOKU > 2 GO TO MAIN-16,
004450 IF W-CHECK1 > 9999
004460 IF WDATE-Y > IIF-HEN
004470 COMPUTE WH2-HEN = WDATE-Y - IIF-HEN + 1
004480 ELSE
004490 MOVE 1 TO WH2-HEN
004500 ELSE
004510 IF WDATE-Y > IIF-HEN
004520 COMPUTE WH2-HEN = WDATE-Y - IIF-HEN
004530 ELSE
004540 MOVE 1 TO WH2-HEN,
004550 MOVE WH2-HEN TO CH2-HEN,
004560 IF CH2-GAKKO = 1
004570 MOVE 'A' TO WH2-HEN,
004580 IF CH2-HEN = 4 MOVE 'ERR' TO WSW-FLAG,
004590 IF CH2-GAKKO = 2
004600 IF CH2-HEN > 2 MOVE 'ERR' TO WSW-FLAG,
004610 MAIN-16,
004620 CLOSE IIF-INST,
004630 MAIN-17,
004640 CLOSE IHSO-FL,
004650 CLOSE OHSO-FL,
004660 IF WSW-FLAG = 'ERR' GO TO MAIN-EXT,
004670 MAIN-30,
004680 .....
004690 ..... 7 0 2 1 5 7 7
004700 .....
004710 OPEN INPUT IHSO-FL,
004720 OPEN OUTPUT OHSO-FL,
004730 MOVE SPACE TO WSW-ERR,
004740 PERFORM WORO-RUN,
004750 CLOSE IHSO-FL,
004760 CLOSE OHSO-FL,
004770 OPEN INPUT IHSO-FL,
004780 OPEN OUTPUT OHSO-FL,
004790 MOVE SPACE TO WSW-ERR,
004800 PERFORM IEXHANE-RUN,
004810 CLOSE IHSO-FL,
004820 CLOSE OHSO-FL,
004830 .....
004840 OPEN INPUT IHSO-FL,
004850 OPEN OUTPUT OHSO-FL,
004860 MOVE SPACE TO WSW-ERR,
004870 PERFORM PLAY-RUN,
004880 CLOSE IHSO-FL,
004890 CLOSE OHSO-FL,
004900 .....
004910 IF WSW-FLAG = 'CAN' GO TO MAIN-30,
004920 IF WSW-FLAG = 'ERR' GO TO MAIN-EXT,
004930 MAIN-30,
004940 .....
004950 ..... 7 0 2 1 5 7 7
004960 .....
004970 OPEN INPUT IHSO-FL,
004980 OPEN OUTPUT OHSO-FL,
004990 MOVE SPACE TO WSW-ERR,
005000 PERFORM FORI-PLAY-RUN,
005010 CLOSE IHSO-FL,
005020 CLOSE OHSO-FL,
005030 .....
005040 IF WSW-FLAG = 'CAN' GO TO MAIN-40,
005050 IF WSW-FLAG = 'ERR' GO TO MAIN-EXT,
005060 IF WSW-FLAG = 'X' GO TO MAIN-30,
005070 .....
005080 ..... 7 0 2 1 5 7 7
005090 .....
005100 OPEN INPUT IHSO-FL,
005110 OPEN OUTPUT OHSO-FL,
005120 MOVE SPACE TO WSW-ERR,
005130 PERFORM MAIN-RUN,
005140 CLOSE IHSO-FL,
005150 CLOSE OHSO-FL,
005160 .....
005170 IF WSW-FLAG = 'CAN' GO TO MAIN-40,
005180 IF WSW-FLAG = 'ERR' GO TO MAIN-EXT,
005190 IF WSW-FLAG = 'X' GO TO MAIN-30,
005200 .....
005210 MOVE 'END' TO WSW-FLAG
005220 GO TO MAIN-END,
005230 GO TO MAIN-60,
005240 MAIN-35,
005250 ..... FORI-RUN 7
005260 IF WSW-FLAG = '3'
005270 MOVE 'Y' TO WH1-RUN
005280 MOVE 'H' TO WH1-FLO
005290 MOVE 'Y' TO WH6-COMPILE
005300 GO TO MAIN-83,
005310 IF WSW-FLAG = '4'
005320 MOVE 'H' TO WH1-RUN
005330 MOVE 'Y' TO WH1-FLO
005340 MOVE 'Y' TO WH6-COMPILE
005350 MOVE 'I' TO WH6-FLO
005360 GO TO MAIN-83,
005370 IF WSW-FLAG = 'X'
005380 MOVE 'END' TO WSW-FLAG
005390 GO TO MAIN-END,
005400 MAIN-60,
005410 .....
005420 EDITOR 4 57A
005430 .....
005440 OPEN INPUT IHSO-FL,
005450 OPEN OUTPUT OHSO-FL,
005460 MOVE SPACE TO WSW-ERR,
005470 PERFORM EDITOR-RUN,
005480 CLOSE IHSO-FL,
005490 CLOSE OHSO-FL,
005500 .....
005510 IF WSW-FLAG = 'CAN' GO TO MAIN-60,
005520 IF WSW-FLAG = 'ERR' GO TO MAIN-EXT,
005530 IF WSW-FLAG = 'X' GO TO MAIN-30,
005540 .....
005550 ..... 7 0 2 1 5 7 7
005560 .....
005570 OPEN INPUT IHSO-FL,
005580 OPEN OUTPUT OHSO-FL,
005590 MOVE SPACE TO WSW-ERR,
005600 PERFORM COMPIL-RUN,
005610 CLOSE IHSO-FL,
005620 CLOSE OHSO-FL,
005630 .....
005640 IF WSW-FLAG = 'CAN' GO TO MAIN-60,
005650 IF WSW-FLAG = 'ERR' GO TO MAIN-EXT,
005660 IF WSW-FLAG = 'X' GO TO MAIN-30,
005670 MAIN-80,
005680 .....
005690 ..... 7 0 2 1 5 7 7
005700 .....
005710 OPEN INPUT IHSO-FL,
005720 OPEN OUTPUT OHSO-FL,
005730 MOVE SPACE TO WSW-ERR,
005740 PERFORM LISTING-RUN,
005750 CLOSE IHSO-FL,
005760 CLOSE OHSO-FL,
005770 .....
005780 IF WSW-FLAG = 'CAN' GO TO MAIN-70,
005790 IF WSW-FLAG = 'ERR' GO TO MAIN-EXT,
005800 IF WSW-FLAG = 'X' GO TO MAIN-30,
005810 .....
005820 ..... 7 0 2 1 5 7 7
005830 .....
005840 .....
005850 OPEN INPUT IHSO-FL,
005860 OPEN OUTPUT OHSO-FL,
005870 MOVE SPACE TO WSW-ERR,
005880 PERFORM FORI-PLAY-RUN,
005890 CLOSE IHSO-FL,
005900 CLOSE OHSO-FL,
005910 .....
005920 IF WSW-FLAG = 'CAN' GO TO MAIN-80,
005930 IF WSW-FLAG = 'ERR' GO TO MAIN-EXT,
005940 MAIN-86,
005950 .....
005960 SUBROUTINE CALL 7

```

```

005720-
005900 OPEN INPUT IHSO-FL.
005900 OPEN OUTPUT OHSO-FL.
006000 MOVE SPACE TO WSV-ERR.
006010 PERFORM FORT-CALL-RIN.
006020 CLOSE IHSO-FL.
006030 CLOSE OHSO-FL.
006040-
006050 IF WSV-FLAG = 'CAH' GO TO MAIN-B3.
006060 IF WSV-FLAG = 'ERR' GO TO MAIN-EXT.
006070 MAIN-90.
006080-
006090 3000 1A 12 T'A
006100-
006110 OPEN INPUT IHSO-FL.
006120 OPEN OUTPUT OHSO-FL.
006130 MOVE SPACE TO WSV-ERR.
006140 PERFORM STEP-RIN.
006150 MAIN-100.
006160-
006170 3000 1A 12 T'A
006180-
006190 MOVE SPACE TO WSV-ERR.
006200 PERFORM HUNYAKU-RIN.
006210 CLOSE IHSO-FL.
006220 CLOSE OHSO-FL.
006230 IF WSV-FLAG = ' ' GO TO MAIN-110.
006240-
006250 3000 1A-
006260-
006270 OPEN INPUT IHSO-FL.
006280 OPEN OUTPUT OHSO-FL.
006290 MOVE SPACE TO WSV-ERR.
006300 PERFORM DISPLAY-RIN.
006310 CLOSE IHSO-FL.
006320 CLOSE OHSO-FL.
006330-
006340 IF WSV-FLAG = 'ERR' GO TO MAIN-EXT.
006350 MAIN-101.
006360-
006370 3000 1A 12 T'A
006380-
006390 GO TO MAIN-50.
006400 MAIN-110
006410 IF (WHS-FLG = '2') AND (WHS-FLG = 4) GO TO MAIN-50.
006420 IF (WHS-FLG = '2') AND (WHIO-LINK = 'H') GO TO MAIN-50.
006430-
006440 3000 1A 12 T'A
006450-
006460 OPEN INPUT IHSO-FL.
006470 OPEN OUTPUT OHSO-FL.
006480 MOVE SPACE TO WSV-FLAG.
006490 PERFORM PRNH-RIN.
006500 CLOSE IHSO-FL.
006510 CLOSE OHSO-FL.
006520-
006530 IF WHO-RUN = 'H' GO TO MAIN-50.
006540 IF WSV-FLAG = 'ERR' GO TO MAIN-EXT.
006550 MAIN-130.
006560-
006570 3000 1A 12 T'A
006580-
006590 MOVE SPACE TO WSV-FLAG.
006600 CALL 'CLPGH'.
006610 GO TO MAIN-50.
006620 MAIN-EXT.
006630 GO TO MAIN-50.
006640 MAIN-END.
006650 EXIT.
.....
006660 READ-RIN SECTION.
006700 READ-EXT.
006710 MOVE 0 TO WSV-ERR.
006720-
006730 READ IHSO-FL AT END
006740 MOVE 3 TO WSV-ERR
006750 PERFORM ERR-RIN
006760 GO TO READ-EXT.
006770 IF STATUS-1 NOT = '00'
006780 MOVE 1 TO WSV-ERR
006790 PERFORM ERR-RIN
006800 GO TO READ-EXT.
006810 MOVE ATTENTION-INV TO WSV-FLAG.
006820 IF ATTENTION-INV = 'E000' GO TO READ-EXT.
006830 IF ATTENTION-INV = 'PCRH' GO TO READ-EXT.
006840 IF ATTENTION-INV = 'PEH0' GO TO READ-EXT.
006850 GO TO READ-EXT.
006860 READ-EXT.
006870 EXIT.
.....
006880 ***** W R I T E *****
006890 ***** W R I T E *****
006900 ***** W R I T E *****
006910 WRITE-RIN SECTION.
006920 MOVE 0 TO WSV-ERR.
006930-
006940 WRITE OHSO-REC INVALID KEY
006950 MOVE 1 TO WSV-ERR
006960-
006970 PERFORM ERR-RIN
006980 GO TO WRITE-EXT.
006990 IF STATUS-0 NOT = '00'
007000 MOVE 2 TO WSV-ERR
007010 PERFORM ERR-RIN
007020 GO TO WRITE-EXT.
007030 WRITE-EXT.
007040 EXIT.
.....
007050 ***** W R I T E *****
007060 ***** W R I T E *****
007070 ***** W R I T E *****
007080 ERR-RIN SECTION.
007090 ERR-EXT.
007100 IF WSV-ERR = 1
007110 DISPLAY 'WRITE INV OF WS = '
007120 WKSIN-NO
007130-
007140 ELSE IF WSV-ERR = 2
007150 DISPLAY 'WRITE SIS OF WS = '
007160 WKSIN-NO
007170-
007180 ELSE IF WSV-ERR = 3
007190 DISPLAY 'READ END OF WS = '

```

```

007200 WKSIN-NO
007210 ELSE
007220 IF WSV-ERR = 4
007230 DISPLAY 'READ SIS OF WS = '
007240 WKSIN-NO
007250 ELSE
007260 NEXT SENTENCE.
007270-
007280 DISPLAY MED-NAME GROUP-NAME STATUS-I STATUS-O REPEAT-NO.
007290 DISPLAY 'STUDY ABEND '.
007300-
007310 ERR-EXT.
007320 EXIT.
007330 ***** W R I T E *****
007340 ***** W R I T E *****
007350 ***** W R I T E *****
007360 ***** W R I T E *****
007370 ***** W R I T E *****
007380 ***** W R I T E *****
007390 ***** W R I T E *****
007400 ***** W R I T E *****
007410 ***** W R I T E *****
007420 ***** W R I T E *****
007430 ***** W R I T E *****
007440 ***** W R I T E *****
007450 ***** W R I T E *****
007460 ***** W R I T E *****
007470 ***** W R I T E *****
007480 ***** W R I T E *****
007490 ***** W R I T E *****
007500 ***** W R I T E *****
007510 ***** W R I T E *****
007520 ***** W R I T E *****
007530 ***** W R I T E *****
007540 ***** W R I T E *****
007550 ***** W R I T E *****
007560 ***** W R I T E *****
007570 ***** W R I T E *****
007580 ***** W R I T E *****
007590 ***** W R I T E *****
007600 ***** W R I T E *****
007610 ***** W R I T E *****
007620 ***** W R I T E *****
007630 ***** W R I T E *****
007640 ***** W R I T E *****
007650 ***** W R I T E *****
007660 ***** W R I T E *****
007670 ***** W R I T E *****
007680 ***** W R I T E *****
007690 ***** W R I T E *****
007700 ***** W R I T E *****
007710 ***** W R I T E *****
007720 ***** W R I T E *****
007730 ***** W R I T E *****
007740 ***** W R I T E *****
007750 ***** W R I T E *****
007760 ***** W R I T E *****
007770 ***** W R I T E *****
007780 ***** W R I T E *****
007790 ***** W R I T E *****
007800 ***** W R I T E *****
007810 ***** W R I T E *****
007820 ***** W R I T E *****
007830 ***** W R I T E *****
007840 ***** W R I T E *****
007850 ***** W R I T E *****
007860 ***** W R I T E *****
007870 ***** W R I T E *****
007880 ***** W R I T E *****
007890 ***** W R I T E *****
007900 ***** W R I T E *****
007910 ***** W R I T E *****
007920 ***** W R I T E *****
007930 ***** W R I T E *****
007940 ***** W R I T E *****
007950 ***** W R I T E *****
007960 ***** W R I T E *****
007970 ***** W R I T E *****
007980 ***** W R I T E *****
007990 ***** W R I T E *****
008000 ***** W R I T E *****
008010 ***** W R I T E *****
008020 ***** W R I T E *****
008030 ***** W R I T E *****
008040 ***** W R I T E *****
008050 ***** W R I T E *****
008060 ***** W R I T E *****
008070 ***** W R I T E *****
008080 ***** W R I T E *****
008090 ***** W R I T E *****
008100 ***** W R I T E *****
008110 ***** W R I T E *****
008120 ***** W R I T E *****
008130 ***** W R I T E *****
008140 ***** W R I T E *****
008150 ***** W R I T E *****
008160 ***** W R I T E *****
008170 ***** W R I T E *****
008180 ***** W R I T E *****
008190 ***** W R I T E *****
008200 ***** W R I T E *****

```