# 



LEVEL 6 SOFTWARE GCOS 6 MOD 400 PROGRAMMER'S GUIDE

## SERIES 60 (LEVEL 6) GCOS 6 MOD 400 PROGRAMMER'S GUIDE

## SUBJECT

Descriptions of Mod 400 Operating Environments, User Access to the System, and Selected Examples of the Use of System Software Components

## SOFTWARE SUPPORTED

This publication supports Release 0100 of the Series 60 (Level 6) GCOS 6 MOD 400 Operating System; see the Manual Directory of the latest GCOS 6 MOD 400 System Concepts manual (Order No. CB20) for information as to later releases supported by this manual.

## Preface

The purpose of this manual is to provide the user with programmer-oriented information regarding the various operating environments available under the GCOS 6 Mod 400 Operating System and programmer procedures for terminal startup and access to the system. Also contained in this manual are examples of the use of various system software components: the editor, macro preprocessor, assembler, COBOL and FORTRAN compilers, and the sort program.

This material is presented in 9 sections, as outlined in the Introduction. The Introduction also presents suggested usages of the Mod 400 manual set for application programmers, system programmers, and operators.

File No.: 1S23

## MANUAL DIRECTORY

The following publications constitute the GCOS 6 manual set. The Manual Directory in the latest GCOS 6 MOD 400 System Concepts manual lists the current revision number and addenda (if any) for each manual in the set.

Order

Manual Title
GCOS 6 Program Preparation
GCOS 6 Commands
GCOS 6 Communications Processing
GCOS 6 Sort/Merge
GCOS 6 Data File Organizations and Formats
GCOS 6 System Messages
GCOS 6 Assembly Language Reference
GCOS 6 System Service Macro Calls
GCOS 6 RPG Reference
GCOS 6 Intermediate COBOL Reference
GCOS 6 MOD 400 System Concepts
GCOS 6 MOD 400 Program Execution and Checkout
GCOS 6 MOD 400 Programmer's Guide
GCOS 6 MOD 400 System Building
GCOS 6 MOD 400 Operator's Guide
GCOS 6 MOD 400 FORTRAN Reference
GCOS 6 MOD 400 Entry-Level COBOL Reference
Remote Batch Facility User's Guide
Data Entry Facility User's Guide
Level 6/Level 6 File Transmission
Level 6/Level 62 File Transmission
Level 6/Level 64 (Release 0300) File Transmission
Level 6/Level 66 File Transmission
Level 6/Series 200/2000 File Transmission
Level 6/BSC 2780 File Transmission
Level 6/Level 64 (Release 0220) File Transmission

In addition, the following documents provide general hardware information:

Order

s The Car

No.	Manual Title
AS22	Honeywell Level 6 Minicomputer Handbook
AT04	Level 6 System and Peripherals Operation Manual

 $\bigcirc$ 

## **Contents**

## Section 1. Introduction

Guide to Using the Manual Set	1-1
Applications Programmer's Manual	
Guide	1-2
System Programmer's Manual Guide	1-2
Operator's Manual Guide	1-2
RBF and DEF User Manual Guide	

## Section 2. Operating Environments

Operator-Only Environment	2-1
All-Online Environment	
Online/Batch Environment	
Dedicated Application Environment	
Mixed Environment	

## Section 3. User Terminal Startup

A WARNESS

Startup with the Login Facility	3-1
Task Group-Specific Terminal Startup	3-1

## Section 4. User Access to the System

Access by Loggin In	4-1	
Direct Login Terminal		
Abbreviated Login Terminal		
Full Login Terminal	4-1 4-1	
Command Processor as		
Lead Task	4-2	
Application as Lead Task	4-2	
Access through the Operator or		
Another User	4-2	
Serial Execution of Application		
Tasks	4-2	
Concurrent Execution of Application		
Tasks		
Concurrent Execution from Several		
Task Groups		
Execution of an Application from the		
Batch Task Group	4-3	
Execution from the Data Entry		
Facility (DEF)	4-4	
Access through the		
Operator Terminal	4-4	
•		
Section 5. Using the Editor		
Editor Directive Description	5-1	
Section 6. Using the Assembler and Macro Preprocessor		
Sample Assembly Language Session		

Macro Preprocessor	
Sample Assembly Language Session	
(SMPMAC)	6-1
Sample Assembly Language Multitask	
Program (BRDCST)	6-6

v

## Section 7. Using the COBOL Compiler

Sample Card-to-Disk Program	7-1
Volume and File Creation	7-2
Source Loading	7-2
Compiling with COBOL	7-2
Linking	7-2
Executing	7-3
Sample COBOL Terminal Session	
(AC8111)	7-3
Calling FORTRAN Routines from an	
Entry-Level COBOL Main Program	7-7
Section 8. Using the FORTRAN Compiler	
Sample FORTRAN Terminal Session	
(MATINV)	8-1
FORTRAN Chaining	8-1

## Section 9. Using the Sort

## **Figures**

1-1	Applications Programmer Guide	
	to Manuals	1-3
1-2	System Programmer Guide to	
	Manuals	1-4
1-3	Operator Guide to Manuals	1-4
1-4	<b>RBF and DEF User Guide to</b>	
	Manuals	1-5
5-1	Sample Editor Directives in File	
	SMPCMDFL	5-1
5-2	Terminal Responses from	
	Sample Editor Directives of	
	Figure 5-1	5-2
5-3	Sample of Unexpanded Assembly	
	Language Program with Macro	
	Calls and Statements	
	(SMPMAC.P)	5-9
5-4	Sample of Unexpanded Macro	
	Routine (SAMPL1) Contained	
	in EXEC_LIB Directory	5-10
5-5	Sample of Unexpanded Macro	
	Routine (SAMPL2) Contained	
	in EXEC_LIB Directory	5-11
6-1	Sample Terminal Session	
	( <b>SMPMAC</b> )	6-1
6-2	Macro Preprocessor Output	
	( <b>SMPMAC</b> )	6-2
6-3	Cross Reference Listing	
	( <b>SMPMAC</b> )	6-3

6-4	Assembler Output Listing (SMPMAC)	6-4
6-5	Linker Output Listing (SMPMAC)	6-5
6-6	Sample Terminal Session	
	(BRDCST)	6-7
7-1	Sample Terminal Session	
	(AC8111)	7-3
7-2	Sample Listings for AC8111	7-4
7-3	COBOL Listing of COBFRT	7-8
7-4	FORTRAN Listing of FRTRAN	7-9
7-5	<b>Operator Terminal Session for</b>	
	COBFRT	7-9
8-1	Sample Terminal Session	
	(MATINV)	8-1

8-2	Source and Linker Output	
	Listing (MATINV)	8-2
8-3	Assembly Listing of Program	
	CHAIN	8-7
8-4	FORTRAN Programs Calling	
	the CHAIN Function	8-7
8-5	Linker Output for Chained	
	Programs	8-9
8-6	Linker Directives for Chained	
	Programs	8-15
8-7	Execution Output from Chained	
	Programs	8-15
9-1	Sample Sort Terminal Session	9-1

# Section 1 Introduction

The GCOS 6 Mod 400 operating system for the Models 6/30 and 6/40 minicomputers provides a comprehensive set of system services which form a base for executing user-written applications, Honeywell-supplied applications, and program development tools. It provides an online, interrupt-driven operation for multiple users and a single, low-priority batch operation typically used for program development and associated activities.

A number of different operating environments are possible, controlled in part by options exercised at system configuration, and in part by options chosen by the system operator at startup or at various times during the operating day. These environments are more fully described in Section 2, "Operating Environments."

Access to the system by users can be achieved in a variety of ways, again depending in part on system configuration options selected. These options are concerned mainly with the definition of local and/or remote terminal devices and how they are connected to the system. These are described in Section 3, "User Terminal Startup." Other access options, normally under the control of the system operator, are concerned with the procedures by which a user identifies himself (logs in) to the system through a connected terminal. This subject is treated in Section 4, "User Access to the System."

The remaining sections comprise descriptions and examples of the use of various system components: the Editor (Section 5), the Assembler and Macro Preprocessor (Section 6), the COBOL Compiler (Section 7), the FORTRAN Compiler (Section 8), and the Sort component (Section 9). Each of these sections presents terminal and/or line printer listings representing the actions performed. In these listings, heading lines may vary in detail depending on the component that initiated the listing or, in some cases, may be omitted. However, in actual use, the user will see heading lines consisting of three major fields of information, as shown below.

1. System Identification: GCOS6 MOD400- S L rrr-mm/dd/hhmm

S — SAF L — LAF rrr — Release number of the operating system mm/dd/hhmm — Date/time when operating system was created (month, day, hour, and minute)

2. Component Identification: xxxxx-rrrr-mm/dd/hhmm

xxxxx — Component name

rrrr — Revision number of component

mm/dd/hhmm — Date/time that specified revision of component was created (month, day, hour, and minute)

3. Time of program execution: yyyy/mm/dd hhmm:ss.t

Date/time of program execution (year, month, day, hour, minute, second, and tenth of second)

## **GUIDE TO USING THE MANUAL SET**

This guide to the manuals is arranged according to functions that might be performed by an applications programmer, a systems programmer, or an operator. As used in this guide, the applications programmer writes applications programs; the system programmer configures the system and defines the environment for each application; and the operator operates the system from the operator terminal. These functions could be performed by three different persons or by the same person serving in the different capacities.

## **APPLICATIONS PROGRAMMER'S MANUAL GUIDE**

Figure 1-1 illustrates the suggested sequence in using the manuals. If you wish to start using the system by writing an application program, begin by using the *Programmer's Guide* manual. It illustrates: (1) various ways to gain access to the system, (2) a sample Editor session, and (3) for application languages, the procedure for performing program preparation and execution. Working with the small subset of commands used in the examples is a good approach to learning the system command set. This approach for getting started assumes that a system programmer has already configured and started up a suitable application environment. While using the system, you may wish to familiarize yourself with the system facilities described in the *System Concepts* manual.

Through examples, the *Programmer's Guide* illustrates how to use the system facilities. Other manuals provide reference material. The *Program Preparation* manual contains Editor directives (statements) to create and update an application language source unit. For each of the languages the appropriate language reference manuals contain the description of the language statements. Operating system dependencies, if any, that affect how you write the application are described in the *Programmer's Guide*. If the application uses communications, refer to the *Communications Processing* manual. Read the *Data File Organizations and Formats* manual if you require a better understanding of a language-supported file organization that is to be used in an application, or if you must calculate the size of a data file. You can use Monitor macro calls, as described in the *System Service Macro Calls* manual, in assembly language programs. Before your program can be entered for execution, it must be linked as described in the *Program*.

For program compilation or assembly and execution, the procedures described in the *Programmer's Guide* might be sufficient. To obtain more control over the execution of your program or utilize the system facilities more completely or efficiently, use the commands described in the *Commands* manual. If you wish to use the operator terminal, read the *Operator's Guide*. In many cases, the description of commands must be supplemented by system concepts described in the *System Concepts* manual. Rather than read all the conceptual material at one time, you may find it more meaningful to refer to it in conjunction with the appropriate reference material. The *Commands* manual also describes the utilities. An assembly language program, the Patch, Debug, and Dump utilities are described in the *Program Execution and Checkout* manual; file transmission from Level 6 to a host system is described in the *File Transmission* manual appropriate to the host system. Error messages and return status codes are listed in the *System Messages* manual.

## SYSTEM PROGRAMMER'S MANUAL GUIDE

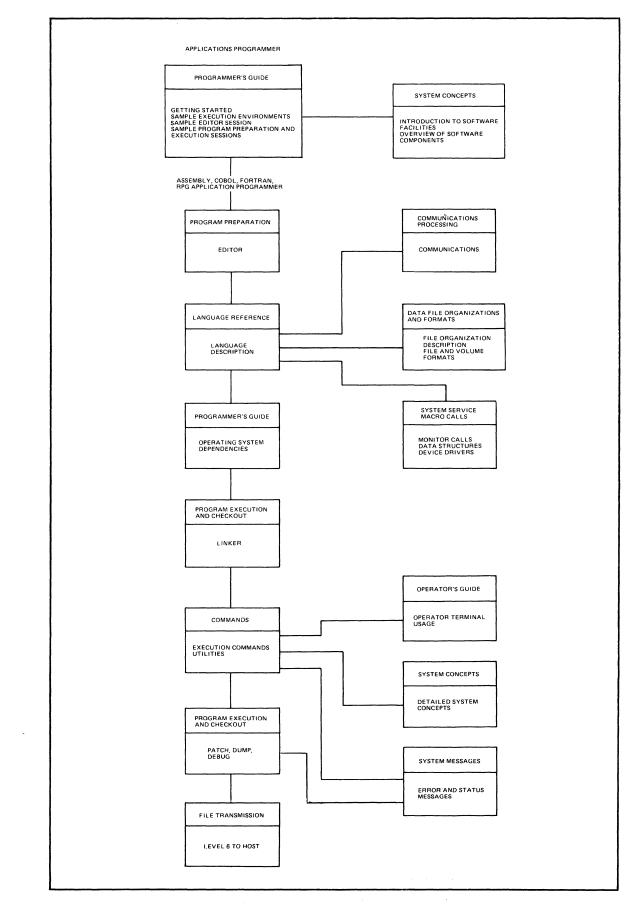
Figure 1-2 illustrates the suggested sequence for using the manuals. The System Building manual provides you with the configuration directives (statements) and startup procedures to configure and start up a MOD 400, a Remote Batch Facility (RBF), or a Data Entry Facility (DEF) system. You must know the conceptual material in the System Concepts manual in order to successfully use the configuration directives. To tailor an applications environment suitable for the intended application, use the operator commands described in the Operator's Guide manual. Error messages are listed in the System Messages manual. If you are working with an application that runs under the BES operating system, the System Concepts manual contains MOD 400 and BES compatibility considerations.

#### **OPERATOR'S MANUAL GUIDE**

Figure 1-3 illustrates the suggested sequence for using the manuals. Specific operator job functions must be determined by each installation; a large system might have a person assigned as an operator; a small system might have each programmer also act as an operator. The *Operator's Guide* indicates the system procedures performed through the operator terminal and describes operator commands used in system operation.

The Programmer's Guide contains examples using commands (described in the Commands manual) that are similar to operator commands. The System Concepts manual provides an understanding of the operating system. Note that the Operator's Guide describes using the

INTRODUCTION





のない

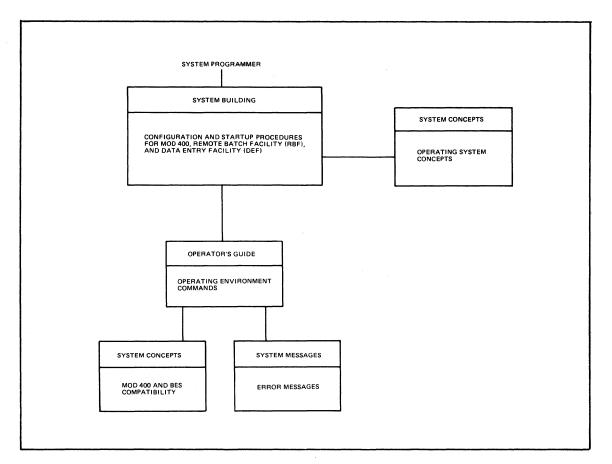


Figure 1-2. System Programmer Guide to Manuals

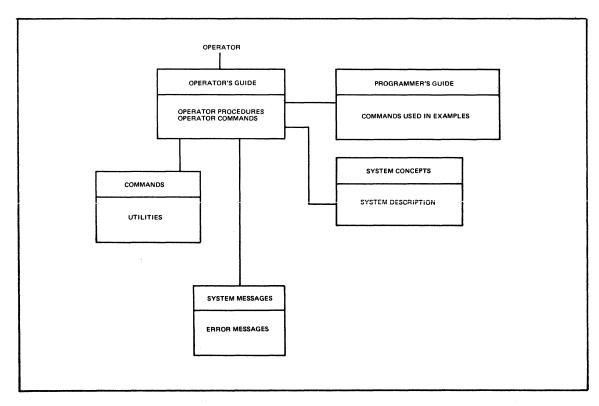


Figure 1-3. Operator Guide to Manuals

INTRODUCTION

operator terminal for operator functions to enter operator commands to the system task group, or for user functions to enter commands to a user task group. To run the utilities, use the commands (described in the *Commands* manual) entered through the operator terminal functioning as a user terminal. Error messages are listed in the *System Messages* manual.

## **RBF AND DEF USER MANUAL GUIDE**

Figure 1-4 illustrates the suggested sequence for using the manuals. The system programmer configuration functions have been done and the system is ready to be used for Remote Batch Facility (RBF) functions or Data Entry Facility (DEF) functions. The *Programmer's Guide* manual provides sample login execution environments typical of ones that might be at your facility. The *Remote Batch Facility User's Guide* is used for RBF operations and the *Data Entry Facility User's Guide* is used for DEF operations.

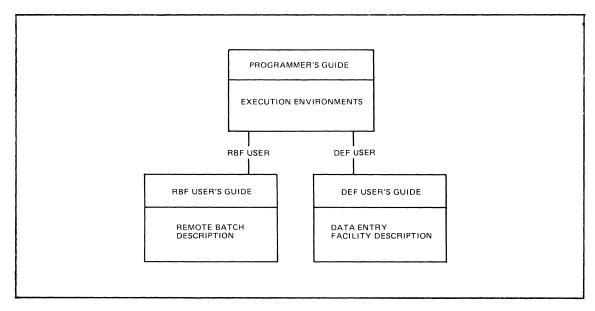
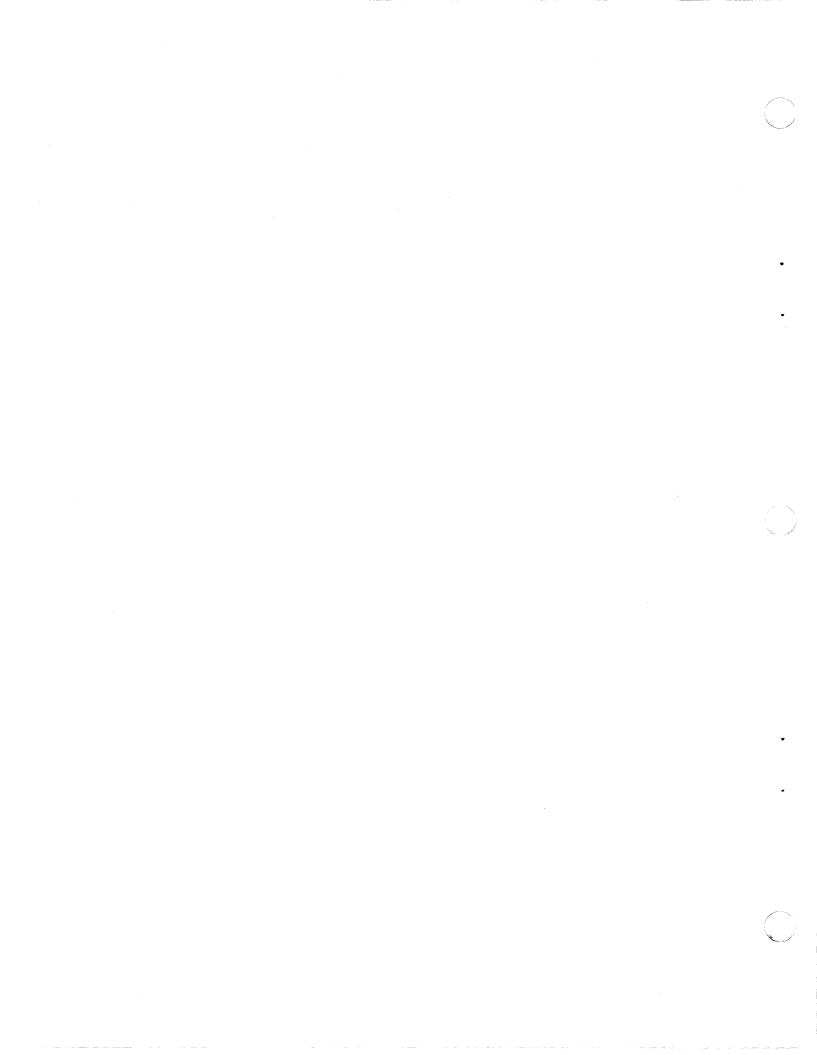


Figure 1-4. RBF and DEF User Guide to Manuals



# Section 2 Operating Environments

The Mod 400 operating system allows a wide variety of operating environments, ranging from a single operator-controlled configuration to one in which the operator, other users, or a combination if both can control the configuration at any time during the operating day. This range of operating environments is described in this section.

## **OPERATOR-ONLY ENVIRONMENT**

This environment is one in which a designated operator and a limited number of users (typically programmers developing application programs) use the system on a first-come first-served basis for developing and testing programs. All work is done through the operator terminal, through either the system task group or a single online task group created by the system startup procedure. Certain functions can be performed through either of the two task groups; others can be done only through the system task group or the online task group — refer to the *Operator's Guide* and the *Commands* manuals for details on which functions can be performed from each task group.

## **ALL-ONLINE ENVIRONMENT**

An all-online environment is one in which one or more users can concurrently use the facilities of the operating system to perform interactive tasks of any kind permitted by the command language described in the *Commands* manual, plus any user applications that can be invoked through the command processor. This latter category consists of user programs in the form of bound units that are called from a task group in which the command processor is declared as the lead task when the task group is created. A task group can also be created by the operator or another online user, declaring the application bound unit as the lead task; in this case the creation of the task group and its activation results directly in the execution of the declared bound unit, without the need to enter its name as a command.

An example of this kind of environment is one in which several task groups have the command processor as lead task and one or more other task groups have specific application programs such as the Data Entry Facility and user-created programs as lead tasks. The former task groups can be used for editing source program files, entering requests for jobs to be run in the batch task group (see below), requesting printouts of files, etc. Concurrent with these activities can be the execution of the user application programs constituting the latter set of task groups. From the user's point of view, each task group has the appearance of having control of the system.

## **ONLINE/BATCH ENVIRONMENT**

This environment differs from the all-online environment only in that, in addition to the creation of the online task groups, a batch task group has also been created by the designated operator from the operator terminal. Once this task group has been created, any online task group having the command processor as its lead task can enter requests for jobs to be run through the use of the EBR (ENTER BATCH REQUEST) command. Typical of such batch jobs would be requests for compilations, links, application program checkout runs, and the like.

Creation and utilization of the batch task group requires the existence of at least the designated operator terminal, through which the batch task group is created and through which requests to it can be entered. Jobs run in the batch task group are normally controlled by a previously created file containing commands directing the execution of the jobs, and not by interactive dialog from a terminal. Section 4 contains additional information on the use of the batch task group.

## **DEDICATED APPLICATION ENVIRONMENT**

This is an environment in which system startup or operator action subsequent to startup results in the creation of one or more task groups in which a user application, and not the command processor, is the lead task. In such an environment no interactive processing takes place; rather, whatever processing occurs is dependent on the nature of the application — e.g., data entry, an inventory application, etc.

## **MIXED ENVIRONMENT**

The Mod 400 system does not restrict the user to any one of the foregoing environments at any given time. Given a large enough system, any of these can be combined with any others to provide concurrent interactive, batch, and dedicated operations on a selected terminal basis. That is, a selected set of terminals can be associated with interactive tasks, while others can be related to the dedicated application tasks.

2-2

# Section 3 User Terminal Startup

Terminal startup procedures vary according to the type of terminal and whether the terminal is a noncommunications or a communications terminal. A noncommunications terminal is one that is connected to the system through the multiple device controller (MDC), while a communications terminal is connected through the multiline communications processor (MLCP). An MLCP-connected terminal can be connected either through a modem or through a dial-up telephone line. In the former case, when the modem is made ready, the terminal is connected and ready for operation. With a dial-up connection, the user must dial the number which connects the telephone to the system and wait for the signal that indicates the connection is made. For an MDC-connected terminal, simply turning on the power to the terminal suffices to connect it to the system. Subsequent actions depend on whether or not the listener/login processor is activated, and whether the terminal is declared by the operator to be associated with a specific task group.

## STARTUP WITH THE LOGIN FACILITY

If the operator has activated the listener/login processor, and the terminal being started up is one which is monitored by the login processor, then the user must log in using the procedures described in the *Operator's Guide* manual after performing the actions required for physically connecting the terminal to the system. When the connection is made to an MLCP-connected terminal, the system will display a system identification message, a message of the day if one is defined, and indicate that it is ready to accept a login request. For an MDC-connected terminal, displays occur only if the terminal is active when the login processor is activated.

## TASK GROUP-SPECIFIC TERMINAL STARTUP

If a terminal is declared by the operator to be associated with a specific task group, and is not monitored by the login processor, then, when the terminal is connected, it is ready to accept whatever input or output is dictated by the logic of the task's execution. If the command processor is the lead task, a ready message will be issued, indicating that the terminal is ready to accept commands. If a user application is the lead task, it should issue a message to the terminal indicating that it has recognized the availability of the terminal and is ready for execution.

 $\bigcirc$ 

 $\left( \right)$ 

\_

## Section 4

## User Access To The System

Once a terminal has been connected to the system as described in Section 3, a user can gain access to the system in any of several ways. Which of these ways is used at any given time depends upon operator actions taken during and after system startup. Examples of various access procedures are given in this section. Each example states any prerequisite operator actions which would have been performed.

## ACCESS BY LOGGING IN

Configuration and system startup have been done. The operator, through the system task group, has activated the login function as described in the *Operator's Guide* manual. The terminal is one which is connected through the MDC, and was active when the login function was activated. The system message of the day has been displayed and the login prompter message has been printed. The user's login procedure at this point depends on the terminal login characteristics for this terminal. Procedures are described below.

## **DIRECT LOGIN TERMINAL**

In addition to the message of the day, if the command processor is the lead task, the ready message will have been displayed, and no further action is required. The user can begin to enter commands.

If an application is the lead task, further action, if any, depends on the characteristics and logic of the application.

#### ABBREVIATED LOGIN TERMINAL

After the login prompter message has been issued, the user enters a one-character abbreviation such as

Α

If the login line corresponding to the abbreviation "A" indicates that the command processor is the lead task, the system responds with the ready message, and the user can then begin to enter commands.

If an application is the lead task, further action, if any, depends on the characteristics and logic of the application. A terminal that accepts an abbreviated login also accepts a full login command line.

## FULL LOGIN TERMINAL

After the login prompter message has been issued, the user must enter a full login line as described in the *Commands* manual. If the login line specifies or implies that the command processor is the lead task, the system responds with the ready message, and the user can begin to enter commands.

If the lead task is an application, further action, if any, depends on the characteristics and logic of the application.

#### COMMAND PROCESSOR AS LEAD TASK

For a user named W. Smith to log in to the system, specifying the command processor as the lead task, a login line such as

L SMITHW -HD ^ VOL22>SMITHW

could be used, where,  $^VOL22>SMITHW$  is the working directory pathname. As soon as the ready message is displayed, the user can begin to enter commands for either serial or concurrent execution. In particular, if the -HD argument was not used, the working directory can be specified with a CWD (CHANGE WORKING DIRECTORY) command

CWD ^ VOL22>SMITHW

## APPLICATION AS LEAD TASK

For the same user to log in specifying a task other than the command processor, his login line could be

L SMITHW -PO MS\_UPDATE -HD ^ VOL22> SMITHW

where  $MS\_UPDATE$  is the name of the bound unit which is to be the lead task. The bound unit is located in the directory ^ VOL22>SMITHW. If it is located in some other directory, a full pathname must be used as the argument, such as

L SMITHW -PO ^ VOL23>MS\_UPDATE -HD ^ VOL22>SMITHW

After the login line is processed, control rests with the application. It is strongly advised that the application issue some kind of message indicating that it has been successfully loaded and is ready to begin, or has begun, execution. It may be simply an informative message or a message requesting some action on the part of the terminal user.

## ACCESS THROUGH THE OPERATOR OR ANOTHER USER

The system operator or an online user can create and activate an online task group through the use of the CG (CREATE GROUP) and EGR (ENTER GROUP REQUEST) commands, or through the SG (SPAWN GROUP) command. The application being run in this task group can be in the form of a series of commands implying either serial or concurrent execution, as shown below.

## SERIAL EXECUTION OF APPLICATION TASKS

The operator or another user has created and activated a new online task group whose lead task is the command processor, and whose command-in file is the MDC- or MLCP-connected terminal being used by the new user.

As soon as the ready message is displayed, the new user can begin to enter commands. After each command request is terminated, indicated by the display of the ready message, control returns to command input level and another command can be entered. The following example shows the entry of commands to initiate a COBOL compilation, the assignment of the user-out file to a line printer, and the printing of the COBOL compilation listings.

COBOL PROGA [ctl\_arg] Invoke the COBOL compiler

compiler responses

4-2

RDY:	Indicates end of compilation
FO >SPD>LPT01	Assign user-out to line printer
RDY:	Indicates assignment complete
PR PROGA.L	Invoke PRINT command to print compilation output
RDY:	Indicates printout complete

## **CONCURRENT EXECUTION OF APPLICATION TASKS**

The operator or another user has created and activated a new online task group whose lead task is the command processor, and whose command-in file is the MDC- or MLCP-connected terminal being used by the new user.

As soon as the ready message is displayed, the new user can begin to enter commands. This example shows the entry of commands to initiate a COBOL compilation, the assignment of the user-out file to a line printer, and the printing of a file which is unrelated to the compilation, and thus has no time dependency upon completion of the compilation.

ST 1 -EFN COBOL [ctl_arg]	1 -EFN COBOL [ctl_arg] Invoke COBOL compiler task at relative level 1				
RDY:	Indicates completion of the ST command; compilation is in progress				
FO >SPD>LPT01 Assign user-out file to line printer					
RDY: Indicates assignment complete					
ST 3 -EFN PR -ARG FILE1 Invoke PRINT task to print FILE1, unrelated to compilation					
RDY:	Indicates completion of the ST command; printing is in progress concurrent with compilation				

This is an example of multitasking. The responses from the COBOL compiler, indicated in the previous example, will be interspersed with other input and output lines, depending on when they occur in relation to these lines. The user should always ensure that a ready message has occurred in response to his last command entry before making another entry.

## CONCURRENT EXECUTION FROM SEVERAL TASK GROUPS

Several task groups have been created and activated, each associated with a different command-in terminal, and each having the command processor as its lead task.

As soon as the ready message appears at each terminal, the user at that terminal can begin to enter commands to do serial or concurrent application execution. The task groups are concurrently active for execution and contend with each other for system resources. Each user appears to have control of the system.

## EXECUTION OF AN APPLICATION FROM THE BATCH TASK GROUP

An application environment has been specified consisting of several online task groups and the batch task group (whose lead task is the command processor).

A user can enter one or more EBR (ENTER BATCH REQUEST) commands from each of the online task groups to obtain processing in the batch task group. These requests are queued and will be satisfied on a first-in first-out basis. The EBR requires a command-in file containing commands to be executed in the batch task group. The file is normally disk-resident in the user's working directory having previously been created. If a terminal were specified as the command-in device, the user at the terminal must wait to enter a command, until the command processor processes this EBR command. Otherwise, batch processing will stall waiting for this batch request to complete. To request is to execute the command file, PAYR\_IN, on directory ^ZSYSO1>IW. The application is to compile, link, and execute an application program PAYROLL.

EBR PAYR\_IN -WD ^ZSYSO1>IW

The file PAYR\_IN contains the following commands:

```
COBOL PAYROLL -LO -COUT >SPD>LPT01
LINKER PAYROLL -COUT >SPD>LPT01
LIB ^ZSYSO2>ZCRT
LINK PAYROLL
MAP;QT
GET DEPT4 2
GET >SPD>LPT02 3
PAYROLL
BYE
```

Any time after the file PAYR\_IN has been created, it can be invoked through an EBR to control batch execution. The command file can contain any combination of legitimate commands, such as compile/link/execute sequences, including any necessary file control commands (GET, REMOVE); or file print/dump commands. The main constraint is that the commands be entered into the file in the same manner as if they were being executed from the online terminal, keeping in mind any time dependencies that might exist among various tasks. Responses from the invoked commands that would normally be written to the user terminal in an online environment are written to a file PAYR\_IN.AO in the working directory of the user who issued the EBR command.

## **EXECUTION FROM THE DATA ENTRY FACILITY (DEF)**

One or more task groups whose lead task is the Data Entry Facility (DEF) have been spawned.

When DEF has indicated at the terminal that it is ready to accept data entry actions, the user can begin to enter directives. No other preliminary actions are required.

Refer to the *Data Entry Facility User's Guide* manual for details on the operation of the Data Entry Facility.

It should be noted that the presence of the Data Entry Facility in no way restricts the presence of other online task groups or the batch task group. These functions can be carried on concurrently as described in the preceding paragraphs.

## ACCESS THROUGH THE OPERATOR TERMINAL

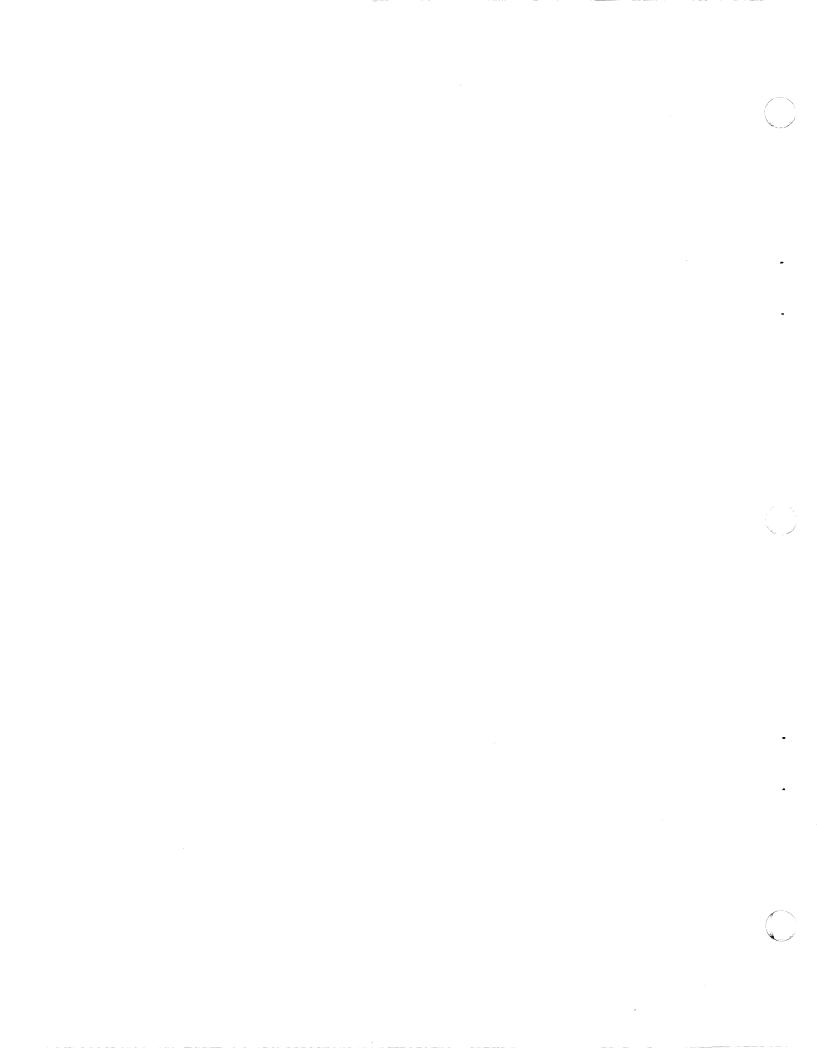
A special case of system access is that in which all interactive and/or batch executions are initiated through the operator terminal. The major difference between this execution mode and those described previously is that the interface to the system is through the Operator Interface Manager (OIM), described in detail in the *Operator's Guide* manual. The most user-visible aspect of this mode is the issuance by the OIM of task group id designations and message numbers, which require, in many cases, task group id and message number entries from the operator terminal in response.

The operator terminal is the only way in which the system as initially delivered to the user can be accessed. Initial startup results in the creation of the system task group (\$S) and one online task group (\$H). For small system environments in which the operator and one or more users (e.g., programmers writing and debugging their own programs through the operator terminal on a first-come, first-served basis) share the operator terminal, this type of startup, appropriately modified for the physical system configuration, may be sufficient.

4-4

Typically, the operator in this kind of configuration could initiate other task groups in any of the combinations described in Section 2 through the system task group, and also use the \$H task group for any function which is not normally done in the system task group (e.g., editing files, assembling or compiling, linking, debugging, and the like). In particular, in the originally-released system, the \$H task group is used to construct new CLM\_USER files for system startup, and STARTUP.EC files for use during the startup process. There is no requirement that the existence of the \$H task group be maintained permanently — the originally-released STARTUP.EC file which results in the creation of the \$H task group can be modified at any time to delete the function of creating this task group.

Any of the operations described above can be done through the operator terminal from an online task group such as \$H, or any other online task group created as a function of system startup or at some later time, and specifying the operator terminal as its command-in file. Most of the examples in Sections 5 through 9 show operations using the online task group \$H. They illustrate the issuance of the task group identification prefix by the OIM in operator terminal typeouts, and in some cases the changing of the OIM default task group identification to \$H, eliminating the need to enter the prefix explicitly when issuing commands to this task group. If these same operations were done in a task group associated with a terminal other than the operator terminal, the prefixes would not be issued nor need to be specified at that terminal.



# Section 5 Using the Editor

This section illustrates how Editor directives are used to modify the contents of four files, merge files into one file, and place macro routines in the macro library directory. The Editor directives are in file SMPCMDFL; the four files to be altered are SMPM01 (example 1), SMPM02 (example 2), SMPM03 (example 3), and SMPM04 (example 4). The examples are shown below. SMPM01 and SMPM02 are altered and written to files SMMPL1 and SMMPL2, respectively, and then combined to form file SMPM03 and SMPM04 are altered and written as files SMMPL3 and SMMPL4, respectively; they are again altered and written as macro library routine files SAMPL1 and SAMPL2, respectively, into the MACRO <EXEC\_LIB directory to be used during macro preprocessing. Editor output directed to the operator's terminal is shown in an operator's terminal typeout.

## EDITOR DIRECTIVE DESCRIPTION

三部の書き

The following is a line-by-line explanation of the action taken by the Editor when it processes the directive file, ^ SYSMAC>SMPCMDFL, of Figure 5-1, and an explanation of the operator terminal typeouts displayed in Figure 5-2. Editor directive lines are identified by line number. In the typeout, the response to these directives begins after the line (\$H) EDIT-0100-11/21/0827. The default working directory is ^ SYSMAC so that either a full pathname of the form ^ SYSMAC>SMPM04 or simple pathname SMPM01 can be used.

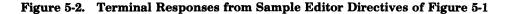
```
1
    R ^SYSMAC>SMPM01
 2
    X
                          PROG2.START2[,]NAME
 3
    6,9CLE
               SETA
 4
    L4
               SETA
                          EQU
 5
                          RESV
    ٤5
               SETA
 6
    L6
               SETA
                          TEXT
 7
    L7
               SETA
                          XDEF
 8
    1F
 9
               LIBM
                          'EXECTLIB', SAMPL1, SAMPL21F1?
    21
    =.=1;$K (SMMPL1)
10
11
    1, $DX
12
13
    R SMPM02
    1,13VL/#L/.=12;13GL/#L/.=9;.S'#L'?L'P=
14
    1, SM (SMMPL2) X
15
16
    R SMPM03
    8,175/SETU/SETA/8,17P
17
    1, SM(SMMPL3) X
18
19
    R SYSMAC>SMPM04
                          ?G7,?LC,IFE1
    X29A
               IFE
20
21
               FAIL
    ENDIT
               IFNL
                          ?P2,?LC,*
55
23
    IFE1
               NULLIF
24
    /SS..LE/L/$$/LD/~~/LD
    1,SK(SMMPL4)
25
    X1,$D
56
    B(SMMPL1)
27
    W "SYSMAC>SMMPL1
28
29
    1, $DX
    B(SMMPL2)
30
31
    W SYSMAC>SMMPL2
32
    1,50
```

#### Figure 5-1. Sample Editor Directives in File SMPCMDFL

```
B(SMMPL3)
33
   W SYSMAC>SMMPL3
34
   1,$D
35
36
   B(SMMPL4)
   W "SYSMAC>SMMPL4
37
   1, SDX
38
39
   R SMMPL1
40
    /INSERT/LD/ADD L7/LD
    15R SMMPL2
41
42
    X52LD
43
   E FO >SPD>LPT00
44
   1,SLW SMPMAC.P
45
   1,$D
   R SMMPL3
46
47
   X/L4/;/LE/S/SETB/SETA/
48
   1,SLW "Z00B02>LDD>MACRO>EXECTLIB>SAMPL1
   1, SD
49
50
   R SMMPL4
   /DLET/D
51
52
   0
53
   1,SLW ~Z00B02>LDD>MACRO>EXECTLIB>SAMPL2
54
   X
   E FO
55
56
   Q
```

Figure 5-1 (cont). Sample Editor Directives in File SMPCMDFL

取 GROUPSD (SD)ON-LINE DEBUG 1115 04 SYSREV. 4014 REV. 1976/11/20 C :SH: RDN (SH)RDY: CWD SYSMAC (SH)RDY: LWD (SH) SYSMAC (SH)RDY: ED -LINE\_LN 75 -IN ^SYSMAC>SMPCMDFL (SH)EDIT 0120 (0) "SYSMAC>SMPM01 (SH) 16 -> (SH)EDIT MODE (SH) 2 18 -> MOD (0) "SYSMAC>SMPM01 (\$H) (SH) 18 (SMMPL1) (0) "SYSMAC>SMPM01 (SH) 0 -> (SH) 18 (SMMPL1) . (SH) 1 \* THESE UNPROTECTED COMMENT LINES WILL BE DROPPED (SH) 2 (SH) \* WHEN MACRO PREPROCESSED. 3 (SH) 4 (SH) ?G4 #L4 ?G1 5 (G1 INITIAL VALUE=\$) (SH) ?IX(#LE, ?PE)?GB?P1 6 #L5 7 (SH) ?G5 #L6 ?P3?VL(35)?P4#LZ#LA (SH) 8 #1.7 ?G4 (SH) 9 #L7 ?P6?P8?GB?AL(?PC)?P7 (SH) 10 #L8 ?SS(?P4,7,1) (SH) 11 #L9 ?VP(11) 766 (\$H) 12 #LB ?G7 (SH) ?P9+?G3 13 ?GA #14 (SH) EN DM (SH) 14 0 -> (0) "SYSMAC>SMPM02 (SH) (SH) 18 (SMMPL1) (SH) (SMMPL2) 36 (SH)L4 SETA ORG (SH)L5 SETA DC (SH)L6 SETA LDR SETA (SH)L7 STR (SH)L8 SETA CALL



USING THE EDITOR

(SH) L9		SETA	
(SH) LA		SETA	
(SH) LB		SETA	
(SH) LC		SETA	
(SH) LD	•	SETA	
(SH)		->	(0) SYSMAC> SMPM03
(\$H)	18		(SMMPL1)
(SH)	36		(SMMPL2)
(SH)	45		(SMMPL3)
(SH)		->	(0) SYSMAC> SMPM04
(SH)	18		(SMMPLI)
(SH)	36		(SMMPL2)
(SH)	45		(SMMPL3)
(SH)	28		7G5 ?PZ ?SS(?LE,1,5)
(SH)	10		DC 'DELETE LINE ENDING IN \$'S
(\$H)		DEL	DC DELETE LINE BEGINNING IN
(SH)			(0) SYSMAC>SMPM04
(\$H)	18		(SMMPL1)
(SH)	36		(SMMPL2)
(SH)	45		(SMMPL3)
(SH)	36		(SMMPL4)
(SH)	-		(0) SYSMAC>SMPM04
(SH) (SH)	-	->	(SMMPL1) *SYSMAC>SMMPL1
	36		(SMMPL2)
(SH)	45		(SMMPL3)
(SH) (SH)	36 0		(SMMPL4) (0) °SYSMAC>SMPM04
(SH)	0		(SMMPL1) *SYSMAC>SMMPL1
(SH)	0		(SMMPL2) *SYSMAC>SMMPL2
(SH)	0		(SMMPL2) STSMAC>SMMPL2 (SMMPL3) SYSMAC>SMMPL3
(SH)	-	->	(SMMPL4) SYSMAC>SMMPL4
(\$H)	3		T LN 2 LIBM STATEMNT BEFORE THIS LN. THEN DEL THIS LN
(SH)			.7 SETA VALUE W/CHANGE FUNCTION THEN DELT THIS LA
(SH)			(0) "SYSMAC>SMPM04
(SH)	ñ		(SMMPL1) SYSMAC>SMMPL1
(SH)	Ő		(SMMPL2) SYSMAC>SMMPL2
(\$H)	Ő		(SMMPL3) SYSMAC>SMMPL3
(SH)	-	-> MOD	(SMMPL4) SYSMAC>SMMPL2
(SH)	52		CDIT READ FUNCT TO ADD "SMPM02" PORTION TO FILE*
(SH)	<u> </u>		(0) °SYSMAC>SMPM04
(SH)	Ő		(SMMPLI) <sup>SYSMAC&gt;SMMPLI</sup>
(SH)	Ő		(SMMPL2) SYSMAC>SMMPL2
(SH)	Ő		(SMMPL3) SYSMAC>SMMPL3
(SH)	-	->	(SMMPL4) SYSMAC>SMMPL3
			TRS EXIST, QUIT DEFERRED
(SH)	0		(0) ^SYSMAC>SMPM04
(SH)	Ő		(SMMPL1) *SYSMAC>SMMPL1
(SH)	Ō		(SMMPL2) SYSMAC>SMMPL2
(SH)	ŏ		(SMMPL3) SYSMAC>SMMPL3
(SH)	-	->	(SMMPL4) <sup>2</sup> Z00B02>LDD>MACRO>EXEC_LIB>SAMPL2
(SH)RDY			

Figure 5-2 (cont). Terminal Responses from Sample Editor Directives of Figure 5-1

1948

Line No.	Editor Directive Description	Terminal Typeout
1	R ^ SYSMAC>SMPM01 Read the 16-line file (example 1) into the current buffer (0).	
2	X Display the status of the current buffer (denoted by $\rightarrow$ ). Sixteen lines were read into current buffer (0).	<b>16-</b> >(0)
3	6,9Ctext Change lines 6 through 9 of the buffer with this text for line 6.	
4	Text for line 7.	
5	Text for line 8.	
6	Text for line 9.	
7	Text, this additional line is inserted after the previous four lines were changed. Note that the Editor recognizes tab characters.	
8	!F Terminate input mode and enter edit mode.	
9	2Itext!F!? Insert text before line 2. Terminate input mode. Display current mode.	EDIT MODE
10		EDIT MODE
10	=1;\$K(SMMPL1) Display current line pointer.	2
	Move the current line pointer back one line to a new current line pointer position. Copy the lines from that position to the last line in the current buffer into auxiliary buffer, SMMPL1.	2
11	X	
	Display status of current and auxiliary buffers. There are 18 lines in current buffer (0), which has been modified (MOD) since it was read in, and 18 lines in auxiliary buffer SMMPL1.	18-> MOD (0) 18 (SMMPL1)
12	1,\$DSX	
	Delete the first through last line of the current buffer (0). Display status of buffers.	0->(0) 18 (SMMPL1)
13	R SMPM02 Read 36-line file (example 2) into the current buffer (0).	
14	1,13VL/#L/12;13GL/#L/9;S'#L'?L'P= For lines 1 through 13, display line numbers and all lines that	1*
	do not contain the expression #L.	2*THESE
		4*
	Move the current line pointer back 12 lines from line 14 to line 2. For lines 2 through 13, display all lines and their line numbers containing the expression #L.	* 5?G4#L4 ·
		13?GA #L4

5-4

C422

Line No.	Editor Directive Description	Terminal Typeout
	Move the current line pointer back nine lines to line 5 and substitute $L$ for #L.	
	Print current line.	ENDM
	Print current line pointer value.	14
15	1,\$M(SMMPL2)X	
	Move line 1 through the last line of the current buffer to auxiliary buffer SMMPL2. The contents of the current buffer (0) are erased. Display the status of the buffers.	0->(0) 18 (SMMPL1) 36(SMMPL2)
16	R SMPM03	
	Read 45-line file (example 3) into current buffer (0).	
17	8,17S/SETB/SETA/8,17P	
	For lines 8 through 17, substitute SETA for SETB.	
	Print lines 8 through 17 without line numbers.	L4 SETA
		•
		LD SETA
18	1,\$M(SMMPL3)X	
	Move line 1 through last line of the current buffer into auxiliary buffer SMMPL3 and erase buffer (0).	
	Display buffer status.	0->(0) 18 (SMMPL1) 36 (SMMPL2) 45 (SMMPL3)
19	R ^ SYSMAC>SMP04	
	Read the 34-line file (example 4) into the current buffer (0).	
20	X29A	
	Display buffer status, 34 lines are currently in buffer (0).	34->(0) 18 (SMMPL1)
		•
		45 (SMMPL3)
	Append, after line 29, four lines of text. Text for line 30.	
21	Text for line 31.	
22	Text for line 32.	
23	text!F	
	Last line of text (line 33).	
	Terminate input mode and enter edit mode.	
24	/SSLE/L/\$\$/LD/ ^ _ /LD	
	Search the current buffer for the first occurrence of the	
	expression SSLE, where are any two characters. List the line and its line number.	009CE 9D7 000/97 5
	List the line and its line number.	28?G5 ?PZ ?SS(?LE.

í

1999 - 1999 - 1999 - 1999 - 1999 - 1999 - 1999 - 1999 - 1999 - 1999 - 1999 - 1999 - 1999 - 1999 - 1999 - 1999 -

(

Line No.	Editor Directive Description	Terminal Typeout
	Locate a line that ends with \$ as the last character. The first dollar sign is escaped using a nonprinting C,i.e.,!C\$\$. The second dollar sign retains its special meaning, and indicates: locate the last character in a line ending in dollar sign.	1 በ ጊ ፲፱፻ ጥቃ ጊ ጊ ጊ ነው
	List the line and its number; then delete the line.	10DELT\$ DCD'\$
	Locate a line beginning with $^{\land}$ . The second circumflex is escaped by using the nonprinting !C,i.e., $^{\land}$ !C $^{\land}$ .	
	List, then delete the line and its line number.	28 ^ DEL DC `DELETE
25	1,\$K(SMMPL4) Copy the current buffer contents from first through last line into auxiliary buffer SMMPL4.	
26	X1,\$D	
	Display the status of the buffers.	36->MOD(0) 18
		36 (SMMPL4)
	Delete first through last line of current buffer.	
27	B(SMMPL1)	
	The auxiliary buffer, SMMPL1, is made the current buffer prior to writing.	
28	W ^ SYSMAC>SMMPL1	
	Write the current buffer contents as a file whose pathname is ^ SYSMAC>SMMPL1.	
29	1,\$DX	
	Delete the first through last line of the current buffer.	
	Display the buffer status. The pointer points to current buffer,	0(0)
	SMMPL1.	0(0) 0->(SMMPL1)
		•
		36(SMMPL4)
30	B(SMMPL2) The auxiliary buffer, SMMPL2, is made the current buffer prior to writing.	
31	W ^ SYSMAC>SMMPL2	
	Write the current buffer contents as a file whose pathname is $^{\circ}$ SYSMAC>SMMPL2.	
32	1,\$D Delete the first through last line of the current buffer.	
33	B(SMMPL3)	
-	The auxiliary buffer, SMMPL3, is made the current buffer prior to writing.	

5-6

.....

.

.

Line No.	Editor Directive Description	Terminal Typeout
34	W ^ SYSMAC>SMMPL3	
	Write the current buffer contents as a file whose pathname is ^ SYSMAC>SMMPL3.	
35	1,\$D	
	Delete the first through last line of the current buffer.	
36	B(SMMPL4) The auxiliary buffer, SMMPL4, is made the current buffer prior to writing.	
37	W ^ SYSMAC>SMMPL4.	
	Write the current buffer contents as a file whose pathname is ^ SYSMAC>SMMPL4.	
38	1,\$DX	
	Delete the first through last line of the current buffer.	
	Display the status of the buffers. SMMPL4 is the current buffer. All the buffers have been cleared.	0 (0) 0 (SMMPL1)
		•
		0->(SMMPL4)
20	R SMMPL1	0->(DIVIIVII LH)
39	R SMMPL1 Read the file SMMPL1 into the current buffer, SMMPL4.	
40	/INSERT/LD//ADD L7/LD	
10	Locate the first line containing the expression, INSERT, list it and its line number, and then delete it.	3* INSERT LN
	Starting at the current line, locate the first line containing the expression, ADD L7, list it and its line number, and then delete it.	11* ADD L7 SETA
41	15R SMMPL2	
	Read the file, SMMPL2, into the current buffer after line 15 of the buffer. Two files are being merged.	
42	X52LD	
	Display the status of the buffers. Current buffer, SMMPL4, now has 52 lines.	0 (0)
		•
		52->MOD(SMMPL4)
	List line 52 then delete it.	52* USED EDIT
43	E FO >SPD>LPT00	
	The Execute directive allows you to execute the ECL command FO to change the output file from the operator's terminal to the line printer.	
44	1,\$LW SMPMAC.P	
	List the first through last line of current buffer on the line printer (Figure 5-3). Write the current buffer as a file whose pathname is SMPMAC.P.	

(

alle al s

(

.ine No.	Editor Directive Description	Terminal Typeout
5	1,\$D Delete the first through last line of the current buffer.	
6	R SMMPL3 Read the file, SMMPL3, into the current buffer, SMMPL4.	
7	X/L4/;/LE/S/SETB/SETA/ Display the status of the buffers.	0 (0)
	Locate the first line containing the expression, L4. Starting with the line containing L4 through the line containing the expression LE, substitute SETA for all occurrences of SETB.	45->(SMMPL4)
3	1,\$LW ^Z00B02>LDD>MACRO>EXEC_LIB>SAMPL1 List the first through last line of the current buffer on the line printer (Figure 5-4).	
	Write the current buffer as a library routine file whose path- name is $^{Z00B02}$ -LDD>MACRO>EXEC_ $^{\%}$ +>SAMPL1.	
	1,\$D Delete the first through last line of the current buffer.	
	R SMMPL4 Read the file, SMMPL4, into the current buffer.	
	/DLET/D Locate and delete the line containing the expression DLET.	
	Q Quit. The quit is deferred since a buffer has been modified and has not been written to a file. You have <i>one</i> more chance to write the contents of the current buffer as a file.	MODIFIED BUFFERS EXIST
3	1,\$LW ^Z00B02>LDD>MACRO>EXEC_LIB>SAMPL2 List the first through last line of the current buffer on the line printer (Figure 5-5). Write the current buffer contents as a library routine file whose pathname is ^Z00B02>LDD>MACRO>EXEC_LIB>SAMPL2.	
	X Display buffer status. Status is always displayed on the operator's terminal even though the output file is the printer.	0 (0) 35->(SMMPL4)
	E FO The Execute directive allows you to execute the ECL comand FO to change the output file from the line printer back to the operator's terminal.	
6	Q Quit. Exit from the Editor.	

USING THE EDITOR

5-8

.

ŧ

-

.

Ĵ

TITLE SMPMAC, '3/1/77' EDITOR/MACRO EXAMPLE 1 'EXECTLIB', SAMPL1, SAMPL2 LIBM 5 P1=0, P2=2, P3='SAMPLE', P4='PRUGRAM', P5=ZER0, P6=(; 3 SMPLM MAC P7=),P8=TW0,P9=\$COMM,PA=A,PB=B,PC=T2,PD=SAMPLE,PE=PROG2 4 \* SET LOCAL VALUES WITHIN MACRO ROUTINE \* 5 SETA ' PROG2.START2[,]NAME' 6 LE 7 ۲4 SETA EQU 8 L5 SETA RESV q L6 SETA TEXT 10 L7 SETA XDEF 1.8 SETA XLOC 11 12 L9 SETA XVAL [2'01'] LA SETA 13 СОММ 14 LB SETA SETA ۰, ۱ 15 LZ 16 \* 17 \* THESE UNPRUTECTED COMMENT LINES WILL BE DROPPED \* WHEN MACRU PREPROCESSED. 18 19 50 ?G4 ?L4 ?61 (G1 INITIAL VALUE=8) ?IX(?LE,?PE)?G8?P1 215 21 ?65 ?P3?VL(35)?P4?LZ?LA 55 ?L6 23 ?17 ?G4 ?P6?P8?G8?AL(?PC)?P7 24 ?L7 ?SS(?P4,7,1) 25 ?L8 219 ?VP(11) 26 ?G6 ?L8 267 27 ?GA ?∟4 ?P9+?G3 28 ENUM 29 30 G 3 SETN 'ZERO' (APOSTROPHE'S DROPPED WHEN SUBSTI.) 31 G4 SETA INAME! G 5 SETA 32 'SCOMM' 33 G6 SETA SETN 100 34 G7 'COM1' 35 GΑ SETA SETA 1,1 36 GВ 37 \* \*\*\*\* THE FOLLOWING PORTION OF CODE IS ADDED FROM "SMPLM" \*\*\*\* 38 39 \* (CALL IN-LINE MACRO ROUTINE) 4 Ŭ SMPLM, 41 \*\*\*\* THE FOLLOWING PORTION OF CODE IS ADDED FROM "SAMPL1" \*\*\*\* 42 43 44 CALL1 SAMPL1 1,,,,,+,150,; ,,START,SC 45 46 47 \*\*\*\* THE FULLOWING PORTION OF CODE IS ADDED FROM "SAMPL2" \*\*\*\* 48 49 CALLS SAMPL2 \$F,,,,,,; ,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,, 50 SMPMAC, START 51 END

Figure 5-3. Sample of Unexpanded Assembly Language Program with Macro Calls and Statements (SMPMAC.P)

STREES.

SAMPL1 P1=0, P2=2, P3='SAMPLE', P4='PROGRAM', P5=ZERO, P6=(, P7=); MAC 1 P8=TW0, P9=\$COMM, PA=A, PB=B, PD=SAMPLE, PE=PROGRAM 5 3 4 5 \* SET LOCAL VALUES WITHIN MACRO ROUTINE \* 6 × 7 \* 8 L4 SETA ORG 9 L5 SETA DC 10 LDR L6 SETA 11 L7 SETA STR 12 L8 SETA CALL 13 L9 SETA LB 14 LA SETA **BBT** 15 LB SETA SLD 1 = 1 16 LC SETA 17 LD SETA [Z'32'] PROG2.START2[,]NAME' 18 LE SETA 19 ٠ 50 \* 21 \* SET GLOBAL VALUES WITHIN MACRO ROUTINE \* 55 \* 23 \* 24 SETA 'ORG INTO COMMON' GH 25 GG SETA 'ORG INTO INTERNAL LOC' 26 GC SETA 'EXTERN VAL REFERENCE' 27 GD 'COMMON REFERENCE' SETA 28 GE SETA 'EXTERNAL LOCATION REFERENCE' 29 GF SETA 'FORWARDS TEMP LABEL REFERENCE' 30 \* 31 \* UNPROTECTED LINES OMITTED WHEN PRE-PROCESSED 35 \* 33 ? 4 ?P9 ?GH 34 ?VR(?P3,?PD)?GB?SR(?P4,?PE) ?L5 35 ?L4 ?G4?P7?P8 **?**GG 36 37 2PC ?L6 SR1, ?LC?PB ?GC ?L7 SR1,<?GA **?G**U 38 [\*] 2PD 39 26 \$R1,<?PA ?GE 40 [\*] 41 ?L8 PROG2. ?SS (?LE, 7, 6) ?GBNAME 42 ?G4?P7?P1?GB?LC?VL(13) ?L9 43 ?LA >?P7\$F ?GF 44 ?LB \$\$1?GB?LCZ'?CH(1,=2)?CH(2,=2)?CH(3,=2)?CH(4,=2)' 45 ENDCL1 ENDM

> Figure 5-4. Sample of Unexpanded Macro Routine (SAMPL1) Contained in EXEC\_LIB Directory

USING THE EDITOR

5-10

1	SAMPL2	MAC	P1=0,P2=2,P3='SAMPLE',P4='PROGRAM',P5=ZER0,P6=(,P7=);			
1 2	P8=TW0,P9=\$CUMM,PA=A,PB=B					
3	* SET LOCAL VALUES WITHIN MACRO ROUTINE *					
4	L4	SETA	>=[2'1300']			
5	LA	SETA	IOLD			
6	LD	SETA	\$R1			
7	LE	SETA	PROG2.START[,]NAME!			
8	LG	SETA	SOR			
9	LC	SETIN	-32768			
10	LP	SETN	32767			
11	LQ	SETN	0			
	LI	SETA	8E Z			
13	LY	SETA	HLT			
14	LZ	SETA	1,1			
15	-		S WITHIN MACKU ROUTINE *			
16	G7	SETN	-32768			
17	G 2	SETA	BACKWARDS TEMP LABEL REFERENCE!			
18	65	SETA	CTRL			
19	*					
50		ECTED LINES	S OMITTED WHEN PRE-PROCESSED			
51	*	<b>.</b>				
55	?P1	?LA	?P5?LZ?L4,=?LD			
23	?P1	?LA	?P5?LZ?L4,=?L0			
24		?LG	?LD,?VG(3)			
25		?LI	?LD,-\$C ?G2			
26		?LY	207 400 (3) 1 1 1			
27		?65 155	?PZ ?SS(?LE,1,5)			
85		IFE FAIL	?G7,?LC,IFE1			
29	5 D T T	IFNL	?P2,?LC,*			
30	ENDIT IFE1	NULL	ir <b>c</b> fillf *			
32	1421	IFNE	?G7,?LP,GTEND			
33		FAIL	; 0/ / ; LF / 0   L NU			
34	GTEND	GOTO	ENUIT			
35	ENDCL2	ENDM				

Figure 5-5. Sample of Unexpanded Macro Routine (SAMPL2) Contained in EXEC\_LIB Directory

Example 1:

湯湯込

File SMPM01 before Editing

```
TITLE SMPMAC, '3/1/77' EDITOR/MACRU EXAMPLE
* INSERT LN 2 LIBM STATEMNT BEFORE THIS LN..THEN DEL THIS LN
 1
 2
                          P1=0, P2=2, P3='SAMPLE', P4='PROGRAM', P5=ZER0, P6=(;
    SMPLM
 3
                MAC
 4
    P7=),P6=TW0,P9=$C0MM,PA=A,P6=6,PC=T2,P0=SAMPLE,PE=PR0G2
 5
    * SET LUCAL VALUES WITHIN MACRO ROUTINE *
                USE CHANGE FUNCTION TO ADD SETA VALUE FOR THIS LN
 6
   LE
                USE CHANGE FUNCTION TO ADD SETA VALUE FOR THIS LN USE CHANGE FUNCTION TO ADD SETA VALUE FOR THIS LN
 7
    L4
 8
    LS
 9
                USE CHANGE FUNCTION TO ADD SETA VALUE FOR THIS LN
    L6
10
    * ADD L7 SETA VALUE W/CHANGE FUNCTION .. THEN DELT THIS LN
                SETA
11
    L8
                            XLOC
12
    L9
                SETA
                            XVAL
                            [Z'01']
                SETA
13
    LA
14
    LВ
                SETA
                            СОММ
                            1, 1
15
    LZ
                SETA
    *USED EDIT READ FUNCT TO ADD "SMPMO2" PORTION TO FILE*
16
```

## Example 2:

File SMPM02 before Editing

```
1
   * THESE UNPROTECTED COMMENT LINES WILL BE OROPPED
 5
 3
    * WHEN MACRO PREPROCESSED.
 4
    *
 5
    264
              おし4
                         ?G1
                                    (G1 IMITIAL VALUE=5)
                         ?1X(#LE, ?PE)?68?P1
               #L5
 6
 7
    265
              #L6
                         ?P3?VL(35)?P4#LZ#LA
 8
              #L7
                         ?G4
 9
              #L7
                         ?P6?P8?G8?AL(?PC)?P7
10
                         ?SS(?P4;7,1)
              #L8
11
               #L9
                         ?VP(11)
    ?G6
              #LB
                         ?G7
12
                         ?P9+?G3
13
    ?GA
              #L4
14
               ENDM
              SETN
15 G3
                         1
                         ZER0'
16
    G 4
               SETA
                                    (APOSTROPHE'S DROPPED WHEN SUBSTI.)
                          INAME!
17
   65
              SETA
                         ISCOMM!
18
    G6
               SETA
19
    67
              SETIN
                         100
20
   GA
              SETA
                          'COM1'
                         1,1
21
   GB
               SETA
55
    *
   **** THE FULLOWING PORTION OF CODE IS ADDED FROM "SMPLM" ****
53
24
    *
25
              SHPLM.
                                    (CALL IN-LINE MACRO ROUTINE)
56
27
   **** THE FOLLOWING PORTION OF CODE IS ADDED FRUM "SAMPL1" ****
58
    *
29
   CALL1
              SAMPL1
                         1,,,,,+,150,;
    , START, SC
30
31
    **** THE FULLOWING PORTION OF CODE IS ADDED FROM "SAMPL2" ****
35
53
34
   CALLS
               SAMPL2
                         $F,,,,,,;
35
                   ,,,,,,,,,,,LINK
    . . . . . . . . . . . . .
36
              END
                         SMPMAC, START
```

5 - 12

## Example 3:

l

编制

File SMPM03 before Editing

1	SAMPL1	MAC	P1=0, P2=2, P3='SAMPLE', P4='PROGR	AM1. P5=7FR0. P6=(. P7=):
ż	-	1	A, P8=B, PD=SAMPLE, PE=PROGRAM	
3	*			
4	*			
5	* SET LOC	AL VALUES	MITHIN MACRO ROUTINE *	
6	·.★			
7	*			
8	L4	SETB	URG	
9	L5	SETB	DC	
10	L6	SETH	LDR	
11	L7	SETB	STR	
12	LB	SETB	CALL	
13	L9	SETB	LB	
14	LA	SETB	BBT	
15	LB	SETH	SLD	
16	LC	SETH		
17	LD	SETB	[2'32']	
18	LE	SETB	PROG2.START2[,]NAME'	
19	*			
20	*			
21		BAL VALUES	WITHIN MACKO ROUTINE *	
53 55	*			
24	* GH	SETA	'ORG INTO COMMON'	
25	GG	SETA	ORG INTO INTERNAL LOC'	
26	GC	SETA	'EXTERN VAL REFERENCE'	
27	GD	SETA	COMMON REFERENCE!	
28	GE	SETA	"EXTERNAL LOCATION REFERENCE"	
2.9	GF	SETA	'FORWARDS TEMP LAALL REFERENCE'	
30	*	0215		
31		OTEN LINES	UMITTED WHEN PRE-PROCESSED	
35		LIED LINES	OFITIED WHEN FRE-FRUCESSED	
33	•	?L4	?Р9 ?Gн	
34		?15	?VR(?P3,?PD)?GB?SH(?P4,?PE)	
35		?L4	?G4?P7?P8 ?GG	
36	?PC	716		GC
37	1 F G	267	SR1, GA ?GU</th <th>GC .</th>	GC .
38	[*]	• • •	unit - 10 - 100	
39	?PD	?L6	5R1, PA ?GE</th <th></th>	
40	(*)			
41		?L8	PRUG2. 255 (2LE, 7, 6) 265 NAME	
42		?L9	?G4?P7?P1?G8?LC?VL(13)	
43		?LA	>?P7\$F ?GF	
44		?LB	\$\$1?GB?LCZ'?CH(1,=2)?CH(2,=2)?C	H(3,-2)?CH(4,-2)!
45	ENDCL1	ENDM		

.

#### **Example 4**:

#### File SMPM04 before Editing

```
P1=0, P2=2, P3='SAMPLE', P4='PROGRAM', P5=ZER0, P6=(, P7=);
    SAMPL2
              MAC
 1
    P8=TWU, P9=SCOMM, PA=A, P8=B
 5
 3
    * SET LOCAL VALUES WITHIN MACRU ROUTINE *
 4
   L4
               SETA
                         >=[Z'1300']
    LA
 5
               SETA
                         1010
    LD
               SETA
                         5Ř1
 6
               SETA
                         PROG2.START [,] NAME!
 7
    LE
 8
    LG
               SETA
                         50k
 9
    LC
               SETIN
                         -32768
               DC "DELETE LINE ENDING IN S'S
   UEL TS
10
11
    LP
               SETN
                         32767
   Lù
               SETN
12
                         0
               SETA
                         BEZ
13
    LI
14
    LY
               SETA
                         HLT
                         ۱,۱
               SETA
15
   LZ
    * SET GLUBAL VALUES WITHIN MACRO-RUNTINE *
16
17
    G7
               SETN
                         -32766
                         BACKWARDS TEMP LABEL REFERENCE!
   6 S
               SETA
18
19
   GS
               SETA
                         CTRL
50
    *
    * UNPROTECTED LINES UMITTED WHEN PRE-PROCESSED
51
55
    *
    ?P1
23
               ?LA
                         ?P5?LZ?L4,=?L0
24
    ?P1
               ?LA
                         ?P5?LZ?L4,=?LD
25
               ?LG
                         ?LD,?VG(3)
                         ?LD,-$C
                                    365
59
               ?LI
27
               ?LY
                         ?PZ ?SS(?LE,1,5)
28
               ?G5
               DC 'DELETE LINE BEGIGNING IN "
29
    *DEL
                         2G7, 2LP, GTEND
30
               IFNE
31
               FAIL
    GTEND
               GUTO
                         ENDIT
32
               DC 'DELETE LINE BEFORE QUIT'
33
    OLET
34
    ENDCL2
               ENDM
```

5-14

### Using the Assembler and Macro Preprocessor

This section illustrates the use of the assembler to construct programs containing macro calls. Two assembly language samples are presented. One illustrates the output of different assembly language program processors. The other illustrates an assembly language program that contains multiple tasks. Both samples provide the operator terminal session listings that contain the commands to invoke the system software.

#### SAMPLE ASSEMBLY LANGUAGE SESSION (SMPMAC)

Figure 6-1 contains a sample operator terminal session to preprocess, assemble with crossreference, and link the assembly language program SMPMAC. It is assumed that the Honeywell-supplied startup has been done prior to this session.

The typeout illustrates the following points. The working directory is changed to SYSMAC where the program's files are located. The file, SMPMAC, that is processed by the Macro Preprocessor is in Figure 5-3. The macro routines, SAMPL1 and SAMPL2, called by the program are listed in Figures 5-4 and 5-5. A listing of the output file SMPMAC. A from the macro preprocessor is shown in Figure 6-2.

```
CWD ↑SYSMAC
($H)RDY:
+ZSYS51>SYSLIB2>MACROP +SYSMAC>SMPMAC -SZ 10
($H)MACROP-0100-11/17/1404
($H)0000 ERR COUNT
($H)RDY:
ASSEM ^SYSMAC>SMPMAC -SZ 10 -SAF -LE -XREF -COUT >SPD>LPTOO
($H)ASSEM-0100-11/17/1346
($H)0000 ERR COUNT
($H)ASSEM: (020105) 1D 1380 0000 0000
($H)>SPD>LPTOO
($H)RDY:
FO >SPD>LPTOO
($H)RDY:
LINKER +SYSMAC>SMPMAC -COUT >SPD>LPTOO -SZ 10
($H)LINKER-0100-11/23/1258
LDEF A,X'100'
VDEF B,X'2'
LINK SMPMAC
MAP
0T
($H)SAF OR SLIC PROG2.0
                           NT FND
($H)SAF OR SLIC PROG2.0
                            NT FND
 $H)ROOT SMPMAC
($H)LINK DONE
($H)RDY:
```

Figure 6-1. Sample Terminal Session (SMPMAC)

「「「「「」」

TITLE SMPMAC, '3/1/77' EDITOR/MACRO EXAMPLE 1 5 3 \*\*\*\* THE FOLLOWING PORTION OF CODE IS ADDED FROM "SMPLM" \*\*\*\* 4 5 ZERO EQU \$ (G1 INITIAL VALUE=S) RESV 2.0 6 7 NAME TEXT 'SAMPLE', 'PROGRAM', Z'01' 8 XDEF ZERO 9 XUEF (TW0,2) 10 XLUC A 11 XVAL 6 12 SCUMM COMM 100 \$COMM+1 13 COM1 Euu 14 \*\*\*\* THE FULLOWING PORTION OF CODE IS ADDED FROM "SAMPL1" \*\*\*\* 15 16 \* 17 ORG \$COMM URG INTO COMMON DC 18 1,2 ORG ZER0+150 URG INTO INTERNAL LOC 19 EXTERN VAL REFERENCE START LDR 20 \$R1,=8 COMMON REFERENCE 21 STR \$R1,<COM1 22 EXTERNAL LUCATION REFERENCE SC. LDR \$R1, <A 23 24 \* CALL PROG2.START2, NAME 25 56 LB ZER0+1,=Z'32' BBT FORWARDS TEMP LABEL REFERENCE 27 >+\$F \$\$1,=Z'01020304' 85 SLD 29 \*\*\*\* THE FOLLOWING PORTION OF CODE IS ADDED FROM "SAMPL2" \*\*\*\* 30 31 32 \$F IOLO ZER0,>=Z'1300',=\$R1 ŝF IOLD ZERO,>=Z'1300',=\$R1 33 34 SUR \$R1,1 \$R1,-\$C BACKWARDS TEMP LAHEL REFERENCE 35 BEZ 36 HLT 37 CTRL LINK PROG2 SMPMAC, START 38 END

Figure 6-2. Macro Preprocessor Output (SMPMAC)

Figure 6-3 illustrates a cross-reference listing produced by the Assembler. See the *Program Preparation and Checkout* manual for an explanation of cross-reference symbols.

Figure 6-4 illustrates the Assembler listing of the assembled Macro Preprocessor output. Figure 6-5 illustrates the link map of the previously assembled program SMPMAC.

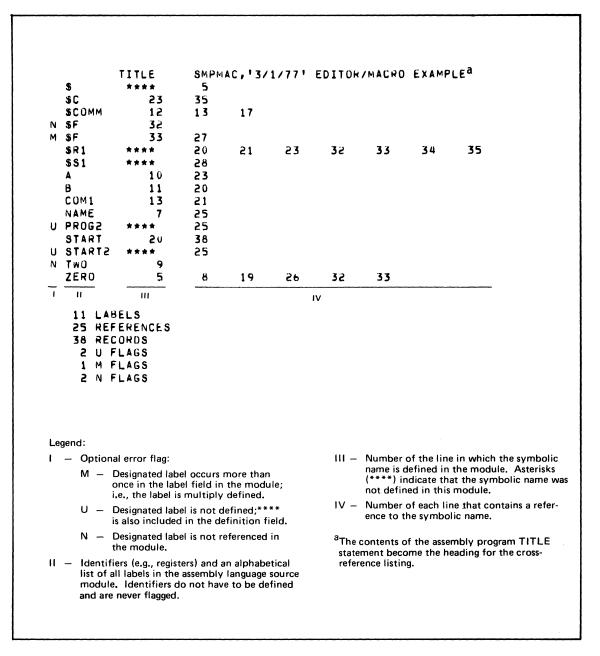


Figure 6-3. Cross-Reference Listing (SMPMAC)

SMPMAC	3/1/77	GCO	DS6/MDT	0101	ASSEMBLER	EDITOR	MACRO EXAMPI	L PAGE 0001	1901/01/01 0026:59.4
000001 000002					•	TITLE	SMPMAC, '3	/1/77' EDITO	OR/MACRO EXAMPLE
000003						FOLLOWIN	G PORTION OF	CODE IS ADD	DED FROM "SMPLM" ****
000005	0000		0000		ZERO	EQU Pesv	\$ 2,0	(G1 INITIAL	L VALUE=\$)
000007	0002 0003 0004 0005 0006 0007 0008	5341 4050 4045 5052 4F47 5241 4001			NAME	TEXT	'SAMPLE',	'PROGRAM',	Z'01'
000008		0000				XDEF	ZERU		
000009		0005				XDEF	(140,2)		
000010		0002				XLOC	A		
000011 000012 000013 000014		0064	0001	ĸ	SCOMM COM1	XVAL Comm Egu	H 100 SCOMM+1		
000015					**** THE	FOLLOWI	IG PORTION OF	CODE IS AD	DED FROM "SAMPL1" ****
000016 000017 000018	0000 0000 0001	0001		ĸ	•	URG UC	\$CUMM 1,2	1	ORG INTO COMMON
000019 000020 000021	0096 0096 0098		6000 0001	X K	START	OFG Lor Str	ZER0+150 SR1,=8 SR1, <c001< td=""><td></td><td>ORG INTU INTERNAL LOC Extern val reference Common reference</td></c001<>		ORG INTU INTERNAL LOC Extern val reference Common reference
250000 250000	0 (° 9 A	9800	0000	×	* \$C	LDK	SR1, <a< td=""><td>1</td><td>EXTERNAL LOCATION REFERENCE</td></a<>	1	EXTERNAL LOCATION REFERENCE
000024 000025	009E	0360	000 <b>3</b> 000 <b>0</b>		*	CALL	PRUG2.STA	RIS,NAME	
000026	00A0 00A1 00A2 00A2	0F80 0002 8200 3200	FFSE	T		Le	ZER0+1,=Z	'32'	
150000 850000 950000		0500	0102 0	1 304		HET SLD	>+\$F \$S1,=7'01		FORWARDS TEMP LABEL REFERENC
000030					**** THE	FULLOWID	G PURTION OF	CODE IS AD	DED FROM "SAMPL2" ****
000032	00A9 00A5 00AC	81C0 1300 0051	FF56		ŝF	1010	2ERU,>=2'	1300',=\$R1	
000033			FF52		۵F	IOLD	ZER0,>=Z'	1300',=\$R1	
0000 <b>3</b> 4 0000 <b>3</b> 5 0000 <b>3</b> 6	0081 0082 0084	1041 1901 0000	FFE7	т		SOR DEZ HLT	\$R1,1 \$R1,-\$C	BACKWARDS	TEMP LABEL REFERENCE
000037 000038 0000 ERR C		0096				CTHL END	LINK PRUG Smpmac,st		
1 11	111		IV	— <u>v</u>				VI	
		Option A C	Numeric	field for	ormaterror sionerror hentoutofrange		tation co of the n	prresponding to umber of logic	digit decimal represen- o the sequential count al records read. git hexadecimal repre-
		E F	Illegal ad	dress e rward	xpression reference		sentation correspo	n of the rela	ative address of the statement on the
		M – N – O –	No matc Illegal op	-define hing le eratio	d symbol ft parenthesis	ər	decimal assembly control	representation / language ins statement sho	3-, or 12-digit hexa- n of the corresponding truction or Assembler own on the right-hand
		Q – R – S –	Address< Illegal re Imprope	<0 or ≩ gister r r statei	≥32K		V – Type fl nonhexa label typ	decimal digit le of the refere	er flag (preferably a :) that specifies the enced symbol:
		U X	constant Undefine Expressi	ed sym on too	-		X — E P — P	emporary xternal	nce to external bol
		If ther	e are mo	re thai	our error flags per n four errors, onl d included in the	y the	VI – Verbatir ments, o	n representat of the source	tion, including com- statement, as defined uage manual.

### Figure 6-4. Assembler Output Listing (SMPMAC)

USING THE ASSEMBLER AND MACRO PREPROCESSOR

6-4

LINKER-0100-11/23/1258 GC0S6 MOD400-S100-12/01/1413 BU= SMPMAC LINKED ON: 1977/12/08 1420:55.9 -SAF SMPMAC 3/1/77 1977/12/08 1420:12.4 ASSFMBLER-0100-11/17/1346 GCOS6 MOD400-S100-12/01/1413 Ρ EDITOR/MACRO EXAMPL SAF OR SLIC PROG2.0 NT FND SAF OR SLIC PROG2.0 NT FND \*\* SMPMAC LINK MAP 1977/12/08 1420:55.9 \*\*START OOFA \*\*LOW 0000 0119 \*\*HIGH \*\*\$COMM 0000 \*\*CURRENT 0119 \*\*EXT DEFS Ρ ZHCOMM 0000 ρ ZHPEL 0000 ۸ 0100 в 0002 ROOT 0000 \*\* SMPMAC 0000 \* С SCOMM. 0000 ZERO 0064 TWO 0002 \*\*UNDEF \* SMPMAC 0000 START2 0103 \*\*\*\*\*\* ROOT SMPMAC \*\*\*\*\*\* HIGHEST OVLY /NUM OF SYMS 0 \*\*\*\*\*\* SAF \*\*\*\*\*\* ROOT SMPMAC BASE 0000 ST OOFA -...UI HIGH=0119 \*\*\*\*\*\*\* \*SIZE OF ROOT AND STATIC OVLYS= 0119 HT REL RCD= 4 \*\*\*\*\*\*\*\* LINK DONF \*\*\*\*\*\* I II III Legend: I - Indicates whether there is a protected symbol, multiply-defined symbol, or symbol that defines the labeled or unlabeled common; designated by P, M, and C, respectively. II - Module and symbol names. (Module names are preceded by \*.) III - Base address of module, address or value of symbol.

Figure 6-5. Linker Output Listing (SMPMAC)

#### SAMPLE ASSEMBLY LANGUAGE MULTITASK PROGRAM (BRDCST)

Figure 6-6 contains a sample terminal session to compile, link, execute, and start debugging the assembly language, program BRDCST, on that system. A specialized system is configured with CLM\_USER containing the following configuration directives:

DEVICE KSR00,5,0,X'0500',CONSOLE MEMPOOL S,,5000 DEVICE CDR00,26,26,X'1300' MEMPOOL E,AA,14336,,BB,11264 MEMPOOL B,,8850

SYS 60,16,SSIP,3 DEVICE DSK01,17,17,X'480' DEVICE DSK02,18,18,X'1200' DEVICE DSK03,19,19,X'1280' DEVICE LPT00,20,20,X'1380',LPT00 QUIT

The typeout illustrates the following points. A task group, \$H, is spawned. Editor is used to print the file containing BRDCST source text, a portion of which is shown in Example 1. The Macro Preprocessor, required for processing of \$IORB, \$CRTSK, and \$RQTSK macro calls, is not on the directory search path and a full pathname must be used. A task group, BC, in which to execute BRDCST is created and BRDCST is loaded for execution.

Example 1 is a listing of BRDCST. It is presented to illustrate how tasks are created and invoked in an assembly language multitask program.

6-6

SG \$H H.L.A 38 >SPD>CONSOLE -OUT >SPD>CONSOLE -POOL AA -WD ^ZOOBOO (\$S)RDY: C :\$H: RDN (\$H)GROUP READY (\$S)GROUP \$H DID NOT ACCEPT INPUT RDN (\$H)RDY: CWD ^ETSCOM>MAN\_EX (\$H)RDY: FO >SPD>LPTOO ED (\$H)EDIT rrrr-mm/dd/hhmm R BRDCST.P 1,\$P Q FO >SPD>CONSOLE (\$H)RDY: ^Zrrr02>SYSLIB2>MACROP BRDCST -SZ 20 -IC (\$H)MACROP rrrr-mm/dd/hhmm (\$H)0000 ERR COUNT (\$H)RDY: ASSEM BRDCST -SIZE 1 -COUT >SPD>LPTOO (\$H)ASSEM rrrr-mm/dd/hhmm (\$H)0000 ERR COUNT (\$H)RDY: LINKER BRDCST -C >SPD>LPTOO -S 2 (\$H)LINKER rrrr BU=BRDCST LINKED ON: yyyy/mm/dd nnmm:ss.t IN ^ETSCOM>MAN\_EX LN BRDCST START BRDCST MP QT. (\$H)ROOT >BRDCST (\$H)LINK DONE (\$H)RDY: C :\$S: CG BC 40 -LRN 30 -POOL BB -EFN ^ETSCOM>MAN EX>BRDCST (\$S)RDY: EGR BC B.E.N >SPD>CONSOLE -WD ^ETSCOM -OUT >SPD>CONSOLE (\$S)RDY:

#### Figure 6-6. Sample Terminal Session (BRDCST)

#### Example 1:

Partial Program Listing of BRDCST

*	TITLE BRDCST LIBM >LDD>MACRO>EXEC_LIB'	
* * * *	THIS TEST PROGRAM IS A MEDIA TRANSCRIPTION TEST. IT CAN EXECUTE AS AN ON-LINE OR BATCH DRIVER TEST	
DEVTBL		RS TO DEVICE I/O BLOCKS

USING THE ASSEMBLER AND MACRO PREPROCESSOR

*	DC DC DC DC	<dskblo <ttyout <asrinp <asrout< th=""><th>LRN 18 LRN 12 LRN 10 LRN 11</th><th>}</th><th>POINTERS TO DEVICE I/O REQUEST BLOCKS</th></asrout<></asrinp </ttyout </dskblo 	LRN 18 LRN 12 LRN 10 LRN 11	}	POINTERS TO DEVICE I/O REQUEST BLOCKS
CRDBLK	\$IORB	26,WAIT,,BUFFER,,8	0		
TTYBLK *	\$IORB	13,WAIT,,BUFFER,,8	0		
DSKBLI *	\$IORB	17,WAIT,,BUFFER,,8	0	>	I/O REQUEST BLOCK DEFINITIONS USING \$IORB MACRO CALLS
DSKBLO	\$IORB	18,WAIT,,BUFFER,,8	0		
^	:	© ₽		)	
BUFFER TILRN	RESV RESV RESV DC	80 80 80 26			BUFFER, LRN AND LEVEL
TO1LRN	DC	27		>	DEFINITIONS
TILVL TO1LVL	DC DC	1 2			
	•	0 • •		)	
* *		TASK 1 (INPUT) REQ	UEST BLOCK LEN 26	\	
* TASK01	\$TRB	26,WAIT,,INSTRT			
*	ψικυ	TASK 2 (OUTPUT) RE	OUEST BLOCK I PN 27		TASK REQUEST BLOCK DEFINITIONS USING
* TASK02	\$TRB	27,WAIT,,OUTSTR	QUEST BLUCK ENN 27		\$TRB MACRO CALLS
*	φικο •	27,WAIT,,00131K			
	:	•		/	
* BRDCST	SET U LDR LDR STH STR	ISER BIT TO INDICATE \$R2,=Z'0021' \$R1, <dskbli+\$af \$R2,=\$R1 \$R1,<dskbli+\$af< td=""><td>E DISK</td><td></td><td>PROGRAM ENTRY POINT AND INITIALIZATION CODE</td></dskbli+\$af<></dskbli+\$af 	E DISK		PROGRAM ENTRY POINT AND INITIALIZATION CODE
	•	•••••••••••			
*	° CDEAT	° E INPUT, OUTPUT TAS	and a second	)	
	\$CRTS	K TILRN,TILVL,INST K TOILRN,TOILVL,OUT	RT		TASK CREATION USING \$CRTSK MACRO CALLS
*	: דווסאד	TASK REQUEST		)	
*	111 01	THOR REQUEST			
INPUTR *	\$RQTS	SK TASKO1			ENTRY OF TASK REQUESTS USING \$RQTSK MACRO CALLS
*	Ουτρι	JT TASK REQUEST			
*	\$RQTS	SK TASKO2			
	:	•			
	T/ •	ASK EXECUTION CODE			
*	•	•		١	
*	XDFF	S AND XLOCS			XDEFS AND XLOCS FOR
*	.XDEF	BRDCST BRDCST		Ś	LINKER PROCESSING

USING THE ASSEMBLER AND MACRO PREPROCESSOR

### Section 7 Using the COBOL Compiler

This section illustrates the use of the COBOL compiler to construct programs written in the COBOL language. It shows how to load a source program from a card deck into a mass storage COBOL source file and how to subsequently invoke the compiler to process the source program from the mass storage file. Two samples are presented; one shows the procedure for running an application, and the other is the ouput listing from an actual compilation and link.

#### SAMPLE CARD-TO-DISK-FILE PROGRAM (CARDIN)

Example 1 is a sample COBOL source program that places data read from cards onto a disk file. The following paragraphs illustrate a procedure for creating application files, loading source, compiling, linking, and executing. System startup has created the application task group \$H. After startup, the current working directory is ^Zrrr01>SYSLIB1. The following summarizes the contents of volumes used in commands:

#### **Volume Device Unit Contents**

Zrrr00	DSK00	Bootstrap, Monitor, Linker
Zrrr01	DSK01	SYSLIB1, SYSLIB2
Zrrr04	DSK02	COBOL Compiler
VOL03	DSK03	Application Files

Example 1:

440.44

Program Listing of CARDIN

000010	IDENTIFICATION DIVISION.
000020	PROGRAM-ID. CARDIN.
000030	ENVIRONMENT DIVISION.
000040	CONFIGURATION SECTION.
<b>00005</b> 0	INPUT-OUTPUT SECTION.
<b>00006</b> 0	FILE-CUNTROL.
<b>00007</b> 0	SELECT CARD ASSIGN TO DA-CARD-READER.
000080	SELECT MASTER ASSIGN TO OC-MSD.
000090	DATA DIVISION.
<b>00010</b> 0	FILE SECTION.
<b>00011</b> 0	FD CARD LABEL RECORDS OMITTED.
000113	C1 CARD-REC PIC X(80).
000116	FD MASTER LABEL RECORDS OMITTED.
<b>00012</b> 0	C1 MASTER-REC PIC X(80).
<b>00013</b> 0	PROCEDURE DIVISION.
<b>00014</b> 0	CARDIN.
<b>00015</b> 0	OPEN INPUT CARD.
<b>00016</b> 0	OPEN OUTPUT MASTER.
<b>00017</b> 0	LOOP.
<b>00018</b> 0	READ CARD RECORD AT END GO TO EOF.
000190	MOVE CARD-REC TO MASTER-REC.
000200	WRITE MASTER-REC.
<b>00021</b> 0	GO TO LOOP.
000220	EOF.
000230	CLOSE CARD.
000240	CLOSE MASTER.
<b>00025</b> 0	STOP RUN.
<b>00026</b> 0	END COBOL

**ATT** 

. ...

$\Delta C \Delta$	:\$H:	
RDN		
CV	>SPD>DSK03 -FT VOL03	Format volume
$\mathbf{C}\mathbf{D}$	^ VOL03>SOURCE	Create director
CD	^ VOL03>OBJECT	source, object a
CD	^ VOL03>FILES	
CF	^ VOL03>FILES>OLD_MASTER -N_REL -RSZ 128	Create user fil
FO	>SPD>LPT00	
LS	-PN ^ VOL03	List contents o
LS	-PN ^ VOL03>OBJECT	directories
$\mathbf{LS}$	-PN ^ VOL03>FILES	
FO		

ie VOL03 ries for and user files

le

of created

#### SOURCE LOADING

The following illustrates loading a source deck using the CP command. Place the source decks for CARDIN in the card reader in the following sequence:

**CARDIN** source deck EOF (11-5-8-9) card

Enter the following command:

CP >SPD>CDR00 ^ VOL03>SOURCE>CARDIN.C

#### **COMPILING WITH COBOL**

In the following ECL commands, the working directory is ^ VOL03>OBJECT. The search path for bound units (executable programs) is current working directory, LIB1, then LIB2, where their pathname is initially ^Zrrr01>SYSLIB1. The COBOL Compiler is not in any directories in the search path; its full pathname must be given. However, the command

COBOL <SOURCE>CARDIN...

can be used if you change the pathname for the directory LIB2 by issuing an operator command to the system task group using:

 $\Delta$ S $\Delta$ CSD -LIB2 ^ Zrrr04 (where  $\Delta$  is exactly one space)

^ VOL03>OBJECT will contain temporary work files required for the compiler and the created object files used by the Linker. The compiler argument LD will list source, data map, and errors; LO will, in addition, list the object text.

To compile CARDIN enter the following:

CWD ^ VOL03>OBJECT COBOL <SOURCE>CARDIN -LO -COUT >SPD>LPT00

#### LINKING

The working directory is still ^ VOL03>OBJECT. The Linker LIB directive directs the Linker to search the secondary directory for COBOL run-time routines (ZCRT) required for linking. To link, enter the following commands:

LINKER CARDIN -COUT >SPD>LPT00 -SIZE 4 LIB ^ Zrrr04>ZCRT LINK CARDIN MAP:QT

7-2

#### EXECUTING

The internal file names 0A and 0C translate to logical file numbers 01 and 03, respectively, and must be associated with the pathnames or the physical devices through a GET or ASSOC command. To execute the program, enter CARDIN.

Enter the following commands:

GET 01 >SPD>CDR00 GET 03 ^VOL03>FILES>OLD\_MASTER CARDIN

#### SAMPLE COBOL TERMINAL SESSION (AC8111)

Figure 7-1 illustrates an operator terminal session in which a system is configured, and the COBOL program AC8111 is compiled, linked, and executed on that system. The Entry Level COBOL compiler is specified in this session. To seeify the Intermediate COBOL compiler, change the COBOL command and the LINKER LIB directive as follows.

COBOLI AC8111 -LO -COUT >SPD>LPT00 LIB ^ZSYS51>ZCIRT;LINK AC8111;MAP;QT

The LINKER LIB directive directs the Linker to search the secondary directory for COBOL run-time routines required for linking. To execute the program, enter AC8111.

RDN (\$H)RDY: CWD ↑STCOB1>SOURCE>ACC208 (\$H)RDY: COBOL AC8111 -LO -COUT >SPD>LPTOO
(\$H)COBOL 0200 11/22/1511 (\$H) 0000 ERRORS (\$H)END COMPILATION (\$H)RDY: LINKER AC8111 -COUT >SPD>LPT00 -SZ 4
(\$H)LINKER-0100-11/23/1258 LIB +ZSYS51>ZCRT;LINK AC8111;MAP;QT (\$H)ROOT AC8111 (\$H)LINK DONE (\$H)RDY: AC8111
(\$H)Q208NUA011001 (\$H)Q208NUA011001 1 2 3 4 (\$H) P P P P (\$H)RDY:

Figure 7-1. Sample Terminal Session (AC8111)

Figure 7-2 is a listing of the program AC8111, its compiled object text, and the output from the Linker. The program was compiled using the entry-level compiler.

SOURCE PROGRAM **IDFNTIFICATION DIVISION.** \*PROGRAM 0208401101.COBOL FROM 0208ACC.ARCHIVE. 2 3 PROGRAM-ID. AC8111. 4 ENVIRONMENT DIVISION. 5 CONFIGURATION SECTION. SOURCE-COMPUTER. LEVEL-6. 6 OBJECT-COMPUTER. LEVEL-6 PROGRAM COLLATING SEQUENCE IS ASCTI. 7 8 DATA DIVISION. 9 WORKING-STORAGE SECTION. 10 01 ODSPLYPEC. 05 ROSPLYFIX. 11 PIC X(13) VALUE "0208NUA011001" 12 10 FILLER 13 PIC XX 10 RTCASE VALUE SPACES. 14 10 FILLER PIC XX VALUE SPACES. 15 PIC XX VALUE SPACES. 10 OSTATUS PIC XX VALUE SPACES. 10 FILLER 16 17 ODSPLYVBL. 05 10 RACTRESLT PIC X(12) VALUE SPACES. 18 19 FILLER PIC XX VALUE SPACES. 10 DEXPRESET PIC X(12) VALUE SPACES. 20 10 21 10 FILLER PIC XX VALUE SPACES. SUMMARYS. 55 01 05 SUM-LINE PIC X(7) VALUE "1 2 3 4". 23 24 RESULTS. 05 25 10 TESTIR PIC XX. 56 TEST2R PIC XX. 10 27 TEST3R PIC XX. 10 28 10 PIC XX. TEST4R 29 \* \* \* TEST GO TO--FORWARD AND BACK \* \* \* 30 PROCEDURE DIVISION. 31 ANFANG. 32 DISPLAY ODSPLYFIX. 33 GO TO PARA-3. 34 WBA1. "GO TO PARA-3" 35 MOVE то **QEXPRESLT**. "FELL THRU" MOVE TO GACTRESLT. 36 "01" "01" TO "F" TO QTCASE. 37 MOVE 38 MOVE TESTIR. 39 DISPLAY NDSPLYREC. 40 PARA-1. "P" TO MOVE TEST3R. 41 42 GO TO. EOJ1. 43 WBA2. 44 MOVE "GO TO FOJ1" TO REXPRESLT. 45 MOVE "FELL THRU" TO DACTRESLT. "04" TO QTCASE. 46 MOVE "F" TO TEST4R. 47 MOVE DISPLAY ODSPLYREC. 48 49 PARA-2. 50 "P" MOVE TO TEST2R. 51 GO TO PARA-1. 52 WBA3. 53 "GO TO PARA-1" TO GEXPRESLT. MOVE IO DACIRESLI. 54 MOVE "FELL THRU" TO QTCASE. 55 MOVE "03" 56 MOVE "F" TO TEST3R. DISPLAY QDSPLYREC. 57 58 PARA-3. нрн MOVE 59 TO TESTIR. 60 TO PARA-2. 60 61 WBA4. MOVE "GO TO PARA-2" TO GEXPRESLT. 62 "FELL THRU" TO QACTRESLT. MOVE 63

#### Figure 7-2. Sample Listings for AC8111

64 MOVE "02" TO OTCASE MOVE "F" TO TEST2R. 65 66 DISPLAY ODSPLYREC. 67 EOJ1. "P" TO MOVE TEST4R. 68 MOVE SPACES TO QTCASE. 69 TO 70 MOVE SPACES QSTAIUS. 71 DISPLAY ODSPLYFIX SUM-LINE. QDSPLYFIX. MOVE SPACES TO 72 DISPLAY RDSPLYFIX RESULTS. 73 STOP RUN. 74 75 FND COBOL. DATA ALLOCATION MAP LHAD AL PICTURE LEVEL NO. NAME WORKING-STORAGE SECTION 0000 X(000049) 01 ODSPLYREC QDSPLYFIX 0000 X(000021) 05 0000 X(000013)10 FILLER 0006 н X(000002) 10 OTCASE 0007 н X(000002) FILLER 10 н 0008 10 QSTATUS X(000005) 0009 H. X(000002) FILLER 10 0 0 0 A X(000028) н 05 **QDSPLYVBL** 0 0 0 A н X(000012) 10 GACTRESLT 0010 H. X(000002) 10 FILLER (DATA ALLOCATION MAP) QEXPRESLT 0011 н X(000012) 10 0017 н X(000005) 10 FILLER SUMMARYS 0019 X(000015) 0 05 0019 X(000007) SUM-LINE 0010 н X(000008) 05 RESULTS 0010 н X(000002) TESTIR 10 X(000002) 001D н 10 TEST2R TEST3R 001E н X(000002) 10 001F X (000002) н... **TEST4R** -10---NO DIAGNOSTICS (PICTURE) (HALF-WORD INDICATOR; DESIGNATED BY H) (STARTING ADDRESS OF DATA) (DATA NAME) (GROUP AND ELEMENTARY ITEM LEVEL NUMBERS) (GROUP LEVEL NUMBERS)

Figure 7-2 (cont). Sample Listings for AC8111

10 B 30

7-5

OBJECT CODE STATEMENT NUMBER 31 0021 0039 DC STATEMENT NUMBER 0039 98C0 FFC6 LAB 003B 9870 0000 LDR 003D E870 0015 LDR 003F D380 0000 \$85,<ZCRTY1 LNJ STATEMENT NUMBER 33 0041 83C8 FFE5 JMP (PARTIAL OBJECT LISTING) STATEMENT NUMBER 34 0022 0043 DC STATEMENT NUMBER 0043 0F87 В 0044 474F DC 0045 2054 DC. 0046 4F20 DC DC. 0047 5041 0048 5241 DC. (SUBROUTINE CALL) -(INSTRUCTION MNEMONIC) -(INSTRUCTION) -(LOCATION OF INSTRUCTION) LINKFR-0100-11/23/1258 GC056 M0D400-S100-11/29/0620 LINKED ON: 1901/01/01 0002:44.1 -SAF BU= AC8111 AC8111 01/01/01 COBOL REV. 0200 DATE 01/01/01 TIME 0000 . ZCRTYU 770208 HRS ASSEMBLER 2.49 06/02/77 1340.3 EDT THU (C) COPYRIGHT 1976 BY HONFYWELL INFORMATION SYSTEMS INC ZCSTOP 770208 HRS ASSEMBLER 2.49 06/02/77 1336.9 EDT THU (C) COPYRIGHT 1976 BY HONFYWELL INFORMATION SYSTEMS INC ZCRTFR 770208 HRS ASSEMBLER 2,49 06/02/77 1934.4 EDT THU (C) COPYRIGHT 1976 BY HONEYWELL INFORMATION SYSTEMS INC LINK MAP 1901/01/01 0002:44.1 \*\* AC8111 \*\*START 0033 \*\*L0W 0000 \*\*HIGH 0381 \*\*CURRENT 0381 \*\*EXT DEFS ZHCOMM 0000 Ρ P ZHREL 0000 ROOT 0000 \*\* AC8111 0000 \* AC8111 0033 7CMAIN 0031 ZCRIYU 0289 \* ZCRTY3 0334 ZCRTY2 0312 7CRTY1 02E1

#### Figure 7-2 (cont). Sample Listings for AC8111

```
ZCSTOP
            033F
    ZCSTOP
            033E
    ZCRTFR
            0341
    ZCRTER
            0353
++UNDEE
    AC8111
            0000
    ZCRTYU
            0289
    ZCSTOP
            033E
    7CRTER
            0341
*******
ROOT AC8111
********
HIGHEST OVLY
                 INUM OF SYMS
                                 1
********
SAF
*****
ROOT ACA111
                        BASE 0000
                                         ST 0033
                                                           .I HIGH=0381
*SIZE OF ROOT AND STATIC OVLYS= 0381
                                             HI REL RCD=
                                                            q
LINK DONE
********
```

Figure 7-2 (cont). Sample Listings for AC8111

#### CALLING FORTRAN ROUTINES FROM AN ENTRY-LEVEL COBOL MAIN PROGRAM

Entry-Level COBOL programs can call FORTRAN subroutines and conversely. This enables a COBOL application to utilize the features of the FORTRAN language, such as the intrinsic routines, and FORTRAN run-time libraries.

The COBOL main program must be linked with all the called FORTRAN routines to form one bound unit. The FORTRAN routines and libraries must either be in the working directory or one of the libraries searched by the Linker, as specified by the Linker LIB and LIBn directives.

Figure 7-3 is a sample Entry-Level COBOL source program, COBFRT, whose function is to calculate and print the square roots of three integers. Since the COBOL library does not have a square root routine, a FORTAN subroutine, FRTRAN in Figure 7-4, is used to convert the passed COBOL integer argument values to read values and call the FORTRAN square root routine.

The commands entered from the operator terminal are listed in Figure 7-5. COBFRT.O and FRTRAN.O are both in the working directory FRTCOB, the COBOL run-time library, ZCRT, is in the directory specified by the Linker directive LIB, and the FORTRAN run-time library, ZFRT, is in the directory specified by LIB2. The system volume, ZSYS51, contains the FORTRAN and COBOL compilers, ZFRT, ZCRT and the operating system software. Volume FRTCOB contains the source modules (COBFRT.C and FRTRAN.F), the object modules (COBFRT.O and FRTRAN.O) and the linked bound unit COBFRT.

SOURCE PROGRAM TDENTIFICATION DIVISION. 1 2 PROGRAM-TD. CORFRT. \* THIS PROGRAM TS AN EXAMPLE OF A COBOL PROGRAM 3 \* CALLING A FORTRAN PROGRAM TO GET THE SQUARE ROUTS 4 \* OF SOME INTERGERS AND RETURNING THAT VALUE TO THE 5 \* COBOL PRUGRAM TO BE DISLPAYED. 6 7 ENVIRONMENT DIVISION. 8 CONFIGURATION SECTION. 9 SUURCE-COMPUTER. HTS-SFRIES-60 LEVEL-6. 10 OBJECT-COMPUTER. HTS-SERTES-60 LEVEL-6. 11 DATA DIVISION. 12 13 WUPKING-SIDRAGE SECTION. 77 WORK COMP-1 VALUE +0. 14 PIC 999 VALUE 625. 15 77 VAL625 PIC 99. 71 ANS25 16 PIC 999 VALUE 144. 17 77 VAL144 PIC 99. 77 ANS12 18 VAL9801 PIC 9999 VALUE 9801. 19 77 20 77 ANS99 PIC 99. 21 01 ANSLN. 25 02 FILLFR PIC XX VALUE SPACES. PIC 9999 VALUE ZFRO. 23 02 TNTVAL PIC X(6) VALUE SPACES. 24 02 FILLER PIC 9999 VALUE ZERO. 25 02 SQVAL 02 FILLER PIC XXX VALUE SPACES. 59 PROCEDURE DIVISION. 27 28 PARA1. 29 MOVE VAL625 TO WORK. CALL "FRTRAN" USING WORK. 30 31 MOVE WORK TO ANS25. MOVE VAL144 TO WORK. 32 CALL "FRTRAN" USING WORK. 33 MOVE WORK TO ANSIZ. 34 35 MOVE VAL9801 TO WORK. CALL "FRTRAN" USING WORK. 36 MOVE WORK TO ANS99. DISPLAY "INTEGER SO. RT.". 37 38 MOVE VAL625 TO INIVAL. 39 40 MOVE ANS25 TO SQUAL. DISPLAY ANSLN. 41 42 MOVE VAL144 TO INTVAL. MOVE ANSIZ TO SQUAL. 43 44 DISPLAY ANSLN. 45 MOVE VAL9801 TO INTVAL. 46 MOVE ANS99 TO SUVAL. 47 DISPLAY ANSLN. 48 STOP RUN. 49 END COROL NO DIAGNOSTICS

Figure 7-3. COBOL Listing of COBFRT

7-8

1 SUBROUTINE FRTRAN(1) 2 J = I3 X = FLOAT(J)4 Y = SQRT(X)5 I = NINT(Y)6 RETURN 7 END 0 DIAGNOSTICS



```
RDN
(AA) RDY:
CWD ^FPTC B
(AA)PDY:
COROL CORFET -COUT >SPD>I PTOO
(AA) COROL 0200 11/22/1511
(AA) 0000 ERPORS
(AA) FND COMPILATION
(AA) PDY:
FORTRAN FRTRAN -COUT >SPD>LPTOO
(AA) FORTRAN M4ED 11/22/1119
(AA) 0000 EPP COUNT FRTPAN
(AA)PDY:
LINKEP COBFRT -COUT >SPD>IPTOO
(AA)LINKER-0100-11/23/1258
LIB ^ZSYS51>ZCRT
LIB2 ^ZSYS51>ZFRT
LINK COBFRT
MAP OT
(AA)ROOT COBFRT
(AA)LINK DONE
(AA)RDY:
COBFRT
(AA) INTEGER
                 SO. RT.
(AA) 0625
(AA) 0144
(AA) 9801
                   0025
                   0012
                   0099
(AA)RDY:
```

Figure 7-5. Operator Terminal Session for COBFRT

7-9

 $\bigcirc$ 

•

### Section 8 Using the FORTRAN Compiler

This section illustrates the use of the FORTRAN compiler to construct programs written in the FORTRAN language, and to perform FORTRAN chaining.

#### SAMPLE FORTRAN TERMINAL SESSION (MATINV)

Figure 8-1 illustrates an operator terminal session in which the FORTRAN program MATINV is compiled, linked, and executed. The FORTRAN compiler is on the search path, and, therefore, a full pathname is not needed to locate the compiler. The Linker LIB directive directs the linker to search the secondary directory for FORTRAN runtime routines (ZFRT) required for linking. Two files, unit number 2 and 3, are associated with device pathnames. To execute the program, enter MATINV. Figure 8-2 shows the MATINV source listing and the linker output when the program was linked.

```
FORTRAN VL782 SUB1 MATINV -COUT > SPD>LPT00
($H) FORTRAN
($H) 0000 ERR COUNT MATINV
($H)BDY:
LINKER VOL2 TEST MATINV -COUT >SPD>LPTOO
($H)
LINKER
LIB ZF0400 ZFRT
LINK MATINV; MP;QT
($H)ROOT MATINV
($H)LINK DONE
($H)RDY:
ASSOC 2 >SPD>CDROO
($H)RDY:
ASSOC 3 >SPD>LPTOO
($H)RDY:
MATINV
($H) STOP
($H)RDY:
```

Figure 8-1. Sample Terminal Session (MATINV)

#### **FORTRAN CHAINING**

A method of creating and controlling execution of overlays within FORTRAN programs to conserve memory space, commonly known as CHAINing, can be used wherein an executable bound unit (overlay) is executed as a chain prior to invoking the next chain.

The source statement for referencing a chain is:

CALL CHAIN(e)

where e is an integer expression resulting in a value greater than or equal to zero, identifying the chain to be loaded. Proper linking results in a bound unit with overlays that require minimum memory for execution.

Although there are no rules for defining the best method of segmenting a FORTRAN application into chains, the following should be considered:

- 1. The largest chain determines the overall memory requirement.
- 2. Any chain may be called by any other chain as many times as required.

1 C		MATRIX INVERSION	
2 C			
3		DIMENSION V(50,50), B(50,50), IDAUL(50), IMDEX(50,50	)),PIVOT(20)
4		WPITE(3,7)	,
5	7	FORMAT(1H1,13Y, MATTNV', //, RX, 16HGIVEN MATRIX A	4,77
6		READ (2,1) N,M	
-7		FORMAT(12,12)	
8	2	FORMAT(7F10.4)	
9		D = T = 1, N	
10	4	READ $(2,2)$ $(A(I,J),J=1,M)$	
11	4 /1	D0 14 1=1,N	
12	14	wRITE(3,2)(A(T,T),J=1,M)	
13		00 11 I=1,M	
15		DO 11 J=1,N 1F(T-T)12,13,12	
16	1.5	B(I,J)=0.	
17	۰ <u>۲</u>	GO TO 11	
18	13	B([,J)=1.0	
1.9		CONTINUE	
.J 05	•	INITIALTZAITON	
21		DO 20 J=1,N	
>2	20	IPVOT(J) = 0	
23		DO 550 T=1,N	
24 C		SFARCH FOR PIVOT FLEMENT	
25		T=0.0	
	(stinted) law	DO 105 J=1,N	
27		IF(TPVOT(J)-1)60,105,60	
28	60	UN 100 K =1,N	
29		IF(TPV0T(K)-1)80,100,740	
30		IF(T**2-(4(J,K))**2)85,100,100	
31	<b>#5</b>	IROW #J	
32	ند خس	ICOL =K	
33		T = A(J,K)	
		CONTINUE	
35	105	CONTINUE	

Figure 8-2. Source and Linker Output Listing (MATINV)

÷.

.,

USING THE FORTRAN COMPILER

8-2

**CB22** 

.

ŧ

	39	1/10	DO 200 L=1,N		
	40		T=A(IROW,L)		
	41		A(IPOW,L) = A(ICOL,L)		
	42	200	A(ICOL,L) =T		
	43		IF(M)260,260,210		
	44	210	D0 250 L=1,M		
فنجعني	45	ZIV	T=B(IPOW,L)		
	-				
	46		B(IPOW,I) =B(TCOL,L)		
	47		B(IrOL,L) =T		
	48	260	INDFX(I,1) = IROW		
	89		INDEX (1,2) =TCOL		
i. Cano e	50		PTVOT(I) =A(ICOL,ICOL)	·	
	51	C	DIVIDE PIVOT ROW BY PIVOT ELEMENT		
	52		ACICOL, TOPL) =1.0		
	53		DD 350 L=1,N		
14 N.	54	350	A(TCOL,L) =A(ICOL,L)/PTVOT(I)		
	55		IF(M)380,380,360		
	56	360	DO 370 L=1,M	· · ·	
з.	57	370	B(ICOL,L) = B(TCOL,L)/PIVOT(T)		
	58	-	REDUCE NON-PIVOT ROWS		
•	59	A CONTRACTOR OF THE OWNER OWNER OF THE OWNER OWNE	D0 550LT=1.N		
	60	0.0	IF(1,I-ICUL) 400,550,400		
	61	400	T = A(LT, TCOL)		
	62	400	A(LT,TCOL) = 0.0	······	
	63		DO 450 I=1,N		
Secto	64	470	$\frac{A(LT,L)=A(LT,L)-A(ICOL,L)*T}{IC(L)=A(LCOL,L)*T}$		
	65		IF(M)550,550,460		
	66		DO 500 L=1,M		
	67		B(LT,I) = B(LT,I) - B(ICOL,L) + T		
5	68		CONTINUE		
50	69	-	INTERCHANGE COLUMNS	· · · ·	
201	70	600	DO 710I=1,N	. *	
	71		L= M-T+1		
	72		IF(TNDEX(1,1)+INDFX(1,2))630,710,63	50	
	73	670	IROW = TNDEX(L, 1)		· · · · · · · · · · · · · · · · · · ·
	74		ICOL = TNDEX(L,2)		
3	75		00 705 K=1,N		
	76		T. =A(K,TROW)		
نصعب	77		A(K, IROW) = A(K, TCOL)		annan ann an Anna Anna Anna Anna Anna A
	78		$A(K,  \Gamma 0 ) = T$		
	79	705	CONTINUE		

Figure 8-2 (cont). Source and Linker Output Listing (MATINV)

USING THE FORTRAN COMPILER

န မ

CB22

.

.

.

### Figure 8-2 (cont). Source and Linker Output Listing (MATINV)

, ,

NULAI WPITE(3,16)	
R2 16 FORMAT(//5X,22HINVERSE OF MATRIX A/)	
$\begin{array}{cccc} R3 & D & 15 & I = 1 \\ R & & D & I = 0 \\ R & & D & I \\ R & & D & I \\ R & & D & I \\ R & & I $	
84 15 APITE(3,2)(A(T,T),J=1,M) 85 APITE(3,444)	
R6 444 FORMAT(1H1)	
6. 87 740 STOP 1	
88 END	
0 DTACNOSTICS	
MATINV 77120200	
FORTRAN M4FD 11/22/1119 1977/12/02 1312:36.9 SAF	
7FYFTU 77111000	
HRS ASSEMBLEP 2.50 11/10/77 1035.9 EST THU	
(C) COPYPICHT 1977 BY HUMEYWELL INFORMATION SYSTEMS INC	
ZESETD 77111000	
HRS ASSEMBLER 2.50 11/10/77 1030.1 EST THU	
(C) CUPYRIGHT 1977 BY HONEYWELL INFORMATION SYSTEMS INC.	
ZFSK01	
7FQFT0 77050100	
HRS ASSEMBLER 2.50 10/03/77 0708.3 EDT MON	
HRS ASSEMBLER 2.50 10/03/77 0708.3 EDT MON (C) COPYRIGHT 1977 BY HONEYWELL INFORMATION SYSTEMS INC	
HRS ASSEMBLER 2.50 10/03/77 0708.3 EDT MON (C) COPYRIGHT 1977 BY HONEYWELL INFORMATION SYSTEMS INC 7FPFTU 77111000	
HRS ASSEMBLEP 2.50       10/03/77       0708.3 EDT MON         (C) COPYRIGHT 1977 BY HONEYWELL INFORMATION SYSTEMS INC         7FPFTU 77111000         HRS ASSEMBLEP 2.50       11/10/77	
HRS ASSEMBLER 2.50 10/03/77 0708.3 EDT MON (C) COPYRIGHT 1977 BY HONEYWELL INFORMATION SYSTEMS INC 7FPFTU 77111000	
HRS ASSEMBLER 2.50 10/03/77 0708.3 EDT MON (C) COPYRIGHT 1977 BY HONEYWELL TNFORMATION SYSTEMS INC 7FPFTU 77111000 HRS ASSEMBLER 2.50 11/10/77 1054.0 EST THU (C) COPYRIGHT 1977 BY HONEYWELL INFORMATION SYSTEMS INC	
HRS ASSEMBLEP 2.50       10/03/77       0708.3 EDT MON         (C) COPYRIGHT 1977 BY HONEYWELL INFORMATION SYSTEMS INC         7FPFTU 77111000         HRS ASSEMBLEP 2.50       11/10/77	
HRS ASSEMBLER 2.50 10/03/77 0708.3 EDT MON         (C) COPYRIGHT 1977 BY HONEYWELL INFORMATION SYSTEMS INC         7FPFTU 77111000         HRS ASSEMBLER 2.50 11/10/77 1054.0 EST THU         (C) COPYRIGHT 1977 BY HONEYWELL INFORMATION SYSTEMS INC         ZEEEIQ 77112100	
HRS ASSEMBLER 2.50       10/03/77       0708.3 EDT MON         (C) COPYRIGHT 1977 BY HONEYWELL INFORMATION SYSTEMS INC         7FPFTU 77111000         HRS ASSEMBLER 2.50       11/10/77       1054.0 EST IMU         (C) COPYRIGHT 1977 BY HONEYWELL INFORMATION SYSTEMS INC         ZEEFIO 77112100         HRS ASSEMBLER 2.50       11/21/77       1539.8 EST MON	
HRS ASSEMBLEP 2.50       10/03/77       0708.3 EDT MON         (C) COPYRIGHT 1977 BY HONEYWELL INFORMATION SYSTEMS INC         7FPFTU       77111000         HRS ASSEMBLEP 2.50       11/10/77         10/03/77       1054.0 EST IMU         (C) COPYRIGHT 1977 BY HONEYWELL INFORMATION SYSTEMS INC         ZEEFIQ       77112100         HRS ASSEMBLEP 2.50       11/21/77         1539.8 EST MON         (C) COPYRIGHT 1977 BY HONEYWELL INFORMATION SYSTEMS INC         ZEEFIQ       75112100         HRS ASSEMBLEP 2.50       11/21/77         1539.8 EST MON         (C) COPYRIGHT 1977 BY HONEYWELL INFORMATION SYSTEMS INC         ZEEK01	
HRS ASSEMBLEP 2.50 10/03/77 0708.3 EDT MON         (C) COPYRIGHT 1977 BY HONEYWELL INFORMATION SYSTEMS INC         7FPFTU 77111000         HRS ASSEMBLEP 2.50 11/10/77 1054.0 EST THU         (C) COPYRIGHT 1977 BY HONEYWELL INFORMATION SYSTEMS INC         ZEEFIQ 77112100         HRS ASSEMBLEP 2.50 11/21/77 1539.8 EST MON         (C) COPYRIGHT 1977 BY HONEYWELL INFORMATION SYSTEMS INC         ZEEFIQ 77112100         HRS ASSEMBLEP 2.50 11/21/77 1539.8 EST MON         (C) COPYRIGHT 1977 BY HONEYWELL INFORMATION SYSTEMS INC         ZEEKO1         ZEFKO1	
HRS ASSEMBLEP 2.50 10/03/77 0708.3 EDT MON         (C) COPYRIGHT 1977 BY HONEYWELL INFORMATION SYSTEMS INC         7FPFTU 77111000         HRS ASSEMBLEP 2.50 11/10/77 1054.0 EST IMU         (C) COPYRIGHT 1977 BY HONEYWELL INFORMATION SYSTEMS INC         ZEEFIQ 77112100         HRS ASSEMBLEP 2.50 11/21/77 1539.8 EST MON         (C) COPYRIGHT 1977 BY HONEYWELL INFORMATION SYSTEMS INC         ZEEFIQ 77112100         HRS ASSEMBLEP 2.50 11/21/77 1539.8 EST MON         (C) COPYRIGHT 1977 BY HONEYWELL INFORMATION SYSTEMS INC         ZFEK 01         ZFEOTE 77072900         HRS ASSEMBLER 2.50 10/13/77 0937.8 EDT THU	
HRS ASSEMBLEP 2.50 10/03/77 0708.3 EDT MON         (C) COPYRIGHT 1977 BY HONEYWELL INFORMATION SYSTEMS INC         7FPFTU 77111000         HRS ASSEMBLEP 2.50 11/10/77 1054.0 EST THU         (C) COPYRIGHT 1977 BY HONEYWELL INFORMATION SYSTEMS INC         ZEEFIQ 77112100         HRS ASSEMBLEP 2.50 11/21/77 1539.8 EST MON         (C) COPYRIGHT 1977 BY HONEYWELL INFORMATION SYSTEMS INC         ZEEFIQ 77112100         HRS ASSEMBLEP 2.50 11/21/77 1539.8 EST MON         (C) COPYRIGHT 1977 BY HONEYWELL INFORMATION SYSTEMS INC         ZEEKO1         ZEFKO1	•

.

.

1 0	1814 1000 1065 1065 1000	7FSRFS	nED5	7F 9W11S	0F11	7F98115		
DEFS           HCOMM           HREL           ONI           IATINV           ZFMRK           FMATN           FYFIO           FSFIO           FSFIO           FSFFIO           FSFFIO           FSMFS           FOFTO	0000 0000 0000 0000 0000 0000 0014 0ERD 0ER5 0ER5 0ER5	7FSRFS	nED5	7F 9W11S	0511	71 8 0110		
DEFS HCOMM HREL ATINV ZFMPK FMATN FYFIO FYXRI FSFIO FSWFS FOFIO FOMRK	0000 0000 0000 0000 0000 0000 0014 0ERD 0ER5 0ER5 0ER5	7FSRFS	nED5	7F 9W11S	0F11	71 8 0 110	2 	
HCOMM HREL DOT ATINV ZFMATK FYFIO FYFIO FSFIO FSFIO FSFIO FSFIO FSFIO	0000 0000 0000 0010 0000 0014 0000 0000	7FSRFS	nED5	7F 9W11S	0511	71 8 0140		
HCOMM HREL DOT ATINV ZFMATK FYFIO FYFIO FSFIO FSFIO FSFIO FSFIO FSFIO	0000 0000 0000 0010 0000 0014 0000 0000	7FSRFS	nED5	7F 9W11S	0611	71 8 0 110		
HREL DOT ZEMPK FMATN FYFIO FYXRI FSFTO FSWFS FSFTO FSWRS	0000 0000 0000 0010 0000 0014 0000 0000	7FSRFS	nED5	7F 9W11S	0611	71 8 0 110		
ETTNV FMATN FYFIO FYRI FSFTO FSWFS FRFTO FRFTO	0000 0010 0814 0880 0885 0885 0885 0885	7FSRFS	nED5	ZESWUS	0F11	71 8 0 110		
ETTNV FMATN FYFIO FYRI FSFTO FSWFS FRFTO FRFTO	0000 0010 0814 0880 0885 0885 0885 0885	7FSRFS	nED5	7F 9 11 13	0F11	71 8000		
ZFNPK FMATN FYFIO FYXRI FSFIO FSWFS FOFIO FONRK	0000 0814 0EAD 0EAD 0EAS 0EAS 0EAS 0F55	7FSRFS	NEN5	7FSWUS	0F11	76 6040		
FMATN FYFIO FYXRI FSFTO FSWFS FDFTO FDARK	0814 0ERD 0ERD 0ER5 0ER5 0F55	7FSRFS	NENS	ZESWUS	0F11	71 6040		
FYXRI FRFIO FRWFS FRFIO FRARK	0ERD 0ERD 0ER5 0ER5 0F55	7FSRFS	NENS	7FSWUS	0F11	71 60110		
FYXRI FRFIO FRWFS FRFIO FRARK	0E8D 0E85 0E85 0E85	7FSRFS	NED5	7FSWUS	0F11	71.6000		
FRWFS Frftu Frakk	1685 1655	7FSRFS	NEDS	7FSWUS	0F11	71.0000		<u> </u>
FOFTO	^F55	7FSRFS	NED5	7FSWIIS	0F11	71 80110	0574	
FONRK						/rakua	0F34	
	0655							
FPFTO								
	0F60					: 		
FPAUS	0F65	7FPSTP	0F60		1		7	() (See G / 2017)
FEFTO	OFAS							
FAN	OFAS	7FFFT0	OFAS	7FFMFI	1288	7FFMF0	128B	
FEMAL	1288	7FFM4U	1288	7FFLFI	1288	7FFLF0	128B	
FFLAI	1285	7FFLA0	1285	ZEFCLI	120E	7FFCLO	120E	
FFCSI	12F1	7FFC90	12F1	7FFCFI	12F4	7FFCF0	12F4	
FFCAI	1209	7FECA0	1209	ZFEWFF	1402	ZFFRFF	1402	
FFIFI	1360	7FEIFO	1360	7FFIAI	1367	7FEIAO	1367	
FFJFI			1360	ZFEJAI	136A	ZFEJAO	136A	
FFKFI			-		-	7FFKA0	1364	
FFRAL		7FFRAD	14FF	7FERFI	1505	7FFRF0	1505	
FTOTE								
		7FFINI	1972					
	FFLAI FFCSI FFCAI FFIFI FFJFI FFXFI FFXFI FFXAI	FFLAI     1285       FFCSI     12F1       FFCAI     1209       FFIFI     1360       FFJFI     1360       FFJFI     1361       FFAFI     135F       FFDFI     1502       FFTI     14FF       FTOTE     1903       FTOTE     1903	FFLAI       1285       7FFLAO         FFCSI       12F1       7FFCOO         FFCAI       12D9       7FFCOO         FFCAI       12D9       7FFCOO         FFJFI       1360       7FFJFO         FFJFI       1360       7FFJFO         FFXFI       135F       7FFKFO         FFXFI       1502       7FFDFO         FFAI       14FF       7FFKAO         FTOTE       19C3       7FFINI	FFLAI       12R5       7FFLAO       12R5         FFCSI       12F1       7FFCAO       12F1         FFCAI       12N9       7FFCAO       12N9         FFIFI       136D       7FFIFO       136D         FFJFI       136D       7FFJFO       136D         FFJFI       136D       7FFJFO       136D         FFJFI       136D       7FFJFO       136D         FFAI       136F       7FFKFO       135F         FFDFI       1502       7FFDFO       1502         FFAI       14FF       7FFKAO       14FF         FTOTE       19C3       7FFINI       19P2	FFLAI       12R5       7FFLAO       12R5       ZFFCLI         FFCSI       12F1       7FFCSO       12F1       7FFCFI         FFCAI       12D9       7FFCAO       12D9       7FEWFF         FFIFI       136D       7FFIFO       136D       7FFIAI         FFJFI       136D       7FFJFO       136D       7FFIAI         FFJFI       136D       7FFJFO       136D       7FFIAI         FFXFI       135F       7FFKFO       136F       7FFKAI         FFOFI       1502       7FFDFO       1502       7FFDAI         FFAI       14FF       7FFKAO       14FF       7FFRFI         FTOTE       19C3       7FFINI       19P2	FFLAI       12A5       7FFLAO       12A5       7FFCLI       12DE         FFCSI       12F1       7FFCSO       12F1       7FFCFI       12F4         FFCAI       12D9       7FFCAO       12D9       7FFCFI       14C2         FFIFI       1360       7FFIFO       1360       7FFIAI       1367         FFJFI       1360       7FFJFO       1360       7FFIAI       1367         FFXFI       1354       7FFKFO       1357       7FFKAI       1364         FFDFI       1502       7FFDFO       1502       7FFDAI       14FC         FFATI       14FF       7FFRAO       14FF       7FFRFI       1505         FTOTE       19C3       7FFINI       19D2       14DP2	FFLAI       12A5       7FFLAO       12A5       ZFFCLI       12DE       7FFCLO         FFCSI       12F1       7FFCSO       12F1       7FFCFI       12F4       7FFCFO         FFCAI       12D9       7FFCAO       12D9       7FFEFFI       14C2       7FFRFF         FFIFI       1360       7FFIFO       136D       7FFIAI       1367       7FEIAO         FFJFI       1360       7FFIFO       136D       7FFIAI       1367       7FEIAO         FFJFI       1360       7FFIFO       136D       7FFIAI       1367       7FEIAO         FFJFI       1360       7FFIFO       136D       7FFIAI       1367       7FEIAO         FFVFI       135F       7FFFFO       136D       7FFIAI       1364       7FFKAO         FFVFI       1502       7FFDFO       1502       7FFDAI       14FC       7FFDAO         FFV0TE       19C3       7FFRAO       14FF       7FFRAO       14FF       7FFRFO         FTOTE       19C3       7FFINI       19P2       14PP       14PP       14PP	FFLAI       12A5       7FFLAO       12A5       7FFCLI       12DE       7FFCLO       12DE         FFCSI       12F1       7FFCSO       12F1       7FFCFI       12F4       7FFCFO       12F4         FFCAI       12D9       7FFCAO       12D9       7FEWFF       14C2       7FFRFF       14C2         FFIFI       1360       7FFIFO       136D       7FFIAI       1367       7FEIAO       1367         FFJFI       1360       7FFJFO       136D       7FFIAI       1367       7FEIAO       1367         FFJFI       1360       7FFJFO       1360       7FFIAI       1364       7FFIAO       1364         FFKFI       135F       7FFKFO       135F       7FFKAI       1364       7FFKAO       1364         FFVFI       1502       7FFDFO       1502       7FFDAI       14FC       7FFDAO       14FC         FFAAI       14FF       7FFKAO       14FF       7FFRFO       1505       7FFRFO       1505         FTOTE       19C3       7FFINI       19P2       19C3       7FFINI       19P2

Figure 8-2 (cont). Source and Linker Output Listing (MATINV)

.

t

.

3

	ZFLB1 7FAMPT	1 ADD 1 ARA	7FRF1 7FRRFD	1 A º D 1 A 4 D	7FUWUF 7FSUP1	1A08 1A58	7FURUF 7FSUB4	1 A 2 A 1 C 0 1	
	7FSUB6	1890							
**U	NDEF								
*	MATINV	0000							
*	7FYFTU	NERD							
*	7FSFTU	AEP5							
*	ZENETU	1F55							
*	ZEPETO	0F60							
*	7FFFT0	OFAS							
*	ZETOTE	1903							
*	7FIIFTO	1400							
	****								
RUD		· · · · · · · · · · · · · · · · · · ·					······ , ····· ,		
	****								
* * *	HEST OVICE	/ //	M DE SYM	S 0					
HIG				×					👝 🖡 san sa
and the second se									
***	******								
*** SA	***** F								
*** SA	****** F *****		BASE	<u>nuno</u>	STO	814	<b>••••</b> I	HIGH=1	C65
*** SA *** RUD	****** F *****	1	BASE	n0n0	ST O	814	<b></b> I	HIGH=1	C 6 5
*** SA *** RUD ***	****** F ****** ] MATINV ******		BASE STATTC OV	-				HIGH=1	C6 <b>5</b>
*** SA *** RUD *** *ST	****** F ****** ] MATINV ******			-		B14 HT REI		HIGH=1	C <b>65</b>
*** SA *** RUD *** *ST ***	****** F ****** I MATINV ****** ZF OF ROC			-				HIGH=1	C <b>65</b>

Figure 8-2 (cont.). Source and Linker Output Listing (MATINV)

.

۲

.

¢

- 3. The first statement executed in a loaded chain is always the first executable statement of the first main program in the chain. (A chain cannot begin with a subroutine.)
- 4. All data passed between chains must be in unlabeled or labeled COMMON blocks that have been defined within the root. Because of Linker constraints, the first occurrence of a COMMON block defines its size, therefore care must be exercised when using COMMON blocks of different sizes.
- 5. Within programs in a chain, either labeled or unlabeled COMMON may be freely used as a means of data communication.
- 6. Data statements (for data not in COMMON) within a program of a chain cause the data to be initialized each time the chain is loaded.
- 7. Files are common to all chains since the run-time work area is defined within the root.

Figure 8-3 shows an assembly-language program, CHAIN, whose function is to load the chains specified in the CALL statements of the FORTRAN programs shown in Figure 8-4. These latter programs call each other at various times and print messages indicating their loading and execution.

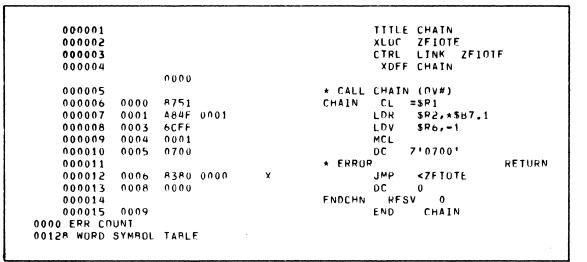
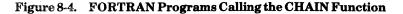


Figure 8-3. Assembly Listing of Program CHAIN

```
PROGRAM PROGOO
1
             COMMON IC
2
3
         COMMON Y, TELR
         DIMENSION ARRY(62)
4
5
          COMMON /LAB1/DUMMY(50)
         COMMON /LAB2/ DUNX(527)
6
7
             1C = 0
          WRITE (3,5)
8
Q
       5 FORMAT('1'/' PROGO APPEARS ON EXECUTE LINE - CALLS CHAIN 0'/)
10
             CALL CHAIN(0)
          END
11
     DIAGNOSTICS
 0
          PROGRAM PROG01
1
2
             COMMON IC,X
3
         COMMON /LAC1/ DATA1(25)
4
         COMMON /LAC2/ DATA2(378)
5
          CHARACTER A*20
         IC=IC+1
6
7
          PEAD(2,215) A
```



音

```
215 FORMAT(420)
8
          WRITE(3,215) A
9
          CALL PROGD
10
          WRITE (3,5)
11
    5
         FORMAT(/' PROG1 IS CHAIN O WHICH CALLS CHAIN 1')
12
13
          CALL CHAIN (1)
14
         END
     DIAGNOSTICS
 0
          SUBROUTINE PROGD
1
2
          CUMMON /LAC1/ DATA1(25)
          CUMMON /LAC2/ DATA2(378)
3
         wRITE(3,305)
4
    305 FORMAT(5x, ' SUBROUTINE PROGD LOADED'/)
5
          RETURN
6
7
          FND
 0
    DIAGNOSTICS
          PROGRAM PROG02
 1
 2
          COMMON /LAB1/DUMMY(50)
          DIMENSTON ARRY(62), ARRY1(157)
 3
 4
          CHARACTER*8 A1, A3, A4
          COMMON IC, Z, TQ
 5
          TF (TC.GT.1) GO TO 10
 6
 7
          TQ=0
          A1=' CHAIN O'
 8
         WRITE(3,5) A1
FORMAT(/' PROG2 IS CHAIN 1 - WHICH CALLS -',A8)
 9
10
    5
11
          CALL CHAIN(0)
12
    10
         A3=' CHAIN 2'
          IF (IQ.EQ.4) GO TO 20
13
14
          WRITE(3,5) A3
15
          CALL CHAIN (2)
16
    20
         A4=' CHAIN 3'
          WRTIF (3,5) A4
17
18
          STOP
19
          END
    DIAGNOSTICS
 0
          PROGRAM PROGO3
 1
          CHARACTER A+20
 2
          WRITE(3,5)
 3
          READ(2,215) A
 4
   215
          FORMAT(A20)
 5
          WRITE(3,215) A
 6
          CALL CHAIN(3)
 7
         FORMAT(/' PROG3 IS CHAIN 2 - WHICH CALLS CHAIN 3'/)
   5
 8
          FND
 9
  0
    DIAGNOSTICS
          PROGRAM PROG04
 1
          CUMMON IC, F, T
 2
          CUMMON /LAR1/ DUMMY(50)
 3
          T=T+1
 4
 5
          K=4
           TF (1.EW.4) K=2
 6
           IF (T.FW.5) GOTO 99
 7
           WRIIF(3,5) K-1
 8
           FURMAT(/' PROG4 IS CHAIN 3 - CALLS CHAIN', T2/)
    5
 9
10
           CALL CHAIN (K-1)
         STUP
    99
11
12
           END
     DIAGNOSTICS
  ٥
```



Figure 8-5 is the output listing resulting from the linking of the programs constituting the chain.

```
LINKER-0100-11/23/1258
                                         GC0S6 MOD400-S100-11/29/0620
BU= TSTCH1
                 LINKED ON: 1977/12/02 1354:06.5 -SAF
PR0G00 77120200
                              1977/12/02 1352:46.3 SAF
  FORTRAN M4ED 11/22/1119
CHAIN
1977/12/02/ 1326:22.5 ASSEMBLER-0100-11/17/1346 GCOS6 MOD400-S100-11/29/0620
ZESEI0 77111000
HRS ASSEMBLER 2.50 11/10/77 1030.1 EST THU
(C) COPYRIGHT 1977 BY HONEYWELL INFORMATION SYSTEMS INC
ZFSK01
ZFQFI0 77050100
HRS ASSEMBLER 2.50 10/03/77 0708.3 EDT MON
(C) COPYRIGHT 1977 BY HONEYWELL INFORMATION SYSTEMS INC
ZFPFI0 77111000
HRS ASSEMBLER 2.50 11/10/77 1054.0 EST THU
(C) COPYRIGHT 1977 BY HUNEYWELL INFORMATION SYSTEMS INC
ZFEFI0 77112100
HRS ASSEMBLER 2.50 11/21/77 1539.8 EST MON
(C) COPYRIGHT 1977 BY HONEYWELL INFORMATION SYSTEMS INC
ZFEK01
ZFIOTE 77072900
HRS ASSEMBLER 2.50 10/13/77 0937.8 EDT THU
(C) COPYRIGHT 1977 BY HONEYWELL INFORMATION SYSTEMS INC
ZEUEIO 77111000
HRS ASSEMBLER 2.50 11/10/77 1623.8 EST THU
(C) COPYRIGHT 1977 BY HONEYWELL INFORMATION SYSTEMS INC
                  LINK MAP 1977/12/02 1354:06.5
** TSTCH1
**START
          0668
**LOW
          0000
**HIGH
          1450
**SCOMM
         0164
**CURRENT 145C
**EXT DEFS
Ρ
   7 HC OMM
           0000
D
   ZHREL
            0000
**
   ROOT
            0000
   PR0G00 0000
*
С
   SZFWRK
           0000
   $COMM
С
            0164
С
   LAB1
            016A
С
   LAB2
            01CE
   PROGOO
           0668
   CHAIN
            06A3
   CHAIN
            06A3
   ZFSFIO
           06AC
                   ZESRES 06CC
                                    ZESWUS 0708
                                                    ZFSRUS 0728
   ZFSWFS
           06AC
   ZFOFIO
           074C
٠
   ZFOWRK
           074C
           0757
   ZFPFIO
    ZFPAUS
                    ZFPSTP
           075C
                            0757
           0790
   ZFEFTO
    ZFAN
            0790
                    ZFEFIO
                            079C
                                    ZFEMEI
                                           58A0
                                                    ZFEMED 0A82
    ZFEMAI
           0A7F
                    7FEMA0
                            0A7F
                                    ZFELEI
                                            S8A0
                                                    ZFELEO
                                                            0482
    ZFELAI
            0A7C
                    ZFELAO
                            0A7C
                                    ZFECLI
                                            0AD5
                                                    ZFECLO
                                                            0AD5
    ZFECSI
                    ZFECSO
                                                    ZFECEO
                                                            OADB
           0AD8
                            0AD8
                                    7FECE1
                                            OADB
```

Figure 8-5. Linker Output for Chained Programs

	1								
1	ZFFCAI	OADO	ZFECAO	OADO	ZFEWFF	OCB9	ZFERFF	0CB9	
	ZFEIEI	0864	7FE1E0	0864	ZFEIAI	OBSE	ZFEIAO	085E	
	7FFJEI	0B64	ZFEJEO	0864	ZFEJAI	0861	ZFEJAO	0861	
1	ZFEKEI	0B56	<b>7FFKEO</b>	0856	ZFEKAI	085B	ZFEKAO	085B	
	ZFEDEI	0CF9		0CF.9			ZFEDAO	0CF3	
			ZFEDEO		ZFEDAI	OCF3			
	ZFERAI	0CF6	ZFERAO	0CF6	ZFEREI	OCFC	ZFEREO	OCFC	
*	ZFIOTE	118A							
	7FIOTE	11BA	ZFFINI	1109					
*	ZEVETO		21 C 2 C 2 C 2	1107					
× 1									
	ZFLB1	1204	ZFGF1	1294	ZEUWUE	1202	ZFURUF	1551	
1	ZFAWRT	1231	ZEBRED	1244	ZFSUB1	124F	ZFSUB4	13F8	
	ZESUB6	1387							
	INDEE								
	JNDEF								
*	PROGOO	0000							
*	CHAIN	0643							
*	7F 5F 10	06AC							
*	ZEGEIO	0740							
*	ZEPETO	0757							
*	ZFEFIO	0790							
*	7FIOTE	118A							
*	ZFUFTU	1202							
,,,	210110	12.02							
l									
<u>PRC</u>	DG01 77	120200							
F	ORTRAN M	44ED 1	1/22/1119	1977	/12/02 13	52:46.7	S SAF		
l '				- / · ·					
0.00	100 <b>7</b> 7	120200							
PRO		150500							
F F	ORTRAN M	14ED 1	1/22/1119	1977	/12/02 13	52:46.3	S SAF		
**	TSTCH1		LINK MAP	1077/1	2/02 1354	• 0 6 5			
			CIMK MAP	17///1	102 1004	•00•5			
		1782							
**L	-OM .	1450							
**	IGH :	180F							
		1164							
	CUPRENT								
**0									
**C **E	EURRENT	180F							
**C **E P	EXT DEFS 7HCOMM	0000							
**C **E	EURRENT	180F							
**C **E P P	CUPRENT EXT DEFS 7HCOMM 7HREL	0000 0000							
**C **E P	EXT DEFS 7HCOMM	0000							
**C **E P P	CUPRENT EXT DEFS 7HCOMM 7HREL	0000 0000							
**C **E P **	CUPRENT EXT DEES 7HCOMM 7HPEL ROOT PROGOO	0000 0000 0000 0000 0000							
**C **E P ** * C	CUPRENT EXT DEES 7HCOMM 7HPEL ROOT PROGOO \$ZEWRK	0000 0000 0000 0000 0000 0000							
**C **E P ** C C	CURRENT SXT DEES ZHCOMM ZHREL ROOT PROGOO \$ZEWRK \$COMM	0000 0000 0000 0000 0000 0164							
**C P P ** * C C C	CURRENT EXT DEES ZHCOMM ZHREL ROOT PROGOO \$ZEWRK \$COMM LAB1	0000 0000 0000 0000 0000 0164 016A							
**C **E P ** C C	CURRENT SXT DEES ZHCOMM ZHREL ROOT PROGOO \$ZEWRK \$COMM	0000 0000 0000 0000 0000 0164							
**C P P ** * C C C	CURRENT THCOMM THREL ROOT PROGOO \$ZFWRK \$COMM LAB1 LAB2	0000 0000 0000 0000 0164 016A 01CE							
**C P P ** C C C C C	CURRENT THCOMM THREL ROOT PROGOO \$ZFWRK \$COMM LAB1 LAB2 PROGOO	0000 0000 0000 0000 0164 0164 0164 0168							
**C P P ** * C C C	CURRENT THCOMM THREL ROOT PROGOO \$ZFWRK \$COMM LAB1 LAB2 PROGOO CHAIN	0000 0000 0000 0000 0000 0164 0164 0164							
**C **E P ** C C C C	CURRENT EXT DEES 7HCOMM 7HPEL ROOT PROGOO \$ZFWRK \$COMM LAB1 LAB2 PROGOO CHAIN CHAIN	0000 0000 0000 0000 0164 0164 0164 0168 0668 0643 0643							
**C P P ** C C C C C	CURRENT EXT DEES 7HCOMM 7HREL ROOT PROGOO \$ZFWRK \$COMM LAB1 LAB2 PROGOO CHAIN CHAIN ZFSFIO	0000 0000 0000 0000 0164 0166 0168 0668 06A3 06A3 06A3							
**C **E P ** C C C C	CURRENT EXT DEES 7HCOMM 7HPEL ROOT PROGOO \$ZFWRK \$COMM LAB1 LAB2 PROGOO CHAIN CHAIN	0000 0000 0000 0000 0164 0164 0164 0168 0668 0643 0643	7F SRF S	<b>06CC</b>	<b>ZFSWUS</b>	0708	ZFSRUS	0728	
**C **E P ** C C C C	CURRENT EXT DEES 7HCOMM 7HREL ROOT PROGOO \$ZFWRK \$COMM LAB1 LAB2 PROGOO CHAIN CHAIN ZFSFIO	0000 0000 0000 0000 0164 0164 0164 0168 0643 0643 0643 0643	ZESRES	06CC	ZFSWUS	0708	ZFSRUS	072B	
* * C P P * * C C C C C C *	CURRENT THEOMM THREL ROOT PROGOO \$ZFWRK \$COMM LAB1 LAB2 PROGOO CHAIN CHAIN ZFSFIO ZFSWFS ZF@FIO	0000 0000 0000 0000 0164 0164 0164 0164	7FSRFS	06CC	7FSw11S	0708	ZFSRUS	072B	
**C P P ** C C C C * *	CURRENT EXT DEES 7HCOMM 7HREL ROOT PROGOO \$ZFWRK \$COMM LAB1 LAB2 PROGOO CHAIN CHAIN ZFSFIO ZFSWFS ZFOFIO ZFOWRK	0000 0000 0000 0000 0164 0164 0164 0164	ZESRES	06CC	ZFSWUS	0708	ZFSRUS	0728	
* * C P P * * C C C C C C *	CURRENT EXT DEES 7HCOMM 7HREL ROOT PROGOO 8ZEWRK \$COMM LAB1 LAB2 PROGOO CHAIN CHAIN ZESEIO ZESWES ZEGWRK ZEPEIO	0000 0000 0000 0164 0164 0164 0164 0166 0668 0643 0643 0643 0643 0643 0643 0643 0643			ZESMUS	0708	ZFSRUS	072B	
**C P P ** C C C C * *	CURRENT EXT DEES 7HCOMM 7HREL ROOT PROGOO \$ZFWRK \$COMM LAB1 LAB2 PROGOO CHAIN CHAIN ZFSFIO ZFSWFS ZFGWRK ZFPFIO ZFPAUS	0000 0000 0000 0000 0164 0164 0164 0164	7FSRFS ZFPSTP	06CC 0757	7FSWIIS	0708	ZFSRUS	0728	
**C P P ** C C C C * *	CURRENT EXT DEES 7HCOMM 7HREL ROOT PROGOO 8ZEWRK \$COMM LAB1 LAB2 PROGOO CHAIN CHAIN ZESEIO ZESWES ZEGWRK ZEPEIO	0000 0000 0000 0164 0164 0164 0164 0166 0668 0643 0643 0643 0643 0643 0643 0643 0643			ZFSWIIS	0708	ZFSRUS	0728	
**C **E P ** C C C C * * *	CURRENT EXT DEFS 7HCOMM 7HREL ROOT PROGOO \$ZFWRK \$COMM LAB1 LAB2 PROGOO CHAIN CHAIN ZFSFIO ZFSWFS <b>ZFOFIO</b> ZFOWRK ZFPFIO ZFPAUS ZFEFIO	0000 0000 0000 0000 0164 0164 0164 0164	ZFPSTP	0757	-				
**C **E P ** C C C C * * *	CURRENT EXT DEFS 7HCOMM 7HREL ROOT PROGOO \$ZFWRK \$COMM LAB1 LAB2 PROGOO CHAIN CHAIN ZFSFIO ZFSWFS <b>ZFOFIO</b> <b>ZFOWRK</b> <b>ZFPFIO</b> <b>ZFPAUS</b> <b>ZFEFIO</b> <b>ZFAN</b>	0000 0000 0000 0000 0164 0164 0164 0164	ZFPSTP ZFEFIO	0757 079C	ZFEMEI	5840	ZFEMEO	2840	
**C **E P ** C C C C * * *	CURRENT EXT DEFS 7HCOMM 7HREL ROOT PROGOO \$ZFWRK \$COMM LAB1 LAB2 PROGOO CHAIN CHAIN ZFSFIO ZFSWFS ZFOFIO ZFPAUS ZFEFIO ZFEMAI	0000 0000 0000 0000 0164 0164 0164 0164	ZFPSTP ZFEFIO ZFEMAO	0757 079C 0A7F	ZFEMEI ZFELEI	0882 0882	ZFEMEO ZFELEO	28A0 28A0	
**C **E P ** C C C C * * *	CURRENT EXT DEES ZHCOMM ZHREL ROOT PROGOO \$ZFWRK \$COMM LAB1 LAB2 PROGOO CHAIN CHAIN ZFSFIO ZFSWFS <b>ZFOFIO</b> <b>ZFOWRK</b> <b>ZFPAUS</b> <b>ZFEFIO</b> <b>ZFEN</b> <b>ZFEN</b>	0000 0000 0000 0000 0164 0164 0164 0164	ZFPSTP ZFEFIO ZFEMAO ZFELAO	0757 079C 0A7F 0A7C	ZFEMEI ZFELEI ZFELEI ZFECLI	0 A 8 2 0 A 8 2 0 A D 5	ZFEMEO Zfeleo Zfelo	0 A 8 2 0 A 8 2 0 A 0 5	
**C **E P ** C C C C * * *	CURRENT EXT DEFS 7HCOMM 7HREL ROOT PROGOO \$ZFWRK \$COMM LAB1 LAB2 PROGOO CHAIN CHAIN ZFSFIO ZFSWFS ZFOFIO ZFPAUS ZFEFIO ZFEMAI	0000 0000 0000 0000 0164 0164 0164 0164	ZFPSTP ZFEFIO ZFEMAO ZFELAO ZFECSO	0757 079C 0A7F	ZFEMEI ZFELEI	0882 0882	ZFEMEO ZFELEO	28A0 28A0	
**C **E P ** C C C C * * *	CURRENT EXT DEES ZHCOMM ZHREL ROOT PROGOO \$ZFWRK \$COMM LAB1 LAB2 PROGOO CHAIN CHAIN ZFSFIO ZFSWFS <b>ZFOFIO</b> <b>ZFOWRK</b> <b>ZFPAUS</b> <b>ZFEFIO</b> <b>ZFEN</b> <b>ZFEN</b>	0000 0000 0000 0000 0164 0164 0164 0164	ZFPSTP ZFEFIO ZFEMAO ZFELAO	0757 079C 0A7F 0A7C	ZFEMEI ZFELEI ZFELEI ZFECLI	0 A 8 2 0 A 8 2 0 A D 5	ZFEMEO Zfeleo Zfelo	0 A 8 2 0 A 8 2 0 A 0 5	
**C **E P ** C C C C * * *	CURRENT T DEFS THCOMM THREL ROOT PROGOO \$ZFWRK \$COMM LAB1 LAB2 PROGOO CHAIN CHAIN ZFSWFS <b>ZFOFIO</b> <b>ZFFFIO</b> <b>ZFPAUS</b> <b>ZFEFIO</b> <b>ZFEMAI</b> <b>ZFEMAI</b> <b>ZFECAI</b>	0000 0000 0000 0000 0164 0164 0164 0164	ZFPSTP ZFEFIO ZFEMAO ZFELAO ZFECSO ZFECAO	0757 079C 0A7F 0A7C 0AD8 0AD0	ZFEMEI ZFELEI ZFECLI ZFECEI ZFEWFF	0A82 0A82 0AD5 0ADB 0CB9	ZFEMEO ZFELEO ZFECLO ZFECEO ZFERFF	0A82 0A82 0AD5 0ADB 0CB9	
**C **E P ** C C C C * * *	CURRENT EXT DEFS 7HCOMM 7HREL ROOT PROGOO \$COMM LAB1 LAB2 PROGOO CHAIN CHAIN CHAIN ZFSFIO ZFGWRK ZFPFIO ZFPAUS ZFEFIO ZFEAN ZFECAI ZFECAI ZFECAI ZFEIEI	0000 0000 0000 0164 0164 0164 0164 0164	ZFPSTP ZFEFIO ZFEMAO ZFELAO ZFECSO ZFECSO ZFECAO ZFEIEO	0757 079C 0A7F 0A7C 0AD8 0AD0 0B64	ZFEMEI ZFELEI ZFECLI ZFECEI ZFEWFF ZFEIAI	0A82 0A82 0AD5 0AD5 0CB9 0B5E	ZFEMEO ZFELEO ZFECLO ZFECEO ZFERFF ZFEIAO	0A82 0A82 0AD5 0ADB 0CB9 0B5E	
**C **E P ** C C C C * * *	CURRENT EXT DEFS 7HCOMM 7HREL ROOT PROGOO \$ZFWRK \$COMM LAB1 LAB2 PROGOO CHAIN CHAIN CHAIN ZFSFIO ZFSWFS ZFOFIO ZFPAUS ZFEFIO ZFEAN ZFELAI ZFELAI ZFELEI ZFEJEI	0000 0000 0000 0000 0164 0164 0164 0164	ZFPSTP ZFEFIO ZFEMAO ZFELAO ZFECSO ZFECAO ZFEIEO ZFEJEO	0757 079C 0A7F 0A7C 0AD8 0AD0 0B64 0B64	ZFEMEI ZFELEI ZFECLI ZFECEI ZFEWFF ZFEIAI ZFEJAI	0A82 0A82 0AD5 0ADB 0CB9 0B5E 0B61	ZFEMEO ZFELEO ZFECLO ZFECEO ZFERFF ZFEIAO ZFEJAO	0A82 0A82 0A05 0AD5 0ADB 0C89 0B5E 0B5E	
**C **E P ** C C C C * * *	CURRENT EXT DEFS 7HCOMM 7HREL ROOT PROGOO \$ZFWRK \$COMM LAB1 LAB2 PROGOO CHAIN CHAIN CHAIN ZFSFIO ZFSWFS ZFGWRK ZFPFIO ZFPAUS ZFEFIO ZFEAN ZFELAI ZFELAI ZFELEI ZFELEI ZFELEI ZFEKEI	0000 0000 0000 0000 0000 0164 0164 0164	ZFPSTP ZFEFIO ZFEMAO ZFELAO ZFECSO ZFECAO ZFEIEO ZFEJEO ZFEJEO ZFEKEO	0757 079C 0A7F 0A7C 0AD8 0AD0 0B64 0B64 0B56	ZFEMEI ZFELEI ZFECLI ZFECEI ZFEWFF ZFEIAI ZFEKAI	0A82 0A82 0AD5 0AD8 0C89 0B5E 0B61 0B5B	ZFEMEO ZFELEO ZFECLO ZFECEO ZFERFF ZFEIAO ZFEJAO ZFEKAO	0A82 0A82 0AD5 0AD5 0CB9 0B5E 0B61 0B5B	
**C **E P ** C C C C * * *	CURRENT EXT DEFS 7HCOMM 7HREL ROOT PROGOO \$ZFWRK \$COMM LAB1 LAB2 PROGOO CHAIN CHAIN CHAIN ZFSFIO ZFSWFS ZFOFIO ZFPAUS ZFEFIO ZFEAN ZFELAI ZFELAI ZFELEI ZFEJEI	0000 0000 0000 0000 0164 0164 0164 0164	ZFPSTP ZFEFIO ZFEMAO ZFELAO ZFECSO ZFECAO ZFEIEO ZFEJEO	0757 079C 0A7F 0A7C 0AD8 0AD0 0B64 0B64	ZFEMEI ZFELEI ZFECLI ZFECEI ZFEWFF ZFEIAI ZFEJAI	0A82 0A82 0AD5 0AD5 0C89 0B5E 0B61 0B5B 0CF3	ZFEMEO ZFELEO ZFECLO ZFECEO ZFERFF ZFEIAO ZFEJAO	0A82 0A82 0A05 0AD5 0ADB 0C89 0B5E 0B5E	
**C **E P ** C C C C * * *	CURRENT EXT DEFS 7HCOMM 7HREL ROOT PROGOO \$ZFWRK \$COMM LAB1 LAB2 PROGOO CHAIN CHAIN CHAIN ZFSFIO ZFSWFS ZFGWRK ZFPFIO ZFPAUS ZFEFIO ZFEAN ZFELAI ZFELAI ZFELEI ZFELEI ZFELEI ZFEKEI	0000 0000 0000 0000 0164 0164 0164 0164	ZFPSTP ZFEFIO ZFEMAO ZFELAO ZFECSO ZFECSO ZFEJEO ZFEJEO ZFEJEO ZFEJEO	0757 079C 0A7F 0A7C 0AD8 0AD0 0B64 0B64 0B64 0B56 0CF9	ZFEMEI ZFELEI ZFECLI ZFECEI ZFEWFF ZFEIAI ZFEJAI ZFEKAI ZFEDAI	0A82 0A82 0AD5 0AD5 0C89 0B5E 0B61 0B5B 0CF3	ZFEMEO ZFELEO ZFECLO ZFECEO ZFERFF ZFEIAO ZFEJAO ZFEKAO ZFEDAO	0A82 0A82 0AD5 0AD5 0C89 0B5E 0B61 0B5B 0CF3	
* ★ E P * * * C C C C C * * * *	CURRENT T DEFS THCOMM THREL ROOT PROGOO \$ZFWRK \$COMM LAB1 LAB2 PROGOO CHAIN CHAIN ZFSFIO ZFSWFS <b>ZFOFIO</b> ZFSWFS <b>ZFOFIO</b> ZFFEII ZFELAI ZFELAI ZFELAI ZFEDEI ZFEDEI ZFERAI	0000 0000 0000 0000 0164 0164 0164 0164	ZFPSTP ZFEFIO ZFEMAO ZFELAO ZFECSO ZFECAO ZFEIEO ZFEJEO ZFEJEO ZFEKEO	0757 079C 0A7F 0A7C 0AD8 0AD0 0B64 0B64 0B56	ZFEMEI ZFELEI ZFECLI ZFECEI ZFEWFF ZFEIAI ZFEKAI	0A82 0A82 0AD5 0AD8 0C89 0B5E 0B61 0B5B	ZFEMEO ZFELEO ZFECLO ZFECEO ZFERFF ZFEIAO ZFEJAO ZFEKAO	0A82 0A82 0AD5 0AD5 0CB9 0B5E 0B61 0B5B	
**C **E P ** C C C C * * *	CURRENT T DEFS THCOMM THREL ROOT PROGOO \$ZFWRK \$COMM LAB1 LAB2 PROGOO CHAIN CHAIN ZFSFIO ZFSWFS ZFOFIO ZFFAUS ZFEFIO ZFEAI ZFELAI	0000 0000 0000 0000 0164 0164 0164 0164	ZFPSTP ZFEFIO ZFEMAO ZFELAO ZFECSO ZFECAO ZFEJEO ZFEJEO ZFEJEO ZFEDEO ZFERAO	0757 079C 0A7F 0A7C 0AD8 0AD0 0B64 0B64 0B64 0B66 0CF9 0CF6	ZFEMEI ZFELEI ZFECLI ZFECEI ZFEWFF ZFEIAI ZFEJAI ZFEKAI ZFEDAI	0A82 0A82 0AD5 0AD5 0C89 0B5E 0B61 0B5B 0CF3	ZFEMEO ZFELEO ZFECLO ZFECEO ZFERFF ZFEIAO ZFEJAO ZFEKAO ZFEDAO	0A82 0A82 0AD5 0AD5 0C89 0B5E 0B61 0B5B 0CF3	
* ★ C P * * * C C C C C * * * *	CURRENT XT DEFS ZHCOMM ZHREL ROOT PROGOO \$ZFWRK \$COMM LAB1 LAB2 PROGOO CHAIN ZFSWFS ZFOFIO ZFSWFS ZFOFIO ZFPAUS ZFEFIO ZFEAN ZFECAI	0000 0000 0000 0000 0164 0164 0164 0164	ZFPSTP ZFEFIO ZFEMAO ZFELAO ZFECSO ZFECSO ZFEJEO ZFEJEO ZFEJEO ZFEJEO	0757 079C 0A7F 0A7C 0AD8 0AD0 0B64 0B64 0B64 0B56 0CF9	ZFEMEI ZFELEI ZFECLI ZFECEI ZFEWFF ZFEIAI ZFEJAI ZFEKAI ZFEDAI	0A82 0A82 0AD5 0AD5 0C89 0B5E 0B61 0B5B 0CF3	ZFEMEO ZFELEO ZFECLO ZFECEO ZFERFF ZFEIAO ZFEJAO ZFEKAO ZFEDAO	0A82 0A82 0AD5 0AD5 0C89 0B5E 0B61 0B5B 0CF3	
* ★ E P * * * C C C C C * * * *	CURRENT T DEFS THCOMM THREL ROOT PROGOO \$ZFWRK \$COMM LAB1 LAB2 PROGOO CHAIN CHAIN ZFSFIO ZFSWFS ZFOFIO ZFFAUS ZFEFIO ZFEAI ZFELAI	0000 0000 0000 0000 0164 0164 0164 0164	ZFPSTP ZFEFIO ZFEMAO ZFELAO ZFECSO ZFECAO ZFEJEO ZFEJEO ZFEJEO ZFEDEO ZFERAO	0757 079C 0A7F 0A7C 0AD8 0AD0 0B64 0B64 0B64 0B66 0CF9 0CF6	ZFEMEI ZFELEI ZFECLI ZFECEI ZFEWFF ZFEIAI ZFEJAI ZFEKAI ZFEDAI	0A82 0A82 0AD5 0AD5 0C89 0B5E 0B61 0B5B 0CF3	ZFEMEO ZFELEO ZFECLO ZFECEO ZFERFF ZFEIAO ZFEJAO ZFEKAO ZFEDAO	0A82 0A82 0AD5 0AD5 0C89 0B5E 0B61 0B5B 0CF3	

Figure 8-5 (cont).	Linker Output for Chained Programs

USING THE FORTRAN COMPILER

	ZFLB1	1204	ZEGE1	1294	ZFUWUF	1202	ZFURUF	1221	
	ZFAWRT	1231	ZFBRED	1244	ZFSUB1	124F	ZF SUB4	13F8	
	ZFSUB6	1387							
Ρ	ENDCHN	145C							
**	XROG00	1450							
*	PROG01	-							
Ĉ	LACI	1450							
	-								
C	LAC2	148E							
1	PROG01								
· *	PROGD	17DF							
1	PROGD	17DF							
	INDEF								
*	PROGOO	0000							
*	CHAIN	06A3							
*	ZESEIO								
*	ZFOFIO	-							
*	ZFPFIO	-							
Â	ZFEFIO								
1									
*	ZFIOTE	11BA							
*	ZFUFIO								
*	PROG01								
*	PROGD	17DF							
ł									
1	GO2 77								
F	ORTRAN	M4ED 11	/22/1119	1977	/12/02 13	52:46.3	5 SAF		
	FIO 77								
			0 10/15/7						
(C)	COPYRI	GHT 1977	' BY HONEY	WELL IN	FORMATION	SYSTEM	IS INC		
**	TSTCH1		LINK MAP	1977/1	2/02 1354	:06.5			
1		1 4 3							
××3	TART	1612							
**U		1612							
**L	OW 2								
**L **H	OW IGH	145C 16EB							
**L **H **S	OW 1 IGH 1 COMM 0	145C 16EB )164							
**L **H **S	OW IGH	145C 16EB )164							
**L **H **\$ **C	OW 1 IGH 1 COMM 0	145C 16EB )164							
**L **H **S **C **E	OW IGH COMM URRENT XT DEFS	L45C L6EB D164 L6EB							
**L **H **S **C **E P	OW IGH COMM URRENT XT DEFS ZHCOMM	145C 16EB 1164 16EB 0000							
**L **H **S **C **E	OW IGH COMM URRENT XT DEFS	L45C L6EB D164 L6EB							
**L **H **S **C **E P P	OW IGH COMM URRENT XT DEFS ZHCOMM 7HREL	145C 16EB 0164 16EB 0000 0000							
**L **H **S **C **E P P **	OW IGH COMM URRENT XT DEFS ZHCOMM 7HREL ROOT	145C 16EB 0164 16EB 0000 0000							
**L **H **S **C **E P P **	OW IGH COMM URRENT XT DEFS ZHCOMM 7HREL ROOT PROGOO	145C 16EB 0164 16EB 0000 0000 0000							
**L **H **S **E **E P P ** * C	OW IGH COMM URRENT XT DEFS ZHCOMM 7HREL ROOT PROGOO \$ZFWRK	145C 16EB 16EB 0000 0000 0000 0000 0000 0000							
**LH ***R ***C #*#E P ** C C	OW IGH COMM URRENT XT DEFS ZHCOMM 7HREL ROOT PROGOO \$ZFWRK \$COMM	0000 0000 0000 0000 0000 0000 0000 0000 0000							
**LH **** *** P P ** C C C	OW IGH COMM URRENT XT DEFS ZHCOMM 7HREL ROOT PROGOO \$ZFWRK	145C 16EB 16EB 0000 0000 0000 0000 0000 0000							
**LH ***R ***C #*#E P ** C C	OW IGH COMM URRENT XT DEFS ZHCOMM 7HREL ROOT PROGOO \$ZFWRK \$COMM	0000 0000 0000 0000 0000 0000 0000 0000 0000							
**LH **** *** P P ** C C C	OW IGH COMM URRENT XT DEFS ZHCOMM 7HREL ROOT PROGOO \$ZFWRK \$COMM LAB1	145C 16EB 0164 16EB 0000 0000 0000 0000 0164 016A 01CE							
**LH **** *** P P ** C C C	OW IGH COMM URRENT XT DEFS ZHCOMM 7HREL ROOT PROGOO \$ZFWRK \$COMM LAB1 LAB2	145C 16EB 0164 16EB 0000 0000 0000 0000 0164 016A 01CE 0668							
**+H **** *** P P ** C C C C	OW IGH COMM URRENT XT DEFS ZHCOMM 7HREL ROOT PROGOO \$ZFWRK \$COMM LAB1 LAB2 PROGOO CHAIN	145C 16EB 0164 16EB 0000 0000 0000 0164 0164 0164 0168 0168 0668 0643							
**************************************	OW IGH COMM URRENT XT DEFS ZHCOMM 7HREL ROOT PROGOO \$ZFWRK \$COMM LAB1 LAB2 PROGOO CHAIN CHAIN	145C 16EB 0164 16EB 0000 0000 0000 0164 0164 0164 0164 0168 0668 0643 0643							
**+H **** *** P P ** C C C C	OW IGH COMM URRENT XT DEFS ZHCOMM 7HREL ROOT PROGOO \$ZFWRK \$COMM LAB1 LAB2 PROGOO CHAIN CHAIN ZFSFTO	145C 16EB 0164 16EB 0000 0000 0000 0000 0164 016A 016A 016A 016A 0668 06A3 06A3 06A3	75 QDE Q	0466	75 Quille	0708	76 00110	0726	
**************************************	OW IGH COMM COMM COMM CHAC THCOMM THREL ROOT PROGOO \$ZFWRK \$COMM LAB1 LAB2 PROGOO CHAIN CHAIN CHAIN ZFSFTO 7FSWFS	145C 16EB 0164 16EB 0000 0000 0000 0000 0164 016A 01CE 0668 06A3 06A3 06A3 06A2 06AC	ZFSRFS	0600	ZFSWUS	0708	ZFSRUS	072B	
**************************************	OW IGH COMM URRENT XT DEFS ZHCOMM 7HREL ROOT PROGOO \$ZFWRK \$COMM LAB1 LAB2 PROGOO CHAIN CHAIN ZFSFIO 7FSWFS ZFOFIO	145C 16EB 16EB 16EB 0000 0000 0000 0000 0164 016A 01CE 0668 06A3 06A3 06A3 06A2 06AC 074C	ZFSRFS	0600	ZFSWUS	0708	ZF SRUS	072B	
**************************************	OW IGH COMM URRENT XT DEFS ZHCOMM 7HREL ROOT PROGOO \$ZFWRK \$COMM LAB1 LAB2 PROGOO CHAIN CHAIN ZFSFIO 7FSWFS ZFOFIO 7FOWRK	145C 16EB 16EB 0000 0000 0000 0000 0000 0164 016A 016A 016A 06A3 06A3 06A3 06A2 06AC 074C 074C	ZFSRFS	0600	ZFSWUS	0708	ZF SRUS	072B	
**************************************	OW IGH COMM URRENT XT DEFS ZHCOMM 7HREL ROOT PROGOO \$ZFWRK \$COMM LAB1 LAB2 PROGOO CHAIN CHAIN ZFSFIO 7FSWFS ZFOFIO 7FOWRK ZFPFTO	145C 16EB 16EB 16EB 0000 0000 0000 0000 0000 0164 0164 0164			ZFSWUS	0708	ZFSRUS	072B	
**************************************	OW IGH COMM COMM COMM CRENT T XT DEFS ZHCOMM THREL ROOT PROGOO SZFWRK \$COMM LAB1 LAB2 PROGOO CHAIN CHAIN CHAIN ZFSFIO ZFSWFS ZFOFIO ZFPAUS	145C 16EB 0164 16EB 0000 0000 0000 0000 0164 0164 0164 0164 0668 0663 0663 0663 0663 0665 0665 0665 0665 0665 06757 075C	ZFSRFS ZFPSTP	06CC 0757	ZFŚWUS	0708	ZFSRUS	072B	
**************************************	OW IGH COMM COMM COMM CRENT T TRENT THE ROOT PROGOO \$ZFWRK \$COMM LAB1 LAB2 PROGOO CHAIN CHAIN CHAIN CHAIN ZFSFTO ZFOFIO ZFPAUS ZFEFTO	145C 16EB 0164 16EB 0000 0000 0000 0000 0164 0164 0164 0164 0164 0164 0663 0663 0663 0663 0663 0665 074C 074C 0757 075C 079C	ZFPSTP	0757					
**************************************	OW IGH COMM COMM COMM CRENT C XT DEFS ZHCOMM THREL ROOT PROGOO SZFWRK SCOMM LAB1 LAB2 PROGOO CHAIN CHAIN ZFSFTO ZFSWFS ZFOFTO ZFPAUS ZFEFTO ZFAN	145C 16EB 0164 16EB 0000 0000 0000 0000 0164 0164 0164 0164 0164 0668 0663 0663 0663 0664 074C 074C 0757 075C 079C 079C	7FPSTP 7FFF10	0757 079C	7FEMEI	5840	ZFEMEO	0482	
**************************************	OW IGH COMM COMM COMM CRENT CXT DEFS ZHCOMM THREL ROOT PROGOO SZFWRK SCOMM LAB1 LAB2 PROGOO CHAIN CHAIN ZFSFIO ZFSFIO ZFSWFS ZFGFIO ZFPAUS ZFEFTO ZFEMAI	145C 16EB 0164 16EB 0000 0000 0000 0000 0000 0164 0164 0164 0663 0663 0663 0664 074C 074C 074C 075C 079C 079C 0A7F	ZFPSTP 7FFFI0 ZFEMA0	0757 079C 0A7F	7FEMEI Zfelfi	0482 0482	ZFEMEO ZFELEO	0A82 0A82	
**************************************	OW IGH COMM COMM COMM CRENT C XT DEFS ZHCOMM THREL ROOT PROGOO SZFWRK SCOMM LAB1 LAB2 PROGOO CHAIN CHAIN ZFSFTO ZFSWFS ZFOFTO ZFPAUS ZFEFTO ZFAN	145C 16EB 0164 16EB 0000 0000 0000 0000 0164 0164 0164 0164 0164 0668 0663 0663 0663 0664 074C 074C 0757 075C 079C 079C	7FPSTP 7FFF10	0757 079C	7FEMEI	5840	ZFEMEO	0482	
**************************************	OW IGH COMM COMM COMM CRENT CXT DEFS ZHCOMM THREL ROOT PROGOO SZFWRK SCOMM LAB1 LAB2 PROGOO CHAIN CHAIN ZFSFIO ZFSFIO ZFSWFS ZFGFIO ZFPAUS ZFEFTO ZFEMAI	145C 16EB 0164 16EB 0000 0000 0000 0000 0000 0164 0164 0164 0663 0663 0663 0664 074C 074C 074C 075C 079C 079C 0A7F	ZFPSTP 7FFFI0 ZFEMA0	0757 079C 0A7F	7FEMEI Zfelfi	0482 0482	ZFEMEO ZFELEO	0A82 0A82	
**************************************	OW IGH COMM COMM COMM CHC CHC CHC CHC CHC CHC CHC CHC CHC C	145C 16EB 0000 0000 0000 0000 0000 0000 0164 0164 0668 0663 0663 0664 074C 074C 075C 079C 079C 075C 079C 075C	7FPSTP 7FEFI0 7FEMA0 7FELA0	0757 079C 0A7F 0A7C	7FEMEI Zfelfi Zfecli	0482 0482 04D5 04D5	ZFEMFO ZFELEU ZFECLO ZFECEO	0 4 8 2 0 4 8 2 0 4 0 5	
**************************************	OW IGH COMM COMM COMM TRENT XT DEFS ZHCOMM 7HREL ROOT PROGOO \$ZFWRK \$COMM LAB1 LAB2 PROGOO CHAIN CHAIN ZFSTO ZFFOID ZFFAID ZFFAID ZFFAIJ ZFELAI ZFELAI ZFECSI	145C 16EB 0000 0000 0000 0000 0000 0000 0164 0164 0668 0663 0663 0663 0663 0663 0665 074C 074C 0757 075C 079C 070C	ZFPSTP ZFEFIO ZFEMAO ZFELAO ZFECSO	0757 079C 0A7F 0A7C 0AD8	7FEMEI ZFELFI ZFECLI ZFECEI ZFECFI ZFEWFF	0482 0482 04D5 04D5 04D8 0C89	ZFEMEO ZFELEO ZFECLO ZFECEO ZFERFF	0 4 8 2 0 4 8 2 0 4 D 5 0 4 D 5 0 4 D B 0 C B 9	
**************************************	OW IGH COMM COMM COMM CHC COMM CHC CHC CHC CHC CHC CHC CHC CHC CHC C	145C 16EB 0000 0000 0000 0000 0000 0000 0164 0164 0164 0668 0663 0663 0663 0664 074C 0757 075C 079C 070C	ZFPSTP ZFEFIO ZFEMAO ZFELAO ZFECSO ZFECSO ZFECAO ZFEIFO	0757 047F 047C 04D8 04D0 0864	7FEMEI ZFELFI ZFECLI ZFECEI ZFEWFF ZFEIAI	0482 0482 0405 0405 0408 0689 0858	ZFEMEO ZFELEO ZFECLO ZFECEO ZFERFF ZFEIAO	0 4 8 2 0 4 8 2 0 4 D 5 0 4 D 5 0 4 D B 0 C B 9 0 B 5 E	
**************************************	OW IGH COMM COMM COMM CHC COMM CHC CHC CHC CHC CHC CHC CHC CHC CHC C	145C 16EB 0000 0000 0000 0000 0000 0000 0164 0164 0164 0164 0668 0663 0663 0663 0664 074C 074C 075C 079C 070C	ZFPSTP ZFEFIO ZFEMAO ZFELAO ZFECSO ZFECSO ZFECAO ZFEIFO ZFEJEO	0757 047F 047F 047C 04D8 04D0 0864 0864	7FEMEI ZFELFI ZFECLI ZFECFI ZFEWFF ZFFIAI ZFFJAI	0482 0482 0405 0408 0689 0858 0858	ZFEMEO ZFELEO ZFECLO ZFECEO ZFERFF ZFEIAO ZFEJAO	0 4 8 2 0 4 8 2 0 4 0 5 0 4 0 5 0 4 0 5 0 6 1	
**************************************	OW IGH COMM URRENT ZHCOMM ZHCOMM ZHCOMM THREL ROOT PROGOO \$ZFWRK \$COMM LAB1 LAB2 PROGOO CHAIN CHAIN CHAIN ZFSFIO ZFFWRS ZFOFIO ZFPAUS ZFEFTO ZFEANI ZFELAI ZFELAI ZFELAI ZFELAI ZFELAI ZFEJFI ZFEJFI ZFEJFI	145C 16EB 0000 0000 0000 0000 0000 0000 0164 0164 0164 0164 0663 0663 0663 0663 0663 0663 0664 074C 075C 079C 079C 079C 079C 079C 079C 079C 079C 079C 0864 0864 0856	ZFPSTP 7FEFI0 7FELA0 7FELA0 7FECS0 7FECS0 7FECA0 7FECA0 7FEJE0 7FEJE0 7FEJE0	0757 047F 047F 047C 04D8 04D0 0864 0864 0856	7FEMEI ZFELFI ZFECLI ZFECEI ZFEWFF ZFFIAI ZFEJAI ZFEJAI	0 A 8 2 0 A 8 2 0 A D 5 0 A D B 0 C B 9 0 B 5 E 0 B 5 B	ZFEMEO ZFELEO ZFECLO ZFECEO ZFERFF ZFEIAO ZFEJAO ZFEKAO	0482 0482 04D5 04D5 04D5 0689 085E 0861 085B	
***** *** ****************************	OW IGH COMM COMM COMM CHC COMM CHC CHC CHC CHC CHC CHC CHC CHC CHC C	145C 16EB 0000 0000 0000 0000 0000 0000 0164 0164 0164 0164 0668 0663 0663 0663 0664 074C 074C 075C 079C 070C	ZFPSTP ZFEFIO ZFEMAO ZFELAO ZFECSO ZFECSO ZFECAO ZFEIFO ZFEJEO	0757 047F 047F 047C 04D8 04D0 0864 0864	7FEMEI ZFELFI ZFECLI ZFECFI ZFEWFF ZFFIAI ZFFJAI	0482 0482 0405 0408 0689 0858 0858	ZFEMEO ZFELEO ZFECLO ZFECEO ZFERFF ZFEIAO ZFEJAO	0 4 8 2 0 4 8 2 0 4 0 5 0 4 0 5 0 4 0 5 0 6 1	

Figure 8-5 (cont). Linker Output for Chained Programs

ZFI	ERAI	0CF6	7FFRAO	0CF6	ZFEREI	OCFC	ZFEREU	OCFC	
	IOTE	11BA							
	IOTE	11BA	ZFFINI	1109					
	UFIO	1202					_		
	LB1	1204	ZFGF1	1294	ZFUWUF	1202	ZFURUF	1221	
	AWRT	1231	ZEBRED	1244	ZESUBI	124F	ZF SUB4	13F8	
	SUB6	1387							
P EN	DCHN	1450							
** XR	0600	1450							
	0602	1450							
	0602	1450							
	0602	1612							
	BFTO	16B3							
21	BCMC	1683							
**UNDE	5								
	, 0600	0000							
	AIN	0643							
	SFIO	0640							
	QFIO	074C							
	PFIO	0740							
	EFIO	0790							
	IOTE	118A							
	UFIO	1202							
	0602	1450							
	BFIO	1683							
			1/22/1119	17//	/12/02 13	72199.3	anr		
FORT									
FORT	TCH1	450	LINK MAP		2/02 1354		-		
<b>Fort</b> ** TS **Star	ТСН1 Т 1								
<b>FORT</b> ** TS **Star **Luw	ТСН1 Т 1 1	45C							
<b>FORT</b> ** TS **STAR **Luw **HIGH	TCH1 T 1 1	45C 45C							
FORT ** TS **STAR **LUW **HIGH	TCH1 T 1 1 M 0	45C 45C 4P6 164							
<b>FORT</b> ** TS **STAR **LUW **HIGH **COM **CUPR	TCH1 T 1 1 M 0 FNT 1	45C 45C 4P6 164							
FORT TS TS TS TS TS TS TS TS TS T	TCH1 T 1 1 M 0 FNT 1 DEFS	45C 45C 486 164 486							
FORT * TS *STAR *LUW *HIGH *CUPR *CUPR *FXT 7H	TCH1 T 1 1 M 0 FNT 1 DEFS CUMM	45C 45C 486 164 486 0000							
FORT * TS *STAR *LUW *HIGH *CUPR *CUPR *FXT 7H	TCH1 T 1 1 M 0 FNT 1 DEFS	45C 45C 486 164 486							
FORT AT TS AT STAR AT STAR	TCH1 T 1 M 0 FNT 1 DEFS CUMM REL	45C 45C 486 164 486 0000							
FORT: ** TS **STAR **LUW **HIGH **CUPR **CUPR **CUPR **FXT 7 H 7 H * R0	TCH1 T 1 M 0 FNT 1 DEFS CUMM REL	450 450 486 164 486 0000 0000							
<b>FORT</b> <b>*</b> TS <b>*</b> STAR <b>*</b> LUW <b>*</b> HIGH <b>*</b> CUPR <b>*</b> CUPR <b>*</b> CUPR <b>*</b> CUPR <b>*</b> CUPR <b>*</b> CUPR <b>*</b> COPR <b>*</b> C	TCH1 T 1 1 M 0 FNT 1 DEFS CUMM REL OT OGOU FWPK	450 450 486 164 486 0000 0000 0000 0000 0000							
FORTI TS TS TS TAR TS TAR TS TAR TS TAR TS TAR TS TAR TS TAR TS TAR TS TAR TS TAR TA TA TA TA TA TA TA TA TA TA	TCH1 T 1 1 M 0 FNT 1 DEFS CUMM REL 0 GOU	45C 45C 486 164 486 0000 0000 0000							
<b>FORT</b> <b>*</b> TS <b>*</b> STAR <b>*</b> LUW <b>*</b> HIGH <b>*</b> CUPR <b>*</b> CUPR <b>*</b> CUPR <b>*</b> CUPR <b>*</b> CUPR <b>*</b> CUPR <b>*</b> CUPR <b>*</b> CUPR <b>*</b> CUPR <b>*</b> COPR <b>*</b> COPCOPR <b>*</b> COPR <b>*</b>	TCH1 T 1 1 M 0 FNT 1 DEFS CUMM REL 0G00 FWPK 0MM B1	450 450 486 164 486 0000 0000 0000 0000 0000							
<b>FORT</b> <b>*</b> TS <b>*</b> TS	TCH1 T 1 H 1 FNT 1 DEFS CUMM REL OT OGOU FWPK OMM R1 P2	45C 476 164 486 0000 0000 0000 0000 0000 0000 0164 0164							
<b>FORT</b> <b>*</b> TS <b>*</b> TS	TCH1 T 1 1 M 0 FNT 1 DEFS CUMM REL 0G00 FWPK 0MM B1	45C 476 164 486 0000 0000 0000 0000 0000 0164 0164 016							
FORTI           ** TS           **SIAR           **LIGH           **COPR           **CUPR           **CUPR           **FXT           7H           **FXT           7H           **FXT           7H           **FXT           7H           **FXT           7H           **COR           **FXT           **The           **The </td <td>TCH1 T 1 H 1 FNT 1 DEFS CUMM REL OT OGOU FWPK OMM R1 P2</td> <td>45C 476 164 486 0000 0000 0000 0000 0000 0000 0164 0164</td> <td></td> <td></td> <td></td> <td></td> <td></td> <td></td> <td></td>	TCH1 T 1 H 1 FNT 1 DEFS CUMM REL OT OGOU FWPK OMM R1 P2	45C 476 164 486 0000 0000 0000 0000 0000 0000 0164 0164							
<b>FORT</b> ** TS **STAR **LUW **LUW **CUPR **CUPR **CUPR **CUPR ** CH	TCH1 T 1 H 1 FNT 1 DEFS CUMM REL OT OGOU FWPK OMM B1 P2 OGOO	450 486 164 486 0000 0000 0000 0000 0164 0166 0166 0668 0683 0683							
FORTI ** TS **STAR **LUW **HIGH **CUPR *	TCH1 T 1 1 M 0 FNT 1 DEFS CUMM RFL OT OT OT OT MM B1 H2 OG00 A IN	450 486 164 486 0000 0000 0000 0000 0000 0164 0164 016							
FORTI ** TS **STAR **LUW **HIGH **COM **CUPR **	TCH1 T 1 1 M 0 FNT 1 DEFS CUMM RFL 0G00 FWPK 0MM B1 P2 0G00 AIN AIN	450 486 164 486 0000 0000 0000 0000 0164 0166 0166 0668 0683 0683					7F SRUS	0728	
FORTI           ** TS           ** TS           ** TS           ** TS           ** TS           ** TS           ** TAR           ** HIGH           ** CUPR           ** FXT           ** TS           ** CUPR           ** FXT           *           *           *           C	TCH1 T 1 1 1 1 1 1 1 1 1 1 1 1 1 1	450 450 486 164 486 0000 0000 0000 0000 0000 0164 0166 0663 0663 0663 0663 06640 0663 06640 06740	LINK ΜΔΡ	1977/1	2/02 1354	1:06.5		0728	
FORTI           ** TS           ** TS           ** TS           ** TS           ** TAR           ** HIGH           ** CUPR           ** FXT           ** FXT           ** TS           ** TS           ** CUPR           ** FXT           *	TCH1 T 1 1 M 0 FNT 1 DEFS CUMM REL 0G00 FWPK 0G10 FWPM AIN SFTU SFTU SFTU RWR	450 476 476 476 476 476 476 476 476 476 476	LINK ΜΔΡ	1977/1	2/02 1354	1:06.5		0.158	
FORTI           ** TS           ** TAR           ** TIG           ** CUPR           ** FXT           ** THIGH           ** TO           ** THIGH           ** CH           *	TCH1 T 1 1 M 0 FNT 1 DEFS CUMM RFL 0G00 FWPK 0GN0 FWPK 0GN0 FWPK 0GN0 SFT0 SFT0 SFT0 SFT0 SFT0 PFT0	450 476 476 476 476 476 476 476 476 476 476	LINK MAP	1977/1 n6CC	2/02 1354	1:06.5		072B	
FORTI           ** TS           ** TAR           ** TG           ** CUPR           ** FXT           ** TS           ** THIGAN           ** TAR           ** CUPR           ** C           ** C           * C           * C           * C           * C           * C           * C           * C           * C           * C           * C           * C           * C           * C           * C           * C           * T           * T           * T           * T           * T           * T           * T           * T           * T           * T           * T           * T           * T           * T           * T     <	TCH1 T 1 I M 0 FNT 1 DEFS CUMM REL OGOU FWPK OMM REL OGOU FWPK OMM SFL OGOU SWFS OGFTU SWFSU OFTU PAHS	450 476 476 476 476 476 476 476 476 476 476	LINK ΜΔΡ	1977/1	2/02 1354	1:06.5		0728	
<b>FORT</b> <b>*</b> TS <b>*</b> CO <b>*</b> CS <b>*</b> CS <b>*</b> CC <b>*</b> TS <b>*</b> TS	TCH1 T 1 1 1 1 1 1 1 1 1 1 1 1 1 1	450 476 476 476 476 476 476 476 476 476 476	LINK MAP 7FSRFS 7FPSTP	1977/1 ∩6СС 0757	2/02 1354 7FSwiis	0708	7F SRUS		
FORTI           ** TS           ** STAR           ** CUPR           ** CUPR           ** STAR           * PR           * PR           C           * C           C           * C           * C           * C           * C           * C           * C           * C           * C           * T	TCH1 T 1 1 1 1 1 1 1 1 1 1 1 1 1 1	450 476 476 164 486 0000 0000 0000 0000 0000 0164 0166 0663 0663 06640 0740 0740 0757 0750 0790	LINK MAP 7FSRFS 7FPSTP 7FFFTU	1977/1 n6CC n75/ n79C	2/02 1354 7FSw118 7FFME1	0708	7FSRIIS ZFFMFU	0482	
<b>FORT</b> <b>*</b> TS <b>*</b> STAR <b>*</b> COPR <b>*</b> COPR	TCH1 T 1 1 1 1 1 1 1 1 1 1 1 1 1 1	450 476 476 164 486 0000 0000 0000 0000 0164 0166 06A3 06A0 0740 0740 0740 0750 0790 0790 047F	LINK MAP 7FSRFS 7FPSTP 7FFFTU 7FFFTU	1977/1 1977/1 06CC 0757 079C 0A7F	2/02 1354 7FSw118 7FFME1 7FFLF1	0708 0708	ZFFMFU ZFFLFU	5840 5840	
<b>FORT</b> <b>*</b> TS <b>*</b> CS <b>*</b> TS <b>*</b> TS	TCH1 T 1 1 1 1 1 1 1 1 1 1 1 1 1 1	450 476 476 476 476 476 476 476 476 476 476	LINK MAP 7FSRFS 7FPSTP 7FFFTU 7FFFTU 7FFLAU	1977/1 1977/1 06CC 0757 079C 0A7F 0A7C	2/02 1354 7FSw115 7FFME1 7FFLF1 7FECL1	0708 0708 0482 0482 0405	7F SRIIS 2F F M F U 7F F L F U 7F F L F U 7F F C L U	0482 0482 0405	
<b>FORT</b> <b>*</b> TS <b>*</b> CO <b>*</b> CC <b>*</b> TS <b>*</b> CC <b>*</b> TS <b>*</b> TS <b>*</b> CC <b>*</b> TS <b>*</b> TS <b>*</b> TS <b>*</b> CC <b>*</b> TS <b>*</b> TS <b>*</b> TS <b>*</b> TS <b>*</b> CC <b>*</b> TS <b>*</b> TS	TCH1 T 1 1 M 1 FNT 1 PEFS CUMM FFS CUM FFS CUM FFS CUM FFS CUM FFS CUM FFS CUM FFS FFS CUM FFS FFS FFS FFS FFS FFS FFS FF	45C 476 476 476 476 476 476 476 476 476 476	LINK MAP 7FSRFS 7FPSTP 7FFFTU 7FFFTU 7FFLAU 7FFLAU	1977/1 1977/1 06CC 0757 079C 0A7F 0A7C 0AD8	2/02 1354 7FSwIIS 7FFME1 7FFLF1 7FECL1 7FFCFI	0708 0708 0482 0482 0405 0405	7F SRIIS 2FFMFU 7FFLFU 7FFCL0 2FFCEU	0482 0482 0405 0405	
<b>FORT</b> <b>t</b> TS <b>t</b> TS	TCH1       1         T       1         M       1         M       1         FNT       1         DEFS       COMM         CUMM       R         FORT       OGON         GON       AIN         SFTUS       SOFTUS         OWNTUS       FELAIN         FLAIS       FCS1         FCAI       FCS1	45C 476 476 476 476 476 476 476 476 476 476	LINK MAP 7FSRES 7FPSTP 7FFFTU 7FFFTU 7FFLAU 7FFCAU	1977/1 1977/1 06CC 0757 047F 047F 0476 04D8 04D0	2/02 1354 7FSw115 7FFME1 7FFLF1 7FECL1	0708 0708 0482 0482 0405	7F SRIIS 2F F M F U 7F F L F U 7F F L F U 7F F C L U	0482 0482 0405	
FORTI           ** TS           ** TAR           ** CUPR           ** FXT           ** TS           ** CUPR           ** C           * C           C           * C           C           C           * C           * C           * C           * C           * C           * C           * C           * C           * C           * TF	TCH1 T 1 1 M 1 FNT 1 DEFS COMM FNT 0 FNT 0	45C 476 476 476 476 476 476 476 476 476 476	LINK MAP 7ESRES 7EPSTP 7EEEAU 7EEAU 7EECSU 7EECAU 7EECO 7EEO	06CC 0757 079C 0A7F 0A7C 0AD8 0AD0 0864	2/02 1354 7FSwIIS 7FFME1 7FFLF1 7FECL1 7FFCFI	0708 0708 0482 0482 0405 0405	7FSR115 7FFRF0 7FFLF0 7FFCL0 7FFCE0 7FFCE0 7FFCF0 7FFRFF 7FF1A0	0482 0482 0405 0405	
<b>FORT</b> <b>**</b> TS <b>**</b> TS <b>**</b> TS <b>**</b> TS <b>**</b> TS <b>**</b> TS <b>**</b> TS <b>**</b> TS <b>**</b> TS <b>**</b> COM <b>**</b> COM <b>**</b> CC <b>*</b> CC <b></b>	TCH1 T 1 1 M 1 FNT 1 DEFS COMM FNT 1 DEFS COMM FNT 1 DEFS COMM FNT 1 DEFS COMM FNT 1 COMM FNT 1 DEFS COMM FNT 1 DEFS COMM FNT 1 DEFS COMM FNT 1 DEFS COMM FNT 1 DEFS COMM FNT 1 COMM FNT 0 FNT 0	45C 476 476 476 476 476 476 476 476 476 476	LINK MAP 7FSRES 7FPSTP 7FFFTU 7FFEA0 7FFLA0 7FFCA0 7FFLO 7FFIE0 7FFJE0	1977/1 1977/1 06CC 0757 047F 047F 0476 04D8 04D0	2/02 1354 7FSwiis 7FSwiis 7FFLF1 7FFCL1 7FFCF1 7FFCF1 7FFCF1	0482 0482 0482 0482 0405 0405 0405 0405	7F SRIIS 7F SRIIS 7F FL FU 7F FL FU 7F FL FU 7F FL FV 7F FR FF	0482 0482 0405 0405 0405 0455 0855 0855	
<b>FORT</b> * TSR * STAW * STAW * * STAW * STAW * * STAW * * STAW * STAW	TCH1       1         T       1         M       0         FNT       1         DEFS       0         RET       0         FWM       0         FWM       0         FWM       0         MAIN       0         MAIN       0         SFFF       0         FLAIN       0      <	450 486 486 486 000 0000 0000 0000 0000 00	LINK MAP 7ESRES 7EPSTP 7EEEAU 7EEAU 7EECSU 7EECAU 7EECO 7EEO	1977/1 1977/1 05CC 0757 079C 0A7F 0A7C 0A76 0A08 0A08 0A08 0A64 0B64 0B64	2/02 1354 7FSwHS 7FSwHS 7FELF1 7FECL1 7FECL1 7FFCF1 7FF1A1	0482 0482 0482 0482 0405 0405 0405 0405 0455	7FSR115 7FFRF0 7FFLF0 7FFCL0 7FFCE0 7FFCE0 7FFCF0 7FFRFF 7FF1A0	0482 0482 0405 0405 0405 0405 0455	
<b>FORT</b> * TS * STAW * * STAW * * STAW * * STAW F * * STAW F * * STAW F * * STAW F F 7 F 7 F 7 F 7 F 7 F 7 F 7 F 7 F 7 F	TCH1 T 11 M 0 FNT 1 PEFS COMPK OF 0 FNT 1 PEFS COMPK OF 0 FNT 1 PEFS COMPK OF 0 FNT 1 PEFS COMPK OF 0 FNT 1 PEFS COMPK OF 0 FNT 1 PEFS COMPK OF 0 FNT 0 FNT 1 PEFS COMPK OF 0 FNT 0	450 486 486 486 486 000 0000 0000 0000 000	LINK MAP 7FSRFS 7FPSTP 7FFFTU 7FFFA0 7FFLA0 7FFCA0 7FFC0 7FFC0 7FFLO 7FFLO 7FFLO 7FFLO 7FFLO	1977/1 1977/1 06CC 075/ 0A7F 0A7C 0A7F 0A7C 0AD8 0AD0 0864 0856 0CF9	2/02 1354 7FFME1 7FFLF1 7FFLF1 7FFCF1 7FFCF1 7FFSF1A1 7FFJA1 7FFJA1 7FFJA1 7FFJA1	0482 0482 0482 0405 0405 0405 0461 0655 0661 0655 0661	7F SRIIS 2F F M F O 7F F L F U 7F F C F U 7F F C F U 7F F C F U 7F F I A U 7F F J A O	0482 0482 0405 0405 0658 0858 0858 0858 0858 0858	
<b>FORT</b> ** TS **STAR **SCUPR **STAR **SCUPR **STAR **SCUPR **STAR **SCUPR **STAR **SCUP	TCH1 T 11 M 0 FNT 1 PEFS COMPA COMPAN AIN AIN AIN SSFTS OW FLAI FLAI FLAI FLAI FLAI FLAI FLFI FLAI FLFI FRAI	450 476 476 476 476 476 476 476 476 476 476	LINK MAP 7FSRFS 7FPSTP 7FFFTU 7FFLAU 7FFLAU 7FFCAU 7FFCAU 7FFLEU 7FFJFU 7FFJFU 7FFLFU	1977/1 1977/1 05CC 0757 079C 0A7F 0A7C 0A76 0A08 0A08 0A08 0A64 0B64 0B64	2/02 1354 7FSwUS 7FSwUS 7FFLF1 7FECL1 7FECL1 7FEVFF 7FF1A1 7FFJA1 7FFKA1	0708 0708 0482 04D5 04D5 04D5 0455 0455 0455	7F SRUS 2FF MFU 7FFLFU 7FFCL 0 7FFCFU 7FFCFU 7FFCFU 7FFLAU 7FFLAU 7FFLAU	0482 0482 0405 0405 0405 055 0855 0855	
** TS **STAR **STAR **STAR **CUPR **CPR ??FFF ??FFF ??FFF ??FFF ??FFF ??FFF ??FFF ??FFF ??FFF ??FFF ??FFF ??FFF ??FFF ??FFF ??FFF ??FFF ??FFF ??FFF ??FFFF ??FFFF ??FFFF ??FFFFFF	TCH1 T 11 M 0 FNT 1 PEFS COMPK OF 0 FNT 1 PEFS COMPK OF 0 FNT 1 PEFS COMPK OF 0 FNT 1 PEFS COMPK OF 0 FNT 1 PEFS COMPK OF 0 FNT 1 PEFS COMPK OF 0 FNT 0 FNT 1 PEFS COMPK OF 0 FNT 0	450 486 486 486 486 000 0000 0000 0000 000	LINK MAP 7FSRFS 7FPSTP 7FFFTU 7FFFA0 7FFLA0 7FFCA0 7FFC0 7FFC0 7FFLO 7FFLO 7FFLO 7FFLO	1977/1 1977/1 06CC 075/ 0A7F 0A7C 0A7F 0A7C 0AD8 0AD0 0864 0856 0CF9	2/02 1354 7FFME1 7FFLF1 7FFLF1 7FFCF1 7FFCF1 7FFSF1A1 7FFJA1 7FFJA1 7FFJA1 7FFJA1	0482 0482 0482 0405 0405 0405 0461 0655 0661 0655 0661	7F SRUS 2FFMF0 7FFLF0 7FFCF0 7FFCF0 7FFCF1A0 7FFLA0 7FFA0 7FFDA0	0482 0482 0405 0405 0658 0858 0858 0858 0858 0858	

Figure 8-5 (cont). Linker Output for Chained Programs

#### USING THE FORTRAN COMPILER

8-12

	ZETOTE		7FFINI	1109					
*	ZEUEIO	1505							
	7FL91	1204	7 F G F 1	1294	ZEUWUE	1505	ZEURUE	1551	
	ZFAWRT	1231	7FRRFD	1244	ZESUB1	124F	ZF SUB4	13F8	
	7FSUR6	1397							
Р	ENDCHN								
•		14.0							
**	XROGOO	1450							
**	XKUCU5	145C							
**	XROG03	1450							
*	PRICOG								
•									
	PROGOS	1450							
**!	INDEF								
*	PROGOO	0000							
*	CHA1N	06A3							
*	ZFSFIU	0640							
*	7FOFTU								
*	ZEPETO								
*	7FEFI0								
*	ZFIOTE								
*	ZFUFIO								
*	PROG03	145C							
PRO	G04 77	120200							
	-		1/22/1119	1977	/12/02 13	52:46.3	SAF		
•	UNTOAN		1, 5, 1, 1, 1, 1, 1	• • • •			•		
				10774	2/02 1354	• 0 4 E			
**			LINK MAP	19/1/1	2/02 1554	:00.5			
~**S	START	1450							
**L	.OW	1450							
**H	IGH	1482							
* * 5	COMM	0164							
	URRENT	-							
	NT BEEN								
	XT DEFS								
Ρ	ZHCOMM	0000							
Ρ	ZHCOMM	0000							
P	ZHCOMM	0000							
P P **	ZHCOMM ZHREL ROOT	0000 0000 0000							
P P **	ZHCOMM ZHREL RONT PRNGNO	0000 0000 0000 0000							
P P ** C	ZHCOMM ZHREL RONT PRNGNO \$ZFWPK	0000 0000 0000 0000 0000							
P P ** C C	ZHCOMM ZHREL PROGIO \$ZFWPK \$COMM	0000 0000 0000 0000 0154							
P ++ C C C	ZHCOMM ZHREL PROGIO \$ZFWPK \$COMM LAB1	0000 0000 0000 0000 0154 0154							
P P ** C C	ZHCOMM ZHREL PROGIO \$ZFWPK \$COMM	0000 0000 0000 0000 0154							
P P ** C C C	ZHCOMM ZHREL PROGIO \$ZFWPK \$COMM LAB1	0000 0000 0000 0000 0154 0154 0154							
P P ** C C C	ZHCOMM ZHREL PROGOO \$ZFWPK \$COMM LAB1 LAB2	0000 0000 0000 0000 0154 0154 0154							
P +* C C C C C	7HCOMM 7HREL ROOT PROGOO \$2FWRK \$COMM LAB1 LAB2 PROGOO CHAIN	0000 0000 0000 0000 0164 0164 0166							
P ** C C C C C	7HCOMM 7HREL ROOT PROGOO \$2FWRK \$COMM LAB1 LAB2 PROGOO CHAIN CHAIN	0000 0000 0000 0164 0164 0164 0168 0683 0683							
P +* C C C C	7HCOMM 7HREL ROOT PROGOO \$2FWRK \$COMM LAB1 LAB2 PROGOO CHAIN CHAIN 7FSFTU	0000 0000 0000 0164 0164 0164 0168 0683 0683 0683	76 SHES	0.50	ZE Swiis	0708	ZESRUS	07215	
P P ** C C C C *	7HCOMM 7HREL PROGOO \$2FWRK \$COMM LAB1 LAB2 PROGOO CHAIN CHAIN 7FSFTU 7FSWFS	0000 0000 0000 0164 0164 0164 0668 0683 0683 0683 0683	7+ 5KF S	ЛъСС	7FSwUS	0708	ZF SRUS	07215	
P ** C C C C C	7HCOMM 7HREL ROOT PROGOO \$2FWPK \$COMM LAB1 LAB2 PROGOO CHAIN CHAIN 7FSFTU 7FSWFS 7FOFTU	0000 0000 0000 0164 0164 0164 0663 0663 0663 0663 0663 0663	7+ 5KF \$	0°CC	7F SWUS	0708	ZESRUS	n728	
P P ** C C C C C * *	7HCOMM 7HREL ROOT PROGOO \$2FWPK \$COMM LAB1 LAB2 PROGOO CHAIN CHAIN 7FSFTU 7FSWFS 7FOFTU 7FOWPK	0000 0000 0000 0164 0164 0164 0663 0663 0663 0663 0663 0663 0663 06	7+SKFS	0°CC	7FSwUS	0708	ZESRUS	072B	
P P ** C C C C *	7HCOMM ZHREL RONT PRNGOO \$2FWRK \$COMM LAB1 LAB2 PRNGOO CHAIN CHAIN 7FSFTU 7FOWRK 7FPFIO	0000 0000 0000 0164 0164 0164 0668 0668 0668 0668 0668 0668 0668 06			7F SwUS	0708	ZF SRUS	n7213	
P P ** C C C C C * *	7HCOMM 7HREL ROOT PROGOO \$2FWPK \$COMM LAB1 LAB2 PROGOO CHAIN CHAIN 7FSFTU 7FSWFS 7FOFTU 7FOWPK	0000 0000 0000 0164 0164 0164 0668 0668 0668 0668 0668 0668 0668 06	7FPSTP	∩ьСС 0757	7FSwUS	0708	ZF SRUS	07213	
P P ** C C C C C * *	7HCOMM ZHREL RONT PRNGOO \$2FWRK \$COMM LAB1 LAB2 PRNGOO CHAIN CHAIN 7FSFTU 7FOWRK 7FPFIO	0000 0000 0000 0164 0164 0164 0164 0668 0668 0663 0663 0663 0663 0663 0663			7FSwUS	0708			
P P ** C C C C C * *	7HCOMM ZHREL ROAT PRAGAO \$2FWRK \$CAMM LAB1 LAB2 PRAGAO CHAIN CHAIN 7FSFTO 7FSWFS 7FOFTO 7FOWRK 7FPFIO 7FPAUS	0000 0000 0000 0164 0164 0164 0164 0668 0668 0663 0663 0663 0664 0740 0740 0757 0750			ZFSWUS ZFFMFI	0708	ZE SRUS ZE SRUS	072B	
P P ** C C C C C * *	7HCOMM 7HREL ROOT PROGOO \$2FWPK \$COMM LAB1 LAR2 PROGOO CHAIN 7FSFTU 7FSWFS 7FOFTU 7FOWPK 7FPFIO 7FPAUS 7FFFTU 7FAN	0000 0000 0000 0164 0164 0164 0668 0663 0663 0663 0664 0740 0740 0757 0750 0790	7FPSTP 7FEFTU	0757 079C	7FFMF1				
P P ** C C C C C * *	7HCOMM 7HREL ROOT PROGOO \$2FWPK \$COMM LAB1 LAB2 PROGOO CHAIN 7FSFTU 7FSWFS 7FOFTU 7FSWFS 7FOFTU 7FFAN 7FFFIO 7FAN 7FFMA1	0000 0000 0000 0164 0164 0164 0668 0643 0643 0640 0740 0740 0740 0757 0750 0790 0790	7FPSTP 7FFFTU 7FFM4U	0757 079C 0A7F	7FEMF1 7FFLEI	5440 5440	ZFEMED ZFELEU	5840 5840	
P P ** C C C C C * *	7HCOMM 7HREL ROOT PROGOO \$2FWPK \$COMM LAB1 LAB1 LAB2 PROGOO CHAIN 7FSFTU 7FSFTU 7FSWFS 7FOFTU 7FSWFS 7FPFIO 7FFAN 7FFAN 7FFAN 7FFAN	0000 0000 0000 0164 0164 0164 0663 0643 0643 0643 0640 0740 0740 0757 0757 0757 0757 0790 047F 0470	7FPSTP 7FFFTU 7FFM40 7FFL40	0757 079C 0A7F 0A7C	7FEMF1 7FFLE1 7FFCL1	0482 0482 0405	ZFEMFO ZFELFU ZFECLO	0485 0485 0402	
P P ** C C C C C * *	7HCOMM ZHREL RONT PRNG00 \$2FWRK \$COMM LAB1 LAB2 PRNG00 CHAIN 7FSFT0 7FSFT0 7FSFT0 7FFF10 7FFAN 7FFF10 7FFAN 7FFFA1 7FFCS1	0000 0000 0000 0164 0164 0164 0663 0663 0663 0663 0663 0663 0663 0740 0740 0740 0757 0750 0790 0790 0676 0790 0676 0790	7FPSTP 7FFFTU 7FFMAU 7FFLAU 7FFLAU 7FFCSU	0757 079C 0A7F 0A7C 0A08	7FEMF1 7FFLE1 7FECL1 7FECE1	0482 0482 0405 0405	ZFEMFO ZFELFO ZFECLO ZFECEO	0485 0485 0402	
P P ** C C C C C * *	7HCOMM ZHREL RONT PRNG00 \$2FWRK \$COMM LAB1 LAB2 PRNG00 CHAIN 7FSFT0 7FSFT0 7FSFT0 7FFF10 7FFAN 7FFF10 7FFAN 7FFFA1 7FFCA1	0000 0000 0000 0164 0164 0164 0663 0663 0663 0663 0663 0663 0663 06	7FPSTP 7FFFTU 7FFMAU 7FFLAU 7FFC9U 7FFC9U 7FFC40	0757 0790 047F 0470 0408 0400	7FEMF1 7FFLE1 7FECL1 7FECE1 7FFCF1 7FFWFF	0482 0482 0405 0408 0689	ZFEMEO ZFELEU ZFECLO ZFECEO ZFEREE	0482 0482 0405 0405 0408	
P P ** C C C C C * *	7HCOMM ZHREL RONT PRNG00 \$2FWRK \$COMM LAB1 LAB2 PRNG00 CHAIN 7FSFT0 7FSFT0 7FSFT0 7FFF10 7FFAN 7FFF10 7FFAN 7FFFA1 7FFCS1	0000 0000 0000 0164 0164 0164 0663 0663 0663 0663 0663 0663 0740 0740 0740 0740 0757 0750 0790 0790 0676 0790 0676 0608 0600	7FPSTP 7FFFTU 7FFMAU 7FFLAU 7FFLAU 7FFCSU	0757 0475 0475 0470 0408 0400 0864	7FEMFI 7FFLEI 7FFCLI 7FECEI 7FECEI 7FFNFF 7FEIAI	0482 0482 0405 0405 0408 0689 0656	7FEMF0 7FELF0 7FFCL0 7FFCE0 7FFCF0 7FFRFF 7FF140	0482 0482 0405 0405 0408 0689 0858	
P P ** C C C C C * *	7HCOMM ZHREL RONT PRNG00 \$2FWRK \$COMM LAB1 LAB2 PRNG00 CHAIN 7FSFT0 7FSFT0 7FSFT0 7FFF10 7FFAN 7FFF10 7FFAN 7FFFA1 7FFCA1	0000 0000 0000 0164 0164 0164 0663 0663 0663 0663 0663 0663 0663 06	7FPSTP 7FFFTU 7FFMAU 7FFLAU 7FFC9U 7FFC9U 7FFC40	0757 0790 047F 0470 0408 0400	7FEMF1 7FFLE1 7FECL1 7FECE1 7FFCF1 7FFWFF	0482 0482 0405 0408 0689	7FEME0 7FELE0 7FECL0 7FFCE0 7FFRFE 7FFRFE 7FF140 7FFJA0	0482 0482 0405 0405 0405 0455 0851	
₽ ₽ ₽ ₽ * * * C C C C C * * *	7HCOMM ZHREL RONT PRNGOO \$2FWPK \$COMM LAB1 LAB2 PRNGOO CHAIN 7FSFTO 7FOFTO 7FOFTO 7FOFTO 7FOFTO 7FFMAI 7FFMAI 7FFCAI 7FFCAI 7FFIFI 7FFJFI	0000 0000 0000 0164 0164 0164 0164 0668 0668 0668 0668 0668 0668 0664 0740 0740 0757 0750 0790 0790 0790 0476 0476 0476 0408 0464 0864	7FPSTP 7FFFTU 7FFMAU 7FFLAU 7FFCSU 7FFCAU 7FFCAU 7FFCAU	0757 0475 0475 0470 0408 0400 0864	7FEMFI 7FFLEI 7FFCLI 7FECEI 7FECEI 7FFNFF 7FEIAI	0482 0482 0405 0405 0408 0689 0656	7FEMF0 7FELF0 7FFCL0 7FFCE0 7FFCF0 7FFRFF 7FF140	0482 0482 0405 0405 0408 0689 0858	
₽ ₽ ₽ ₽ ₽ * * * C C C C C * * *	7HCOMM ZHREL RONT PRNG00 \$2FWPK \$COMM LAB1 LAB2 PRNG00 CHAIN 7FSFT0 7FSFT0 7FSFT0 7FOFT0 7FOWR 7FPF10 7FPAN 7FFMA1 7FFLA1 7FFCS1 7FFCS1 7FFCS1 7FFCS1 7FFCS1 7FFLF1 7FFLF1	0000 0000 0000 0164 0164 0164 0668 0663 0663 0663 0664 0740 0757 0750 0790 0790 0790 0790 0790 079	7FPSTP 7FFFTU 7FFLAU 7FFLAU 7FFCSU 7FFCAU 7FFCAU 7FFJFU 7FFJFU	0757 047F 047F 047C 0408 0400 0864 0864 0856	7FEMFI 7FFLEI 7FFCLI 7FFCEI 7FFWFF 7FFIAI 7FFJAI 7FFJAI	0482 0482 0405 0405 0406 0089 0856 0856	7FEME0 7FELE0 7FECL0 7FFCE0 7FFRFE 7FFRFE 7FF140 7FFJA0	0482 0482 0405 0405 0405 0455 0851	
₽ ₽ ₽ * * * C C C C C * * *	7HCOMM ZHREL ROAT PRAGAD \$2FWPK \$COMM LAB1 LAB2 PRAGAO CHAIN 7FSFTU 7FSWFS 7FOFTU 7FSWFS 7FFFIO 7FAN 7FFAN 7FFAN 7FFCAI 7FFCAI 7FFCAI 7FFCAI 7FFCAI 7FFCAI 7FFCAI 7FFCAI 7FFCAI	0000 0000 0000 0164 0164 0164 0663 0663 0663 0664 0740 0757 0750 0790 0790 0790 0790 0790 079	7FPSTP 7FFFTU 7FFLAU 7FFLAU 7FFCSU 7FFCSU 7FFCSU 7FFJFU 7FFJFU 7FFDFU	0757 047F 047C 047C 047C 047C 047C 047C 047C 047C	7FEMFI 7FFLEI 7FFCLI 7FFCEI 7FFWFF 7FFIAI 7FFJAI 7FFDAI	0482 0482 0405 0405 0405 0689 0856 0856 0856 0653	ZFEMEU ZFELEU ZFECLU ZFECEU ZFEREE ZFEIAU ZFEJAU ZFEKAU ZFEDAU	0482 0482 0405 0405 0408 0458 0858 0658 0653	
P * * * C C C C C * * * * *	7HCOMM 7HREL ROOT PROGOO \$2FWPK \$COMM LAB1 LAR2 PROGOO CHAIN 7FSWFS 7FOFTO 7FSWFS 7FOFTO 7FOWPK 7FFFIO 7FFAN 7FFAN 7FFAN 7FFCAI 7FFCAI 7FFCAI 7FFCAI 7FFCAI 7FFCAI 7FFCAI	0000 0000 0000 0164 0166 0663 0663 0663 0664 0740 0740 0740 0757 0757 0757 0757 075	7FPSTP 7FFFTU 7FFLAU 7FFLAU 7FFCSU 7FFCAU 7FFCAU 7FFJFU 7FFJFU	0757 047F 047F 047C 0408 0400 0864 0864 0856	7FEMFI 7FFLEI 7FFCLI 7FFCEI 7FFWFF 7FFIAI 7FFJAI 7FFJAI	0482 0482 0405 0405 0406 0089 0856 0856	7FEME0 7FELE0 7FECE0 7FECE0 7FEREE 7FFRFE 7FFIA0 7FFLA0 7FFKA0	0482 0482 0405 0405 0406 0689 0861 0861 0858	
₽ ₽ ₽ ₽ * * * C C C C C * * *	7HCOMM ZHREL ROOT PROGOO \$2FWPK \$COMM LAB1 LAB1 LAB2 PROGOO CHAIN 7FSFTU 7FSFTU 7FSWFS 7FOFTU 7FSWFS 7FFFIO 7FAN 7FFAN 7FFAN 7FFLAI 7FFCAI 7FFCAI 7FFCAI 7FFLFI 7FFL 7FFL	0000 0000 0000 0164 0166 0663 0663 0664 0740 0740 0740 07757 0757 0757 0757 07	7FPSTP 7FFF40 7FFL40 7FFC40 7FFC40 7FFC40 7FF1F0 7FF1F0 7FFJF0 7FFDF0 7FFR40	0757 047F 047C 0408 0400 0864 0864 0864 0856 0CF9 0CF6	7FEMFI 7FFLEI 7FFCLI 7FFCEI 7FFWFF 7FFIAI 7FFJAI 7FFDAI	0482 0482 0405 0405 0405 0689 0856 0856 0856 0653	ZFEMEU ZFELEU ZFECLU ZFECEU ZFEREE ZFEIAU ZFEJAU ZFEKAU ZFEDAU	0482 0482 0405 0405 0408 0158 0858 0858 0053	
P P ** C C C C C C C * * *	ZHCOMM ZHREL RONT PRNG00 %ZFWRK %COMM LAB1 LAR2 PRNG00 CHAIN ZFFNG00 ZFAIN ZFFFI0 ZFFFI0 ZFFFI0 ZFFFI0 ZFFFI1 ZFFFA1 ZFFCA1 ZFFCA1 ZFFCFI ZFFCFI ZFFCA1 ZFFCFI ZFFCA1 ZFFCFI ZFFCA1	0000 0000 0000 0164 0168 0683 0683 0683 0683 0683 0740 0740 0740 0740 07750 07750 07750 07750 0864 0856 06559 06559 06559 06559 06559 06559 06559 06559 06559 07560 0000 000	7FPSTP 7FFFTU 7FFLAU 7FFLAU 7FFCSU 7FFCSU 7FFCSU 7FFJFU 7FFJFU 7FFDFU	0757 047F 047C 047C 047C 047C 047C 047C 047C 047C	7FEMFI 7FFLEI 7FFCLI 7FFCEI 7FFWFF 7FFIAI 7FFJAI 7FFDAI	0482 0482 0405 0405 0405 0689 0856 0856 0856 0653	ZFEMEU ZFELEU ZFECLU ZFECEU ZFEREE ZFEIAU ZFEJAU ZFEKAU ZFEDAU	0482 0482 0405 0405 0408 0158 0858 0858 0053	
P P ** C C C C C C C * * *	7HCOMM ZHREL ROOT PROGOO \$2FWPK \$COMM LAB1 LAB1 LAB2 PROGOO CHAIN 7FSFTU 7FSFTU 7FSWFS 7FOFTU 7FSWFS 7FFFIO 7FAN 7FFAN 7FFAN 7FFLAI 7FFCAI 7FFCAI 7FFCAI 7FFLFI 7FFL 7FFL	0000 0000 0000 0164 0168 0683 0683 0683 0683 0683 0740 0740 0740 0740 07750 07750 07750 07750 0864 0856 06559 06559 06559 06559 06559 06559 06559 06559 06559 07560 0000 000	7FPSTP 7FFF40 7FFL40 7FFC40 7FFC40 7FFC40 7FF1F0 7FF1F0 7FFJF0 7FFDF0 7FFR40	0757 047F 047C 0408 0400 0864 0864 0864 0856 0CF9 0CF6	7FEMFI 7FFLEI 7FFCLI 7FFCEI 7FFWFF 7FFIAI 7FFJAI 7FFDAI	0482 0482 0405 0405 0405 0689 0856 0856 0856 0653	ZFEMEU ZFELEU ZFECLU ZFECEU ZFEREE ZFEIAU ZFEJAU ZFEKAU ZFEDAU	0482 0482 0405 0405 0408 0158 0858 0858 0053	
P P ** C C C C C C * * * *	ZHCOMM ZHREL RONT PRNG00 %ZFWRK %COMM LAB1 LAR2 PRNG00 CHAIN ZFFNG00 ZFAIN ZFFFI0 ZFFFI0 ZFFFI0 ZFFFI0 ZFFFI1 ZFFFA1 ZFFCA1 ZFFCA1 ZFFCFI ZFFCFI ZFFCA1 ZFFCFI ZFFCA1 ZFFCFI ZFFCA1	0000 0000 0000 0164 0168 0683 0683 0683 0683 0683 0740 0740 0740 0740 07750 07750 07750 07750 0864 0856 06559 06559 06559 06559 06559 06559 06559 06559 06559 07560 0000 000	7FPSTP 7FFF40 7FFL40 7FFC40 7FFC40 7FFC40 7FF1F0 7FF1F0 7FFJF0 7FFDF0 7FFR40	0757 047F 047C 0408 0400 0864 0864 0864 0856 0CF9 0CF6	7FEMFI 7FFLEI 7FFCLI 7FFCEI 7FFWFF 7FFIAI 7FFJAI 7FFDAI	0482 0482 0405 0405 0405 0689 0856 0856 0856 0653	ZFEMEU ZFELEU ZFECLU ZFECEU ZFEREE ZFEIAU ZFEJAU ZFEKAU ZFEDAU	0482 0482 0405 0405 0408 0158 0858 0858 0053	

Figure 8-5 (cont).	Linker Output f	or Chained Programs
--------------------	-----------------	---------------------

USING THE FORTRAN COMPILER

(

State of the second

(

8-13

	7FL81	1204	ZEGE1	1294	ZEUWUE	1202	7FURUE 1221
	7FAWRT	1231	ZEBRED	1244	7FSUB1	124F	ZESUR4 13E8
	7FSUB6	1387					
Р	ENDCHN	1450					
**	XKUGUO	1450					
**	XROG02	1450					
**	XROG03	1450					
**	XROG04	1450					
*	PROG04	145C					
	PROG04	1450					
**U	INDEF						
*	PROGOO	0000					
*	CHAIN	0643					
*	7FSFTU	06AC					
*	ZERETO	074C					
*	ZEPETO	0757					
*	7FFFT0	0/90					
*	ZETOTE	<u>118</u> A					
*	7FUFT0	1505					
*	PROG04	1450					
	******						
	T TSTCH	1					
	******						
	HEST OVE	Y 57N	UM OF SYM	S 0			
	******						
SA	F *****						
RON		1	BASE	0000	ST	0668	I HIGH=1450
*** 0VL	****** Y XROGO	0	# 00 BASE	1450	ST 1	782	1 HIGH=180F
***	******						
0VL	Y XROG0	2	# 01 BASE	1450	ST 1	612	I HIGH=16FB
OVL	Y XROGO	3	# 02 BASE	1450	ST 1	450	I HIGH=1486
	****** Y XRNG0	4	# 03 BASE	145C	ST 1	450	I HTGH=1482
	******			· · •		-	
	-	UT AND	STATTC OV	LYS= 18	10F	HT PEL	RCD= 62
	******						
	K DONE						
***	******						

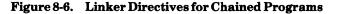
Figure 8-5 (cont). Linker Output for Chained Programs

Figure 8-6 illustrates the linker directives required to create the bound unit TSTCH1, comprising the programs listed in Figures 8-3 and 8-4.

Figure 8-7 shows the output resulting from the execution of the chained programs.

8-14

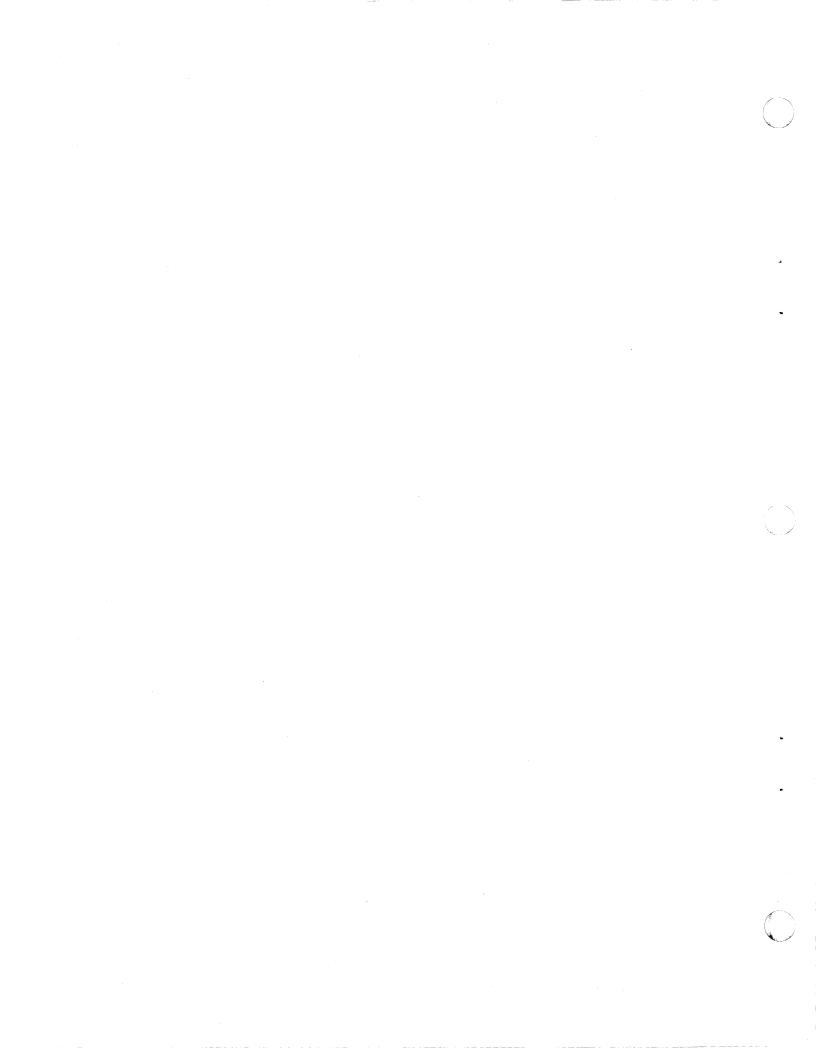
	^VL5901>LDD>0BJI PROGOO	ECT>FRIO	र		
OVLY LDEF PROT BASE	XROGOO ENDCHN,\$ ENDCHN ENDCHN PROGO1			O (CHAIN OVERLAYS	
BASE OVLY	ENDCHN XROGO1 PROGO2	DEFINES	OVERLAY	1 (CHAIN	1)
OVLY BASE	XROGO2 ENDCHN PROGO3	DEFINES	OVERLAY	2 (CHAIN	2)
OVLY BASE	XROGO3 ENDCHN PROGO4	DEFINES	OVERLAY	3 (CHAIN	3)



```
PROGO APPEARS ON FXFCUTE LINE - CALLS CHAIN O
CARD 1
     SUBROUTINE PROGD LOADED
PROG1 IS CHAIN O WHICH CALLS CHAIN 1
PROG2 TS CHAIN 1 - WHICH CALLS - CHAIN 0
CARD 2
     SUBROUTINE PROGD LOADED
PROG1 IS CHAIN O WHICH CALLS CHAIN 1
PROG2 IS CHAIN 1 - WHICH CALLS - CHAIN 2
PROG3 IS CHAIN 2 - WHICH CALLS CHAIN 3
CARD 3
PROG4 IS CHAIN 3 - CALLS CHAIN 1
PROG2 IS CHAIN 1 - WHICH CALLS - CHAIN 3
```

#### Figure 8-7. Execution Output from Chained Programs

USING THE FORTRAN COMPILER



### Section 9 Using the Sort

Figure 9-1 contains a sample session at the operator terminal to sort a file using the Sort utility. Sort descriptors are entered through the operator terminal. Refer to the *Sort/Merge* manual for details on the use of the Sort component.

C :\$H: SD "1977/02/15 1428" (\$H)RDY: CWD ^SRTI02 (\$H)RDY: Zrrr06>SORT (\$H)ENTER SORT DESCRIPTION FILES: -IF IDSF06 -OF ODSF02 -WF ^SRTCW2>WDSF02 ; KEYS: CHAR (6) 78 D, CHAR 4 36 ; QUIT (\$H)MOUNT ^SRTCW2>WDSF02 (\$H)SORT-rrrr-∆mm/dd/hhmm \$H)INPUT FILE : ^SRTI02>IDSF06 \$H)RECORDS READ 000350 \$H)OUTPUT FILE:^SRTIO2>ODSF02 \$H)RECORDS WRITTEN 000350 \$H)RECORDS DELETED 000000 \$H)RDY:

Figure 9-1. Sample Sort Terminal Session

The Sort utility is on volume Zrrr06 and the application files are on SRTI02. The Sort is invoked by entering the pathname,  $^{7}$ Zrrr06>SORT. The Sort description statements are then entered. In this example, the work file is not mounted, and a message to mount  $^{7}$ SRTCW2>WDSF02 is issued.

der des

.

•

INDEX

ACCESS ACCESS BY LOGGING IN, 4-1 ACCESS THROUGH THE OPERATOR OR ANOTHER USER, 4-2 ACCESS THROUGH THE OPERATOR TERMINAL, 4-4 USER ACCESS TO THE SYSTEM, 4-1 ASSEMBLER ASSEMBLER OUTPUT LISTING (SMPMAC) (FIG), 6-4USING THE ASSEMBLER AND MACRO PREPROCESSOR, 6-1 BATCH EXECUTION OF AN APPLICATION FROM THE BATCH TASK GROUP, 4-3 CALL CALL FORTRAN ROUTINES FROM AN ENTRY-LEVEL COBOL MAIN PROGRAM, 7-7 CALLING FORTRAN PROGRAMS CALLING THE CHAIN FUNCTION (FIG), 8-7 CHAINING FORTRAN CHAINING, 8-1 COBOL CALL FORTRAN ROUTINES FROM AN ENTRY-LEVEL COBOL MAIN PROGRAM, 7-7 COBOL LISTING OF COBFRT (FIG), 7-8 COMPILING WITH COBOL, 7-2 SAMPLE COBOL TERMINAL SESSION (AC8111), 7-3 USING THE COBOL COMPILER, 7-1 COMMAND COMMAND PROCESSOR AS LEAD TASK, 4-2COMPILER USING THE COBOL COMPILER, 7-1 USING THE FORTRAN COMPILER, 8-1 CONCURRENT CONCURRENT EXECUTION FROM SEVERAL TASK GROUPS, 4-3 CONCURRENT EXECUTION OF APPLICATION TASKS, 4-3 DATA EXECUTION FROM THE DATA ENTRY FACILITY (DEF), 4-4 DEDICATED DEDICATED APPLICATION ENVIRONMENT, 2-2 DEF RBF AND DEF USER GUIDE TO MANUALS (FIG), 1-5 RBF AND DEF USER MANUAL GUIDE, 2-1

EDITOR EDITOR DIRECTIVE DESCRIPTION, 5-1 SAMPLE EDITOR DIRECTIVES IN FILE SMPCMDFL (FIG), 5-1 ENVIRONMENT ALL-ONLINE ENVIRONMENT, 2-1 DEDICATED APPLICATION ENVIRONMENT, 2-2 MIXED ENVIRONMENT, 2-2 ONLINE/BATCH ENVIRONMENT, 2-2 OPERATOR-ONLY ENVIRONMENT, 2-1 EXECUTION CONCURRENT EXECUTION FROM SEVERAL TASK GROUPS, 4-3 CONCURRENT EXECUTION OF APPLICATION TASKS, 4-3EXECUTION FROM THE DATA ENTRY FACILITY (DEF), 4-4EXECUTION OF AN APPLICATION FROM THE BATCH TASK GROUP, 4-3 EXECUTION OUTPUT FROM CHAINED PROGRAMS (FIG), 8-15 SERIAL EXECUTION OF APPLICATION TASKS, 4-2FORTRAN CALL FORTRAN ROUTINES FROM AN ENTRY-LEVEL COBOL MAIN PROGRAM, 7-7 FORTRAN CHAINING, 8-1 FORTRAN LISTING OF FRTRAN (FIG), 7-9 FORTRAN PROGRAMS CALLING THE CHAIN FUNCTION (FIG), 8-7 SAMPLE FORTRAN TERMINAL SESSION (MATINV), 8-1 USING THE FORTRAN COMPILER, 8-1 GUIDE APPLICATIONS PROGRAMMER'S MANUAL GUIDE, 1-2APPLICATIONS PROGRAMMER GUIDE TO MANUALS (FIG), 1-3 GUIDE TO USING THE MANUAL SET, 1-1 OPERATOR'S MANUAL GUIDE, 1-5 OPERATOR GUIDE TO MANUALS (FIG), 1-4RBF AND DEF USER GUIDE TO MANUALS (FIG), 1-5RBF AND DEF USER MANUAL GUIDE, 2-1 SYSTEM PROGRAMMER'S MANUAL GUIDE, 1-2SYSTEM PROGRAMMER GUIDE TO MANUALS (FIG), 1-4LEAD TASK APPLICATION AS LEAD TASK, 4-2

INDEX

LINKER LINKER DIRECTIVE'S FOR CHAINED PROGRAMS (FIG), 8-15 LINKER OUTPUT FOR CHAINED PROGRAMS (FIG), 8-9 LINKER OUTPUT LISTING (SMPMAC) (FIG), 6-5SOURCE AND LINKER OUTPUT LISTING (MATINV) (FIG), 8-2 LISTING ASSEMBLER OUTPUT LISTING (SMPMAC) (FIG), 6-4ASSEMBLY LISTING OF PROGRAM CHAIN (FIG), 8-7COBOL LISTING OF COBFRT (FIG), 7-8 CROSS REFERENCE LISTING (SMPMAC) (FIG), 6-3FORTRAN LISTING OF FRTRAN (FIG), 7-9 LINKER OUTPUT LISTING (SMPMAC) (FIG), 6-5 SAMPLE LISTINGS FOR AC8111 (FIG), 7-4 SOURCE AND LINKER OUTPUT LISTING (MATINV) (FIG), 8-2 LOADING SOURCE LOADING, 7-2 LOGIN ABBREVIATED LOGIN TERMINAL, 4-1 DIRECT LOGIN TERMINAL, 4-1 FULL LOGIN TERMINAL, 4-1 STARTUP WITH THE LOGIN FACILITY, 3-1 MACRO MACRO PREPROCESSOR OUTPUT (SMPMAC) (FIG), 6-2 SAMPLE OF UNEXPANDED MACRO ROUTINE (SAMPL1) (FIG), 5-10 SAMPLE OF UNEXPANDED MACRO ROUTINE (SAMPL2) (FIG), 5-11 SAMPLE OF UNEXPANDED PROGRAM WITH MACRO CALLS (SMPMACP) (FIG), 5-9 USING THE ASSEMBLER AND MACRO PREPROCESSOR, 6-1 MANUAL APPLICATIONS PROGRAMMER'S MANUAL GUIDE, 1-2GUIDE TO USING THE MANUAL SET, 1-1 OPERATOR'S MANUAL GUIDE, 1-5 RBF AND DEF USER MANUAL GUIDE, 2-1 SYSTEM PROGRAMMER'S MANUAL GUIDE, 1-2MANUALS APPLICATIONS PROGRAMMER GUIDE TO MANUALS (FIG), 1-3OPERATOR GUIDE TO MANUALS (FIG), 1-4RBF AND DEF USER GUIDE TO MANUALS (FIG), 1-5

MANUALS (CONT) SYSTEM PROGRAMMER GUIDE TO MANUALS (FIG), 1-4MULTITASK SAMPLE ASSEMBLY LANGUAGE MULTITASK PROGRAM (BRDCST), 6-6 ONELINE/BATCH ONLINE/BATCH ENVIRONMENT, 2-2 OPERATING OPERATING ENVIRONMENT, 2-1 OPERATOR ACCESS THROUGH THE OPERATOR OR ANOTHER USER, 4-2ACCESS THROUGH THE OPERATOR TERMINAL, 4-4 OPERATOR GUIDE TO MANUALS (FIG), 1-4 OPERATOR TERMINAL SESSION FOR COBFRT (FIG), 7-9 OUTPUT ASSEMBLER OUTPUT LISTING (SMPMAC) 8 (FIG), 6-4EXECUTION OUTPUT FROM CHAINED PROGRAMS (FIG), 8-15 LINKER OUTPUT FOR CHAINED PROGRAMS (FIG), 8-9 LINKER OUTPUT LISTING (SMPMAC) (FIG), 6-5 MACRO PREPROCESSOR OUTPUT (SMPMAC) (FIG), 6-2 SOURCE AND LINKER OUTPUT LISTING (MATINV) (FIG), 8-2 PROGRAM ASSEMBLY LISTING OF PROGRAM CHAIN (FIG) 8-7 CALL FORTRAN ROUTINES FROM AN ENTRY-LEVEL COBOL MAIN PROGRAM, 7-7 SAMPLE ASSEMBLY LANGUAGE MULTITASK PROGRAM (BRDCST), 6-6 SAMPLE CARD-TO-DISK PROGRAM, 7-1 SAMPLE OF UNEXPANDED PROGRAM WITH MACRO CALLS (SMPMACP) (FIG), 5-9 PROGRAMS EXECUTION OUTPUT FROM CHAINED PROGRAMS (FIG), 8-15 FORTRAN PROGRAMS CALLING THE CHAIN FUNCTION (FIG), 8-7 LINKER DIRECTIVES FOR CHAINED PROGRAMS (FIG), 8-15 LINKER OUTPUT FOR CHAINED PROGRAMS (FIG), 8-9 RBF RBF AND DEF USER GUIDE TO MANUALS (FIG), 1-5RBF AND DEF USER MANUAL GUIDE, 2-1

i-2

INDEX

SAMPLE SAMPLE ASSEMBLY LANGUAGE MULTITASK PROGRAM (BRDCST), 6-6 SAMPLE ASSEMBLY LANGUAGE SESSION (SMPMAC), 6-1 SAMPLE CARD-TO-DISK PROGRAM, 7-1 SAMPLE COBOL TERMINAL SESSION (AC8111), 7-3 SAMPLE EDITOR DIRECTIVES IN FILE SMPCMDFL (FIG), 5-1 SAMPLE FORTRAN TERMINAL SESSION (MATINV), 8-1 SAMPLE LISTINGS FOR AC8111 (FIG), 7-4 SAMPLE OF UNEXPANDED MACRO ROUTINE (SAMPL1) (FIG), 5-10 SAMPLE OF UNEXPANDED MACRO ROUTINE (SAMPL2) (FIG), 5-11 SAMPLE OF UNEXPANDED PROGRAM WITH MACRO CALLS (SMPMACP) (FIG), 5-9 SAMPLE SORT TERMINAL SESSION (FIG), 9-1 SAMPLE TERMINAL SESSION (AC8111) (FIG), 7-3 SAMPLE TERMINAL SESSION (BRDCST) (FIG), 6-7 SAMPLE TERMINAL SESSION (MATINV) (FIG), 8-1 SAMPLE TERMINAL SESSION (SMPMAC) (FIG), 6-1SERIAL SERIAL EXECUTION OF APPLICATION TASKS, 4-2 SORT SAMPLE SORT TERMINAL SESSION (FIG), 9-1 USING THE SORT (FIG), 1-4 SOURCE SOURCE AND LINKER OUTPUT LISTING (MATINV) (FIG), 8-2 SOURCE LOADING, 7-2 STARTUP STARTUP WITH THE LOGIN FACILITY, 3-1 TASK GROUP-SPECIFIC TERMINAL STARTUP, 3-1 USER TERMINAL STARTUP, 3-1 SYSTEM SYSTEM PROGRAMMER'S MANUAL GUIDE, 1-2SYSTEM PROGRAMMER GUIDE TO MANUALS (FIG), 1-4USER ACCESS TO THE SYSTEM, 4-1 TASK APPLICATION AS LEAD TASK, 4-2 COMMAND PROCESSOR AS LEAD TASK, 4-2 CONCURRENT EXECUTION FROM SEVERAL TASK GROUPS, 4-3

TASK (CONT) EXECUTION OF AN APPLICATION FROM THE BATCH TASK GROUP, 4-3 TASK GROUP-SPECIFIC TERMINAL STARTUP, 3-1 TASKS CONCURRENT EXECUTION OF APPLICATION TASKS, 4-3TERMINAL ABBREVIATED LOGIN TERMINAL, 4-1 ACCESS THROUGH THE OPERATOR TERMINAL, 4-4 DIRECT LOGIN TERMINAL, 4-1 FULL LOGIN TERMINAL, 4-1 OPERATOR TERMINAL SESSION FOR COBFRT (FIG), 7-9 SAMPLE COBOL TERMINAL SESSION (AC8111), 7-3 SAMPLE FORTRAN TERMINAL SESSION (MATINV), 8-1 SAMPLE SORT TERMINAL SESSION (FIG), 9-1 SAMPLE TERMINAL SESSION (AC8111) (FIG), 703 SAMPLE TERMINAL SESSION (BRDCST) (FIG), 6-7 SAMPLE TERMINAL SESSION (MATINV) (FIG), 8-1 SAMPLE TERMINAL SESSION (SMPMAC) (FIG), 6-1 TASK GROUP-SPECIFIC TERMINAL STARTUP, 3-1 TERMINAL RESPONSES FROM DIRECTIVES OF FIG 5-1 (FIG), 5-2 USER TERMINAL STARTUP, 3-1 VOLUME VOLUME AND FILE CREATION, 7-2

#### HONEYWELL INFORMATION SYSTEMS

Technical Publications Remarks Form

#### TITLE

- CUT ALONG LINE

「「「「」

#### SERIES 60 (LEVEL 6) GCOS 6 MOD 400 PROGRAMMER'S GUIDE

ORDER NO.

CB22, REV. 0

DATED JANUARY 1978

ERRORS IN PUBLICATION

SUGGESTIONS FOR IMPROVEMENT TO PUBLICATION

						7		
as requir	omments will be pr red. If you require	e a written rep	ily, check here a	nd furnish comple	te mailing addre	ss below.		
as requir	red. If you require	a written rep	ly, check here a	nd furnish comple	te mailing addre	ss below.	<sup>en</sup>	
as requir	red. If you require : NAME	a written rep	ly, check here a	nd furnish comple	te mailing addre	ss below.		

PLEASE FOLD AND TAPE – NOTE: U. S. Postal Service will not deliver stapled forms

> Business Reply Mail Postage Stamp Not Necessary if Mailed in the United States

Postage Will Be Paid By:

HONEYWELL INFORMATION SYSTEMS 200 SMITH STREET WALTHAM, MA 02154

**ATTENTION: PUBLICATIONS, MS 486** 

## Honeywell

×-----

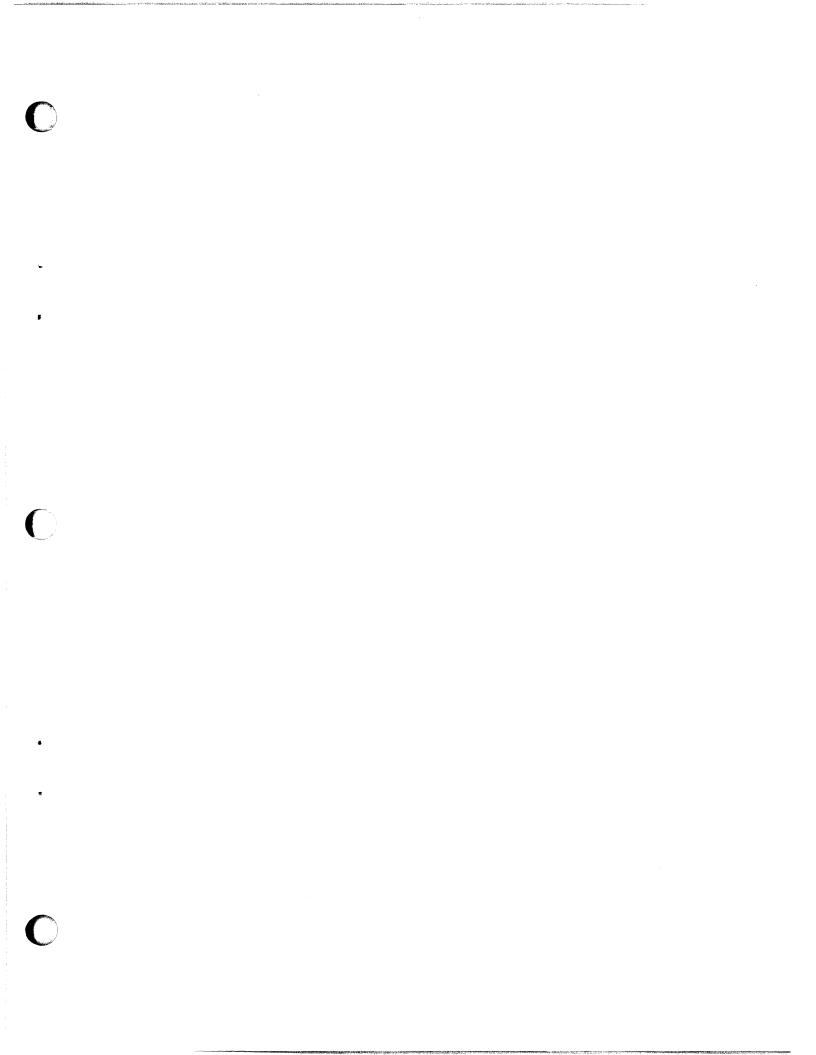
FOLD ALONG LINE

CUT ALONG LINE

FOLD ALONG LINE

FIRST CLASS PERMIT NO. 39531 WALTHAM, MA

02154



# Honeywell

Honeywell Information Systems In the U.S.A.: 200 Smith Street, MS 486, Waltham, Massachusetts 02154 In Canada: 2025 Sheppard Avenue East, Willowdale, Ontario M2J 1W5 In Mexico: Avenida Nuevo Leon 250, Mexico 11, D.F.

20682, 5578, Printed in U.S.A.

CB22, Rev. 0