

MAN2602

PRIMOS INTERACTIVE  
User Guide

Revision A  
2 January 77

**PRIME**  
Computer, Inc.

145 Pennsylvania Ave.  
Framingham, Mass. 01701

Copyright 1977 by  
Prime Computer, Incorporated  
145 Pennsylvania Avenue  
Framingham, Massachusetts 01701

Performance characteristics are  
subject to change without notice.

# CONTENTS

	PAGE
	----
SECTION 1      INTRODUCTION	
SYSTEMS DEFINITIONS	1-1
SYSTEMS CONFIGURATIONS	1-1
PRIMOS II FEATURES	1-2
PRIMOS III FEATURES	1-3
PRIMOS IV FEATUJRES	1-3
SECTION 2      USING PRIMOS	
INTRODUCTION	2-1
SUBSET OF COMMANDS AND	2-1
SAMPLE USER SESSIONS	
PREPARING FOR PRIMOS USAGE	2-2
ACCESS TO SYSTEM	
BEGINNING A SESSION AT THE TERMINAL	
SYSTEM RESPONSE AND COMMAND FORMAT	
COMMANDS FOR MANIPULATING	
FILES AND DIRECTORIES	2-
THE CONCEPT OF ATTACHMENT	2-1
FURTHER FILE AND DIRECTORY	
MANIPULATIONS	2-20
PROGRAM DEVELOPMENT UNDER	
PRIMOS II OR III	2-23
PROGRAM DEVELOPMENT UNDER	
PRIMOS IV	2-28
DEBUGGING	2-28
USING FILE UTILITY (FUTIL)	
SECTION 3      COMMANDS	
COMMAND STRUCTURE	3-1
SUMMARY AND INTRODUCTION TO COMMANDS	3-4
COMMAND DESCRIPTIONS	3-
ADDISK	3-7
AMLC	3-7
ASRCWD	3-8
ASSIGN	3-9
ATTACH	3-13
AVAIL	3-16
BASIC	3-18
BASINP	3-18
BINARY	3-19
CLOSE	3-19
CMPRES	3-20

# CONTENTS (Cont)

CNAME	3-21
CNVIMA	3-21
COMINPUT	3-22
CONFIG	3-25
COPY	3-25
CREATE	3-26
CRMPC	3-27
CRSER	3-27
DBASIC	3-27
DELAY	3-28
DELETE	3-29
ED	3-30
EDB	3-32
EXPAND	3-32
FILMEM	3-32
FILVER	3-33
FIXRAT	3-35
FTN	3-36
FUTIL	3-37
HILOAD	3-39
INPUT	3-39
I	3-40
	3-40
	3-41
	3-42
	3-43
	3-44
	3-45
	3-46
IK	3-47
MAC RST	3-51
MAGSAV	3-54
MAKE	3-54
MUG	3-54
ML	3-54
MESSAGE	3-55
MTASK	3-56
NUMBER	3-56
OPEN	3-57
OPRPRI	3-58
PASSWL	3-59
PHANTOM	3-60
PM	3-63
PMA	3-64
PRERR	3-65
PROTECT	3-66
PRVER	3-69
PSD	3-69
PSD20	3-69
PSD160	3-69
PTCPY	3-70
PUSS	3-71
RESTOR	3-73
RESUME	3-76



## ILLUSTRATIONS

<u>Figure No.</u>	<u>Title</u>	<u>Page</u>
2-1	Typical Terminal Keyboard	2-3
2-2	Typical Portion of PRIMOS File System, Showing: MFD, Command UFD, and User UFD.	2-14
2-3	Attaching to UFD on Another Disk	2-15
2-4	Attaching Subdirectories and Returning to User Level Directory (JDOAKS).	2-16
4-1	Set up of Line Configurator	4-5

## CONTENTS (Cont)

RUNOFF	3-77
SAVE	3-80
SED	3-80
SEG	3-81
SETIME	3-83
SFTN	3-83
SHUTDN	3-83
SIZE	3-84
SLIST	3-85
SORT	3-86
SPOOL	3-88
START	3-93
STARTUP	3-94
STATUS	3-95
SVCSW	3-101
TAP	3-101
TIME	3-102
TRAMLC	3-103
UDOS64	3-104
UNASSIGN	3-105
UPCASE	3-106
USERS	3-106
USRCWD	3-107
VRTSSW	3-108
*	3-108

SECTION 4	PRIMOS IN NETWORKS
-----------	--------------------

APPENDIX A	SUMMARY OF ALL COMMANDS
APPENDIX B	PRIMOS ERROR MESSAGES
APPENDIX C	ERRVEC CONTENTS
APPENDIX D	CHARACTER SET
APPENDIX E	I-O VIRTUALIZATION

## TABLES

<u>Table No.</u>	<u>Title</u>	<u>Page</u>
2-1	Terminal Keys with Special Significance	2-4
3-1	Value for Virtual Control Word and Port Assignment.	3-8
3-2	Device Names.	3-10
3-3	Subset of ED (Editor) Commands.	3-31
3-4	FUTIL Commands.	3-38
3-5	RUNOFF Commands	3-78

## FOREWORD

This manual provides information for the Prime Disk Operating System (PRIMOS II), the Prime Virtual Memory Disk Operating System (PRIMOS III) and the operating system for the Prime 400 computer (PRIMOS IV). The systems described are as implemented on master disk Revision 11 (Rev. 11).

We assume users are familiar with FORTRAN IV, COBOL, or Prime macro-assembly language programming. This manual emphasizes user/system dialogs at a terminal of a multi-user system.

Commands and operations that are entered only at the system console (the monitoring terminal at the computer installation) are described in the PRIMOS Computer Room User Guide, MAN2603.

PRIMOS implements a comprehensive file management system that is utilized transparently by all system software. PRIMOS file handling routines and primitives are also accessible at the user program level. The file system organization and facilities are described in detail in the PRIMOS File System User Guide, MAN2604.

Information in this manual is organized as follows:

- Section 1     Provides general information on PRIMOS, PRIMOS II relationships to other operating systems, and features of PRIMOS III and PRIMOS IV.
- Section 2     Is an illustrative overview of how to use PRIMOS as a user at an interactive terminal. This section is designed to provide a new user with enough information to use the PRIMOS system effectively.
- Section 3     Describes PRIMOS user commands. First, there is a review of the commands and functions. This is followed by a detailed description of commands normally used for programming development and production, arranged alphabetically.
- Section 4     Describes use of PRIMOS in multi-processor networks.

## FOREWORD

## APPENDICES

- Appendix A Summarizes the PRIMOS commands and their formats.
- Appendix B Describes the contents of ERRVEC and the system error vector, for both error and normal return.
- Appendix C Lists the PRIMOS error messages and value of the disk status word.
- Appendix D Lists the Prime ASCII character sets.

## RELATED PUBLICATIONS

The following Prime documents should be available for reference:

<u>Title</u>	<u>Manual No.</u>
Prime CPU System Reference Manual (instruction set, addressing modes, input/output programming)	MAN1671
Prime CPU Operator's Guide (Console and peripheral device operation)	MAN1672
Macro Assembler Language Reference Manual	MAN1673
FORTRAN IV Language Reference Manual	MAN1674
Program Development Software Manual (Editor, Loader, TAP, etc.)	MAN1879
Library Subroutine Manual	MAN1880

## FOREWORD

### SYMBOLS AND ABBREVIATIONS

Symbols and abbreviations and special characters used frequently in the rest of this handbook are defined below.

<u>Symbol</u>	<u>Meaning</u>
Number representations:	
1000	1000 decimal.
'1000	1000 octal.
\$1000	1000 hexadecimal.
Terminal functions:	
CR	Carriage Return.
LF	Line Feed.
\	Backslash (upper case L) used as tab character (Editors only).
"	Delete character (cancels last typed character); may be used in PRIMOS III and IV command lines (but not in PRIMOS II (use kill character).
?	Kill character (deletes all characters in current command line).
::	In the editor, ED, separates multiple commands on a line.
↑	Escape character (identifies octal codes of non-printing characters while using Text Editor).
Miscellaneous:	
SA	Starting address of program or memory block.
EA	Ending address of program or memory

## FOREWORD

block.

### ALLCAPS

A literal that must be included verbatim. Underlines indicate acceptable abbreviations.

### Initialcaps

A parameter to be selected by the user according to the accompanying text.

### [ ]

Brackets enclose optional parameters in command strings.

### | | | |

Vertical bars enclose a selection of entries of which one must be chosen.

### K

1024

### M

1,000,000

### Underlining

Indicates user input in examples.

Spaces (in command strings).

Blanks or space characters  
(in Hollerith or ASCII strings).

### Altrtn

Alternate return program step in case of I/O errors, missing EOF, etc.

### Ba

Buffer Address

### CPU

Central Processor Unit (the Prime computer proper as opposed to peripheral devices or main memory).

### DSKRAT

Disk Record Availability Table.

### Filename

A PRIMOS filename (in the in the current UFD, unless otherwise specified).

### Funit

PRIMOS File unit (1-16).

### Ldisk

Logical disk unit number, as assigned by STARTUP command.

### Lunit

Logical unit number, (1-15) as used in FORTRAN READ and WRITE statements. (Same as IOCS [Input/Output

## FOREWORD

	Control System] unit number.)
MFD	Master File Directory.
Password	A PRIMOS password.
Punit	IOCS Physical unit number (1-16).
UFD	User File Directory.
Ufd	A UFD name in a parameter string (for example, in a FORTRAN calling sequence: pointer to a UFD name in the form of a Hollerith expression or 3-word array).

### Filename Conventions

B<-XXXX	Binary (Object) file.
L<-XXXX	Listing file.
C<-XXXX	Command file.
XXXXXX	Source file.
*XXXXX	SAVED (Executable) file.

### NOTE

Throughout this document, the two character sequence <- in a filename represents only a single character. This character is the back-arrow on the terminal keyboard which prints as an underscore on output devices. Since this is inconvenient in text examples that are underlined, the "<-" convention has been adopted.



## SECTION 1

## INTRODUCTION

The current Prime computer operating systems are PRIMOS II, PRIMOS III, and PRIMOS IV. In this user guide, the generic term, PRIMOS, includes all three systems.

## SYSTEMS DEFINITIONS

PRIMOS II is the single-user Disk Operating System for the Prime family of computers. It is a memory-resident operating system that provides a complete working environment for the user's software development process and for user program development and production use of the various Prime disk options.

PRIMOS III has the same capabilities as PRIMOS II; in addition, this version of PRIMOS allows a sharing of the computer resources among a community of up to 31 simultaneous users and a variety of peripheral devices. PRIMOS III also gives each user a virtual memory.

PRIMOS IV, in addition to having all the capabilities of PRIMOS III, fully utilizes Prime 400 computer system architecture, including such features as: cache, 32-bit ALU, rapid control unit, a live register set consisting of 128 32-bit registers, and interleaved memory. PRIMOS IV allows the specification of a vast amount of virtual memory per user, access to a very sophisticated file system and more efficient execution of user programs.

## SYSTEMS CONFIGURATIONS

PRIMOS II may function in any of the possible Prime computer system configurations and allows direct memory addressing of up to 64K. It operates under control of a Prime 100, 200, 300, or 400 central processor with or without available options. A broad range of disks can be supported by PRIMOS II. All disk units are supported interchangeably by PRIMOS II and other Prime system software.

Up to four disk units, each with a capacity of 60 million bytes, can be attached to a disk controller Type 4001/4002, which handles up to four disk pack units as well as one fixed head disk and has a 256K or 512K byte capacity. Alternatively, mass storage (disk) configurations supported by PRIMOS II include moving head cartridge disks providing 3 - 6 and 12 million byte capacities, and high-speed fixed-head disks storing 256K, 512K, or 1024K bytes (using a Type 4001 Disk Controller). Diskette drives (floppy disks) are supported by PRIMOS II and PRIMOS III via a diskette controller; large disks (storage modules) of 40 million word capacity and up, are also supported by PRIMOS II.

PRIMOS II configurations may also include a high-speed paper tape reader. The PRIMOS II Supervisor Terminal can be any Teletype, a compatible terminal, or a CRT-type terminal attached either to a serial interface or to the system option controller, and running at 110 baud. Peripherals that are supported by IOCS running under PRIMOS II control include: up to four 7- or 9-track magnetic tape transports on one controller; a card reader (one per system); a character printer (one per system, connected to the system option controller); a line printer (one per system); and a paper tape reader/punch (one per system). For further details about PRIMOS II configuration, refer to the Computer Room User Guide (MAN 2603).

All disks and peripheral devices supported by PRIMOS II are supported by PRIMOS III.

The minimum configuration upon which PRIMOS II operates is a Prime computer with a Teletype for a Supervisor Terminal, 32K bytes of memory, and mass storage consisting of diskettes. PRIMOS II is upward-compatible, and operates on any sophisticated Prime computer system configuration.

The PRIMOS III operating system requires a Prime 300 system with a minimum of: 64K bytes of memory, disk, system terminal, and 1 to 31 user terminals on a Prime 300 configuration and 1 to 63 terminals on a Prime 400 configuration. PRIMOS III fully supports virtual memory and up to 256K of real memory. For details of the PRIMOS III and IV configurations, refer to the Computer Room User Guide (MAN 2603).

PRIMOS IV supports all disks and peripheral devices supported by PRIMOS II and III, and requires a Prime 400 configuration, to provide an address space of 0.5 billion bytes, an interleaving main memory system of up to eight million bytes with a 2K bipolar cache memory, and a disk capacity that can exceed 2.4 billion bytes. Currently, PRIMOS IV supports up to 512K bytes of main memory and provides each user with a 2-million byte virtual address space. Future versions will be extended toward 512N bytes per user.

## FEATURES

### PRIMOS II

PRIMOS II operates in several environments. Because the Prime RTOS (Real Time Operating System) and the PRIMOS III (Virtual Memory Operating System) are started from PRIMOS II, users of these systems must know how to start up PRIMOS II start up their systems from PRIMOS II and shut down PRIMOS II. Once RTOS or PRIMOS III is running, PRIMOS II can be run as a background job in RTOS, or PRIMOS II can be started up from PRIMOS III.

The fundamental unit with which most PRIMOS II commands and concepts

deal is the file. Each disk is organized into a system of files, thus permitting the user to reference programs and data by file name only. Consequently, there is no need for the user to identify specific physical records or to have knowledge of the format of the disk. An overview of files and the associated file system is provided in the File System User Guide (MAN 2604).

PRIMOS II provides an interactive command language for summoning programs and manipulating the file system. The command language interfaces the user to PRIMOS II by simple commands entered at the terminal. The same command functions may also be performed by programs, reducing the amount of operator involvement. Software written for stand-alone execution may be run under PRIMOS II with no changes. (See Section 3 of this document for command descriptions.)

All standard Prime software is available under PRIMOS II and makes use of its command structure and file-handling abilities.

### PRIMOS III

In addition to all the features of PRIMOS II, PRIMOS III gives the user, through the implementation of virtual memory via a demand-paging memory management technique, up to 65K words of virtual memory. The user can therefore write large programs or maintain extensive data bases without being limited by the amount of available physical memory. Because the PRIMOS III supervisor makes the necessary concerns of memory management transparent, the user need not be concerned with where PRIMOS is loaded in memory, the physical location of FORTRAN COMMON, or the possibility that user programs may overwrite vital parts of the system or other-user programs. Through the use of attachment and assignment techniques, all users may take advantage of the computer's resources and can share all files on the system. Sharing is controlled by the PRIMOS III supervisor, so that all users obtain their fair share of the computer's resources. Also, user-implemented password and file protection attributes allow the user to control the manner in which user-owned files are to be shared.

### PRIMOS IV

Because PRIMOS IV allows a user access to all the features of PRIMOS III, any command or user program that previously ran under PRIMOS III can run under PRIMOS IV. In addition, PRIMOS IV is an embedded operating system. Furthermore, PRIMOS IV enables users to implement the advantages of the Prime 400, including new instructions and facilities for a 16-segment program, where each segment is 64K words (128K bytes) long. These large programs may be written in FORTRAN, COBOL, or PMA, which are available on all versions of PRIMOS.



## SECTION 2

## USING PRIMOS

## INTRODUCTION

The PRIMOS operating system enables users to execute their programs. Users interact with the computer through remote input/output terminals. PRIMOS II is a single user disk operating system.

PRIMOS III and IV operating systems allow up to 31 users to share a Prime computer configuration. They provide a virtual memory environment, and regulate and coordinate the activities of the users and system resources. Each system occupies its own virtual memory space. PRIMOS III provides each active user with up to 130K of virtual memory space. PRIMOS IV provides each user with up to 4 million bytes of virtual address space. PRIMOS III and IV enable user (and supervisory) programs to be paged in and out of real memory and perform necessary input/output functions on an as-needed basis. The individual user at a terminal need concern himself only with the running of his commands and programs.

The purpose of this section is to acquaint the user with the process of interacting with the computer from a multi-user terminal while operating under PRIMOS. A subset of PRIMOS commands is described in simplified form, and many of the options are omitted. For additional information, consult the user guides listed in the Foreword under the heading RELATED DOCUMENTS.

This section is primarily intended for new users - either those who are experienced in programming but have little knowledge of Prime's computer systems, or those who have no programming experience but need to use a terminal. It does not discuss all the features of PRIMOS, nor does it describe in full detail those features that are discussed. Complete familiarity with the computer and PRIMOS system will come with experience and thorough familiarity with the documents listed in the bibliography. The user should be able to read this section once, and seldom refer to it again. This section provides simplified information enabling a new user to get started; and also contains a collection of reference material to help the new user learn more about PRIMOS.

## SUBSET OF COMMANDS AND SAMPLE USER SESSIONS

## REMINDER

In the examples, user input is underscored and system response is not underscored. Also, all command lines are terminated by a CARRIAGE RETURN. This CARRIAGE RETURN is not shown in the examples, but the user should be aware that it is implied in the sample command lines.

## PREPARING FOR PRIMOS USAGE

If you are a new user of Prime's PRIMOS operating system, your first impression of the system will be the user terminal and its keyboard. Figure 2-1 is an illustration of a typical terminal keyboard. In the next paragraph, the character set and some characters that have special meaning to the system are discussed. However, for now, it is necessary to confront the problem of how to get started using the system. First, the user must be sure to turn on the terminal power and ensure that the terminal is working properly. The terminal should echo characters when they are typed by the user.

Terminal

The user issues commands by typing them on a terminal connected in the system configuration. To input a command, the user is generally concerned with the typewriter keyboard at the terminal. There is no need to list all the available characters on the keyboard; the user will learn them soon enough. However, the characters (keys) listed in Table 2-1 have special significance:

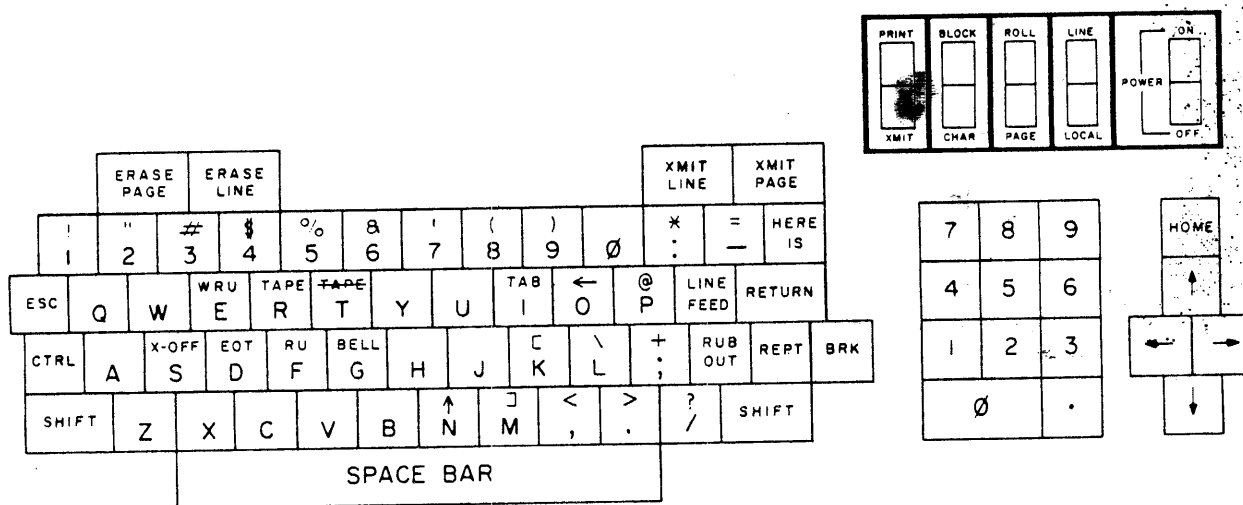


Figure 2-1. Typical Terminal Keyboard

Table 2-1. Terminal Keys with Special Significance

<u>Key</u>	<u>Significance</u>
[RETURN]	Used to signify end of line. This indicates to PRIMOS either the end of a command line or the end of the line input to a file.
[CTL]	Used simultaneously with the P-key to QUIT from a command or program execution and return to system level. The user's terminal-input buffer is also cleared.
[SPACE BAR]	Used in typing spaces.
ATTN [INTRPT] or BREAK	Used to interrupt or terminate execution of a command (i.e., equivalent of CTL-P).
\	TAB (when in Editor INPUT mode). For convenience in writing FORTRAN programs, tabs are initially set by the system.  The TABSET command, available when interacting under control of the Editor, allows the user to change the TAB settings.
?	Kill character. Deletes all characters on the line typed previously up to and including the ?.
"	Erase character. Erases the character typed before the ". If a consecutive number (n) of erase characters is typed, then that number of characters (i.e., n-characters) preceding the first " is erased.

## ACCESS TO SYSTEM

To use PRIMOS III and IV, the user must first gain access to it; that is, he must first log in. Generally, the user accomplishes this by turning on his terminal and typing the appropriate form of the LOGIN command. If a user is at a terminal location that is remote from the computer site; he must first dial up the system to connect the terminal to the computer via an acoustic coupler or other modem. The PRIMOS III & IV system either accepts or rejects the attempt to log in, depending on the identity of the user and the access rights of the user. Through judicious use of passwords, the user can control his own access and the access of other users to his files and directories. Refer to the File System User Guide (MAN 2604) and to Section 3 for a discussion of access protection.



## BEGINNING A SESSION AT THE TERMINAL

The user begins each session by typing the LOGIN command. Logging-in is accomplished by simply typing:

LOGIN Ufdnam (enter with a carriage return)

where: Ufdnam is the name of the user login UFD. (It is assumed that user already knows the name of a UFD to log into. If not, he should check with a system administrator or other person familiar with the system. It is also assumed that the user knows that each command line must be terminated by a carriage return).

Sometimes, if the user has made a mistake in logging-in, the system will reply with the error message:

LOGIN PLEASE

or the message:

Ufdnam NOT FOUND

In either case, the user must try to log in again, being careful that the LOGIN command is specified with the correctly spelled UFD name.

A typical LOGIN sequence is shown in the following example.

Example:

```
LOGIN JDOAKS
JDOAKS (2) LOGGED IN AT 13'18 02197
OK,
```

In this example, JDOAKS is the name of a UFD (User File Directory). This UFD is contained in the MFD (Master File Directory). The MFD is discussed in more detail later in this section.

## Commands

The user at an input/output terminal issues instructions, called commands, to the system by typing the name of the command and any arguments associated with the command. The PRIMOS system then loads and executes any program or programs that are required to perform the command's functions.

## Files

Most often, the arguments of commands are names of files. A file is a logical set of information. This information may represent a source program, an object program, a set of data, a program listing, text of an on-line document, or anything for the user to define and express in the available symbols. File contents may be typed and input to the

computer system from the user's terminal keyboard. Files are normally stored on the disks attached to the computer configuration. On some configurations, files may also be stored on magnetic tape for backup or for archiving. No detailed knowledge of the physical location of a file is required because the user, through PRIMOS commands, refers to files by name. PRIMOS programming features maintain a separate user file directory (UFD) for each possible user to avoid conflicts that might arise in assignment of filenames. A master file directory (MFD) is maintained by PRIMOS for each disk connected to the system configuration. The MFD contains information about the location of each User File Directory (UFD) on the disk. In turn, each UFD contains information about the location and content of each file that is filed within the scope of that directory. For a primer on the file system, a description of the PRIMOS file systems and a description of the ordering of information within files, refer to the File System User Guide (MAN 2604).

#### SYSTEM RESPONSE AND COMMAND FORMAT

When the PRIMOS III and IV systems receive and complete execution of a command, they acknowledge that the command ran to completion by printing:

OK,

(PRIMOS II acknowledges command completion by typing OK: (OK followed by a colon). This is one way of telling which operating system has control.)

The OK, message signifies that PRIMOS III and IV are ready for another command.

If an error occurs, the system generally types a message and then types the response:

ER!

Each individual command must be terminated by a CARRIAGE RETURN. In all examples in this document, the presence of a CARRIAGE RETURN character terminating a command is assumed, and not shown.

OK, DELETE MYFILE  
OK,

#### Spaces in Commands

Spaces in a command line are significant. A space character is used to separate a command from its arguments and to separate one argument from another.

Example:

OK, DEL ETE MYFILE

In the original delete command, the space in the middle of the word DELETE was incorrect and caused an error message.

```
BAD DEL
ER! DELETE MYFILE
```

OK,

### Characters per Word

If a portion of a command line delimited by spaces (i.e., a word) is longer than six characters, PRIMOS III and IV truncate the word to six characters.

Example:

User types     CREATE THEREISNOSUCHFILE

Response       OK,

User types     LISTF (to list file names)

Typical  
Response       XXXXXX YYYYYY THEREI  
                 OK,

Notice the name THEREI is truncated and is the name that has been accepted by PRIMOS.

### Number of Words per Command

The format of a command line is:

```
COMMAND ASCII1 ASCII2 1 2 3 4 5 6 7 8 9 10 11
```

where:

COMMAND     Is the command name  
              (for example, ATTACH), an ASCII  
              string of up to six characters.

ASCII1       Is generally the name of  
              a file or UFD; sometimes  
              it is a character with special  
              meaning (such as \*).

ASCII2       Is also an ASCII string of  
              up to six characters; often  
              is a password.

1,2...11     Are 16-bit parameters.  
              These are often used as keys.

Erase and Kill Characters

The double quote (") character is interpreted as a single character eraser. Thus N consecutive quote characters erase the previously typed N characters on a command line back to, but not including, the previous CARRIAGE RETURN. Spaces are also recognized as characters; therefore, the quote characters may be used to erase on to N spaces.

Examples (while typing in program):

```
ABCD""C = DE F""F
```

results in the line:

```
ABC = DEF
```

The question mark (?) is the kill character. When this character is typed, the entire line back to, but not including, the previous CARRIAGE RETURN is deleted.

```
P = LOC(B) ?P = LOC(B)
```

The resulting line is:

```
P = LOC(B)
```

Example (Error in Command Line):

```
LISS"TF
```

is interpreted as LISTF

```
LIS?SLIST
```

is interpreted as SLIST.

Error Messages

If the user commits a typographical error when using a command, or if the system cannot execute a command for some reason, the system may print a diagnostic message at the terminal.

Examples:

```
QDELET NOT FOUND  
ER!
```

```
FILEAA NOT A UFD  
ER!
```

Also, subsystems such as FORTRAN or a loader (LOAD) may print

diagnostics. Usually, these diagnostics consist of a two character code for some error condition; however, other types of diagnostics may also be constructed and printed by the subsystems.

Examples:

```
CM,
SS,
NO ERRORS:
MTU ERROR
<STATUS>
```

PRIMOS IV error messages are processed via a standard error handler and give more detailed information than PRIMOS II or III messages.

## COMMANDS FOR MANIPULATING FILES AND DIRECTORIES

Arguments to the commands described in the following paragraphs are often filenames. Filenames specify the name of a file or of a directory. Filenames consist of one to six characters. A filename may contain any printable character except " (ERASE) and ? (KILL); p3 a filename must start with a non-numeric character. It is strongly recommended that the only special characters used to start a filename be the characters \* (asterisk), <- (left arrow), - (hyphen), and # (number sign), because other special characters are not allowed in new style partitions of the file system at a later revision. An example of a filename is PROG1.

### Creating a File

The Editor command to invoke the system Editor, ED, is one way to create a file, and it is the way that is most likely to be used by the new user. The new PRIMOS user will probably be writing an ASCII file (such as a source program file). The Prime character set is described in Appendix D.

All of these characters described in Appendix D appear on the user terminal keyboard. The fundamental requirements for creating a file using the System Editor are:

INPUT lines of text

LOCATE a specific item (string) within existing text

INSERT or DELETE lines of text

CHANGE character strings within a line of text

FILE away edited text as a file in the system.

The following simple examples show how to use Editor to accomplish the

above functions.

Example: Creating a File:@Notes

```
OK, ED
GO
INPUT
THIS IS A NEW FILE CREATED AS AN EXAMPLE
LINE 2 = DUMMY LINE FOR EXAMPLE.
LINE 3 = DUMMY LINE FOR EXAMPLE.
```

```
;
EDIT
FILE NEWFIL
```

When the command, ED, is given, Editor begins in Input Mode. User types lines in the file.

Typing a line with just a semicolon or CR switches Editor Mode, in this case to Edit Mode.

```
OK, LISTF
```

```
UFD=JDOAKS
```

```
NEWFIL
```

```
OK,
```

The Editor's FILE command files the edited text in the

filename specified. In this case, NEWFIL. Also, FILE command causes a return from the Editor to PRIMOS III command level.

The LISTF command shows that the file NEWFIL has been created in the Editor and is an entry (named file) residing in the user's current UFD.

### THE CONCEPT OF ATTACHMENT

The next question that might occur to a user, once he is successfully logged in, is just what it is that he is logged into. Obviously, the previous paragraph on LOGIN has informed the user that he is logged into the system and resides within the confines of something called the User File Directory (UFD). A UFD is a directory (catalog) that contains entries that point to files or to other UFD's (subUFD's). Both files and UFD's contained within any directory are specified by mnemonic names. A complete discussion of files and directories is given in the File System User Guide (MAN 2604). The user must be aware that files and directories are arranged in a hierarchy generally referred to as the file system. This hierarchy is a tree structure,

and it is made up of branches that are directories or subdirectories and nodes that are files. Subsequent listing (LISTF) and attaching (ATTACH) examples in this section show how files and directories are related in the file system hierarchy (tree structure). To find out what files or directories are cataloged within the current UFD, the LISTF command may be invoked. The following paragraph gives a simple example of LISTF.

#### Listing Files in a UFD

When the user logs in, the UFD name that is specified becomes the current UFD. It is used as a working directory for the manipulation of files and any subordinate directories (referred to as subdirectories or subUFD's). File manipulation includes operations such as CREATION, DELETION, READING, WRITING, COPYING, MODIFYING, and LISTING. The LISTF command can be used to see what files exist in the current UFD (or any UFD).

Example:

```
OK, LISTF
UFD=JDOAKS  0  0
SHELP  L_SHEL  B_SHEL  *HELP
OK,
```

This LISTF example shows that the UFD named JDOAKS has four files contained within it: SHELP, L\_SHEL, B\_SHEL, and \*HELP.

#### The Concept of ATTACHing to a UFD

Another definition of the current UFD is that UFD to which the user is currently ATTACHED. If the user wants to go to another UFD and make that newly referenced UFD his working directory, it is necessary to use the ATTACH command.

Example:

Assume the current UFD is JDOAKS. Suppose the user wants to access the UFD named CMDNC0 to see what files are contained in that directory. The following underlined example command lines are typed into the system by the user:

```
OK, ATTACH CMDNC0
OK, LISTF
```

```
UFD=CMDNC0 0 N
```

```
FILMEM FILBLK OSORT  MSG      NUMBER  TAP      RT128F  BASINP
SIZE   LOADAP FILVER CMPRES  EXPAND  PSD20   FILCPY  UFDPCY
```

PMA	DOSEXT	PTBOOT	CRSER	CRMPC	PRMPC	PRSER	DBASIC
MTDSK	FLG	LOAD7	FTN	MDL	CPUT2	HSMT1	PUSS
HILOAD	DATE	RUN	MAG	SCAN	MAKE	FIXRAT	OCOPY
AVAIL	COUNT	BASIC	PRLST	VDOS32	NFTN	SPOOL	PSD
HPSD	EDB	HELP	SLIST	LYT	FUTIL	DIGIT	SORT
PTCPY	TAP56	OLDED	LOGIN	MAGSAV	LATE	MAGRST	COPY
PRIMOS	LOAD	CMD	CLEAN	ED	P2	P3	GRINDE

## NOTE:

CMDNC0 on each disk in every PRIMOS configuration is a special UFD. It contains memory image or run files of every command available for that configuration. This set of commands may be system-supplied or user modified.

Attaching to The Master File Directory

At this stage of discussion of the ATTACH command, it is useful to consider another special kind of directory, the MFD (Master File Directory). The MFD is the root of the tree hierarchy on each disk or partition of a disk configured to the PRIMOS system (see Figure 2-2). There is an MFD for each disk device or partition, which contains some essential files for system operation such as DSKRAT, BOOT, DOS, and CMDNC0. The MFD also contains UFD's. These UFD's generally have names that the user uses as LOGIN-names with the LOGIN command. To attach to the MFD, the user must know a correct password.

Example:

ATTACH MFD XXXXXX 3

In this ATTACH example, the user ATTACHes to the MFD on the disk that has the logical disk number 3; the password specified is XXXXXX. (The logical disk number may be discovered by use of the STATUS command. This command is described later.) For curiosity's sake, the user should try the following command sequence:

ATTACH MFD XXXXXX  
LISTF

Attaching to a UFD on Another Disk

If the name of the UFD that is specified in the ATTACH command (target UFD) is unique, the function of the ATTACH command is independent of the UFD-resident disk. If the UFD name of the target UFD is the same as the name of some other UFD in the file system hierarchy, then the user must specify the logical disk number in order to attach to the specific UFD desired.

To attach to a UFD residing in another disk, first establish which disks are configured to the system; and use the STATUS command.

Example:



OK, STATUS

USER=JDOAKS

FUNITS

DISK	LDEV	PDEV
TS	0	250
SPOOLD	1	252
ETCH	2	40250
PMFII	3	100250
ADMIN	4	110250
PRIMOS	5	60250
DUD	6	20250

OK,

In the above example of the STATUS command, it can readily be seen that the logical disk number is given for each named disk in the column labeled LDEV. The column named PDEV is the physical disk or partition number. (PDEV need not be of concern to the user at a terminal at this time.)

In addition to the filename argument, a password (if any) and a logical disk number may be specified.

Example:

OK, A JDOAKS US 6

ATTACHes to the UFD named JDOAKS that has an associated password US and the disk that was found to be logical 6 when the STATUS command was invoked. See Figure 2-3 for a graphic illustration of this process.

### ATTACHing to a Subdirectory

For the purposes of this example, assume that there is a series of subdirectories (subUFD's) subordinate to the UFD named JDOAKS. To attach to a subUFD that is several levels down the tree hierarchy from the UFD JDOAKS, it is necessary to perform a series of individual ATTACH commands in succession. Figure 2-4 also shows the series of attachments needed to work one's way down the tree structure to the UFD CELTIC from the UFD JDOAKS. These steps are:

OK, A JDOAKS  
 OK, A JOE 1/2  
 OK, A JOESUB 1/2  
 OK, A CELTIC 1/2  
 OK,

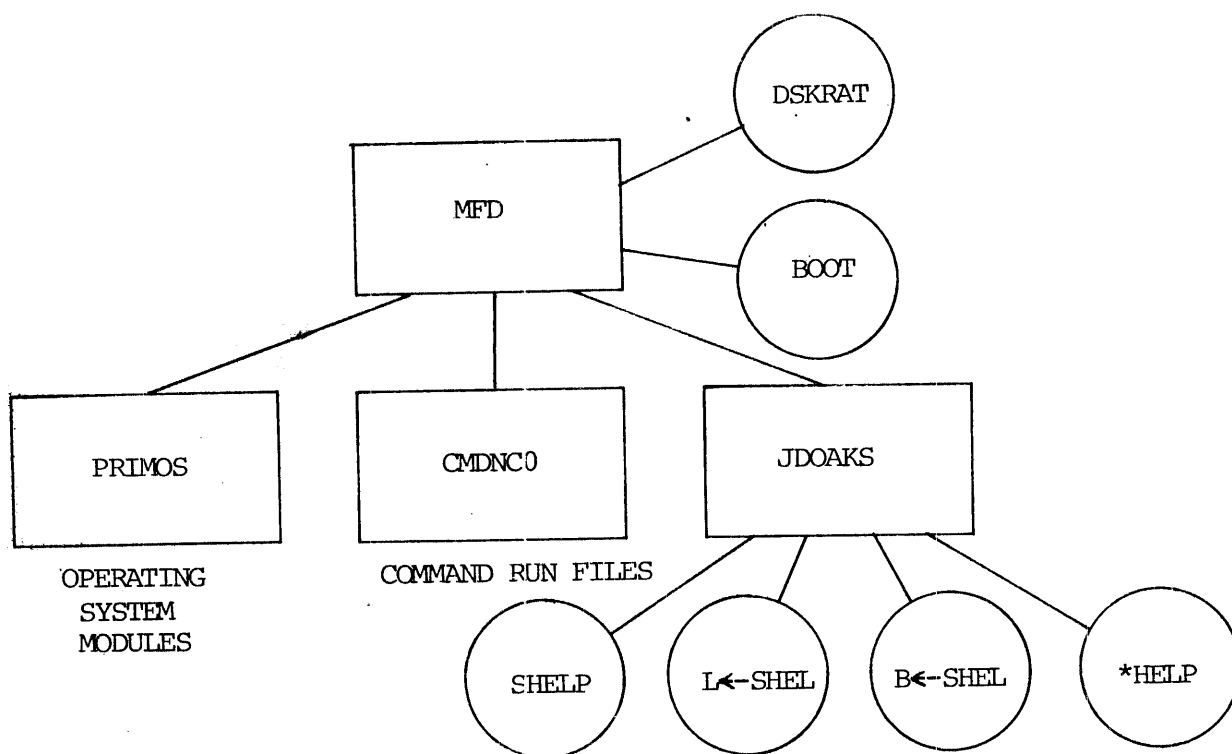


Figure 2-2. Typical portion of PRIMOS File System, Showing: MFD, Command UFD, and User UFD.

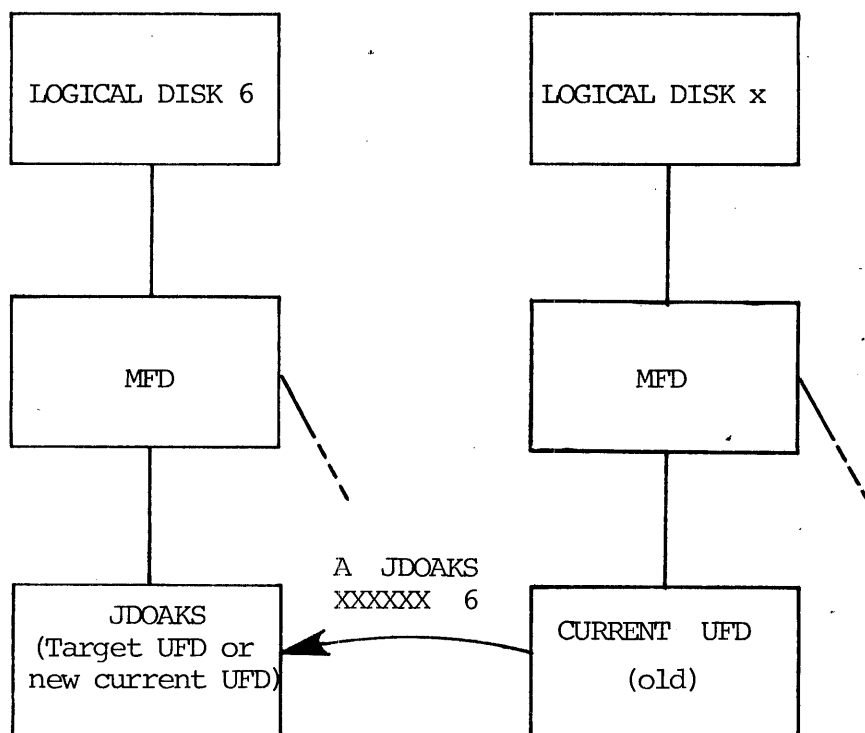


Figure 2-3. Attaching to UFD on Another Disk.

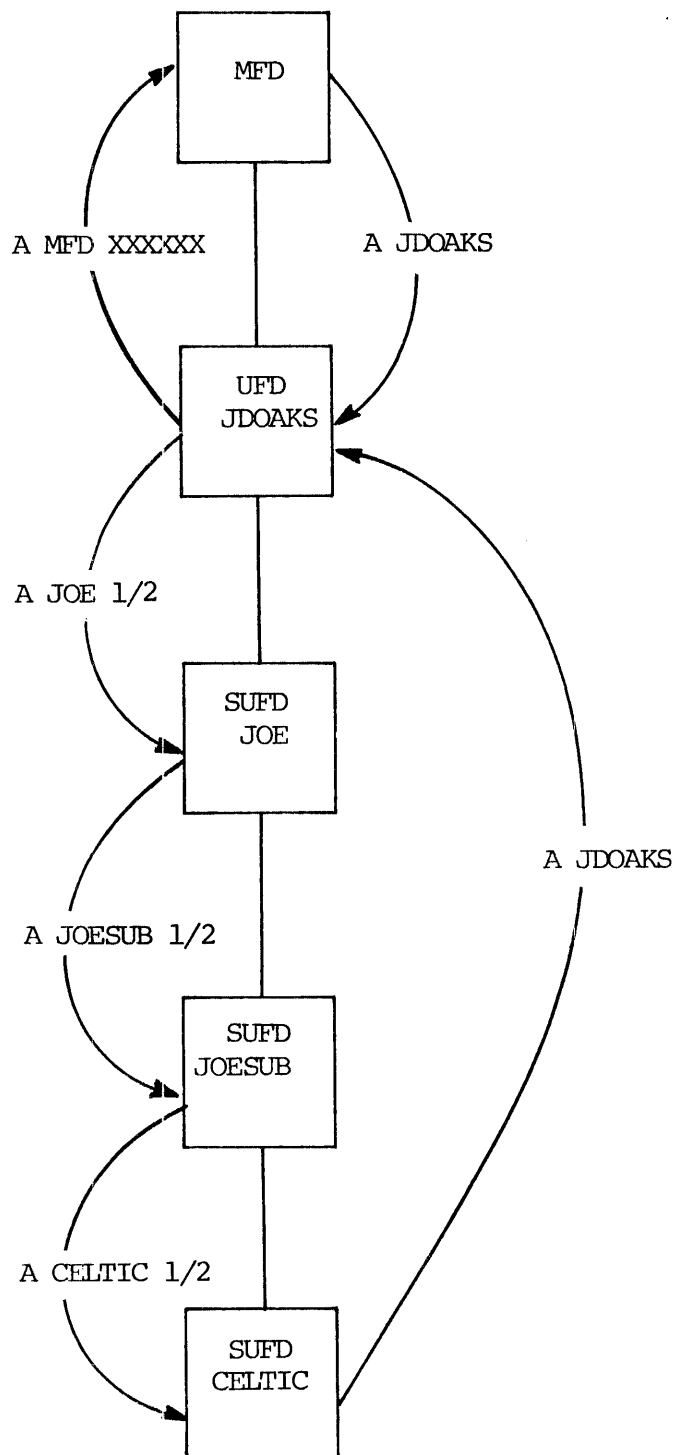


Figure 2-4. Attaching Subdirectories and Returning to User Level Directory (JDOAKS) .

It is important to note that in directories subordinate to the UFD JDOAKS, which is at the level in the tree that is just subordinate to the MFD, it is necessary to specify an additional parameter to the ATTACH command. This parameter is a key that sets the A register in such a way that the user's home UFD is changed to the UFD ATTACHED to. The specification of some key is necessary in all cases of ATTACHing to a subUFD. It is optional in the case of the MFD or the UFD'S immediately subordinate to the MFD. For complete details about keys in the ATTACH command, refer to the description of the ATTACH command in Section 3 and to the File System User Guide (MAN 2604).

#### What Happens if You Make an Error While ATTACHing

Making an error when issuing the ATTACH command may lead the user to some confusion. If an error message is returned when an ATTACH command is given (for example, if a password is given), then the user is in a state where he is not ATTACHED to any UFD. In fact, if a command such as a LISTF is input while in this state, the message:

NO UFD ATTACHED

is returned. To get out of this state, the user must issue a command to ATTACH to some UFD immediately subordinate to the MFD, such as:

A JDOAKS

If one were attempting to ATTACH a subUFD at the time of the error, it would be necessary to again work one's way down the tree through a series of ATTACHments, as shown in the example in Figure 2-4.

In the case of a UFD NOT FOUND message, the user is left ATTACHED to the directory that he was ATTACHED to before issuing the ATTACH command that failed.

#### FURTHER FILE AND DIRECTORY MANIPULATIONS

##### Deleting A File

To delete a file, simply type the DELETE command.

Example:

OK, LISTF

UFD=JDOAKS 6 0

TIMNGF	FUNCTF	OCTALF	SFTNDX	PROGS	INPUT	MNEMOSNAMES
ALGEBR	C	ARRS	MAGTST	NAMES1	MI2	

OK, DELETE MI2

OK, LISTF

UFD=JDOAKS 6 0

TIMNGF	FUNCTFG	OCTALF	SFTNDX	PROGS	INPUT	MNEMOSNAMES
ALGEBR	C	ARRS	MAGTST	NAMESI		

### Renaming a File

To change the name of a file, use the CNAME command. The CNAME command takes two arguments. The first is the filename as it currently exists; the second is the new name that is to be given to the file.

Example:

OK, CNAME ARRS ARRAYS  
OK, LISTF

UFD=JDOAKS 6 0

TIMNGF	FUNCTF	OCTALF	SFTNDX	PROGS	INPUT	MNEMOSNAMES
ALGEBR	C	ARRAYS	MAGTST	NAMESI		

### Executing a Series of Commands from a File

A user frequently uses a series of commands to do a repetitive job. He may use the Editor to create a file that consists of this frequently used series of command lines, one command per line of text, and can run this command file by issuing one command, COMINP.

Example:

The contents of the command file COMMDT are as follows:

```

USERS
*TELLS.NUMBER.OF.USERS.LOGGED.IN

AVAIL SYSTM
*GIVE.NUMBER.OF.RECORDS.LEFT.ON.DISK.NOTE

*NAME.OF.DISK.NOT.NUMBER.WAS.SPECIFIED

SIZE ZZZ
*GIVE.SIZE.OF.SPECIFIED.FILE

COMINPUT TTY

```

The following shows the result of using this file in the COMINPUT command:

OK, <u>COMINP COMMDT</u>	Invokes COMINP to run the command file COMMDT
OK, USERS	As command lines in the file are processed, COMINP prints each one
USERS= 5	USERS= 13 result of USERS command
OK, * TELLS.NUMBER.OF.USERS.LOGGED.IN ds.	Comment lines, note use of periods to separate words.  Note also that format of comment lines is * followed by a space.
OK, AVAIL SYSTM GO 77 RECORDS AVAILABLE 98.8	
OK, * GIVES.NUMBER.OF.RECORDS.LEFT.ON.DISK.NOTE OK, * NAME.OF.DISK.NOT.NUMBER.WAS.SPECIFIED OK, SIZE 777 GO 30 RECORDS IN SAMFIL	
OK, *GIVES.SIZE.OF.SPECIFIED.FILE OK, COMINPUT TTY OK,	

### Printing a File

A file may be printed at the terminal by using either the SLIST command or the Editor.

Example:

OK, <u>SLIST PROGS</u> GO 10 DEFINE FILE #1='INPUT' 11 S\$= 20 DEFINE FILE #2='OUTPUT' 21 WRITE #2,'NAME','UPD','VOL','FUNCT' 22 WRITE #2, S\$ 23 WRITE #2, S\$ 30 READ #1,A\$,B\$,D\$,C\$,E\$ 40 WRITE #2,B\$,D\$,C\$,E\$ 60 GOTO 30 99 END RUN QUIT	BASIC Language Program         BASIC commands
--	---

in contents of the file.

OK,

A file may be printed off-line on the system line printer, if available, by use of the SPOOL command.

Example:

```
OK, SPOOL L<-PROG
GO
YOUR SPOOL FILE IS PRNT10
OK,
```

#### NOTE

In the above example, the two characters "<-" are an editorial representation of the back arrow or underscore on the terminal keyboard. This convention is used throughout this document.

SPOOL creates a copy of the user's file in the UFD SPOOL with the name PRNTXX (where XX is a sequential number). Thus, the user does not have to wait for the file to be printed before issuing another command, and he can pick up a copy of the file at the system printer after his terminal session is finished. Optionally, the user can delete the copy of a file that has been requested to be printed by the SPOOL command, since a copy is retained in the UFD, SPOOL, until the copy is printed.

#### Printing a File with PRIMOS II

Users that wish to print a file on PRIMOS II off line may do so by using the PRSER or PRMPC commands, depending on whether the printer configured to the system is a serial line printer or a parallel interface line printer, respectively. These commands are also available on PRIMOS III and IV, but are not generally used, since it is necessary to ASSIGN the printer to the individual user at the expense of other users on the system. For details of the PRSER, PRMPC, and ASSIGN commands, refer to Section 3.

#### PROGRAM DEVELOPMENT UNDER PRIMOS II OR III

The following example is an annotated example of how to create a source program, compile the object from the source, load the object and execute the object program. Errors were intentionally introduced so that the user could be shown some of the mechanics of the process of program development; user input is underscored.



```

OK: A MIKE [1]
OK: LISTF

UFD=MIKE 0 [2]

OK: ED
GO [3]
INPUT
C\DUMMT"Y FORTRAN PROGRAM EXAMPLE [4]
\ DIMENSION BUFF(32,32)
\ COMMON BUFF
A=?\ A = 3.
\ B = 4
\ C = SQRT(A**2 + B**2
\ WRITE (1,1000) A,B,C
1000\ FORMAT(.HYPOT. OF TRIANGLE WITH SIDES 'F8.4,'AND'F8.4,'IS 'F8.4)
\ CALL EXTI

\ END
i [5]
EDIT
T,L = 4,P
B = 4
C4/4. [6]
BAD C [7]
C/4/4./,P [8]
B = 4. [9]
T,P20 [10]
.NULL.

C DUMMY FORTRAN PROGRAM EXAMPLE
  DIMENSION BUFF(32,32)
  COMMON BUFF
  A = 3.
  B = 4.
  C = SQRT(A** + B**2
  WRITE (1,1000) A,B,C
1000 FORMAT('HYPOT. OF TRIANGLE WITH SIDES 'F8.4,'AND'F8.4,'IS'F8.4)
  CALL EXTI
  END

BOTTOM
FILE TEST [11]

OK: FTN TEST [12]
GO
(0006) C = SQRT(A** + B**2)
**** LINE 0006 [ SQRT(A**+B ] ILL. UNARY OP USAGE
(0007) WRITE (1,1000) A,B,C
(0008) 1000 FORMAT('HYPOT. OF TRIANGLE WITH SIDES 'F8.4,'AND'F8.4,'IS
'F8.4)
**** LINE 0008 [ 4,'IS'F8.4 ] SYNTAX ERROR

OK: ED TEST

```

## MAN2602

2 - 22

OK: LOAD

GO

\$ LOAD TEST?LO B TEST

[25]

\$ LIB

LC

[26]

\$ MAP

\*START 001000 \*LOW 000074 \*HIGH 010545 \*PBRK 010546

\*CMLow 057752 \*CMHGH 063752 \*SYM 057216 \*UII 000005  
[27]

\*BASE 000200 000352 000771 000777

LIST	000001	SQRT	001127	E\$21	001221	F\$WN	001273
F\$WNX	001302	F\$10	001360	F\$A1	001714	F\$A3	001714
F\$A2	001720	F\$A5	001720	F\$A6	001725	F\$IORS	002201
F\$CB	002202	F\$FLEX	004155	F\$ER	004323	F\$HT	004330
AD1	004410	AC2	004411	AC3	004412	AC4	004413
AC5	004414	RDASC	004415	RDBIN	004421	WRASC	004425
WRBIN	004431	CONTRL	004532	ATTDEV	004603	SETIOS	004632
RATBL	004713	RBTBL	004723	WATBL	004733	WBTBL	004743
CNTBL	004753	LUTBL	004763	PUTBL	004774	I\$DASC	005005
I\$DVMS	005153	O\$DASC	005155	O\$DBIN	005346	I\$DBIN	005415
O\$LASC	005462	I\$CASC	006247	IDCASC	006247	I\$AASC	006545
I\$PASC	006553	O\$AASC	006714	O\$PASC	006720	I\$ABIN	007004
I\$PBIN	007025	O\$ABIN	007336	O\$PBIN	007350	READ	007624
WRITE	007655	SEARCH	007706	EXIT	007711	OPSCHK	007716
PUTC	007747	TLIN	010005	TLOU	010076	TONL	010112
TNOUA	010116	TNOU	010133	TOOCT	010067	PLIN	010223
PLOU	010244	C\$P	010262	C\$A	010356	PL0B	010456
PLIB	010462	TLIB	010467	TLOB	010474	F\$AT	010501
F\$AT1	010503		057752				

\$ LIB UII

[28]

LC

\$ SA \*TEST

[29]

\$ EX

[30]

HYPOTENUSE OF TRIANGLE WITH SIDES 3.0000 AND 4.0000 IS 5.0000

OK: R \*TEST

[31]

GO

HYPOTENUSE OF TRIANGLE WITH SIDES 3.0000 AND 4.0000 IS 5.0000

OK: PM

[32]

SA,EA,P,A,B,X,K=

000066 012252 001107 120240 006726 000000 006203

OK:

## NOTES FOR COMPILE AND RUN EXAMPLE

- [1] Attach to UFD Mike for program development.
- [2] Note LISTF of 'empty' UFD; now let's enter and create a file.
- [3] In response to ED, PRIMOS loads editor and puts user in INPUT mode. Anything typed is stored in editor's buffer as text.
- [4] We enter trivial FORTRAN example. Backslash (shift L) is tab character. " erases last character, ? kills line up to that point.
- [5] Typing semicolon is one way to switch from EDIT to INPUT mode.
- [6] To put a decimal point after the 4.
- [7] Incorrectly specified delimiting /.
- [8] Change made correctly.
- [9] Confirmed by print.
- [10] To print entire program for cursory inspection.
- [11] Looks OK. Let's file it and try compiling it.
- [12] Errors.
- [13] Forgot to close the parenthesis after the 2.
- [14] After the editor's verify mode, N command caused next line to be printed automatically.
- [15] Second try compiled OK. Let's load it. FILMEM loads unoccupied memory with zeroes; it is useful for making an MDL tape after loading the program, and for starting in a known state. B\_TEST is binary file generated by compiler.
- [16] \$ is LOAD prompt.
- [17] Load the program B TEST.
- [18] Loads library.
- [19] No load complete (LC) message; try a load map.
- [20] Load UII.
- [21] Check to see if all subroutines are loaded.
- [22] Specified non-existent subroutine.

[23] Leave loader.

[24] No need to specify name if ED was invoked with a Filename argument.

[25] To load B<-TEST. This line shows use of ? to cancel incorrect command.

[26] LC, Load Complete, this means all external references are satisfied. Check this time by making a load map.

[27] Indicates UII if not 0. Note, only a MAP 1 was required to find this out.

[28] Load UII package.

[29] Use of loader SAVE command to save \*TEST.

[30] To execute \*TEST.

[31] Resume \*TEST for the fun of it.

[32] Examples of a few more commands; e.g., PM.

For a detailed discussion of FUTIL, refer to the File System User Guide (MAN 2604).

#### PROGRAM DEVELOPMENT UNDER PRIMOS IV

The following examples illustrate the use of SEG and FTN under PRIMOS IV. In this example, the program TEST, written in FORTRAN, is compiled, loaded, and executed.

The source text of the program TEST is:

```

      WRITE(1,10)
10   FORMAT('HELLO THERE')
      CALL EXIT
      END

OK, FTN TEST 2/400 [1]
GO
0000 ERRORS (FTN-1082.L13)
OK, SEG [2]
GO
#LOAD
SAVE FILE TREE NAME: *TEST [3]
$ LOAD B<-TEST [4]
$ LIB [5]
LC

```

MAP [6]  
 START 004002 000001 \*STACK 004017 000001 \*SYM 016016

SEG. #	TYPE	LOW	HIGH	TOP
004001	PROC	000200	010265	010265
004002	DATA	000000	001774	001774

\*BASE 004001 000200 000242 000777 000777

ROUTINE	EDB	PROCEDURE	ST. SIZE	LINK FR.		
####	4002	000001	4001	001000	000012	177400
F\$WB	4002	000415	4001	001056	000056	177427
F\$RB	4002	000355	4001	001061	000056	177427
F\$DE	4002	000435	4001	001067	000056	177427
F\$EN	4002	000455	4001	001072	000056	177427
F\$WA	4002	000375	4001	001112	000056	177427
F\$RA	4002	000335	4001	001115	000056	177427
F\$A1	4002	000475	4001	001476	000056	177427
F\$A2	4002	000515	4001	001501	000056	177427
F\$A6	4002	000535	4001	001504	000056	177427
F\$A7	4002	000555	4001	001507	000056	177427
F\$CB	4002	000575	4001	001735	000056	177427
F\$ERX	4002	000643	4001	004612	000020	000243
RDASC	4002	000673	4001	004640	000026	000273
RDBIN	4002	000713	4001	004705	000026	000273
WRASC	4002	000733	4001	004725	000026	000273
WRBIN	4002	000753	4001	004745	000026	000273
IOCS\$	4002	001001	4001	004773	000040	000401
IOCS\$T	4002	001151	4001	005103	000040	000425
ATTDEV	4002	001171	4001	005155	000040	000425
I\$AD07	4002	001233	4001	005615	000026	000633
O\$AD07	4002	001255	4001	006067	000026	000655
I\$BD07	4002	001277	4001	006341	000036	000677
O\$BD07	4002	001321	4001	006457	000034	000721
O\$AD08	4002	001343	4002	006556	000034	000743
I\$AA12	4002	001365	4001	006650	000052	000765
CNIN\$	4002	001411	4001	007073	000024	001011
O\$AA01	4002	001433	4001	007303	000030	001033
F\$IOER	4002	001457	4001	007337	000014	001057
TNOU	4002	001505	4001	007412	000020	001105
TNOUA	4002	001531	4001	007424	000020	001131
TONL	4002	001551	4001	007561	000012	001151
TLOU	4002	001573	4001	007567	000016	001173
PRWFIL	4002	001615	4001	007607	000034	991215
SEARCH	4002	001635	4001	007711	000026	001235
EXIT	4002	001655	4001	007762	000012	001255
GETERR	4002	001675	4001	007765	000020	001275
ERRSET	4002	001715	4001	010030	000032	001315
TLIB	4002	001735	4001	010252	000012	001335
TLOB	4002	001755	4001	010260	000012	001335
F\$IOBF	4002	000227	F\$A3	4002	000475	F\$A5 4002 000515

```

$ SAVE
$ QUIT [7]
OK, SEG *TEST
GO [8]
HELLO THERE
OK, FUTIL
GO [9]
> L 100
FROM-DIR = *
TO-DIR = *

BEGIN *
TEST B<-TEST

BEGIN *TEST

( 0, 0) ( 0, 1) ( 0, 2) ( 0, 3)
( 0, 33)

END *TEST
END
> TREDEL *TEST [10]
> Q
> TREDEL *TEST
> Q
OK,

```

#### NOTES TO PREVIOUS PRIMOS IV EXAMPLE

- [1] 2/400 tells FTN to generate 64-V mode code.
- [2] SEG is used to load PRIMOS IV Prime 400 programs.
- [3] Run file (in this case, \*TEST) must be specified here.
- [4] LOAD command of SEG.
- [5] The library loaded by SEG's LIB command is VFTNLB.  
A copy of VFTNLB must be present in the UFD name LIB  
or an error will occur.
- [6] Load map is typed. Note the differences between  
the PRIMOS IV (Prime 400) and PRIMOS II and III load maps  
(if you are familiar with these maps).
- [7] Saves run file image and quits SEG.
- [8] Executes the program saved in \*TEST.
- [9] FUTIL is invoked here to show the user something  
interesting about \*TEST.

- [10] Do not delete \*TEST with the DELETE command;  
instead, use the FUTIL TREDEL command to delete  
it. \*TEST is a segment directory.

## DEBUGGING

The PRIMOS operating system provides the user with on-line octal and symbolic debugging facilities (TAP and PSD). These are discussed in detail in the Program Development Software User Guide.

## USING FILE UTILITY (FUTIL)

A file utility command, FUTIL, provides commands for the user to copy, delete, and list both files and directories. FUTIL also allows directories and subdirectories to be referred to by treenames, and it allows the manipulation of both SAM (Sequential Access Method) and DAM (Direct Access Method) type files. Annotated examples of the use of FUTIL are described in the following paragraphs:

### Simple Copying of a File

```

FUTIL
>FROM HOPKIN
>TO JDOAKS
>COPY BIGFIL
>QUIT

```

'>' is the prompt printed by FUTIL to indicate it is ready to receive commands. Specifies another UFD from which to copy a file. The TO command specifies a UFD to which files are to be copied. If no TO-directory is specified, the current home UFD at the time FUTIL was invoked is taken to be the TO-directory.

The COPY command copies the file specified (in this case a file named BIGFIL).

QUIT command returns from FUTIL to command level.

### Copying a UFD

```

OK, A NEWUFD0 2

```

Attaches to a new UFD that is a sub-UFD under the current one. Note keys (in this case, 0 and 2) must be used when attaching to sub-UFD.



OK, FUTIL

GO

>FROM MSMITH

>UFDCPY

Copies FROM-directory,  
UFD MSMITH, and all  
its contained files and  
subdirectories.

>QUIT

OK,

For a detailed discussion of FUTIL, refer to the File System User Guide  
(MAN 2604).



## SECTION 3

## COMMANDS

## COMMAND STRUCTURE

This section defines the form of the various command lines that may be input to the operating system and describes the effect of PRIMOS commands, both internal (processed by the operating system) and external (executed by system-level programs that are called by PRIMOS). Commands unique to PRIMOS III or IV are indicated by a banner line. Commands that are normally issued from the supervisor terminal are briefly described in this section and detailed in the Computer Room User Guide (MAN 2603).

When properly loaded and started, PRIMOS II prompts the operator with the message OK:, and PRIMOS III or IV prompts the operator with the message OK, . These responses indicate that the operating system is ready to receive and process a command string. All commands consist of a command name and an optional list of arguments typed on a single line and entered by the carriage return key. The operating system analyzes and executes the command, if possible. Blank lines are ignored. Errors in the command string or the programs that execute external commands result in an error message.

A series of PRIMOS commands may be prepared by the Text Editor and stored in a command file for automatic execution under control of the COMINPUT or PHANTOM command.

Command Format

The general format of a PRIMOS command is:

```
COMMAND Name1 Name2 Arg1 Arg2 ... Arg9 (CR)
```

where COMMAND is the command name. Usually each Name is a Filename or UFD name (or is a meaningful identifier) and each Arg is an octal argument, or parameter, of up to six octal digits (maximum is 177777). If more than six digits are specified, the last six are used. Up to three names and nine arguments are allowed. Spaces must be used following the command name and between each Filename or argument. The ellipsis (...) indicates that the preceding item can be repeated. All commands are terminated by a carriage return (CR). Normally, this carriage return character is implied and not shown in the examples and formats contained in this document. The following examples demonstrate the notation used in this section to represent command formats:

```
RESUME Filename [Pc] [A] [B] [X] [Keys]
```

In this example, RESUME is the command name. The letter R is

underlined as the acceptable abbreviation. Generally, if an abbreviation for a command exists, it will be specifically noted in the description of the command. The RESUME command, shown in the example, must specify one of several legal filenames existing in the UFD to which the user is currently attached. The remaining items in the command string are the RVEC parameters (described later). Items enclosed in brackets are optional. Parameters are identified by the operating system according to their position in the command string; parameters that are omitted are assumed to be zero.

An ordinal value followed by a slash and a value can be used to set a selected octal parameter.

For example:

```
R FILENAM 3/1000
```

sets the value of the RVEC parameter, X. (i.e., skip three octal parameters and set the fourth to '1000.')

Items enclosed in vertical lines are alternatives, of which one must be chosen, as in

```
COMINP  |TTY      |
         |CONTINUE |
         |PAUSE    |
         |Filename |
```

Items in all capital letters (e.g., CONTINUE) must be entered literally. Items in initial caps (e.g., Filename) are variables to be assigned.

### Levels of Communication

There are two levels of communication between a user at a terminal and PRIMOS. The user either interacts with the supervisor, or with a program currently being executed under control of the operating system. When interacting with the supervisor, terminal input is interpreted at the command level as system commands. If the user is interacting with a program that is running under control of PRIMOS, terminal input is interpreted as data significant to that program, and it is passed from the supervisor to the running program. (The LINE FEED character is ignored by PRIMOS III and IV). In PRIMOS III and IV, there is one exception to the interpretation of terminal input. When the CONTROL-P character is input, it is always interpreted by the supervisor as a QUIT character. Whenever a user program or system command has completed execution, the user returns to command level ready to communicate with PRIMOS III and IV. Upon normal completion of a command or program, PRIMOS III and IV print the prompt:

OK,

or PRIMOS II prints the prompt:

OK:

If an error occurs, the operating system prints an error message and the prompt:

ER!

#### PRIMOS II Commands Allowed in PRIMOS III and IV

All user commands for PRIMOS II described in this section are available for use with PRIMOS III and IV, but the commands **STARTUP** and **SHUTDOWN** are not allowed for invocation by users at user terminals. In addition, the supervisor terminal commands described in The Computer Room User Guide are needed or are useful, for PRIMOS III and IV system operation.

#### PRIMOS III Commands Allowed in PRIMOS IV

All commands used in PRIMOS III (described in this section and The Computer Room User Guide) are available for use with PRIMOS IV.

#### Command Line Error Correction

Errors typed into the command line may be corrected by using the kill character (question mark character). It deletes everything previously typed on the line; the command must be retyped in entirety. Do not use the editor's erase character (double quote character) to rub out single characters in a command line under PRIMOS II (i.e., at PRIMOS II command level). Under PRIMOS III and IV, usage of both question mark and double quote in command lines is permitted.

#### PRIMOS Names

PRIMOS names, or UFD names, consist of one to six ASCII characters. For compatibility with the command string interpreter and the text editors, the first character must be non-numeric; the others may be any printing character except the question mark or quotation mark.

Examples:

<u>Legal</u>	<u>Illegal</u>	
CMDNC2	2CMDNC	(Begins with numeral)
LDR\$A	LDR A	(Contains space)
TEST1	TESTER1	(First six characters
TEST2	TESTER2	not unique)

#### Disk vs PRIMOS Units

PRIMOS file units (1-16), referenced by the **BINARY**, **CLOSE**, **INPUT**,

LISTING, and OPEN commands, are identified by the abbreviation Funit. These are not to be confused with the logical disk units referenced by the abbreviation Ldisk in the ATTACH command. Physical disk drives are assigned logical disk unit numbers by the STARTUP command (refer to The Computer User Guide(MAN 2603); thereafter, only the logical unit numbers are meaningful to PRIMOS.

## SUMMARY AND INTRODUCTION TO COMMANDS

### Internal Commands

Internal commands are executed in the address space occupied by PRIMOS itself, as opposed to those commands which are external to the operating system and execute in user space. Most internal commands are concerned with the file handling and with saving or restoring of filed programs and associated register values. PRIMOS internal commands are described here in alphabetical order, and also listed in Appendix A. PRIMOS III and IV operator commands are described in the Computer Room User Guide (MAN2603). Detailed information about each internal command is given in the last part of this section. The descriptions include all commands, both internal and external, arranged in alphabetical order. The internal commands described are:

ASRCWD	INPUT	PRERR	SVCSW
ASSIGN	LISTF	PROTECT	TIME
ATTACH	LISTING	RESTORE	UNASSIGN
BINARY	LOGIN	RESUME	USERS
CLOSE	LOGOUT	SAVE	VRTSSW
COMINPUT	MESSAGE	SHUTDN	
DELAY	OPEN	START	
DELETE	PHANTCM	STARTUP	

### Hybrid Commands

PRIMOS II recognizes four external commands that are restored into low memory superimposed upon user address space. These commands accept an internal command line interpretation, but destroy user memory space. Furthermore, in PRIMOS III and IV, they function as internal commands. The file containing hybrid commands is in CMDNC0 and is named DOEXT. All versions of PRIMOS II use the same hybrid command file. The hybrid commands are: CNAME, CREATE, PASSWD, and STATUS. The hybrid commands CREATE, PASSWD, CNAME, and STATUS act as external commands in PRIMOS II and act as internal commands in PRIMOS III and IV. They are listed in Appendix A. Detailed descriptions of the hybrid commands are given in last part of this section, arranged alphabetically with all the other command descriptions.

### External Commands

The external commands serve to load and start system programs in the command UFD (e.g., CMDNC0). They are external to the operating system

and may execute in user address space (i.e., they may reside in a UFD). In general, these programs include the translators, utilities, and debugging programs used in Prime application program development. However, the memory image of any type of program can be saved and filed in CMDNC0 and called for thereafter by filename. Some of the external commands, like FUTIL, CRSER, and PRSER, control data transfers to or from peripheral devices. The user may want to add programs to CMDNC0 to perform functions unique to his system. Unless otherwise specified, programs invoked by external commands return to command level after they have completed execution.

The external commands are listed along with hybrid commands in Appendix A. The external commands and the hybrid commands listed alphabetically (as are descriptions of all commands) are arranged in the last part of this section as follows:

AVAIL	FILMEM	MCG	PUS
BASIC	FILVER	MDL	RUNOFF
BASINP	FIXRAT	MTDISK	SED
CMPRES	FTN	NUMBER	SFTN
CNAME	FUTIL	PASSWD	SIZE
CNVIMA	HILOAD	PMA	SLIST
COPY	LATE	PRMPC	SORT
CREATE	LBASIC	PRSER	SPMA
CRMPC	LOAD	PRVER	SPOOL
CRSER	LOAD20	PSD	STATUS
DBASIC	MACHK	PSD20	TAP
ED	MAGRST	PSD160	TRAMLC
EXPAND	MAKE	PTCPY	UDOS64
EDB	MAGSAV	PTBOOT	UPCASE

#### Contents of Command UFD

The following is a typical example the command lines issued to print the contents of the command file CMDNC0.

Example:

```
OK: ATTACH CMDNC0
OK: LISTF
```

UFD = CMDNC0

```
FILBLK RTOSRA MCG    NUMBER TAP    RT128F BASINP SIZE    LOADAP CNVT45
FILVER CMPRES EXPAND COPYB  AVAIL  PSD    FILCPY UFDCPY PMA    FTN
PTBOOT COPY  MAGSAV MAGRST LFTN  CRSER  CRMPC  PRMPC  PRSER  BASIC
LBASIC DBASIC SLIST  MTDISK SPOOL  LOAD   LOAD20 MDL    FILMEM PSD20
ED    PRTE  VDOS32 DOSEXT PTCPY  FUTIL  FIXRAT HILOAD DOSVM  EDB
MACHK  MAKE    SORT
```

## COMMAND DESCRIPTIONS

All command descriptions (internal, external and hybrid) are arranged in alphabetical order in the following paragraphs. Programs that have operating procedures or an extensive command repertoire may be described in detail in appendices, or in other user guides, such as the Program Development Software User Guide (MAN 1879). In the following detailed descriptions of commands, the elements in all capital letters are command names. Elements in initial capital letters are arguments for which Parameters are substituted. If an argument is enclosed in square brackets, the argument is optional. The abbreviation of a command name is given in the description, and commands unique to PRIMOS III and IV are flagged.



```
*****  
* ADDISK *          ***PRIMOS III and IV***  
*****
```

The command format is:

ADDISK Pdisk1 Pdisk2 ... [Pdisk]

ADDISK may be used to start up the disk specified by the physical disk (Pdisk) arguments. It is an operator command, and it is issued from the supervisor terminal. For further information, refer to the Computer Room User Guide (MAN 2603).

Example:

ADDISK 52 54

```
*****  
* AMLC *           ***PRIMOS III and IV***  
*****
```

The command format is:

AMLC [Protocol] Line [Config] [Ldisk]

The AMLC command may be used to start up an AMLC line. AMLC is an operator command, and it is issued from the supervisor terminal. For further information, refer to the Computer Room User's Guide (MAN 2603).

\*\*\*\*\*  
 \* ASRCWD \*  
 \*\*\*\*\*

\*\*\*PRIMOS III and IV\*\*\*

The ASRCWD command allows changing the virtual control word (refer to I/O Virtualization in Appendix E). This control word is used to select one of four devices as effective output and one of four for input. The control word sets the device's output by the OTA 4 instruction and the device input by the INA 4 instruction (refer to the System Reference Manual MAN 1671 ). The format of ASRCWD is:

ASRCWD XXXXXX

where XXXXXX is an input or output number as specified in Table 3-1.

Table 3-1. Value for Virtual Control Word and Port Assignment.

Device or Port No.	Input (Bits 11, 12)	Output 13-16)
1	00 User terminal	00 or 10 User terminal
2	01 Reserved	Octal 4 CENPR (J2)
3	10 Reserved	Octal 2 CE2PR (J3)
4	11 CARDR (J4)	Octal 1 PUN (J4)

If ASRCWD is issued with no arguments the value 0 is assumed by PRIMOS.

The ASRCWD command is chiefly useful to users that have a serial line printer in their Prime computer system configuration.

Example:

Assume output was being sent to the serial line printer and a user program abort or a CNTRL-P condition occurred. At this point, the user would not be able to get output (from the editor ED for example) printed or displayed at the users terminal. Issuing the command line: ASR 0 would then allow the user to recover and get output at the terminal.

\*\*\*\*\*  
\* ASSIGN \*  
\*\*\*\*\*

\*\*\*PRIMOS III and IV\*\*\*

The ASSIGN command obtains complete control over a disk or a peripheral device (e.g., printer, paper tape reader) from the user terminal.

The format is:

```
ASSIGN | Device [WAIT] |  
      | DISK [WAIT] Line |  
      | AMLC [Protocol] Line [Number] [Config] |  
      | SMLC [WAIT] Line |
```

where Device is an available device.

All assignable devices are named as shown in Table 3-2.

Table 3--2. Device Names.

Code	Device
<u>CARDR</u>	General Card Reader (AMLC Line No. 6)
<u>CENPR</u>	First Centronics Printer (System Option Controller, port No. 2)
<u>CE2PR</u>	Second Centronics Printer (System Option Controller, port No. 3)
<u>CR1</u>	MPC Parallel Card Reader
<u>DISK 0</u>	Physical Disk 0
<u>DISK 1</u>	Physical Disk 1
<u>DISK 2</u>	Physical Disk 2
<u>DISK 3</u>	Physical Disk 3
<u>DISK 4</u>	Physical Disk 4
...	
<u>DISK 57</u>	Physical Disk 57
...	
<u>DISK 3256</u>	Physical Disk 5256
<u>DISK 002452</u>	Physical Disk partition
<u>MT0</u>	Magnetic Tape Unit 0 Dial = 0
<u>MT1</u>	Magnetic Tape Unit 1 Dial = 1
<u>MT2</u>	Magnetic Tape Unit 2 Dial = 2
<u>MT3</u>	Magnetic Tape Unit 3 Dial = 3
<u>PRI</u>	MPC Parallel Interface Line Printer
<u>PTR</u>	Paper Tape Reader
<u>PUNCH</u>	Paper Tape Punch
<u>PLOT</u>	Versatec Printer-Plotter
<u>SMLC 1</u>	Communications Line 1
<u>SMLC 2</u>	Communications Line 2
<u>SMLC 3</u>	Communications Line 3

For complete disk-assignment details, refer to the Computer Room User Guide.

A user may ASSIGN only a disk that is not already assigned and which appears in the Assignable Disks Table. This table which is initially empty, is altered from the supervisor terminal using the DISKS command. This restriction provides a degree of system integrity because it prevents users from assigning a disk without the supervisor terminal operator's knowledge, or from assigning disks or partitions the operator wishes to reserve for special use.

For a disk to be ASSIGNED to a user, it must not be the paging disk nor ASSIGNED to another user, nor a disk specified in a previous STARTUP command. To ASSIGN a disk that has been started by STARTUP, it must first be shut down by the SHUTDOWN command at the supervisor terminal.

If the device is currently assigned to another user, the system replies:

DEVICE IN USE

ER!

unless the optional argument WAIT was supplied. In this case, the ASSIGN command is queued until the device is UNASSIGNED by another user, or until the user presses the CTRL-P or BREAK key.

If the user does not ASSIGN a device and attempts to perform I/O to or from the device, the error message:

DEVICE NOT ASSIGNED

ER!

is printed at the terminal.

The terminal issuing the ASSIGN command is unavailable for use until the device assigned is again available for assignment or until the CTRL-P or BREAK key is pressed by the user at the terminal.

Disks or devices ASSIGNED by another user are released when the user invokes the UNASSIGN command and/or when the user invokes the LOGOUT command.

Examples:

ASSIGN CENPR WAIT

Assigns the Centronics printer and queues the assignment if the printer is already assigned.

AS PTR

assigns the paper-tape reader.

AS DISK 460

AS DISK 54

each assigns disk drives as defined in Table 3-2.

The maximum number of disk drives that may be ASSIGNED to all users at any one time is ten. If an attempt is made to ASSIGN too many disks, the message:

ASSIGN TABLE FULL

is printed.

A user may assign an AMLC line as follows:

ASSIGN AMLC [Protocol] Line [Config]

where Line specifies a line number. Using this form of the ASSIGN command, a user may assign the AMLC line number and may set a terminal protocol and line configuration word for the line specified by the line number, Line. Refer to the description of the AMLC command in the Computer Room User Guide (MAN2603) for a description of possible parameters for the arguments Protocol and Config.

A user may only ASSIGN an AMLC line if it has been configured to be ASSIGNED and if it not ASSIGNED to another user. A user terminal line may be ASSIGNED if first the command:

AMLC TTYNOP

has been given at the supervisor terminal.

```
*****
* ATTACH *
*****
```

The format of the ATTACH command is:

```
ATTACH Ufd [Password] [Ldisk] [Key]
```

To access files, a PRIMOS user must be attached to some User File Directory. Being attached implies PRIMOS has been supplied with the proper file directory and either the owner or nonowner password, and PRIMOS has found and saved the location of the UFD named in the ATTACH command. After a successful attach, the name, location and owner/non-owner status of the UFD is referred to as the current UFD. As an option, this information may be copied to another place in PRIMOS, referred to as the home UFD. Unless otherwise specified in the ATTACH command, the current UFD is also the home UFD. The user obtains owner status if the owner password is specified, or nonowner status if the nonowner password is specified. The owner of a file directory can declare on a per-file-basis what rights a nonowner has over the owner's files. The nonowner password may be specified only under PRIMOS III or IV (refer to the File System User Guide (MAN 2604) and the commands PASSWD and PROTEC for more information). In attaching to a directory, ATTACH specifies a file directory in the Master File Directory (MFD) on a particular logical disk, or a file directory in the current UFD, or the home UFD as the directory to be attached. The most common form of the ATTACH command is:

```
ATTACH Ufdnam Passwd
```

The meaning of this command line is: search for UFD in the MFD on all started-up logical devices 0, 1, 2 ... n, and attach to the UFD specified by Ufdnam that appears in the MFD of the lowest numbered logical device. Also, the command line indicates attach to Ufdnam only if Passwd matches the password of UFD Ufdnam. After the attach operation is completed, ATTACH then sets the home UFD to Ufdnam.

The user may specify the logical disk of the MFD to be searched, as in the command:

```
ATTACH Ufd Password Ldisk
```

Ldisk is specified as an octal integer.

Finally, the user may specify a key as in the command:

```
ATTACH Ufd Password Ldisk Key
```

If Key is 177777, the MFD of the currently attached disk is searched for Ufdnam. If Key is 100000, all disks are searched in logical order.

The keys are as follows:

<u>Key</u>	<u>Meaning</u>
0	Attach to Ufd in MFD on Ldisk; set home UFD.
1	Attach to Ufd in current UFD; do not set home UFD.
2	Attach to Ufd in current UFD; set home UFD to current UFD.
<del>100000</del>	<del>Attach</del> to Ufd in MFD on Ldisk; do not set home UFD.

To attach to the home UFD, use ATTACH (blanks).

If the user specifies a UFD that the ATTACH command cannot find, the message

Ufdname NOT FOUND

is printed at the terminal. The user remains ATTACHED to the UFD that was the current UFD at the time that the ATTACH command was issued.

If the user specifies an incorrect password, the message:

BAD PASSWORD

is printed at the terminal. The user is unATTACHED from the UFD that was the current UFD at the time of the ATTACH command. When this occurs, any subsequent command that attempts to reference files in the original current UFD fails and results in the message:

NO UFD ATTACHED

Examples:

ATTACH GOUDY ABCABC

Search for GOUDY in the MFD on all started up disks. Attach to GOUDY on the lowest numbered logical disk where found. Check the password. Set home UFD.

ATTACH

Attach to home UFD (GOUDY).

ATTACH CARLSO XXXXX 7

Attach to CARLSO. Look for CARLSO with a password of XXXXX in the MFD of logical disk 7. Set home UFD to CARLSO.

Attach is an internal command.



Annotated Example, ATTACHing to UFD and Sub UFD's:

OK, A MFD XXXXXX 10 Attaches to the MFD on logical unit:  
10 (with a password of XXXXX) to start  
example at the top of the file system  
tree.

OK, A UFDMAX This command line attaches to  
an already existing UFD, MAX (treename  
is: MFD > UFDMAX.) The home UFD and  
current UFD are both UFDMAX.

OK, ATTACH MAX1 0 2 Attaches user to UFD = MAX1  
and changes the home UFD to MAX1.  
Note that the key had to be  
specified in this case; i.e., attaching  
to a UFD subordinate and another  
UFD (as opposed to attaching to a UFD  
subordinate to the MFD).

OK, CREATE MAX2 Creates a UFD named MAX2  
subordinate to MAX1. (The  
treename of MAX2 is:  
MFD>UFDMAX>MAX1>MAX2.)

OK, A MAX2 0 1 Attaches user to UFD=MAX2. Note  
that this time the home UFD was  
not changed, only the current UFD  
(i.e., key=1).

...

OK, A Attaches the user back  
to the home UFD with blank arguments,  
in this case, MAX1.

\*\*\*\*\*  
 \* AVAIL \*  
 \*\*\*\*\*

The AVAIL command gives the number of disk records available for use in the specified logical disk (in decimal). The format is:

```

  AVAIL [ | ZERO      | | ]
          | ONE       | |
          | TWO       | |
          | ...       | |
          | SEVENTEEN | |
          | Packname  | |
          | *         | |

```

If no argument is specified, AVAIL types the number of available records on the current logical disk and the percentage of space used up on that disk, provided that the password on the MFD is XXXXXX. If the MFD's password is other than XXXXXX, the message NO RIGHT is returned when AVAIL is invoked with no argument. However, the user can check availability with a Packname argument, regardless of the MFD's password. If Packname is specified instead of ZERO...NINE, the number of available records on the logical disk with DSKRAT name Packname is printed. AVAIL is an external command.

Examples:

```

  OK, AVAIL
  GO
      3684 RECORDS AVAILABLE
      71.6 PERCENT FULL.
  OK,

```

The above example illustrates that when no arguments are specified, AVAIL prints status information for the current logical disk (in this case, DOCUME).

Example:

```

  OK, AVAIL DOCUME
  GO
      3684 RECORDS AVAILABLE
      71.6 PERCENT FULL
  OK,

```

The previous examples show use of AVAIL with a packname. The following is an example of using AVAIL with a unit number (spelled out).

Example:

```

  OK, AVAIL TEN
  GO
      6328 RECORDS AVAILABLE

```

## 2.6 PERCENT FULL

OK,

The following example shows the result of specifying \*. With an \*, a UFD SYSTEM is looked for and a file called DISCS must contain a list of disks which may be configured (started-up) on a system. This information is printed one disk per line as shown in the following example.

Example:

AVAIL \*  
GO

VOLUME	PHYSCL	TOTAL	FREE	PERCENT
ID	DEVICE	RECS	RECS	FULL
TS	50	6496	54	99.2
MFGINV	12450	32480	301	99.1
MFGDSK	61450	19488	2354	87.9
SPOOLD	110050	6496	5975	8.0
SOFTWR	3452	45472	4421	90.3
HARDWR	71452	19488	3156	83.8
ADMIN	1054	12992	2313	82.2
ETCH	21054	12992	2553	80.3
DOCUME	41054	12992	3528	72.8

## NOTE

If the MFD owner password is not XXXXXX, the AVAIL command will not work under PRIMOS II. Under PRIMOS III and IV, AVAIL will work with the Packname and \* optional arguments if:

The MFD nonowner password is set to XXXXXX.

The Packname (name of DSKRAT file) in the MFD is given nonowner PROTECTION rights of 1.  
The following is an example of how to do this:

A MFD XXXXXX

PASSWD NEW XXXXXX New password for MFD

PROTEC PACKNA 7 1 New protection rights for  
DSKRAT file named PACKNA

\*\*\*\*\*  
\* BASIC \*  
\*\*\*\*\*

The BASIC command loads the Prime BASIC Language interpreter. On the original master disk, the version of BASIC that has both the matrix functions and print-using functions is named LBASIC; the version that does not have these features is named BASIC on the master disk. Single-precision arithmetic is standard. BASIC is an external command. For further information, refer to the BASIC User Guide.

\*\*\*\*\*  
\* BASINP \*  
\*\*\*\*\*

The format of the BASINP command is:

BASINP Filename

The BASINP command invokes a program that loads, from paper tape, a BASIC program that has been written for a computer system other than a Prime computer. Filename is the name of the file into which the contents of the paper tape are to be read. BASINP is an external command.

```
*****
* BINARY *
*****
```

The format of the BINARY command is:

```
BINARY Filename
```

BINARY opens a file for writing on PRIMOS File Unit 3, usually as a binary output file for use by the compiler or assembler. The file is assigned the name Filename in the current UFD. This command has the same effect as OPEN Filename 3 2. BINARY is an internal command.

PMA and FTN automatically open a file named B XXXX as the binary output file (XXXX are the first four letters of the Input (source) filename). A BINARY command is required only if the user wants the output file to have a different name.

```
*****
* CLOSE *
*****
```

The format of the CLOSE command is:

```
CLOSE | [Filename] [Funit] ... [Funit] |
      | ALL |
```

The CLOSE command closes the named files and specified file units. The form: C ALL closes all files and units. (In a command file, specify each item to be closed; do not use C ALL or the command file itself will be closed.) CLOSE is an internal command.

The CLOSE ALL command also makes sure that buffers are retrieved properly and resets the state of the file system. If the user is even slightly uncertain about the state of the file system, he should enter a CLOSE ALL. (The STATUS command prints the state of the file system.)

If the file named cannot be found, an error message is printed and the CLOSE command returns to operating system command level.

The command line:

```
CLOSE ALL
```

must be given following a CNTRL-P to QUIT (interrupt) a program if the user wishes to avoid difficulties later attempting to use an open file (or files).

```
*****
* CMPRES *
*****
```

The format of the CMPRES command is:

CMPRES Filename1 [Filename2]

The input ASCII file, Filename1, is translated into the output ASCII file, Filename2, using the relative copy character (^220). The byte following the relative copy character specifies the number of characters to copy from corresponding positions in the preceding line. If Filename2 is omitted, the output replaces Filename1. The amount of space saved is a function of the structure of Filename1. CMPRES handles a line size of up to 720 characters. CMPRES is an external command.

Example (contents of typical Filename1 named STEST):

```
C      PROGRAM TO TEST DSQRT
C
      DOUBLE PRECISION A,B
      READ (1,1) A
      B; DSQRT (A)
      WRITE (1,2) A
      STOP
```

Command line:

CMPRES STEST CTEST

Example (contents of Filename2, CTEST):

```
C      PROGRAM TO TEST DSQRT
C
      DOUBLE PRECISION A,B
      ^221,1^007READ (0) A
      ^220^007B; DSQRT (A)
      ^220^007WRITE (1,2) A
      ^220^007STOP
```

To reverse the effect of the CMPRES command, use the EXPAND command.

\*\*\*\*\*  
\* CNAME \*  
\*\*\*\*\*

The format of the CNAME command is:

CNAME Oldname Newname

CNAME changes the name of the file (or UFD) named Oldname to Newname. This command operates within the current UFD, and can be used to change the name of a UFD that is immediately subordinate to the current UFD, as well as changing the names of files. Thus, if the user is attached to the MFD, CNAME can be used to change the names of UFDs residing in the MFD, as well as changing the names of specialized files such as DSKRAT. The user is cautioned not to change names of special UFD's such as CMDNC0. CNAME is a hybrid command under PRIMOS II. Under PRIMOS III and IV, CNAME requires owner status to the UFD.

Example:

OK: A MFD  
OK: CNAME SPARE2 JHND0E

assigns a new UFD name JHND0E in the place of the older name SPARE2.

\*\*\*\*\*  
\* CNVIMA \*  
\*\*\*\*\*

CNVIMA converts a memory map to an ASCII file image and may be useful in debugging. For further information, refer to the description of PSD in the Program Development Software User Guide (MAN 1880).

\*\*\*\*\*  
 \*COMINPUT\*  
 \*\*\*\*\*

The formats of the COMINPUT command are:

COMINPUT	TTY	[Funit]
	Filename	
	CONTINUE	
	PAUSE	

CO Filename causes PRIMOS to read terminal input from in the current UFD, rather than from the terminal. The file is usually prepared and filed by the text editor (ED). This type of file is referred to as a command file. Command files may be chained. If the last line in a COMINPUT command file is of the form:

CO Filename

the current command file is closed and PRIMOS reads commands from the new command file Filenameex. This feature allows chaining of command files. The last command in the last command file in the chain must be CO TTY to return control to the terminal. Note that 'TTY', 'CONTINUE', and PAUSE are reserved words for PRIMOS and must not be used for other purposes.

COMINPUT is an internal command.

PRIMOS reads commands from the command file, Filename, by opening File Unit 6 and reading, then executing, one line at a time. When the command CO TTY is encountered, PRIMOS takes subsequent commands from the terminal. The user must specify a file unit (Funit) for COMINPUT TTY if not using the default unit. Any error message causes command input to be returned to the terminal. However, the command input file is left open, which allows a user to retype the command that caused the error message and then continue reading from the command input file by typing:

CO CONTINUE

Use of the command CLOSE ALL in a command input file closes the command input unit and causes the message COMINP FILE EOF to be printed.

The form:

COMINP Filename Funit

has the additional capability of specifying the file unit upon which the command file is to be opened. Thus, the user can set up a complex set of interacting command files.

The form:



## COMINPUT PAUSE

leaves the current command input unit open and returns to command level. Thus, a user can invoke other commands or use COMINPUT (on another file unit) to start another command file on another unit before issuing a COMINPUT CONTINUE line to continue the original command file.

Note that external commands such as the loader and the editor may get input from a command file. FORTRAN programs using Unit 7 to read from the terminal get input from the command file instead, if a command file is invoked. User programs may also request input from a command file directly by calling (7IN, COMANL, CMREAD, CMIN\$, and ISAAL2). See the File System User Guide and the Subroutine Library User Guide (MAN 1880) for details.

Command files may be controlled from user programs using the COMINP subroutine. Example:

Assume the command file PMLIST contains the following:

```
PM
LIST
COMINPUT PMPM 7
CLOSE 7
PM
COMINPUT TTY
```

and the command file PMPM contains the following lines:

```
PM
STATUS
PM
COMINPUT CONTINUE 6
```

Then, typing the command line:

```
COMINPUT PMLIST
```

from PRIMOS command level causes both command files to be run.

The COMINPUT command is useful for updating large programs that consist of many files, use several library files, or require special loading procedures. For example, suppose a user with the UFD USER1 has a program consisting of three FORTRAN source files MAIN, SUB1, and SUB2.

## Examples:

Assume this program requires two libraries, MATHLB and FTNLIB.  
A user makes up the following command input file named DPROG:

```
FTN MAIN
FTN SUB1
FTN SUB2
FILMEM
LOAD
LO B MAIN
LO B SUB1
LO B SUB2
LIB MATHLB
LIB
MAP
QUIT
CO TTY
```

After the programs are corrected and ready to be compiled, the user enters the command CO DPROG. The DPROG file then provides the commands that cause the programs to be compiled, loaded, and a load map printed. DPROG also documents the source files and loading procedure.

\*\*\*\*\*  
\* CONFIG \*           \*\*\*PRIMOS III and IV\*\*\*  
\*\*\*\*\*

The CONFIG command defines system parameters. Its format is:

CONFIG Nusr Pdev1 Comdev [Availm] [Pdev2] [Namlc] [Nphan]

CONFIG is specified once per system session. It is an operator command, and it is issued at the supervisor terminal. For further information, refer to the Computer Room Operators Guide (MAN 2603).

\*\*\*\*\*  
\* COPY \*  
\*\*\*\*\*

COPY is an external command that copies and verifies any disk to any other disk, under PRIMOS. Under PRIMOS III and IV, both disks must be ASSIGNED before invoking COPY. When invoked, the COPY command responds with numerous queries and instructions to the user. Since COPY is a command used most frequently by system operators, it is discussed in full detail in the Computer Room User Guide (MAN 2603).

\*\*\*\*\*  
 \* CREATE \*  
 \*\*\*\*\*

The format is:

CREATE Newufd

The CREATE command creates a new UFD named Newufd in the current UFD. CREATE is a hybrid command under PRIMOS II. The passwords of the new UFD are: owner password is Blank, and the nonowner password is Zero (any password will match). Also the protection keys are set to 7 7 when CREATE is invoked for the new UFD.

OK: A MFD XXXXXX  
 OK: CREATE BETTY

OK: LISTF  
 UFD=MFD      0

TSDISK	MFD	BOOT	CMDNC0	PODISK	JBRWNS	GABOON	CREATE
BARBOU	STUMP	GYLES	LIB	SPQRXR	BASIC	PRIMOS	WEYLER
M.JOHN	AROSS	KRQY	GRABIN	DEMO	JSKOL	KAY	CURREV
JCVB	DAVIS	EDIN	BROVN	SEV	LEVIS	PRNGL	BUTTER
BRIGGS	COHEN	ROWDY	DUMAS	BRODIE	CARLSO	PLANIT	WEBB
ETTA	KUNDQV	RUNDQ	BETTY				

OK:

For an example of creation of a sub UFD, refer to the example following the description of the ATTACH command.

\*\*\*\*\*  
 \* CRMPC \*  
 \*\*\*\*\*

The format of the CRMPC command is:

CRMPC Filename

CRMPC reads cards from the parallel interface card reader connected to the MPC controller and loads card image ASCII data into the file Filename. Reading continues until the end of the deck or a \$E is read in columns 1-2 of a card. The \$E causes a return to PRIMOS and closes the file. If the reader runs out of cards before a \$E card is read, the processor returns to the operating system but the file is not closed. The user can load more cards and enter S (i.e., START) to resume reading cards into the same file. At completion of reading, if there was no \$E card, enter CLOSE ALL to close the open file. \$6 in column 1 indicates 026 keypunch code. \$9 in column 1 indicates 029 keypunch code. CRMPC is an external command. Under PRIMOS III and IV, CR1 must be assigned before the CRMPC command is given.

\*\*\*\*\*  
 \* CRSER \*  
 \*\*\*\*\*

The format of the CRSER command is:

CRSER Filename

CRSER reads cards from the serial interface card reader. Card deck format and card formats is the same as for CRMPC, described previously. Under PRIMOS III and IV, CARDR must be assigned before the CRSER command may be given. The CRSER command is an external command.

\*\*\*\*\*  
 \* DBASIC \*  
 \*\*\*\*\*

DBASIC loads the Prime BASIC interpretative DBASIC language version that has double-precision arithmetic capabilities. DBASIC is an external command.

```
*****  
* DELAY *          ***PRIMOS III and IV***  
*****
```

The DELAY command defines a time function to be used to delay the printing of a character after a line feed (LF) has been output to a terminal. The format is:

DELAY [Minimum] [Maximum] [Rmargin]

Minimum defines the number of character-times (time it takes the system to type a character on a line) to delay when CR (carriage return) is output at the left margin. Maximum defines the number of character-times to delay when CR is output at the right. Rmargin defines the number of characters required to move to the right margin. If a CR is typed at some point within a line, the time delay is proportional to the number of characters typed. If Rmargin is not specified, 72 is assumed; if Maximum is not specified, 12 is assumed. If the command, DELAY, is given with no parameters, the default values 6, 12, and 72 are assumed; these values are adequate for most 30 cps terminals.

Example:

```
DELAY 0 10 100.
```

The DELAY command may be issued from the system terminal as well as the user terminal. In this case, the DELAY command must be issued while the system terminal is designated to be user 1 (refer to the USRASR command).

Another example:

Delay is particularly useful if the user has a terminal with a non-standard line speed. In this case the command.

DELAY 10

should be sufficient to allow the terminal to function in the Prime computer configuration.

\*\*\*\*\*  
\* DELETE \*  
\*\*\*\*\*

The format is:

DELETE Filename

DELETE frees the disk storage space used by Filename and removes the name from the current UFD. DELETE is an internal command.

CAUTION:

Do not delete a directory until all files within the directory have been deleted. Otherwise, available disk storage space is lost until the next time FIXRAT is run. To delete a directory, use of the TREDEL subcommand of the FUTIL command is recommended.

```
*****  
*  ED  *  
*****
```

The format is:

ED [Filename]

This command loads and starts ED, the most commonly used version of the text editor. If a filename is specified, the file specified by filename is loaded into the editor's text buffer in memory, and the editor is started in EDIT mode. Otherwise, the editor is started in high-speed INPUT mode with an empty text buffer. Files and units are automatically opened and closed. ED is an external command. For details of ED operation, refer to the Program Development Software User Guide (MAN 1879).

#### Restarting Editor

If the user accidentally returns control to PRIMOS (for example, by a QUIT), the user can restart ED without losing any of the text buffer by issuing the command:

START 1000

Refer to the EDITOR section of the Program Development Software User Guide for exceptions and details of Recovery Procedures.

#### Versions of ED

ED, on the master disk, exists in two versions: SED for small editor and ED for large editor. ED requires a 24K memory and PRIMOS II (LED corresponds to EDG and SED corresponds to EDLIN, refer to the Program Development Software User Guide). It is anticipated that when the master disk is installed for use, SED will be renamed ED for small systems. All editor documentation refers to the editor command as ED.

#### Summary of Editor Subcommands

For a complete description of all the subcommands available for text editing under control of ED, refer to Appendix A. The following table, Table 3-3, is a subset consisting of the simplest and most useful subcommands.



Table 3-3. Subset of ED (Editor) Commands

<u>Commands</u>	<u>Command Function</u>
<u>QUIT</u>	Return to PRIMOS command level.
<u>APEND</u> String1	Append the contents of the string specified to the current line being edited.
<u>CHANGE</u> /String 1/String 2/	Change the contents of String1 to String2 in a line.
<u>DELETE</u>	Delete the current line.
<u>FILE</u> Filename	Preserve the contents of the editor's text buffer by copying them to the file specified by Filename. If Filename already exists, its contents will be replaced by the current contents of the editor's text buffer.
<u>INSERT</u> String1	Insert a new line after the current line; its contents are specified by String1.
<u>LOCATE</u> String1	Locate the first instance of String1 on a line that follows current line of the editor's text buffer.

```
*****
* EDB *
*****
```

The format of the EDB command is:

EDB Inputfile [Outputfile]

The EDB command loads and starts EDB, the binary editor, which prints ENTER and waits for command input. The input and output files may be on disk or paper tape. If paper tape is used for either file, use the filename (PTR). If an output filename is specified, a file of that name is created in the current UFD. If the filename already exists, it is overwritten by the output file. EDB is an external command. For details, see the Program Development Software User Guide (MAN 1879). The EDB editor that runs under PRIMOS IV understands the object files generated by the PRIMOS IV language processors.

```
*****
* EXPAND *
*****
```

EXPAND reverses the operation of CMPRES. The format of EXPAND is:

EXPAND Filename1 [Filename2]

where Filename1 is the file to be expanded and Filename2 is an optionally specified output file. If Filename2 is omitted, output is placed in Filename1. EXPAND handles line sizes up to 720 characters. EXPAND is an external command.

```
*****
* FILMEM *
*****
```

The format is:

```
FILMEM
or
FILMEM ALL
```

Under PRIMOS III and IV, FILMEM with no argument fills the memory locations 100 to the top of 32K with zeros. If running under PRIMOS II, FILMEM clears 100 to the top of 64K bytes, except for those locations occupied by PRIMOS II.

FILMEM ALL clears all of the user space (up to 128K bytes).

\*\*\*\*\*  
 \* FILVER \*  
 \*\*\*\*\*

The format is:

FILVER Filename1 Filename2

When Filename1 and Filename2 are used, FILVER causes them to be compared for equivalence. If any differences exist, a message is printed indicating failure to verify. If the file Filename1 and Filename2 are exactly the same, a message is printed that confirms successful verification. FILVER is an external command.

When FILVER is specified with no filename arguments, FILVER accepts treename arguments. If the user elects to use treenames, FILVER responds by asking the user to specify the treename of each file to be compared.

Example:

```
OK, FILVER
FILE 1: >ROWDY>OBJ>DATA1
FILE 2: >ROWDY>OBJ>DATA2
```

OK,

In addition, to confirm verification, FILVER displays any differences and offers the user the option of continuing file verification. Up to 23 differences can be displayed on the user's terminal. If more differences exist, the user is asked whether he wishes to continue. For further information on the concept of treename, refer to the File System User Guide (MAN 2604).

Example:

```
OK, FILVER
GO
FILE 1: GOUDY > LIB4S2
FILE 2: TEKMAN > LIBMAN > S2
DIFF 000000 000043 120245 122706
DIFF 000000 000044 143311 144726
DIFF 000000 000045 153305 142723
DIFF 000000 000046 151645 122612
DIFF 000000 000047 105000 127360
DIFF 000000 000050 127360 160747
DIFF 000000 000051 160747 120261
DIFF 000000 000052 120261 105000
DIFF 000000 000053 105000 127357
DIFF 000000 000054 127357 163240
DIFF 000000 000055 163240 127657
DIFF 000000 000056 127657 122723
DIFF 000000 000057 122723 142703
```

```

DIFF 000000 000060 142703 122621
DIFF 000000 000061 122621 002255
DIFF 000000 000062 002255 121657
DIFF 000000 000063 121657 122704
DIFF 000000 000064 122704 140724
DIFF 000000 000065 140724 142645
DIFF 000000 000066 142645 127612
DIFF 000000 000067 127612 127345
DIFF 000000 000070 127345 163240
DIFF 000000 000071 163240 127722
CONTINUE= NO
OK,

```

The differences are reported in a form useful for comparing run files. It is suggested that the user also have listings of both files compared to make use of the difference information printed by FILVER. Four numbers are displayed for each difference:

```
DIFF wwwwww xxxxxx yyyyyy zzzzzz
```

where wwwwww xxxxxx describes the position of the file: wwwwww is a sector number (in octal); xxxxxx is the offset within the file (in octal). The user must take into account the nine-word header in run files and any offset from a sector boundary in the starting location of the run file. The parameter yyyyyy is the value of the differing word in FILE 1, zzzzzz is the value of the word in FILE 2; both of these parameters are in octal.

\*\*\*\*\*  
\* FIXRAT \*  
\*\*\*\*\*

The FIXRAT command may be specified as follows:

FIXRAT [OPTIONS]

FIXRAT is a maintenance program that checks the file integrity of any disk pack. Under PRIMOS III and IV, the disk to be checked must be ASSIGNED before invoking FIXRAT. FIXRAT is an external command. If the optional keyword OPTIONS is typed, FIXRAT requests printout options; otherwise, it defaults to printing the name and decimal number of records used in the MFD and each directory file in the UFD. After the command line is typed, FIXRAT asks the question: FIX DISK?. If the answer is YES followed by a CARRIAGE RETURN (CR), FIXRAT truncates or deletes defective files and generates a corrected DSKRAT file. FIXRAT truncates or deletes files in the MFD as well as files in other directories. FIXRAT then asks the question: PHYSICAL DISK DRIVE =. The user must respond by entering the number of the physical disk drive on which FIXRAT is to be run in octal followed by a CR. A complete discussion of FIXRAT, along with examples is given in the Computer Room User Guide (MAN 2603).

```
*****
*  FTN  *
*****
```

The format is:

FTN Filename [1/A] [2/B]

The FTN command loads the Prime FORTRAN IV Compiler and starts compilation of an object program from an ASCII source file, Filename, in the current UFD. FTN is an external command. A is the A-Register setting. If no A-Register value is specified, a default value is used; typically, 1707. (List errors on terminal, use disk for all input/output.) Other common options are:

1/1777	List errors on terminal, generate listing file.
1/40777	Generate listing file that includes symbolic listing.
2/10	List errors on terminal and create cross-reference listing.
2/400	Generate Prime 400 64V mode code.

Unless it is preceded by BINARY and LISTING commands, the compiler will automatically open Unit 3 to write a binary file named B XXXX, and open Unit 2 to write a listing file named L XXXX, where XXXX is the first four letters of the input filename. The compiler closes any units that it opens (Units opened by BINARY and LISTING commands are not closed.) The listing file can be printed by using the text editor or the PRMPC, PRSER, SLIST, or SPOOL commands.

PRIMOS II 16K and 24K users must use SFTN, a small version of the FORTRAN compiler.

#### FORTTRAN for the Prime 400

FTN generates code for either Prime 300 or Prime 400, depending upon the setting of a bit in the B register. For a simple example of writing, loading, and executing a program using FTN and LOAD on the Prime 400, refer to the description of the SEG command.

For more information, refer to the Program Development Software User Guide (MAN 1879), the Subroutine Library User (MAN 1880) and the FORTRAN Reference User Guide.

\*\*\*\*\*  
\* FUTIL \*  
\*\*\*\*\*

FUTIL invokes a file utility command that provides subsystem commands for the user to copy, delete, and list both files and directories. FUTIL also has an ATTACH command that allows attaching to subdirectories by giving a directory treename from either the MFD or home UFD to the specified subdirectory. FUTIL allows operations not only with files within UFD's, but also files within segment directories. FUTIL may be run from a command file.

For a detailed discussion of subsystem commands available under control of FUTIL, refer to the File System User Guide (MAN 2604).

A summary of FUTIL commands is listed in Table 3-4.

Table 3-4. FUTIL Commands

<u>Commands</u>	<u>Command Function</u>
QUIT	Returns to PRIMOS command level.
FROM Directory Treename	Defines FROM directory.
TO Directory Treename	Defines TO directory.
ATTACH Directory Treename	Moves the Home UFD to the directory defined by Directory-Treename.
COPY File1 [,File2] [,File3] [,File4].	Copies a file or files in the FROM directory to the TO directory
COPYSAM File1 [,File2] [,File3] [,File4] ,...	Same as COPY, but sets file type of file TO directory to SAM.
COPYDAM File1 [,File2] [,File3] [,File4] ,...	Same as COPY, but sets file type of file in TO directory to DAM.
TRECPY Dir1 [,Dir2] [,Dir3] [,Dir4] ...	Copies directory tree specified.
UFDCPY	Copies all files and directories in the FROM directory to the TO directory.
DELETE File1 [,File2] ...	Deletes the files specified.
TREDEL Dir1 [,Dir2] ...	Deletes specified directory tree.
UFDDEL	Deletes all files and directory trees within the FROM directory.
LISTF [level] [LISTFIL] [PROTECT] [SIZE] [TYPE]	Lists at the terminal the FROM directory treename; the TO treename; and all files and directories in the FROM directory.
SCAN File [Level] [LISTFIL] [PROTECT] [TYPE] [FIRST] [SIZE]	Searches for a file named File in the directory hierarchy and prints the filename and directory treename if a file is found.



\*\*\*\*\*  
\* HILOAD \*  
\*\*\*\*\*

See LOAD.

\*\*\*\*\*  
\* INPUT \*  
\*\*\*\*\*

The format of the INPUT command is:

INPUT Filename

INPUT opens a source file on File Unit 1 for reading. The file is assigned the name Filename in the current UFD. This command has the same effect as OPEN Filename 1 1. (For FMA and FTN, the source filename is usually provided with the command that starts assembly or compilation.) INPUT is an internal command..

```

*****
*   LATE   *           ***PRIMOS III and IV***
*****

```

LATE requests the time at which the next command is to be accepted.  
The format is:

LATE

The LATE command responds as follows:

TYPE IN TIME OF DAY DESIRED TO EXECUTE NEXT COMMAND IN HHMM FORMAT

The user then types in the time of day. The time of the next command is expressed as a number of the form HHMM. HH is the hour (00 through 23), and MM is the minute (00 through 59). LATE responds to this input with the message:

NEXT COMMAND WILL BE EXECUTED AT        HHMM

LATE calls the subroutine RECYCL until the specified time is reached, then it returns to PRIMOS III and IV. No other commands can be executed until this time. LATE is useful if a user wishes to defer execution of a process, such as a command file, until a time when it is expected that system load will be light, such as during the second and third shift.

Example:

```

LATE
TYPE IN TIME OF DAY DESIRED TO EXECUTE NEXT COMMAND
1715
NEXT COMMAND WILL BE EXECUTED AT        1715

```

```

*****
* LBASIC *
*****

```

LBASIC invokes a version of the BASIC interpretive language that contains both MAT functions and PRINT USING functions. Refer to the BASIC User Guide. LBASIC is an external command.

```
*****
* LISTF *
*****
```

LISTF prints the current UFD name, the logical device upon which the UFD resides, and all filenames in the UFD at the terminal. LISTF is an internal command. Attributes of files such as type, size, and protection may be examined using the LISTF subcommand of the FUTIL command.

Example LISTF for PRIMOS II:

OK: LISTF

UFD=JDOAKS 1

FDAT	FATI	PMAT	FLN	ARG	B FDAT	DRAT	B DFAT
B FATI	B ARG	B FLN	SLITE	N66	N22	DIV	B DSUB
B DADD	B MPY	B DIV	B UIIT	UII	MA4	UIIT	NEWMAP
P221	MT1	COMIOC	IOCS	B COMI	B IOCS	C PMA	MYEMA
OLAPRN	B OLAP	PMV2	*UIIT	TRR	P211	F\$UII	QNEWMA
MC1	B FTN2	MYFORT	FTST				

OK:

For PRIMOS III and IV, the LISTF command prints the letter O followed by the device number upon which the UFD resides, if the user is an owner. If the user is a nonowner, the LISTF command prints the letter N followed by the device number upon which the UFD resides. The concept of owner and nonowner is described in the File System User Guide (MAN 2604) under the heading 'File Access', and is associated with the commands PASSWD and PROTEC.

Example LISTF for PRIMOS III or IV:

OK, LISTF

UFD=MAX 5 0

LIB7    PROG#1    PROG#2    DATAFL

OK,

\*\*\*\*\*  
\*LISTING \*  
\*\*\*\*\*

The format is:

LISTING Filename

LISTING opens a file for writing on File Unit 2, usually as a listing output file for the compiler or assembler. The file is assigned the name Filename in the current UFD. LISTING is an internal command, and has the same effect as OPEN Filename 2 2.

#### NOTE

If no LISTING command has been entered, PMA and FTN automatically open a file named L<-XXXX as the listing file (if listing is requested). (XXXX is the first four letters of the source filename.)

\*\*\*\*\*  
 \* LOAD \*  
 \*\*\*\*\*

This command loads and starts LOAD, Prime's Linking Loader for PRIMOS. LOAD has a command structure and, therefore, a single entry point. LOAD is an external command. For an example of the use of LOAD, refer to Section 2. For a complete discussion of LOAD, refer to the Program Development Software User Guide (MAN 1879).

A number of versions of loader are available on the original master disk. The versions of the loader are:

<u>Name</u>	<u>Function</u>
LOAD	(Loader 600000-63777) P Register = 61000 Normally used with 32K PRIMOS II.
LOAD20	(Loader 200000-23777) P Register = 21000 Normally used with 16K PRIMOS II.
HILOAD	(Loader 174000-177777) P Register = 175000 Normally used to load programs longer than 32K.

Other than the function and configuration differences noted above, the internal function and user interface is the same. Any version of the loaders may be used on a system configuration equal to or greater than the one specified. If the user chooses, he can rename his particular load command using the CNAME command after deleting the old LOAD. (e.g., CNAME LOAD20 LOAD).

LOAD loads programs for all addressing modes except 64V segmented mode, which may be generated by PMA or FORTRAN. To load segmented code, use the command SEG.

```
*****
* LOGIN *          ***PRIMOS III and IV***
*****
```

LOGIN is the command the user must type at the terminal to obtain access to the PRIMOS III or IV system. The format is:

```
LOGIN Ufdnam [Passwd] [Ldisk]
```

where Ufdnam must be a valid UFD name on any of the disks available to the system, Passwd is an optional argument that specifies the owner or nonowner password, and Ldisk is an optional argument that specifies logical device numbers to be searched for Ufdnam.

If the UFD has a password, the user may supply it at LOGIN time.

When LOGIN is successful, the user is attached to the UFD specified by Ufdnam. The time-accounting registers for the user are cleared, and some initialization is performed on the user's 'virtual machine' (i.e., VRTSSW, ASRCWD, and SVCSW are initialized), then a login message is printed at the terminal and at the supervisor terminal.

Examples of LOGIN command:

```
LOGIN JHNDOE
```

logs in the user and the UFD, JHNDOE, is attached.

```
LOGIN JHNDOE GEMINI
```

logs in the user and attaches the UFD, JHNDOE, if the password GEMINI is correct. A typical system response to this login at the terminal and at the system supervisor terminal is:

```
JHNDOE(2) LOGGED IN AT 12'39 03046
```

The number in parenthesis is the user number of the user terminal (e.g., in this case, (2)).

The prompt:

```
OK,
```

is printed at the terminal in addition to the login message.

The user may give the ATTACH command, as under PRIMOS II. The UFD name given in the argument to the LOGIN command is remembered and printed upon LOGOUT, no matter which UFD is currently attached. (This information also may be printed by use of the STATUS command.)

```
*****
* LOGOUT *          ***PRIMOS III and IV***
*****
```

LOGOUT is the last command the user issues when giving up access to the system.

During LOGOUT, all user files are closed, all devices ASSIGNED to the user's terminal are released, the UFD is detached, and a logout message is printed at the user's terminal and at the supervisor terminal. Under PRIMOS IV, all segments that were used by the user are returned to the supervisor when the user logs out.

### LOGOUT [-UU]

When the optional parameter -UU is specified at the user's terminal, the user specified by the user number UU will be logged out if the UU is a phantom user that has the same login name as the user at the terminal. When the optional parameter -UU is specified at the system terminal, the user specified by UU will be logged out, regardless of whether UU specifies a phantom or a terminal user.

Example:

### LO

Typical response at the user terminal and also at the supervisor terminal:

```
JHND0E (2) LOGGED OUT AT 13'16 03046
TIME USED = 00'37 03'01 00'54
```

The first number after 'TIME USED = ' is the connect time in hours and minutes; the second number is CPU time in minutes and seconds, and the third number is disk I/O time in minutes and seconds. For more details about time accounting, refer to the TIME command.

If a user is logged out because the inactivity time has run out, the message TIMEOUT is printed at both user terminal and the supervisor terminal, followed by the normal logout message. If the user is logged out by a command issued at the supervisor terminal, the message FORCED LOGOUT is printed at both the user terminal and the supervisor terminal, followed by the normal logout message. If a phantom user terminates because of an error, the message PHANTOM TTY REQUEST is printed at the supervisor terminal followed by the error message that describes the condition that caused the phantom user to terminate.

Example:

```
PHANTOM TTY REQUEST
FILE NOT FOUND
```

```
*****  
* MACHK *      ***PRIMOS II only***  
*****
```

MACHK causes the Prime computer that PRIMOS is configured upon to be run in machine check mode. MACHK is an external command. Unless the command specifies otherwise, PRIMOS II normally operates out of machine check mode; PRIMOS III and IV normally operate in machine check mode.

The MACHK command works only for PRIMOS II.



\*\*\*\*\*  
 \* MAGRST \*  
 \*\*\*\*\*

## GENERAL INFORMATION FOR MAGNETIC TAPE - FILE UTILITIES

MAGSAV and MAGRST are utility programs that move files on any disk including the storage module, to a 9-track magnetic tape and vice versa. The files may be SAM, DAM, segment directories, UFD's, or an entire disk. Whenever a directory is specified, the directory and all components (the subtree) are transferred.

### Logical Tapes

A logical tape consists of a header record, a file mark, file records, and two file marks. A logical tape may span multiple physical tapes or a single physical tape may contain multiple logical tapes. The header record contains the tape name, data, and revision number. All tape records are 512 words long.

### Tree Names

A disk file appears on tape as a record containing a tree name, followed by as many data records as are required for the file. The tree name contains the path from the file specified by the user to the current file. When an entire disk is saved, all tree names begin in the MFD. For example, an ordinary SAM file might have a tree name of MFD>UFD>JUNK> or MFD>UFD>SUBUFD>JUNK.

### USE OF MAGRST

All restore operations take place in the home UFD. MAGRST asks for the tape unit and logical tape number. MAGRST then prints the name, date, and revision on the user terminal and asks:

READY TO RESTORE: The responses are YES, NO, PARTIAL (abbreviated Y, No, PA), \$I Level or NW level. YES restores the entire tape. NO causes a request for another tape unit and logical tape combination. PARTIAL permits a restore of part of the tape. NW followed by a level number gives an index of the magnetic tape, but does not write it to disk.

TREE NAME: This is typed when a partial restore is requested. The response is in the form:

NAME1>NAME2>....NAME<sub>n</sub>

A file on the tape with a tree name beginning with the sequence entered is restored.

## Example:

Tree names in a save of the entire disk begin with MFD. The tree name to restore UFD would be MFD>Ufdnam. The tree name to restore a file would be MFD>Ufdnam>Filename.

MAGRST provides the ability to enter multiple tree names for a partial restore. For example, the tree names are typed in response to the query TREE NAME: as follows:

```
TREE NAME:      MFD>LIB>FTNLIB
TREE NAME:      MFD>LIB>JUNK
TREE NAME:      MFD>CMDNC0>PRINT
TREE NAME:      null line
```

After each file is restored, the message:

FILE COMPLETE

is printed at the terminal. For a partial restore, files that have bad records are omitted. The tree names of these files are printed along with an error message. The message:

RESTORE COMPLETE

is printed when the end of logical tape is reached.

MAGRST checks for conflicting file types when a file is going to be overwritten. Conflicts generate an error message, and the file is skipped.

MAGRST has been modified to avoid having to read through all logical tapes when restoring sequential logical tapes. After MAGRST has exited to the PRIMOS supervisor, the magnetic tape is not rewound. Instead, it is positioned at the location before the beginning of the next logical tape in sequence. In the case of sequential logical tapes, the user must run MAGRST again and specify 0 for logical tape number where LOGICAL TAPE NO: is requested.

Then, the next logical tape is restored without rewinding and reading through the preceding logical tapes.

### Index

MAGRST allows a user to index a tape and direct the listing of the index to a disk file rather than the user terminal. To use this feature, follow the NW or the I command with a filename, and then with the number of index levels. (MAGSAV does not support this feature.)

Example of index with MAGRST:

READY TO RESTORE: NW TAPE#1 5 Filename

In this case, MAGRST writes an index to the file specified by Filename.

#### Physical End of Tape

When physical END OF TAPE is encountered in either MAGSAV or MAGRST, a message is logged on the user terminal and a new tape unit is requested. The new unit may be the same as the old unit.

#### Errors

Tape read or write errors are retried five times and are then considered unrecoverable. Both recovered and unrecovered errors are logged. The first record on a tape is not retried.

#### Assigning Tapes

When running MAGSAV under PRIMOS III or IV, the magnetic tape drive must be assigned. Refer to the description of the ASSIGN command for further details. Users must avoid RESTORing files into UFD's that are in use by other users because this action could either confuse the other users or could cause either MAGSAV or the user program to abort with the message:

FILE IN USE

In this case, the abort is caused by two different programs attempting to gain access to the same file at the same time.

## Examples:

OK, MAGRST  
GO

TAPE UNIT: 1  
ENTER LOGICAL TAPE NUMBER: 1  
NAME: MCARCH  
DATE: 05-06-75  
REV NO: 1  
REEL NO: 1  
YOU ARE NOT ATTACHED TO AN MFD  
READY TO RESTORE: NW 3

Response to NW 3 is to list the contents of MCARCH as follows:

MFD > CMDC0  
MFD > PRIMOS II  
MFD > CPU  
MFD > CPU > OP3FLT  
  
MFD > SMC > M5374K  
INDEX COMPLETE  
OK:

~~Tape~~ is not restored to disk in this case.

Another example of MAGRST:

OK: MAGRST  
GO  
TAPE UNIT: 1  
ENTER LOGICAL TAPE NUMBER: 1  
NAME: ADMIN  
DATE: 05-23-75  
REV NO: 1  
REEL NO: 1  
READY TO RESTORE: YES  
  
OK:

\*\*\*\*\*  
 \* MAGSAV \*  
 \*\*\*\*\*

The same general information that applies to MAGRST also applies to MAGSAV.

MAGSAV requests information in the following order:

TAPE UNIT:	The proper response is the physical unit number of the tape (0-3).
ENTER LOGICAL TAPE NUMBER:	The response is 1 for the first logical tape, 2 for the second, etc. MAGSAV rewinds the tape, then positions itself correctly. A response of 0 implies the tape is already positioned correctly and MAGSAV takes no action.
TAPE NAME:	Any six-character name.
DATE:	The response format is MMbDDbYY where b represents a space and MM=month, DD=day and YY=year. The date is checked for validity and rejected if it is not valid. For example, 07 35 03 would be rejected.
REV NO:	An arbitrary number.
NAME:	NAME asks the user what to save. The response is either a file name or one of the alternate action commands: \$A, \$I, \$Q. \$A changes the home UFD via an ATTACH; e.g., \$A USER3 PASSWD5. \$Q and \$R each terminates a logical tape and returns to the operating system. \$R also rewinds the tape. \$I causes an index to be printed. \$I followed by a blank and level number indexes to the level indicated. Thus, MAGSAV has the ability to index tape by tree name to any level. For example, \$I 3 prints an index of the MFD, any UFD's and any Filenames. NAME: is printed whenever writing has been completed, so that further writing may be requested or the current logical tape may be terminated. If the user does not respond correctly to the query NAME, or when the operation is complete; MAGSAV again asks NAME:. The user must then give another action command.

To save an entire disk, the user must respond to the query NAME with MFD. To save a UFD, the user must attach (\$A) to the MFD and give the

name of the UFD that is to be saved. To save a file with the UFD, the user must attach to the UFD (e.g., \$A Ufdnam) and give the name of the file. MAGSAV also saves a disk that contains nested segment directories.

MAGSAV handles the 1040-word record size on the storage module as well as PRIMOS 448-word record size. MAGSAV also works with UFD's that contain up to 169 entries.

## NOTE

UFDs with more than 72 entries must be restored to a 1040-word record size device (i.e., storage module). Otherwise, MAGRST loops.

If running under PRIMOS III or IV, the magnetic tape must first be assigned using the ASSIGN command. Files or directories are in simultaneous use by other users must not be accessed by MAGSAV. An attempt to do so results in a backup tape with files that contain either partially written or partially updated information. Furthermore, there is a high probability that MAGSAV or other user programs will abort with the message:

## FILE IN USE

since the two programs compete over which program is to gain access to the same file.

## Examples:

```
OK: MAGSAV
GO
TAPE UNIT: 1
ENTER LOGICAL TAPE NUMBER: 0
TAPE NAME: DUD
DATE: 05 23 75
NAME: MFD
NAME: $R
OK:
```

## Another example of MAGSAV:

User input is underscored.

```
OK, STARTUP 50 52
OK, A MFD XXXXXX
OK, MAGSAV
GO
```

```
TAPE UNIT: 0
ENTER LOGICAL TAPE NUMBER: 0
```

TAPE NAME: MD10V1  
DATE: 05 18 75  
REV NO: 2  
NAME: \$I 3  
NAME: MFD

MAGSAV responds to each command input and again asks for name.

During the save, MAGSAV lists directories and files saved.

Example:

MFD  
MFD>CMDNC0  
MFD>PRIMOS  
MFD>CPU  
MFD>CPU>OP3FLT  
MFD>CPU>P221B  
...  
MFD>LPRCDR  
MFD>LPRCDR>PCRDO3

When the listing is complete, MAGSAV again asks:

NAME: \$R                      The MAGSAV rewind command causes the tape  
to be rewound and exits to command level.

OK,

\*\*\*\*\*  
 \* MAKE \*  
 \*\*\*\*\*

MAKE creates a disk for any disk type supported by PRIMOS. MAKE may be run from a command file. Under PRIMOS III or IV, the disk to be created by the MAKE command must be ASSIGNED before MAKE is invoked. MAKE should be used to create a partition of a large disk to be used for paging, as well as partitions and disks to be used for files. MAKE creates a PRIMOS disk that has the following:

DSKRAT  
 MFD  
 BOOT  
 DOS (which is empty)  
 CMDNC0 (which is empty)

The MAKE program writes the bootstrap (BOOT) into Record 0 of the disk.

To run MAKE, type the command:

MAKE

Since MAKE is a command most frequently used by system operators, complete details of MAKE operation and examples are given in the Computer Room User Guide (MAN 2603). For further information, refer to the Computer Room User Guide.

\*\*\*\*\*  
 \* MCG \*  
 \*\*\*\*\*

The format of the MCG command is:

MCG Filename

MCG translates results of microcode assembly into proper code for the ROM simulator. MCG is an external command. For information on microcode, refer to the Microcoders Handbook (MAN 1857).

\*\*\*\*\*  
 \* MDL \*  
 \*\*\*\*\*

MDL punches paper tapes of specified sections of memory in a self-loading format that can be read by the panel LOAD operation (or equivalent operation). MDL tapes load into the same memory locations from which they are punched. Tapes can be punched using locations as low as '34. MDL is an external command; refer to the Program Development Software User Guide (MAN 1879).



```
*****  
* MESSAGE *      ***PRIMOS III and IV***  
*****
```

The command:

```
MESSAGE  
text of message
```

issued by the user at the user terminal results in two lines of printed information at the system terminal. The format of this line is:

```
***UU HH'MM  
text of message
```

where UU is a user number and HH'MM is the time of day in hours and minutes.

Example:

Assume the user inputs the following at the user terminal:

```
MESSAGE  
PLEASE MOUNT SCRATCH TAPE
```

The following message is printed at the system terminal:

```
*** 25 17'50  
PLEASE MOUNT SCRATCH TAPE
```

For further information, refer to The computer Room User Guide (MAN2603).

The MESSAGE command may be used to send messages across the computer network from the terminal user. For example, the command:

```
MESSAGE SYSA
```

followed by the text of the message will send the message from the user terminal to the supervisor terminal of the remote system, if it is SYSA.

If the MESSAGE command is issued at the supervisor terminal of a given system, it can only be used to send a message to all the users in the remote system, but not a particular user.

\*\*\*\*\*  
\* MTDSK \*  
\*\*\*\*\*

MTDSK is a command to read and write magnetic tape to and from disk, especially convenient to users of 7-track tapes. When invoked, MTDSK asks the user a series of questions; his responses direct the tape-to-disk or disk-to-tape operation.

\*\*\*\*\*  
\* NUMBER \*  
\*\*\*\*\*

NUMBER invokes a utility program that numbers or re-numbers statements in a BASIC program. NUMBER is an external command. For further information, and for an example, refer to the BASIC User Guide.

\*\*\*\*\*  
 \* OPEN \*  
 \*\*\*\*\*

The format of the OPEN command is:

OPEN Filename Funit Key

OPEN opens the specified File Unit (1-16), associates it with the specified Filename, and assigns a Status according to the Key. OPEN is an internal command.

The argument, Key, for OPEN is significant. Key parameters consist of octal values for the type of file, and the action to be taken when the file is opened. The format of Key is as follows:

<u>Bits</u>	<u>Meaning</u>
1-5	New file (file type) key, octal values are: 000X File is sequential threaded file (SAM) 200X File is sequential directed file (DAM) 400X File is a SAM segment directory 600X File is a DAM segment directory 1000X File is a Ufd
11-16	Action Key, octal values are: 1 Open for reading 2 Open for writing 3 Open for reading and writing

```
*****  
* OPRPRI *      ***PRIMOS III and IV***  
*****
```

When the command OPRPRI (operator priorities) is issued at the systems terminal, it allows certain other commands to be issued at the same terminal. The format of this command is:

<u>OPRPRI</u>		1	
		0	

Under default conditions, the commands START, RESTOR, RESUME, LOGIN, LOGOUT, and any external command may not be issued at the system terminal without the command line OPRPRI 1. The command line, OPRPRI 0, resets the protection against these commands.

\*\*\*\*\*  
\* PASSWD \*  
\*\*\*\*\*

The format of the PASSWD command is:

PASSWD Owner-password [Nonowner-password]

The password command replaces any existing passwords in the current UFD. However, under PRIMOS II, only the owner password may be given.

Example:

OK: ATTACH JHNDOE  
PASSWD US

Under PRIMOS III and IV, the PASSWD command replaces any existing passwords in the current UFD with two new passwords. The first is the owner password: the second is the nonowner password. The nonowner password is optional. If it is not specified, the nonowner password becomes blanks. The PASSWD command must be given by the owner while attached to the UFD. A nonowner cannot give this command.

Example:

OK, A JHNDOE OLDPW  
OK, PASSWD US THEM

If a nonowner attempted the above PASSWD command, the message:

NO RIGHT

is printed.

#### CAUTION

Unless the user takes special precautions, MAGRST cannot restore UFD's with passwords unknown to the user of MAGRST.

```
*****
* PHANTOM *          ***PRIMOS III and IV***
*****
```

The format is:

PHANTOM Cifile [Unit]

where Cifile is a command file and Unit is an optional file unit number.

Under PRIMOS III and IV, a user may initiate, at the user terminal, a phantom user to perform a job. A phantom user is similar to any other PRIMOS user, except that the phantom user has no terminal associated with it; all controlling input is read from a command file instead of a user terminal. The file specifies the sequence of commands and/or user program invocations and necessary input data specifications to complete a given job.

The phantom user feature is useful for running programs that are not interactive, and therefore do not require the services of a terminal.

When PRIMOS III and IV are started up, a fixed number of phantom users is specified by the CONFIG command. For example, the line printer spooler, SPLMPC, or the card reader spooler, SPLCRD, may be run as phantom user, thereby releasing as many as two terminals for interactive work. Also, a user may run a long FORTRAN compilation or a long SORT as a phantom user and free one terminal for interactive work.

#### Startup of Phantom

The operator must specify the number of phantom users by use of the CONFIG command. The format of this is as follows:

```
CONFIG Trmusers Pagedev Comdev ... 6/Pusers
```

where:

Trmusers	Specifies the number of terminal users (in octal).
Pagedev	Is the paging device (refer to the Computer Room User Guide MAN 2603).
Comdev	(Refer to The Computer Room User Guide).
Pusers	Specifies the number of phantom users (in octal).

The parameters Trmusers + Pusers cannot exceed 16 for 15-user PRIMOS III or 32 for the 31-user PRIMOS III or IV. For each Puser specified at CONFIG time, half a page of memory is locked; therefore, it is not

available for terminal users. Phantom users require disk paging space, the same as any other user. Furthermore, when phantom users run, they get a full-time slice of CPU time and compete with other users for memory in which to run. The administrator of the PRIMOS system should be aware of the costs when specifying the number of phantom users.

#### User Invocation of PHANTOM

The command to start a Phantom user is:

PHANTOM Cifile [Unit]

where Cifile is the filename of the command input file, and Unit is an optional file unit number.

The PHANTOM command checks if there is a user number available for a phantom user to be logged in. If no user numbers are available, the message:

NO FREE PHANTOMS

is printed at the user terminal. If an address space is available for use by a phantom user, the phantom user is logged into the user file directory that is the login directory of the user who invoked the phantom. At this point, an ATTACH operation is made by the phantom feature of the operating system to the user's current UFD. The command file (Cifile) specified in the phantom command is opened on File Unit 6, or is opened on the File Unit specified by the optional argument Unit. PRIMOS III and IV take all further commands from the file specified by Cifile, similar to the COMINPUT command operation.

A program running as a phantom user must not perform any terminal I/O. An attempt to read from a terminal causes the command file, Cifile, to abort and causes the phantom user to be logged out. If this condition occurs, the logout message at the system terminal is preceded by the line:

PHANTOM TTY REQUEST

Any terminal output that is generated by the phantom user program or directed to the user terminal by system commands, such as LISTF, is ignored.

An error that causes the command input file, Cifile, to abort also causes the phantom user to be logged out. The last command in Cifile must be LOGOUT (instead of COMINP TTY).

A user may monitor the status of any phantom user that is initiated at the user terminal by invoking the STATUS command. For each phantom user that is logged into the same login UFD, STATUS prints the UFD name followed by the user number of that phantom in decimal. When the user PHANTOM job is complete, it is logged out. Therefore, it will no longer appear in the output printed by the STATUS command. If a user

wishes to stop a phantom user that was started at his terminal, the command:

LOGOUT -UU

must be issued, where UU is the number of the phantom user as reported in the STATUS command.

A user may log out, return later, and log in to the same UFD. The STATUS and LOGOUT commands may be used as before to control the phantom.

Any phantom or user may be logged out by use of the LOGOUT command at the supervisor terminal.

The PHANTOM command may be issued from a command file. Command files running in phantoms may also include PHANTOM commands.



```
*****
*   PM   *
*****
```

### Overview

The PM(POST MORTEM) command is used to print the contents of the RVEC vector (described later in this section). PRIMOS first prints labels for the items in RVEC, then prints the values on the line in the same order. PM is an internal command.

Example:

```
OK: REST CSETV1
OK: PM
SA,EA,P,A,B,X,K=
000100 011100 001000 000000 000000 000000 000000
OK:
```

### PM for PRIMOS IV

The Prime 400 contains more registers than other Prime CPU's that must be SAVED and RESTORED when PRIMOS switches to a new user. The PM command for the PRIMOS IV operating system is modified to display some of these additional registers: the procedure base register (PB), the stack base register (SB), the link base register (LB), and the temporary base register (XB). These 32-bit registers are displayed at the user terminal on a text line separate from the other registers. Each of the Prime 400 registers is displayed as two 16-bit octal numbers separated by a slash (/) character.

Example:

```
OK, PM
SA, EA, P, A, B, X, K, =
200 12111 3043 0 0 0 6000
PB, SB, LB, XB:
64000/3043 0/0 0/0 0/0
```

The above example of PM under PRIMOS IV shows a PB of 64000/3043, which indicates: ring 3, segment 4000 octal. The word number portion of PB indicates the same number as the P parameter of PM. This number, which is the same as the P parameter, specifies the location within the segment to execute the next instruction upon possible receipt of a START command. The other base registers shown in the example contain a 0, indicating that they have not been used since LOGIN. Programs that run in one of the Prime 300 addressing modes use segment 4000 ring 3, and give values as a result of invoking PM in the form shown by the example.

```
*****  
* PMA *  
*****
```

The format is:

PMA Filename [1/A]

PMA loads the Prime Macro Assembler and starts assembly of a source file Filename from the current UFD. A is the A-register setting that specifies listing detail and input/output devices. If A is not specified, the default value is:

000777 Normal listing detail, all input  
and output files on disk.

For other values, refer to the Program Development Software User Guide.

Unless it is preceded by BINARY and LISTING commands, the assembler automatically opens Unit 2 to write a binary file named B XXXX, and opens Unit 3 to write a listing file named L XXXX, where XXXX is the first four letters of the input filename. The assembler closes any units that it opens. (Units opened by BINARY and LISTING commands are not closed.) PMA is an external command.

\*\*\*\*\*  
\* PRERR \*  
\*\*\*\*\*

PRERR prints the message stored in ERRVEC and the first six locations of ERRVEC in octal. The PRERR command is useful in debugging a program. On encountering an error condition, PRIMOS sets up an internal vector called ERRVEC with several pieces of information. One of these pieces is an error message. Refer to Appendix C for a description of ERRVEC.

Using the system subroutine ERRSET (refer to the File System User Guide), a user may set the content of the error message and have the message printed or not printed, depending on the alternate return being zero or nonzero in a user subroutine. If the user routine was the last routine to set ERRVEC, PRERR prints the user-stored message.

Example:

```
ER! PRERR
140710 0 155311 146303 120240
FILEA NOT FOUND
```

```
*****
* PROTECT*          ***PRIMOS III and IV***
*****
```

A user (hereafter called the owner) has the ability to open his file directories to other users, giving restricted access rights to his files. This declaration of access rights can be made on a per file basis. Access rights to a file are declared and specified through the PASSWD and PROTEC commands. The format of this command is:

PROTEC Filename Key1 Key2

Filename Is the name of the file to be protected.

Key1 Is an integer that specifies the owner's access rights to Filename.

Key2 Is an integer that specifies the nonowner's access rights to Filename.

Possible values and their meaning for Key1 and Key2 are:

- 0 No access of any kind allowed
- 1 Read only
- 2 Write only
- 3 Read and write
- 4 and truncate
- 5 Delete, truncate and read
- 6 Delete, truncate and write
- 7 All

Example:

```
OK, PROTEC MYPROG 7 1
OK, PROTEC OLDIS 7 7
```

Gives the owner all access rights to MYPROG, nonowners read-only access rights to MYPROG, and gives both owners and nonowners all access rights to the file OLDIS.

#### CAUTION

MAGRST does not restore protected files. The default protection keys associated with any newly created file or UFD, 7 7 (owner and nonowner are given all rights).

The following example is intended to give a user an idea of the use of the PASSWD and PROTEC commands:

LOGIN JHNDOE

JHNDOE (2) LOGGED IN AT 10'25 02255

OK, PASSWD JHNDOE US THEM

Gives owner password US and  
nonowner password THEM to  
UFD JHNDOE.

OK, LISTF

UFD=JHNDOE 2 0

TIMING FUNCT MNEMOS MYPROG OLD

OK, PROTEC TIMING 7 0

Gives JHNDOE all access,  
nonowners no access to TIMING.

OK, PROTEC MYPROG 7 1

Gives owner access = all,  
nonowners access = read.

OK, PROTEC OLD 7 7

nonowners access = all,  
owner access = all. (These are also  
the default values of PROTECT if  
not specified)

OK, LO

JHNDOE (2) LOGGED OUT AT 10'34 02255

TIME USED = 00'03 00'04 00'01

OK, LOGIN MSMYTH

MSMYTH (2) LOGGED IN AT 11'34 02255

OK, ATTACH JHNDOE THEM

OK, LISTF

UFD = JHNDOE 2 N

TIMING FUNCT MNEMOS MYPROG OLD

OK, DELETE TIMING

TIMING NO RIGHT

MSMYTH, a nonowner, cannot  
even read timing since he has no  
access.

ER! ED MYPROG

MSMYTH can enter editor and  
read MYPROG since read access  
has been granted.

GO

P2;

.NULL.

C PROGRAM TO TEST DATA

C JOHN DOE 02 02 75

INPUT

C WITH CHANGES INSERTED  
BY MSMYTH

MSMYTH attempts to change  
MYPROG; he seems to have  
succeeded.

EDIT

FILE MYPROG

MYPROG NO RIGHT

?

Q

OK,

ED OLD

GO

i

INPUT

C CHANGES BY MSMYTH WILL BE RECORDED HERE

i

EDFT

FILE OLD

OK,

Cannot change file because  
write access is denied  
by JHNDOE.  
Might as well quit.

As all access has been  
granted for OLD, changes  
are made successfully.

```
*****
* PRVER *          ***PRIMOS III and IV***
*****
```

PRVER prints a file on a Versatec Printer/Plotter that may be configured to a Prime computer system running under PRIMOS III or IV. The format is:

PRVER Filename

Details of use are the same as for the PRMPC command described in this section. The plotter must be assigned before the PRVER command can be issued. The PRVER command does not run under PRIMOS II.

Example:

```
OK, ASSIGN PLOT
OK, PRVER FILEA
```

assigns the Versatec Printer/Plotter and prints the file, FILEA, at the printer/plotter.

```
*****
* PSD *
*****
```

PSD loads and starts Prime Symbolic Debug, an interactive debugging program that assumes control and waits for a command string. For details, refer to the Program Development Software User Guide. To return to PRIMOS, type Q and carriage return at the user terminal. PSD occupies location 6000 to 64777.

```
*****
* PSD20 *
*****
```

PSD20 is a version of PSD for 16K PRIMOS II. PSD20 occupies locations 20000 - 24777. PSD20 also supports the TRACE feature.

```
*****
* PSD160 *          ***PRIMOS IV***
*****
```

PSD160 invokes a version of PSD for patching PRIMOS III and IV run files.

\*\*\*\*\*  
\* PTCPY \*  
\*\*\*\*\*

PTCPY is a utility program that duplicates and verifies paper tapes using the high-speed reader-punch. Operation is controlled by P-register and sense switch settings. PTCPY is an external command. For details, see the Program Development Software Guide.

To return to PRIMOS II, restart the processor from the systems terminal at location '30000, '50000, or '70000, or `170,000 depending on system c configuration. Under PRIMOS III or IV, the command ASSIGN PTR and ASSIGN PUN must be given before PTCPY is invoked.



```
*****
*  PUSS  *
*****
```

The format is:

### PUSS

The PUSS command then prints questions for user response (refer to the example below).

PUSS is a source compare program. It generates a difference file that describes the differences between two input files. The difference file consists of commands and text that could be used to transform the old input file to the new input file. The following is a sample difference file:

```
#OMIT 11, 13
#NCOPY
THESE TWO LINES ARE IN THE OLD FILE
BUT NOT THE NEW ONE
#COPY
THESE THREE LINES REPLACED THE TWO
LINES AT THE SAME POSITION IN THE
OLD FILE
#NSRT 55
THESE LINES ARE IN THE NEW FILE
BUT NOT THE OLD FILE
DONE
```

The above difference file is interpreted as follows: lines 11 through 13 of the old input file containing the text following #NCOPY command are replaced in the new file by the lines following the #COPY command. The two lines following #NSRT 55 are inserted in the new input file following line 55 of the old input file. #DONE indicates the end of the difference file. The following is an example of using the PUSS command.

Example:

```
user:      PUSS
response:  GO
           DIFF FILE, OMISSIONS?
user:      DIFFIL, YES
response:  OLD-FILE TREE NAME:
user:      OPROG
response:  NEW-FILE TREE NAME:
user:      NPROG
response:  (after a Pause) DONE
OK,
```

The response to the request for file tree name may be the name of a file in the current UFD, or it may be a general tree name. The form of a general tree name is given in the File System User Guide (MAN 2604). PUSS has a buffer that contains the differences between the old and new files i.e., #NCOPY, #COPY, or #NSRT. If the differences between the files are too large, the buffer becomes full. PUSS then prints the last three lines of the buffer and asks:

IS IT A REPLACE, INSERT, OR SHOULD I QUIT?

The user must respond: REPLACE, INSERT, or QUIT. Upon REPLACE or INSERT, PUSS dumps its buffer to the difference file and continues processing.

```
*****
* RESTORE *
*****
```

The format is:

RESTOR Filename

The RESTOR command restores a program Filename in the current UFD from disk to high-speed memory using the SA and EA values SAVED with the file. The SAVED RVEC parameters (refer to next side head) are also loaded into RVEC to be ready for a START command. RESTORE is an internal command.

Do not use RESTOR to restore a 64V segmented mode run file. Use the RESTOR command that is available under SEG instead. Refer to SEG, in the section, for further details.

Example:

```
OK, REST *GENFIL
OK: PM
SA: EA,P,A,B,X,K=
000200 011710 001000 075072 000001 177771.006001 .
OK:
```

#### RVEC Parameters

The commands RESTORE, RESUME, SAVE, PM, and START process a group of optional parameters associated with the PRIMOS RVEC vector. These parameters are stored on disk along with a starting address (SA) and ending address (EA), for every program saved by the SAVE command.

Initial values for the RVEC parameters are usually specified in the PRIMOS SAVE command, or by the loader's SAVE command that stored the program on disk.

Each parameter is a 16-bit processor word identify, represented by up to six octal digits.

Prime 100/200/300/400 Memory		
<u>Parameter</u>	<u>Location</u>	<u>Definition</u>
SA	-	Starting Address (first memory word used by program)
EA	-	Ending Address (last memory word used by program)
PC	7	P Register (Program Counter)
A	1	A Register (Arithmetic)
B	2	B Register (Arithmetic)
X	0	Index Register
Keys	--	Status keys associated with INK, OTK instructions OTK instructions

The RVEC parameters are optional in the command string. Any item that is specified replaces the previous value in RVEC, which is saved with the program. Thus, for any parameters that are not specified, the value previously stored in RVEC is saved with the program.

RVEC parameters specified in RESUME or START commands replace the previous values in RVEC. Also, when a program returns to PRIMOS through the EXIT subroutine, RVEC is loaded from the processor values in effect at the time of exit. Only the SAVE command alters the values of RVEC stored on disk with the program.

RESTORE returns a program from disk to memory and loads the SAVE parameters into RVEC in preparation for a START command.

RESUME combines the functions of RESTORE and START.

PM lists the current values of the RVEC parameters.

External commands have RVEC parameters that can be modified at the time the command is started (e.g., PMA Filename 1/740).

### Keys

The item [Keys] among the RVEC parameters refer to the processor status keys handled by the INK and OTK instructions. (Refer to the Systems Reference Manual, MAN 1671). These are represented by a single 16-bit word in the following format:

C		P		*		ADR		*		shift count						
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	

where:

C = State of C (Carry) bit

P = Arithmetic mode; 0 - single precision,  
1 double precision

\* = Must be zero

ADR = Addressing Mode:

Bits 5-16 Mode

0XXX 16K Sector

2XXX 32K Sector

6XXX 32K Relative

4XXX 64K Relative

Shift Count = Bits 9-16 of Location 6, which may  
contain a normalized shift count

If [Keys] are not specified, they are unchanged.

\*\*\*\*\*  
\* RESUME \*  
\*\*\*\*\*

The format of RESUME is:

RESUME Filename [PC] [A] [B] [X] [KEYS]

RESUME is equivalent to a combined RESTORE and START command. The program Filename in the current UFD is loaded from disk to memory, using the SAVED values of SA and EA. RVEC is loaded from the SAVED RVEC parameters or from any new values specified in the command string. The processor registers and keys are then set from RVEC and the program is started at location PC. RESUME is an internal command.

Do not use RESUME to resume a 64V segmented mode program. Use SEG instead. For further details, refer to SEG in this section.

\*\*\*\*\*  
\* RUNOFF \*  
\*\*\*\*\*

RUNOFF, Prime's Text Formatter, accepts commands on-line or edited into text to control margins, indention, line spacing, column width, page numbering, running heads, and many other features of an ASCII source file. It produces a formatted output to SPOOL or a designated disk file. Table 3-5 is an alphabetical list of all RUNOFF commands. For detailed information, refer to the Text Processing User Guide (MAN).

Table 3-5. Alphabetical List of RUNOFF Commands

Command	BRK	EJT	Default	Meaning
.NULL.				In command mode only.
.* text				Start processing
.+ text	Y			Comment Line
.> text	Y			Enter verbatim text
./-/-/-/	Y			Center text
.. text				Apportion text as header
.A-DJUST	Y		Yes	Command escape (.. ->.)
.B-REAK	Y			Enter adjust and fill mode
.BL-ANK			.NULL.	Break
				Define blank substitute char
.BM-ARGIN	Y	Y	5	Set bottom margin
.C-COLUMNS	Y	Y	1	Set number of columns
.CM-ARGIN	Y	Y	5	Set column margin
.D-EFINE sym value				Define symbol with value
.DD-OWN text	Y		No	Down a decimal level
.DDS-UP text	Y		No	Down a decimal level
.DI-NDENT l m1 m2				Specify decimal indent increment
.DL-EVEL "l"				Go to decimal level specified by "l"
.DN-EXT text				Next block on current decimal level
.DNS-UP text				Next block on current level
.DS-KIP l n1 n2				Specify decimal skip amounts
.DU-P k				Go up decimal level(s)
.EF-OTER /-/-/-/				Define footer on even pages
.EH-EADER /-/-/-/				Define header on even pages
.E-JECT	Y	Y		Page eject
.ER-ASE				Define command mode erase char
.ERR-GO				No prompt on error
.F-ILL	Y		Yes	Enter fill mode
.FO-OTER /-/-/-/				Define footer on all pages
.FILE fname			fname=blank	Define output file
.FR-OM i			1	First page number to print
.H-EADER /-/-/-/				Define header on all pages
.HY-PHEN			.RUBOUT.	Define phantom hyphen char
.I-NDENT			5	Indent left margin
.INDEX string				Write string + page # to index
.INS-ERT fname/i			fname=blank	Insert input file
.IX-FILE fname			unit 16 open	Define index



.K-ILL			?	file
				Define command mode kill character
.L-ENGTH	Y	Y	66	Physical page length
.N-FILL	Y			Leave file and adjust mode
.NFILE				No output to file
.NA-DJUST	Y			Leave adjust mode
.NE-ED	?	?	1	Need n printing lines for text
.NER-RGO				Prompt on error
.NIX				Do not process index commands, runoff text only
.NP-ARAGRAPH			Yes	No paragraph indentation
.NPAU-SE			Yes	Do not pause between pages
.NPE-RFORATE			Yes	Do not print perforation mark
.NT-TY			Yes	Do not print on terminal
.OF-OOTER /-/-/-/				Define footer on odd pages
.O-HEADER /-/-/-/				Define header on odd pages
.P-ARAGRAPH m	Y		previous mode	Start paragraph, indent m
.PAG-EN			1	Define starting page number
.PAU-SE				Pause between pages
.PE-RFORATE				Print perforation mark
.PI-CTURE n	?	?	1	Need n lines for picture
.Q-UIT	Y	Y		Exit

The following commands, if typed at the terminal, are the same as not typing any commands (i.e., each specifies a default condition explicitly).

```

$ LENGTH 66
$ WIDTH 85
$ COLUMNS 1
$ TMARGIN 7
$ BMARGIN 5
$ SMARGIN 7
$ PAGEN 1
$ FROM 1
$ TO 32767
$ SPACE 1
$ ADJUST
$ TAB
$ NPARAGRAPH
$ FILE
$ NTTY
$ PAUSE
$ NPERFORATE
$ PARAGRAPH
$ .null (C/R only)

```

```
*****
*  SAVE  *
*****
```

The format of SAVE is:

SAVE Filename SA EA [PC] [A] [B] [X] [KEYS]

The SAVE command saves the content of memory from SA (starting address) to EA (ending address) as a file named Filename in the current UFD. SAVE is an internal command.

As discussed, under RESTORE, the contents of the PRIMOS vector RVEC are saved along with the program. RVEC may be altered by new parameters specified in the SAVE command string before the program and parameters are stored. For any parameters that are not specified, the previous values of RVEC remain in effect and are stored with the program. The RVEC parameters are used to initialize the processor registers and keys when the program is RESTORED or RESUMED.

Example:

```
SAVE PROG1 200 2600 1000 0 0 0 0
```

This command saves the program PROG1 from locations '200 to '2600. Execution starts at 1000, the A, B, and X registers are set to 0, and all bits of the keys are set to 0 (carry bit is 0, arithmetic mode is single-precision, addressing mode is 16K sector, and shift count is zero). Start this save at '200 to preserve address links in Sector 0.

All FORTRAN programs begin with ELM, Enter Load Mode. If macro assembler (PMA) users have ELM as the first instruction in the program, there is no need to set the keys after loading. The preferred way to save a memory image is to use the loader SAVE command.

Do not use SAVE to save 64V segmented mode run files. Use the SAVE subcommand fo SEG instead. Refer to SEG for details.

```
*****
*  SED  *
*****
```

SED invokes a smaller version of the editor. Users of 16K and 24K PRIMOS II must use SED. It is assumed that when SED is copied from the master disk, it has been renamed ED. For further details, refer to the description of ED in this section.

\*\*\*\*\*

\* SEG \*

\*\*\*PRIMOS IV\*\*\*

\*\*\*\*\*

The SEG command invokes a utility program for loading and running segmented programs or making modifications to segmented run files.

Its format is:

SEG Filename

This command RESTORes and executes the program or segmented run file specified by Filename, that was previously LOAded and SAVED by the SEG command. The action of SEG is similar to RESUME, except that the file RESTORed and executed is a PRIMOS IV saved memory image; it is not a PRIMOS II or III saved memory image.

SEG is used under PRIMOS IV only. PRIMOS IV assigns memory segments to a user as they are accessed, and they are retained by that user until log-out. There are 64 segments available for assignment. Up to 16 may be assigned to a single user. Unless a user is executing or examining a segmented program, segments should not be invoked. Most of the sub-commands that operate under control of SEG use only one 64K Prime 300-type segment. Operations that RESTOR a run file use more than one segment (refer to the example in the following paragraphs).

When a user initially gives the LOGIN command, he has no segments assigned. When a user runs a program or invokes an external command, a hardware segment trap occurs and the supervisor allocates a segment to a user if available and retreats his program. If no segments are available, the error message NO AVAILABLE SEGMENTS is given. Additional segments are allocated to a user if his program addresses them. All segments belonging to a user are released to the supervisor on a LOGOUT.

The segments 0 to 2047 are reserved for the supervisor and 2048 to 4095 are reserved for each user. At Rev 11, the supervisor uses segments 0 and 1, and segments 2048 to 2603 are available to each user. A user may address the supervisor segments but, if he does, gets the error message ACCESS VIOLATION. This occurs because the user runs in ring three, and the segment descriptor words for supervisor segments allow no access for ring 3 procedures. A user attempt to address non-existent segments generates the error message ILLEGAL SEGMENT NO. Note that the PRIMOS PM command and the load map generated by SEG print the segment number in octal. The segments available to users in octal are 4000 to 4017. Normally, the user does not care in which segments his program is loaded. On large programs, the PRIME 400 loader VLOAD will automatically assign segment numbers as needed. Users may be interested in these numbers during debugging. SEG is a command that can be invoked in either of two ways:

SEG Filename

is equivalent to

R Filename

for a Prime 300 run file. The run file is loaded into segmented memory and execution started.

If the user enters:

SEG

only, the other commands available under control of SEG may be invoked. These commands are similar to the LOAD commands and are (abbreviations are underlined):

<u>PARAMS</u>	Display SAVE parameters.
<u>SAVE</u> Filename	Modify and SAVE the run file.
<u>PSD</u>	Execute the resident debugger.
<u>MAP</u>	Generate a load map.
<u>RESTOR</u> Filename	RESTORE but do not execute the program.
<u>RESUME</u> Filename	RESTORE (if necessary) and execute the program.
<u>LOAD</u> Filename	Invoke the loader to load the specified binary file.
<u>TIME</u>	Report date and time of last file save.
<u>HELP</u>	List the SEG commands.
<u>QUIT</u>	Return to PRIMOS IV command level.

For further information about SEG, refer to the Program Development Software User Guide (MAN1879).

For a detailed example of the use of SEG, refer to section 2.

```
*****
* SETIME *          ***PRIMOS III and IV***
*****
```

The format of the SETIME command is:

```
SETIME -mmddy -hhmm
```

The SETIME command sets date and time. It is an operator command and it is issued at the supervisor terminal. For further information, refer to the Computer Room User Guide (MAN 2603).

```
*****
* SFTN *
*****
```

The SFTN command invokes a smaller version of the FTN compiler without sector 0 optimization, 32-bit integers, and 64V address mode. SFTN must be used in the 16K and 24K versions of PRIMOS II.

```
*****
* SHUTDN *
*****
```

The PRIMOS II command SHUTDN is an internal command which performs tasks necessary to shutting down PRIMOS II in an orderly manner. Refer to the examples in the Computer Room User Guide (MAN 2603) for use of the PRIMOS II SHUTDN command. SHUTDN must also be entered before closing down a PRIMOS II system or changing disk packs. The command does some incidental PRIMOS II housekeeping that ensures all the information in memory buffers and is properly transferred to disk.

The PRIMOS III and IV SHUTDN command is described in the Computer Room User Guide (MAN 2603). Under PRIMOS III and IV the SHUTDN command can only be given at the supervision terminal. Also under PRIMOS III and IV the command line:

```
SHUTDN ALL
```

shuts down all disk devices simultaneously.

\*\*\*\*\*  
\* SIZE \*  
\*\*\*\*\*

The format is:

SIZE Filename

SIZE gives the data size of Filename in records (i.e., the decimal number of records). The number of records in a file is defined to be the number of data words in a file divided by 440, rounded up. The exception to this rule is that a zero word-length file always contains one record. Example:

OK: SIZE PRPLOT

GO

22 RECORDS IN FILE

OK:

\*\*\*\*\*  
\* SLIST \*  
\*\*\*\*\*

The format of SLIST is:

SLIST [Filename]

SLIST prints the content of the file Filename at the user's terminal. SLIST is often used to obtain source listings of short program or data files. If the name of the file specified to be listed is blank, SLIST expects a tree filename to be specified (refer to the File System User Guide (MAN 2604) for a definition of tree filenames). SLIST is an external command. (SLIST with no argument).

Example:

SLIST

GO,

UFDNAM > SUUFD> FILNAM

User inputs the tree name of the file to be printed, then SLIST prints its contents.

\*\*\*\*\*  
 \* SORT \*  
 \*\*\*\*\*

The SORT command sorts an ASCII file and writes the sorted file in the current UFD. The SORT program requests input and output files, number of columns, and starting and ending columns for the sorting operation. Up to ten (10) sort keys may be specified. The maximum characters/SORT record is 144 characters. SORT is an external command. The format is:

```

  SORT [ | BRIEF | ]
        | SPACE |
        | MERGE |

```

The options BRIEF, SPACE, or MERGE, or a combination of these options, may be entered following the command SORT. Only two options can be implemented at a time. (Note: the names of the options may be abbreviated: BR, SP, or ME.) The meaning of these options when specified is as follows:

<u>Option</u>	<u>Meaning</u>
BRIEF	SORT program messages are not printed at the user's terminal.
SPACE	Any blank lines are deleted from the SORT output file.
MERGE	A maximum of ten unsorted files can be merged at a time. The SORT program asks for the names of the merged files. The user at the terminal types the filenames on one line, separated by spaces.

#### Reverse Sorting

Sorting can be specified to be in descending order by typing the letter R separated by a space after the ending column of the desired keys.

#### Command File

The SORT command can be run from a command file, since it does not close Unit 6.

Example:

```

OK, SORT
GO
SORT PROGRAM PARAMETERS ARE:

```

INPUT FILE NAME -- OUTPUT FILE NAME FOLLOWED BY



NUMBER OF PAIRS OF STARTING AND ENDING COLUMNS.

INFILE OUTPUT 3

INPUT PAIRS OF STARTING AND ENDING COLUMNS

ONE PAIR PER LINE -- SEPARATED BY A SPACE.

FOR REVERSE SORTING ENTER 'R' AFTER DESIRED

ENDING COLUMN -- SEPARATED BY A SPACE.

1 5

15 25

30 35

BEGINNING SORT

PASSES     3    ITEMS     266

OK,

Respond to the first inquiry with the input file name, output file name, and number of pairs of columns.

Respond to each inquiry with the appropriate starting and ending column numbers (character positions).

During operation of the SORT program, the console DATA indicators display a count of the number of passes completed. When the sort is completed, SORT prints the number of passes and number of items (lines in the input file), and returns to PRIMOS.

\*\*\*\*\*  
\* SPOOL \*        \*\*\*PRIMOS III and IV\*\*\*  
\*\*\*\*\*

The format of the SPOOL command is:

SPOOL [Filename] [(F)]

SPOOL queues a copy of a file in the UFD SPOOL for off-line printing on the PRIMOS III and IV system configuration line printer. SPOOL allows a user to get output printed without specifically ASSIGNing the printer and then waiting until the printing operation is complete before being able to issue another command at the terminal.

When the optional argument (F is given, the spooler is instructed to use the CARRIAGE CONTROL character in the file specified by filename. Furthermore, the spooler is instructed not to paginate (put page numbers on the output pages). In other words, the FORTRAN forms control information will be missing.

Example:

OK, FTN MYPROG

OK, SPOOL L<-MYPRO

GO

YOUR SPOOL FILE IS PRNT10

OK, FTN NEWPRG

Using SPOOL, terminals are not tied up waiting for the printer, and terminals and files can be used while copies of the files are being printed.

Unless the PHANTOM command is invoked, one terminal in the PRIMOS III and IV configurations must be dedicated to running SPOOL (i.e., SPOOL must be logged in as if it were a user).

The SPOOL program copies the specified file Filename into the UFD SPOOL and changes its filename to prevent naming conflicts. Each file in the UFD SPOOL is deleted after it is printed.

There are two programs in the UFD SPOOL that control printer output. The SPLCEN program prints files in UFD SPOOL on the Centronics Line Printer, and the SPLMPC program prints files in the UFD SPOOL on the high-speed line printer. Only entire files can be printed using SPOOL.

After a successful file copy of Filename to the UFD SPOOL, SPOOL responds:

YOUR SPOOL FILENAME IS PRNTnn

where nn is a two-digit decimal integer that is part of the new filename in the SPOOL directory.

### SPOOL Output Format

The SPOOL filename (PRNT nn) is printed on the header page before the file is printed. A header page with the User Login Name and Filename is generated as the first page of each SPOOL job.

### Errors

If a FULL DISK error occurs when SPOOL attempts to print a file, an error message is printed and the file is deleted.

### SPOOL Argument

SPOOL typed with no Filename argument opens File Unit 2 for writing in the SPOOL directory. SPOOL responds by typing the SPOOL filename. A user program may then write directly to File Unit 2. When the program finishes, the user may close File Unit 2 with the command. the command:

CLOSE 2

or may close the unit at the end of the program. The file produced in this manner is subsequently printed by SPLCEN or SPLMPC, whichever is appropriate.

Example 1:

```
OK, PMA FILE
GO
0000 ERRORS (PMA 1080.018)
OK, SPOOL L<-FILE
GO
YOUR SPOOL FILENAME IS PRNT10
OK, DELETE L<-FILE
OK,
```

In the above example, a user assembles the program named FILE and generates the listing, L FILE, and a binary file B File. The user then issues the command: SPOOL L FILE. This command causes L FILE to be copied to PRNT10 in the UFD SPOOL. If SPLMPC or SPLCEN is running in UFD SPOOL (logged in on another terminal), the file PRNT10 is printed on the line printer. The user may then DELETE the file L FILE, since what is desired is a printed copy of the listing.

## Example 2:

```

OK, SPOOL
GO
YOUR SPOOL FILENAME IS PRNT10
OK, PMA FILE
GO
0000 ERRORS (PMA 1080.018)
OK, CLOSE ALL

```

In this example, the user issues the command SPOOL with no Filename argument before invoking the assembler. SPOOL opens PRNT10 in UFD SPOOL for writing on File Unit 2. The command: PMA FILE first checks if Unit 2 is open. Because Unit is open, PMA does not open and write L FILE in the user's UFD; instead, it outputs the assembly listing to the file already open on File Unit 2, which happens to be PRNT10 in the UFD SPOOL. When the assembly is done and PMA returns to command level, PMA leaves File Unit 2 open. The user gives the CLOSE ALL command, which closes Unit 2.

## NOTE

The user may also invoke a series of assemblies or compilations before giving the CLOSE ALL command, with the result that a listing-file that contains a series of listings would be created.

After File Unit 2 is closed, and if SPLCEN or SPLMPC is running, the file PRNT10 is printed on the line printer.

Logging in and Starting-Up SPOOL

To start SPLMPC or SPLCEN at a terminal, proceed as follows (user input is underlined):

```

LOGIN SPOOL SPOUT
OK, ASSIGN CENPR
OK, RESUME SPLCEN
GO

```

For the high-speed printer:

```

LOGIN SPOOL SPOUT
OK, ASSIGN PRI
OK, RESUME SPLMPC
GO

```

The SPLMPC or SPLCEN program looks for files with names: PRNT10, PRNT11, PRNT12, etc., in the UFD named spool and prints them if any exist. SPLMPC or SPLCEN always processes files first-in, first-out (FIFO). An INPUT FILE ERROR or a LINE SIZE ERROR results in an error message at the terminal from which SPOOL was logged in. The spool-file (PRNT10, 11, ... etc.) is deleted and the next file is processed.

Stopping a File Print by SPOOL

If a user decides not to print a file that is queued for printing by SPOOL, it is possible to ATTACH to the UFD SPOOL and DELETE the appropriate file (named PRNTnn where nn is a number 10, 11, 12, etc.).

If printing of the file to be deleted has started, the attempt to DELETE it fails. However, the user can request the operator at the terminal from which SPOOL was logged-in to stop the file from printing; or the operator may stop printing a file if he perceives that the file is incorrect. The operator or user proceeds to do this by:

CONTROL P	(Operator presses QUIT)
QUIT,	
CLOSE ALL	
OK, <u>DELETE PRNTnn</u>	(nn = number for SPOOL request)
OK, <u>RESUME SPLMPC</u>	(or <u>RESUME SPLCEN</u> )
GO	

CAUTION

SPOOL reuses available names. Thus, after PRNT10 is printed and deleted, the name PRNT10 is available for use by SPOOL again and may be given to a subsequent Filename argument in a subsequent SPOOL request. If deleting file from the UFD SPOOL, be sure you are deleting the right one.

Card Spooler (CRDSPL)

A run file in the UFD=SPOOL named CRDSPL provides a card reader spooler facility. This run file allows users to read cards from the card reader via the use of the SPOOL utility. To use, proceed as follows:

LOGIN SPOOL SPOUT

ASSIGN CR1

RESUME CRDSPL

Directions for reading cards are outlined in the Card Reader User Guide.

Spooler for Printer/Plotters

A special version of the line printer spooler has been written for printing and plotting on the Versatec or the Gould Printer/Plotter. The spooler is called SPLVER and is found in UFD SPOOL. To start SPLVER at a terminal, do the following:

LOGIN SPOOL SPOUT

ASSIGN PLOT

RESUME SPLVER

To print a file on the printer/plotter from any other user terminal, give the command:

SPOOL Filename

To plot a file on the printer/plotter from any user terminal, give the command:

SPOOL Filename (PLOT)

The file must consist of a series of records, each record 70 words long. Each record is sent to the printer/plotter as one line of plot information. The records can be generated by setting a 70-word memory buffer with the appropriate information and calling PRWFIL repeatedly to output 70 words to a file.

\*\*\*\*\*  
\* START \*  
\*\*\*\*\*

The format of START is:

START [PC] [A] [B] [X] [Keys]

START initializes the processor's registers and keys from the command line (or from RVEC, for any values not specified in the command line) and starts execution at location PC. This command assumes a program has been loaded into memory by a previous RESTORE, RESUME, or LOAD command. START is an internal command.

START can also restart a program that has returned control to PRIMOS (for example, because of an error, a FORTRAN PAUSE or CALL EXIT statement). If START is typed without a value for PC, the program resumes at the PC value at which execution was interrupted. To restart the program at a different point, specify an octal starting location as the PC value.

Example:

```
OK, ED MYFILE
GO
P55
QUIT (user has pressed -CNTR-P-)
START 1000
```

The above command sequence restarts the editor in a state to receive the next command, without continuing the unwanted output.

\*\*\*\*\*  
\* STARTUP \*  
\*\*\*\*\*

Under PRIMOS III and IV, the STARTUP command can be given only at the supervisor terminal. The format of STARTUP is:

STARTUP Pdisk0 [Pdisk1] [Pdisk2] [Pdisk3] ... [Pdisk8]

STARTUP initializes the configuration of disk drives by relating physical disk drive numbers to PRIMOS logical disk unit numbers. STARTUP is an internal command. Physical device numbers for disks are shown in Table 3-1 and the Computer Room User Guide (MAN 2603).

The logical-to-physical assignment depends on the order in which the physical device numbers are listed as parameters in the STARTUP command. The physical device number specified in the Pdisk0 position is assigned as logical disk unit 0, the physical device number specified in the Pdisk1 position is assigned as logical disk unit 1, and so on.

The number of parameters indicates to PRIMOS the number of logical drives assigned to the system.

Example:

STARTUP 51 52 53

This command makes the following logical/physical disk assignments:

<u>Logical Unit</u>	<u>Physical Unit</u>
0	51
1	52
2	53
3	Not Assigned

STARTUP has some extended capability in PRIMOS III and IV; refer to the Computer Room User Guide (MAN 2603).



\*\*\*\*\*  
 \* STATUS \*  
 \*\*\*\*\*

STATUS lists the login UFD - the logical device upon which the UFD resides, the low boundary of PRIMOS II plus buffers, the open file units, and the physical-to-logical device correspondence. STATUS also lists physical device numbers, as described in the Computer Room User Guide.

Example: (for PRIMOS II)

OK: STATUS

USR=GOUDY 0

DOSLO 67000 FUNITS

LDEV	PDEV
0	51
1	50
2	53

In PRIMOS III and IV, the STATUS command prints the packnames of the disks also. Rather than typing the current UFD, the login UFD is typed.

Example: (for PRIMOS III and IV)

OK, STATUS

USR=GOUDY 0

FUNITS

DISK	LDEV	PDEV
TSDISK	0	50
COMMAND	1	50
DUD	2	52

USER	N0	LIN	PDEVS
GOUDY	7	5	41054

The disk name (Packname) is the name of the DSKRAT on that disk pack. The DSKRAT name can be changed by the CNAME command. UBR is the login name of the user; NO is the user number; LIN is the AMLC line number of the user terminal; and PDEVS represents the physical devices the user is currently using. User numbers are printed in decimal and AMLC line numbers are printed in octal. PDEVS includes disks that the user is using under PRIMOS III or IV, and any assigned devices. All assigned devices have the same name as those specified in the ASSIGN command, with the exception of assigned disks and AMLC lines. Assigned disks are indicated by printing the letters DK followed by the disk

number. Assigned AMLC lines are indicated by printing of the letters AL followed by the line number. AMLC numbers are printed in octal. If the terminal line is connected to the system through a serial interface (bit-banger) line, it is given a number between 50 and 53, and that number is printed.

The STATUS command input at the system terminal prints the above information for each user. Phantom users are identified by an AMLC line number of 77. A user connected to the system at the system terminal through the USRASR command is given an AMLC line number of 76. Furthermore, if the user (or any other users logged in under the same name), has phantom processes running, the STATUS command prints the word PHANTOM followed by the user numbers of all such users.

The STATUS command may be used to monitor the usage of PRIMOS III or IV. When entered at the system terminal, the STATUS command prints status information that includes user terminal data, the paging device, the command device, a list of current logged-in users, and the devices that each user has currently assigned. Disks assigned to a user are printed as: DISK <octal number>. Following each user name in the list, the user terminal number and the numbers of the physical disks currently being used are printed. A disk is considered to be in use by a user (1) if his home UFD or current UFD resides on the disk or (2) if the user has opened a file on that disk. Some typical instances where the STATUS command must be used are:

1. Prior to mounting a new disk pack (to determine what physical disk assignments are available).
2. After a request that all users release a given disk or disks (to determine that they have done so before shutting down the given disk or disks).
3. As a check that all users have logged out before shutting down PRIMOS III or IV. (No harm to the system results if the users of a particular disk are still logged-in when the disk or the system is shutdown. However, users will be disconnected and the message: DISK d DETACHED; YOUR FILES CLOSED will be printed at their terminal, where d represents the physical disk number.)

Example: STATUS Command and Response When Issued at System Terminal.

STAT

USR = SYSTEM

## FUNITS

DISK	LDEV	PDEV
TS#1	0	250
DUD#2	1	40250
COMAND	2	52
ETCH3	3	50
TS#2	4	20250
MD6V2	5	60
TRANS	6	50250
PRIMOS	7	60250

PAGDEV = 10250 COMDEV = 250

USER	NO	LINE	PDEVS
SPOOL	3	1	40250 PR1
PDAVIS	4	2	20250 MT0 DK1
PDAVIS	5	3	20250
GRUBIN	7	5	50 DK20
GREATA	8	6	250
SPQRXR	9	7	20250

STATUS for PRIMOS III and IV

On PRIMOS III and PRIMOS IV, the STATUS command has the following formats:

STATUS  
 STATUS DISKS  
 STATUS USERS  
 STATUS ALL  
 STATUS NETWORK  
 STATUS PHANTOMS

The results of the these various parameters are specified by the following examples:

1. STAT with no parameters specified:

OK, STAT

USR=GOUDY                      SYSB

## FUNITS

DISK	LDEV	PDEV	SYSN
TS/B	0	50	
SPOOLB	1	10050	

MKTDB	2	22450	
DOCUME	3	31452	
HARDWR	4	62052	
ADMIN	5	1452	
TS/A	6	50	SYSA
MARKET	7	22450	SYSA
ETCH	10	460	SYSA
MFGDSK	11	462	SYSA
MFGINV	12	10463	SYSA
OSFTWR	13	10461	SYSA
BIKE1	14	54	

NODE	STATE
SYSA	UP
SYSC	UP
SYSD	DOWN
SYSE	DOWN

USER	NO	LIN	PDEVS
GOUDY	18	20	31452

OK,

## 2. STATUS DISKS

OK,      STAT DISKS

DISK	LDEV	PDEV	SYSN
TS/B	0/50		
SPPOLB	1	10050	
MKTDB	2	22450	
DOCUME	3	31452	
HARDWR	4	62052	
ADMIN	5	1452	
TS/A/6/50/SYSA			
MARKET	7	22450	SYSA
ETCH	10	460	SYSA
MFGDSK	11	462	SYSA
MFGINV	12	10463	SYSA
OSFTWR	13	10461	SYSA
BIKE1	14	54	

NODE	STATE
SYSA	UP
SYSC	UP
SYSD	DOWN
SYSE	DOWN.sk

OK,

## 3. STATUS USERS

OK, STAT USERS

USER	N	O	LINPDEVS
DONNA	5	3	22450
SALES	8	6	22450
TEKMAN	14	14	31452
CARR	15	15	1452
GOUDY	18	20	31452
SYSTEM	26	77	5010050PRO
SYSTEM	27	77	10050
FAM	28	77	
SYSTEM	29	77	10050

OK,

## 4. STATUS ALL

OK, STAT ALL

USR=GOUDY SYSB

FUNITS

DISK	LDEV	PDEV	SYSN
TS/B	0	50	
SPOOLB	1	10050	
MKTDB	2	22450	
DOCUME	3	31452	
HARDWR	4	62052	
ADMIN	5	1452	
TS/A	6	50	SYSA
MARKET	7	22450	SYSA
ETCH	10	460	SYSA
MFGDSK	11	462	SYSA
MFGINV	12	10463	SYSA
OSFTWR	13	10461	SYSA

NODE	STATE
SYSA	UP
SYSC	UP
SYSD	DOWN
SYSE	DOWN

USER	NO	LIN	PDEVS
DONNA	5	3	22450
SALES	8	6	22450
TEKMAN	14	14	31452
CARR	15	15	1452
GOUDY	18	20	31452
SYSTEM	26	77	50 10050 PR0

SYSTEM 27 77 10050  
FAM 28 77  
SYSTEM 29 77 10050

OK,

#### 5. STATUS NETWORK

OK, STAT NETWORK

NODE	STATE
SYSA	UP
SYSC	UP
SYSD	DOWN
SYSE	DOWN

#### 6. STATUS PHANTOMS

OK, STAT PHANTOMS

USER = JDOAKS

FUNITS

DISK	LDEV	PDEV	SYS
SYSTEM	0	0	
SPOOL	1	50	
USERX	2	22450	

USER	NO	LIN	PDEVS
JDOAKS	5	3	22450
JDOAKS	18	20	22450

\*\*\*\*\*  
 \* SVCSW \*  
 \*\*\*\*\*

\*\*\*PRIMOS III\*\*\*

The SVCSW command controls the handling of SVC instructions in a virtual memory environment. The format is:

```
SVCSW | 0 |
      | 1 |
```

The normal mode (SVC 0) causes all SVC instructions to be trapped and processed by the system supervisor. If the SVC SWITCH is ON (SVC 1), almost all SVC instructions cause a virtual trap, and SVC instructions are handled through the user's location 65. The class of SVC instructions always processed by the PRIMOS III operating system, regardless of the SVCSW command, are those determined by FUNCTION code 5XX. Currently the SVC's are RREC, WREC (for reading and writing to disk, TIMDAT (for obtaining the time and date from PRIMOS III), and RECYCL.

The SVC switch is initialized to 0 by the LOGIN command. The SVCSW command allows a special version of PRIMOS II called VDOS32 to be run under PRIMOS III. (SVCSW does not run under PRIMOS IV.)

\*\*\*\*\*  
 \* TAP \*  
 \*\*\*\*\*

TAP is an octal mode debugging routine that permits the operator to access memory locations, process memory blocks, and trace program execution dynamically. Its format is:

TAP

For complete details about TAP commands, usage, and features, refer to the Program Development Software User Guide(MAN 1879).

Under PRIMOS IV, TAP is no longer loaded at 56000 for patching the PRIMOSL or PRIMOSU run files (refer to the Computer Room User Guide MAN 2603 for a description of these files). To patch PRIMOS (PRIMOS IV), use the command PSD160.

#### CAUTION

The command PSD160 cannot be used when running under 64K PRIMOS II.

```
*****
*   TIME   *           ***PRIMOS III and IV***
*****
```

The TIME command prints the current value stored in the time accounting registers. The three values printed are the same as the three values in the logout message, namely:

Connect Time	(hours, minutes)	Time since LOGIN.
Compute Time	(minutes, seconds)	Time accumulated executing commands or using programs (does not include disk I/O time).
Disk I/O time	(minutes, seconds)	Time accumulated for disk input/output.

The disk I/O time includes not only user-requested I/O to files, but also paging I/O time generated on the user's behalf. All times include supervisor overhead, such as the time spent executing supervisor subroutines on the user's behalf. Some supervisor overhead associated with the PRIMOS scheduler is charged to the supervisor (at the supervisor's terminal) and not the user. When the system is idle, CPU time is charged to the supervisor. Compute time does not include I/O time for diskette or for disks that are connected to a type 4000 controller.

Examples:

TIME Command Issued At User Terminal:

```
OK, TI
2'40" 0'07 0'06
OK,
```

TIME Command Issued at Supervisor Terminal:

```
OK, TI
11'31" 96'07 0'01
OK,
```



\*\*\*\*\*

\* TRAMLC \*

\*\*\*PRIMOS III and IV\*\*\*

\*\*\*\*\*

TRAMLC transmits or receives a file over an assigned AMLC line between two Prime computer systems operating under PRIMOS III or IV, using transparent protocol. The format is:

### TRAMLC

When TRAMLC is invoked, the user may specify transmission or reception by the command lines:

TRANSMIT Filename AMLC-Number [T]

typed on a user terminal connected to one PRIMOS III or IV system; or

RECEIVE Filename AMLC-Number [T]

typed on a user terminal connected to the other PRIMOS III or IV system.

The parameter T is an optional value that provides a milestone message at time intervals that are multiples of T.

AMLC-Number is the logical device number of the AMLC.

The file specified by Filename, to be transmitted or received, is divided into 64-word blocks. Each block is transmitted until it is received without errors. The transmitter and receiver must be running at the same baud rate. When the entire file, specified by Filename, has been transmitted or received, the message:

FILE COMPLETE

is printed at the user's terminal.

### NOTE

Either the transmitter or receiver program can be started first.

Error messages give the reason for the error that occurred and the block number of the failure.

Example:

To Transmit:

OK, ASSIGN AMLC TRANS 1

Assign line and protocol.

OK, TRAMLC

Start program.

GO

TRANSMIT FILET 1 [T]

Direction, filename,

and AMLC line number  
(in octal, time  
interval).

To Receive:

<u>OK, ASSIGN AMLC TRANS 1</u>	Assign line and protocol.
<u>OK, TRAMLC</u>	Start program.
<u>RECEIVE FILER 1[T]</u>	Direction, filename, and AMLC line number (in octal time interval).

\*\*\*\*\*  
\* UDOS64 \*  
\*\*\*\*\*

UDOS64 starts a version of PRIMOS II from PRIMOS IV. This version of virtual PRIMOS II is loaded at the top of 64K.

Example:

OK, LOGIN SYSTEM  
SYSTEM (3) LOGGED IN AT 9'33 01027  
WELCOME SYSTEM

OK, UDOS64  
GO

PRIMOS II REV. 11.0 07/15/76 (AT BAD SVC  
ER!

SVC 1  
OK, UDOS64  
GO

PRIMOS II REV. 11.0 07/15/76 (AT 170000)

\*\*\*\*\*

\*UNASSIGN\*

\*\*\*PRIMOS III and IV\*\*\*

\*\*\*\*\*

The UNASSIGN command may be entered at the user terminal (to which a device is currently ASSIGNED) or at the supervisor terminal. The UNASSIGN command, entered at the system terminal, unconditionally deassigns the peripheral assigned to any user. Entered from a user terminal, UNASSIGN deassigns only the device that was previously assigned to the user. On selected devices, this command turns off the device and clears the associated I/O buffers. The format is:

UNASSIGN Device

where Device is a previously assigned device, named as shown in Table 3-1.

From the system terminal, this command is useful to release a device if the user who assigned it has forgotten to log out and has left his terminal.

Examples:

UNASSIGN CENPR

unassigns the Centronics Printer.

UN PTR

unassigns the paper-tape reader.

Before a disk may be assigned to a terminal, it must not be assigned to PRIMOS III, IV, or another user and the disk must be specified by an entry in the assignable Disks Table (refer to ASSIGN). If the disk is assigned to PRIMOS III or IV, it must be released, using the SHUTDN command at the supervisor terminal. A disk that has been ASSIGNED by a user cannot be entered as an argument in the STARTUP command. The supervisor terminal can UNASSIGN a device that may be assigned. Devices ASSIGNED by another user are released when the LOGOUT command is invoked by that user.

\*\*\*\*\*  
\* UPCASE \*  
\*\*\*\*\*

UPCASE reformats files that contain lower-case alphabetic characters, making them suitable for output to a device with only upper case alphabetic characters. UPCASE scans through an input file, replacing all occurrences of lower-case characters with their upper-case counterparts. The command format is:

UPCASE Infil Outfil

where:

Infil is the input file, and  
Outfil is the output file.

If Outfil is not specified, it is assumed that the user has previously opened a file on Unit 2, as may be done by use of the LISTING or SPOOL commands.

\*\*\*\*\*  
\* USERS \*                    \*\*\*PRIMOS III and IV\*\*\*  
\*\*\*\*\*

The USERS command prints the number of users currently logged into PRIMOS III or IV. A user needing a fast response time may decide on the basis of this command whether or not to run a long program.

Example:

OK, USERS  
USERS = 18

\*\*\*\*\*

\* USRCWD \*

\*\*\*PRIMOS III and IV\*\*\*

\*\*\*\*\*

The format of USRASR is:

USRASR Userno

The USRASR command allows the supervisor terminal to act as user terminal. USRASR is an operator command, and it is issued at the supervisor terminal. For further information, refer to the Computer Room User Guide (MAN 2603).

```
*****  
* VRTSSW *          ***PRIMOS III and IV***  
*****
```

The VRTSSW command allows setting the virtual sense switches. The 16-bit configuration, specified by the numeric parameter of the VRTSSW command, is stored and made available. The format is:

VRTSSW [XXXXXX]

where XXXXXX is an octal number that specifies a 16-bit configuration; when XXXXXX is not specified, its value is 0.

The 16-bit configuration specified by the numeric parameter of VRTSSW is stored and made available to the user when a program written in PMA executes an INA 1620 (read sense switches) instruction. For further details, see the Assembly Language Reference Manual.

Example:

V 10100

The virtual sense switches are initialized to 0 by the LOGIN command.

#### WARNING

The instructions, skip on sense switch, always refers to the actual sense switches, not to the virtual sense switches.

```
*****  
* * * [Comment]  
*****
```

The internal command name, \*, indicates the beginning of a comment line. \* must be followed by a space and have no other spaces in the text on the comment line.

Example:

\* PROGRAM.1..JULY.14.1974

This command is useful for including comment lines in COMMAND files.

## SECTION 4

## PRIMOS IN NETWORKS

The network software is available as an integral part of and PRIMOS III and IV (15-user version only). This software and a symbiont called FAM (which runs as a PHANTOM on both computers), combine to provide a file-sharing capability to all users of the network. The network consists of two Prime 300 and/or Prime 400 systems connected via an Inter-Processor Communicator (IPC). The file sharing capability allows the users of the two systems to implement the standard file system calls (ATTACH, SEARCH, PRWFIL) to manipulate files on a set of disks that may be physically connected to either of the two systems. Thus, file utilities such as FUTIL, PMA, FTN, SORT, SPOOL, etc., can be used on files located anywhere in the network without modification. Optional network software is also available.

As all file system calls and most user commands work on logical devices, the user's view is transparent with respect to the network. For example, the following command sequence works exactly the same whether logical device ten is started locally or remotely:

```
A UFDX 10
LISTF
A SUBUFDX 0 2
FTN TEST
LOAD
LO B<-TEST
LIB
SA *TEST
QUIT
R *TEST
```

Remote file I/O such as the above always runs into and out of the user's address space. Therefore, R \*TEST runs the program \*TEST in the address space where the user logged in.

The commands that are not implemented for remote disks are:

```
LOGIN
COMINP
PHANTOM
CREATE
PASSWD
PROTECT
ASSIGN
```

The error message, \*\*ILL REMOTE REF\*\*, occurs if these are used on remote disks.

## NOTE:

Both user and phantom command files can access remotely started disks, but cannot emanate from a remote disk.

Configuring the Networks

The CONFIG command includes the node name: CONFIG SYSA, etc. The node name corresponds to the physical slave address on the IPC. The default names are:

<u>Name</u>	<u>IPC Slave Address</u>
SYSA	1
SYSB	2

These names can be six characters long. To change the names, check the load map for location of NAMDEF. The first name is at NAMDEF+4; the second is at NAMDEF+10.

Once the CONFIG has been given, the system comes up and the network software attempts to bring up the IPC line. This status is shown when the command STATUS NET is typed.

If the node name is not given in the CONFIG command, the network software is disabled and not locked into memory.

Starting FAM

FAM stands for the File Access Manager. Before any remote activity can be attempted, the FAM must be started as a PHANTOM. The following sequence should be used:

```
ATTACH FAM  
PHANTOM C<-PFAM  
ATTACH CMDNCO
```

When FAM is started, it attempts to establish a dialogue with its counterpart in the other system. When this is accomplished, FAM sends a message to the supervisor terminal indicating that the two FAM's are operational. If the FAM was cold-started, its counterpart prints a cold start message on the supervisor terminal.



## NOTE:

If the PRIMOS system is warm started and the FAM logs out, FAM must be restarted by executing the above sequence. Setting the PRIMOS system variable DONSTP to one allows all PHANTOMS to continue after a warm start. DONSTP is in LOC FIGCOM+2.

Starting Remote Disks

To start up a remote disk, the remote node name is given with the STARTUP or ADDISK command:

```
STARTUP SYSB 2/50M  
ADDISK SYSB 50 52 53
```

To shut down a remote disk, simply type:

```
SHUTDN SYSB 50
```

To start up or shut down a local disk, only the physical device number is given.

The STATUS command displays all the started disks. The disks with a system name (SYSN) are remote; those with a blank system name are local.

## NOTE:

To STARTUP a remote disk, e.g., STARTUP SYSB 50, device 50 must already be started as a local disk on system SYSB, and FAM must also have reported that its counterpart is UP.



## APPENDIX A

## SUMMARY OF COMMANDS

<u>Command Syntax</u>	<u>Function</u>
<u>ADDISK</u>	Adds a logical disk or partition to the logical Disks Table. (PRIMOS III and IV only.)
<u>AMLC</u>	Changes AMLC status for a given AMLC line.
<u>ASRCWD</u> Number	Changes the virtual control word to select one of four devices for effective I/O. (PRIMOS III and IV only.)
<u>ASSIGN</u>   Device [WAIT]     DISK Number	Obtains complete control over a disk or peripheral device from the user terminal (refer to Table 3-2 for device names and Table 3-1 for disk numbers). WAIT queues the assignment until the device is ready. The disk must be an assignable disk. (PRIMOS III and IV only.)
<u>ATTACH</u> Ufd [Password] [Ldisk] [Key]	Attach PRIMOS to the specified UFD. For PRIMOS III and IV, password may be either owner or nonowner password.
<u>AVAIL</u>   [ZERO]     ONE     TWO     ...     NINE     Packname     *	Prints the number of disk records available for use on (1) the current logical disk; (2) the specified logical disk; (3) the logical disk specified by Packname.
<u>BASIC</u> <u>BASIC</u> Filename	Invokes the BASIC language interpreter to write and execute programs in BASIC.
<u>BASINP</u>	Loads a paper tape containing programs written in BASIC language on a computer other than a Prime computer.
<u>BINARY</u> Filename	Opens file specified by Filename for writing on File Unit 3, usually as a binary output file.

<u>CLOSE</u>   Filename [Funit...]   ALL	Closes the named files and specified file units; or if ALL is specified, closes all FILES AND UNITS.
<u>COMPRES</u> Filename [Filename2]	Translates an input file into an output ASCII file, using the relative copy character ('200). EXPAND is the opposite of COMPRES.
<u>CNAME</u> Oldname Newname	Changes name of a file named Oldname to Newname.
<u>CNVIMA</u>	Converts load map into PSD input format.
<u>COMINPUT</u> Filename [Funit]	Reads commands from the file specified by Filename in the current UFD or logical unit specified, rather than from the terminal.
<u>COMINPUT</u> CONTINUE	Continues reading commands from a command file after a pause or interruption.
<u>COMINPUT</u> PAUSE	Leaves the current command input unit open and returns to operating system command level.
<u>COMINPUT</u> TTY	Read subsequent commands from the terminal. COMINPUT must be the last line in the command file or the last line in the last command file of a chain.
<u>CONFIG</u>	Operator command to configure the system (refer to MAN 2603).
<u>COPY</u> *	Copies and verifies a disk.
<u>CREATE</u> Newufd	Create a new Ufd, Newufd, in the current UFD.
<u>CRMPC</u> Filename	Read cards from the parallel interface card reader and places their image in the file specified by Filename. The first card in the deck must be \$E.
<u>CRSER</u> Filename	Reads cards from the serial interface card reader and places their image in the file specified by Filename. The first card in the deck must be \$E.
<u>DBASIC</u>	Invokes a version of BASIC that provides double-precision arithmetic capabilities.

DELAY [Minimum] [Maximum]  
[Rmargin]

Defines a time function to be used to delay the printing of a character after a LINE FEED has been output to the terminal.

DELETE Filename

Deletes file specified by Filename from current UFD.

DISKS

Adds disks to system at system terminal only.

ED [Filename]

Loads and starts the system text editor (in INPUT mode if no Filename is specified; in EDIT mode if Filename is specified). Editor commands are:

LINE Mode  
Editor Commands:

APPEND String  
BOTTOM  
BRIEF  
CHANGE/String1/String2/[Gn]  
DELETE [n]  
DELETE To String  
DUNLOAD Filename To String  
ERASE Char  
FILE  
FILE Filename  
FIND String  
INPUT Device  
INSERT String  
LOAD Filename  
LOCATE String  
MODE PRUPPER  
MODE PRALL  
MODE PROMPT  
MODE NPROMPT  
MODE LINE  
MODE BOX  
MODIFY/String1/String2/[Gn]  
MODE Buffer1 Buffer2  
NEXT [n]  
OUTPUT TTY  
OUTPUT  
OVERLAY String  
PAUSE  
PRINT [n]  
PTABSET Tab...  
PUNCH (ASR) n  
PUNCH (PTP) n  
QUIT

RETYPE String  
 SYMBOL Name Char  
 TABSET Tab...  
 TOP  
 UNLOAD Filename [n]  
 UNLOAD Filename To String  
 VERIFY  
 WHERE  
 XEQ Buffer  
 \* [n]

BOX ModeEditor Commands:

BOX v h ^D# ^D#  
 BOXIN Filename (MODIFY)  
 BOXIN Filename (OVERLAY)  
 BOXOUT  
 BRIEF  
 DISPLAY  
 ERASE Char  
 FILE Filename  
 FIND String  
 KILL Char  
 MODE PRUPPER  
 MODE PRALL  
 MODE PROMPT  
 MODE NPROMPT  
 MODE LINE  
 MODE BOX v h  
 MODIFY/String1/String2/[G]  
 MOVE Buffer1 Buffer2  
 OUTPUT  
 OVERLAY String  
 POINT v h ^D# ^D#  
 PRINT  
 PTABSET  
 QUIT  
 RFIND String  
 RLOCATE String  
 RPOINT v h ^D# ^D#  
 SYMBOL Name Char  
 VERIFY  
 WHERE  
 XEQ  
 \* [n]

EDB Inputfile [Outputfile]  
EDB (PTR) [(PTP)]

Loads and starts the binary  
 editor; EDB Commands are:

BRIEF  
 COPY Name  
 COPY ALL  
 ET

FIND Name  
 FIND ALL  
 GENET [G]  
 INSERT Name  
 NEWINF [Name]  
 OMITET [G]  
 OPEN [Name]  
 QUIT  
 RFL  
 SFL  
 TERSE  
 TOP  
 VERIFY

EXPAND Filename1 [Filename2] Inverts the operation of CMPRES.

FILMEM Fills memory locations with zeroes from '100 to the top of 32K, except for those locations occupied by PRIMOS II. Under PRIMOS III and IV, all locations from '100 to the top of 32K are filled with zeroes.

FILVER Filename1 Filename2 Compares contents of file specified by Filename1 with contents of file specified by Filename2 for equivalence, and prints message that verification is either confirmed or not confirmed.

FILVER specified with no filename arguments allows the user to subsequently specify a series of treenames of files to be verified.

FIXRAT  
FIXRAT OPTIONS

Loads and starts a maintenance program that checks file integrity of any disk pack. For PRIMOS III and IV, the disk being checked out must be ASSIGNED. Refer to MAN 2603 for complete details about FIXRAT.

FTN Filename [1/A] [2/B]

Loads Prime FORTRAN IV and starts compilation of a program.

FUTIL

Invokes a file utility that provides subsystem commands to copy, delete, and list both files and directories. FUTIL commands are:

ATTACH Directory Pathname  
 COPY File1 [,File2...]  
 COPYDAM File1 [,File2...]  
 COPYSAM File1 [,File2...]  
 DELETE File1 [,File2...]

```

FROM Directory Pathname
LISTF [level] [LISTFIL]
      [PROTECT] [SIZE] [TYPE]
QUIT
TO Directory Pathname
TRECPY Dir1 [,Dir2...]
TREDEL Dir1 [,Dir2...]
UFDCPY
UFDDEL

```

Refer to MAN 2604 for a detailed description of FUTIL.

### HILOAD

See LOAD.

INPUT Filename

Opens an ASCII source file on Unit 1 for reading by a compiler or assembler.

### LATE

Delays execution of commands until a specified time.

### LBASIC

Invoke a version of BASIC with MAT and PRINT USING.

### LISTF

Print the current UFD name, the logical device, and all Filenames in the UFD at the terminal. For PRIMOS III and IV, LISTF also prints O or N for owner or nonowner status.

LISTING Filename

Opens the file specified by Filename for writing on File Unit 2,

### LOAD

Loads and starts Prime's Linking Loader. (Loader 60000-63777; P-Register = 61000.) LOAD provides the following commands:

(LOAD will now send maps to Disk Unit 2. Unit 2 must be open for writing.)

```

Attach [Ufd] [Password] [Ldisk]
      [Key]
CCommon Address
EXecute [AReg] [BReg] [XReg]
FOrce Filename [Loadpoint]
      [Linkstart] [Linkrange]
HHardware Definition
IInitialize [Filename] [Loadpoint]
      [Linkstart] [Linkrange]
LOad Filename [Loadpoint]
      [Linkstart] [Linkrange]
LLibrary [Filename]
MAP [Option]

```



Mode Mode  
 QUIT  
 RECOVER  
 SAVE Filename [AReg] [BReg] [XReg]  
 SETbase Linkstart Linkrange  
 VIRTUALbase Startlinks To sector

LOAD20 (See LOAD.)

(Loader 200000-23777; P-Register = 210000.)

LOGIN

Connects to the PRIMOS III and IV systems only for a terminal session.

LOGOUT

Gives up user-access to the PRIMOS III and IV systems only. (Exit from a terminal-session).

MACHK

For PRIMOS II, uses computer to operate in machine check mode. PRIMOS III and IV default is machine check mode.

MAGSAV

Writes all or part of the contents of a disk to magnetic tape.

MAGRST

Reads the contents of a magnetic tape to a disk or portion of a disk.

MAKE

Creates a disk supported by PRIMOS that contains the following:

DSKRAT  
 MFD  
 BOOT  
 DOS (empty disk)  
 CMDNC0 (empty disk)

MCG Filename

Translates results of microcode assembly into proper code for the ROM simulator.

MDL

Punches paper tapes of specified sections of memory in a self-loading format.

MESSAG [ | -UU | ]

Sends message to operator, user | ALL | -UU, or broadcasts message from operator to ALL users.

MTDSK

Performs record for record copy of disk to magnetic tape. Useful to users of 7-track magnetic tape.

NUMBER

Utility to number or renumber

a BASIC program.

<u>OPEN</u> Filename Funit Key	Opens the file specified by Filename on the File Unit, Funit; Key specifies type of file and action to be taken.
<u>OPRPRI</u> 0 <u>OPRPRI</u> 1	Sets (or resets) operator privilege status at supervisor terminal. Allows external commands to be given at supervisor terminal.
<u>PASSWD</u> Owner Password	Replaces any existing Password in the [Non-owner Password] current UFD with a new owner (and optionally non-owner) password. This version of PASSWORD is the PRIMOS II version of the command.
<u>PASSWD</u>	Replaces existing passwords with null (no) password for all PRIMOS systems.
<u>PHANTOM</u> Cifile	Starts up and runs phantom user job; specified by command file, Cifile .
<u>PM</u>	Prints contents of the RVEC vector.
<u>PMA</u> Filename [1/A] [2	Loads the macro assembler and starts assembly of Filename in the current UFD. Default value of the A Register is: 000777 which signifies -- normal listing detail, all input and output files on disk.
<u>PRERR</u>	Prints message stored in ERRVEC.
<u>PRMPC</u> Filename	Prints file on MPC line printer.
<u>PROTECT</u> Filename Key1 [Key2]	Open file directory giving restricted access rights to Filename as specified by Key1 (owner) and Key2 (non-owner). The following is a list of values for Key1 or Key2:  0 = No access 1 = Read only 2 = Read and write 3 = Delete only 4 = Delete, truncate and read 5 = Delete, truncate and write 6 or 7 = All access (PRIMOS II and IV only.)
<u>PRSER</u> Filename	Prints file on Serial line printer.
<u>PRVER</u> Filename	Starts and runs printer/plotter spooler.

<u>PSD</u>	Loads and starts symbolic debugging program. program.
<u>PSD20</u>	Invoke version of PSD for 16K PRIMOS II.
<u>PSD160</u> <u>PTBOOT</u>	Starts and runs paper-tape loader. This command assumes SLT (self-loading tape).
<u>PTCPY</u>	Loads a utility program that duplicates and verifies paper tapes.
<u>PUSS</u>	Invokes source file comparison.
<u>RESTORE</u> Filename	Restores Filename in the current UFD to high-speed memory, using the SA and EA values; SAVED with Filename.
<u>RESUME</u> Filename [PC] [A] [B] [X] [Keys]	Starts execution of a program.
<u>RUNOFF</u> [Filename]	Starts and runs text formatter.
<u>SAVE</u> Filename SA EA [PC] [A] [B] [X] [Keys]	Saves the content of high-speed memory using SA (starting address) to EA ending address) on a file named Filename in the current UFD.
<u>SEG</u>	Loads and executes PRIMOS IV programs or modifies PRIMOS IV run file.
<u>SETIME</u> -mm-ddyy-tttt	Sets system time and date usually done at system startup.
<u>SFTN</u>	Invokes FORTRAN IV compiler for less than 16K of memory.
<u>SHUTDN</u>   [Pdisk]     ALL	For PRIMOS II, shuts down the system (no parameters), the specified physical disk (Pdisk), or the entire system (ALL).
<u>SIZE</u> Filename	Prints the size of Filename in records, at the terminal.
<u>SLIST</u> Filename	Prints the contents of Filename at the terminal.
<u>SORT</u>   [BRIEF]     SPACE	Sorts an ASCII file and writes the sorted file

MERGE	in the current UFD.
	BRIEF: No messages.
	SPACE: Delete blank lines from output.
	The SORT command gives the user instructions on the items to specify as the command progresses.
<u>SPOOL</u> [Filename] [F]	Queues a copy of Filename in the UFD SPOOL for off-line printing. SPOOL typed with no Filename opens File Unit 2 for writing in the UFD SPOOL and prints them after they are closed (either by the user or the end of the program). Using spool with no Filename argument is a convenient way to get PRIMOS listings and LOAD maps printed. (PRIMOS III and IV only.)
<u>START</u> [PC] [A] [B] [X] [Keys]	Initializes the registers and keys from the command line (or from RVEC) and starts execution at the location PC. START can also be used to restart a program (in fact, this is its most
<u>STARTUP</u> Pdisk [Pdisk1...]	Initialize the configuration of disk drives by relating physical disks to logical disk unit number. 'Refer to Section 4 for extended capabilities of STARTUP in PRIMOS III and IV.)
<u>STATUS</u>	Print status information at the terminal. (Status information varies for PRIMOS II and III; for details, refer to STATUS ALL Sections 3 and 4.)
<u>SVCSW</u> i	Controls the handling of SVC instructions in the virtual memory environment (PRIMOS III only.)
<u>TAP</u>	Enter Trace and Patch Debugger.
<u>TIME</u>	Prints the current value of the time accounting registers. (PRIMOS III and IV only.)
<u>TRAMLC</u>	Transmits and receives data over an AMLC.
<u>VSDOS64</u>	Starts a version of PRIMOS II that may be run under PRIMOS III or IV.

<u>UNASSIGN</u> Device	Deassigns peripheral devices or disks. UNASSIGN may be entered from either a PRIMOS II only.)
<u>UPCASE</u> I file O file	Translate lower-case file to upper-case file.
<u>USERS</u>	Print number of users currently on the system.
<u>USRASR</u> User-No	Sets system terminal to different user number.
<u>VRTSSW</u>	Allows setting of the virtual sense switches. (PRIMOS III and IV.)
<u>*</u>	Indicates comment line by a space and has the correct command line form (1 to 3 names followed by 0 to 9 parameters).



## APPENDIX B

## PRIMOS ERROR MESSAGES

<u>Message</u>	<u>Remarks</u>
BAD <COMMAND-NAME>	Example: bad startup.
BAD CALL TO SEARCH	
BAD DAM FILE	
BAD PARAMETER	
BAD PASSWORD	
BAD RTNREC	
BAD SVC	Bad supervisor call.
DEVICE IN USE	
DISK<X> NON DOS	(DOS is former name of PRIMOS II)
DISK FULL	
DK ERR	See disk error explanation below.
DUPLICATE NAME	
FATAL ERROR IN DOEXT	
<FILENAME> NOT FOUND	
<FILENAME> IN USE	
<FILENAME> ALREADY EXISTS	
ILLEGAL INSTRUCTION AT	
<OCTAL LOC.>	
<NAME> NOT ASSIGNED	
NO UFD ATTACHED	
NO VECTOR	User has gotten a UII, PSU, or FLEX, or trap to a location that is 0, or SVC switch is on and user got an SVC trap and location '65 was 0.
NOT A UFD	
POINTER MISMATCH	Run FIXRAT.
PROGRAM HALT AT	
<OCTAL LOC.>	
PRWFIL BOF	
PRWFIL EOF	
PRWFIL POINTER MISMATCH	
PRWFIL UNIT NOT OPEN	
SEG-DIR ER	
UFD FULL	
UFD OVERFLOW	
UNIT <X> CLOSED	This line and the next two lines are part of the same message.
DISK <X> CLOSED	
YOUR FILES DETACHED	
UNIT IN USE	
UNIT NOT OPEN	
UNIT OPEN ON DELETE	

## DISK ERRORS

DK ER P# Physical-device-# PRIMOS-record-address Cra disk-status-word

Cra is only valid on read requests. It is the identifier of the record that is read. Cra should match the requested record address.

There is no alternate return caused by a detected disk error. A message is printed and the operation is retried continuously, in PRIMOS II; in PRIMOS III and IV the operation is tried ten times.

Status Word

The status word typed as the third octal number of a disk error depends on the type of controller as follows:

4000 Controller

<u>Status Word</u>	<u>Meaning</u>
177777	Bad record identifier
177776	Device not ready
100000	Data transfer complete (good if present)
040000	Read/write past end of record
040000	Seek complete (good if present)
002000	Write protect violation
000400	Command error
000200	Checksum error
000100	DMX overrun
000040	Stack overflow

4001 Controller

<u>Status Word</u>	<u>Meaning</u>
177777	Bad record identifier
177776	Device not ready
100000	Bit 1 always set
040000	DMX overrun
020000	Disk is write protected
010000	Checksum error
000100	Disk drive seeking
000040	Disk drive seeking
000020	Disk drive seeking
000010	Disk drive seeking
000004	Illegal seek
000002	Malfunction detected



Diskette Controller

<u>Status Word</u>	<u>Meaning</u>
177777	Bad record identifier
177776	Device not ready
100000	Normal end of instruction (good if present)
040000	Sector not found
020000	Checksum error on sector ID
010000	Track error; head is mispositioned
002000	Deleted data mark read
001000	DMX overrun
000400	Checksum error, write protect
	Violation of file inoperable on
	Write or format

Storage Module Error Correcting Code

The storage module controller writes a two-word error-detecting and correcting code checksum on each record. It is a fire code with generator polynomial  $G(X) = (X^{11} + X^2 + 1)(X^{21} + 1)$ . The code together with the correction logic in the PRIMOS storage module driver is capable of detecting any two-error bursts of combined length 22 bits, or one error of length 32 bits, or any odd number of errors, and is capable of correcting any single error burst of up to 11 bits. Error correction is attempted only after ten attempts to read a record have failed.

The message printed when error correction is attempted and fails is 'UNCORRECTABLE'. The message printed when error correction succeeds is 'WORDNO ERROR ERROR', where WORDNO is the offset relative to the beginning of the record of the beginning of the correction, and ERROR ERROR is the 32-bit correction pattern (of which at most 11 consecutive bits are nonzero).

The disk status word for the Storage Module disk is as follows:

<u>Status Word</u>	<u>Meaning</u>
177777	bad record identifier
177776	device not ready
100000	always set
040000	DMX overrun
010000	check error
004000	checksum error
002000	header check failure
000010	disk drive seeking
000004	illegal seek
000002	select error
000001	not available or not ready



## APPENDIX C

## ERRVEC CONTENTS

ERRVEC consists of eight words; their contents are as follows:

<u>Word</u>	<u>Content</u>	<u>Remarks</u>
ERRVEC (1)	Code	Indicates origin of error and nature of error.
(2)	Value	On alternate return, this is the value of the A-register. On normal return, this may have special meaning, (e.g., refer to PRWFIL and SEARCH error codes).
(3)	X X	ERRVEC (3), ERRVEC (4), ERRVEC (5), and ERRVEC (6) contain a six-character Filename of the routine that caused the error [ERRVEC (6) is available for expansion of names].
(4)	X X	
(5)	X X	
(6)	X X	
(7)	Pointer To Message	For PRIMOS supervisor usage.
(8)	Message Length	For PRIMOS supervisor usage.

PRWFIL Error Codes

PD	UNIT NOT OPEN	
PE	PRWFIL EOF (End of File)	Number of words left. (Information is in ERRVEC(2))
PG	PRWFIL EOF (Beginning of File)	Number of words left. (Information is in ERRVEC(2))

PRWFIL Normal Return

ERRVEC (3) = Record Number

ERRVEC (4) = Word Number

PRWFIL Read-Convenient

ERRVEC (2) = Number of words read.

SEARCH Error Codes

<u>ERRVEC (1) =</u>	<u>Meaning</u>
SA	SEARCH, BAD PARAMETER
SD	UNIT NOT OPEN (truncate)
SD	UNIT OPEN ON DELETE
SH	<Filename> NOT FOUND
SI	UNIT IN USE
SK	UFD FULL
SL	NO UFD ATTACHED
SQ	SEG-DIR-ER
DJ	DISK FULL

SEARCH Normal Return

ERRVEC (2) = Type where Type has the following values:

<u>Type =</u>	<u>Meaning</u>
0	File is SAM
1	File is DAM
2	Segment Directory is SAM
3	Segment Directory is DAM
4	UFD is SAM

## APPENDIX D

## CHARACTER SET

The standard character set used by Prime is the ANSI, ASCII 7-bit set shown in Figure D-1. Control characters are described in Table D-1.

## PRIME USAGE

Prime hardware and software uses standard ASCII for communications with devices. The following points are particularly important to Prime usage.

1. Output Parity is normally transmitted as a zero (space) unless the device requires otherwise, in which case software will compute transmitted parity. Some controllers (e.g., MLC) may have hardware to assist in parity generations.
2. Input Parity is ignored by hardware and by standard software. Input drivers are responsible for making the parity bit suit the host software requirements. Some controllers (e.g., MLC) may assist in parity error detection
3. The Prime internal standard for the parity bit is one.

## INTERNAL STANDARDS

The following standards apply to internal usage of character codes, excluding communications and control functions. Internal Standards are composed of Storage Definitions; Table D-2 explains the internally redefined codes for the characters shown in Figure D-2.

	0	1	2	3	4	5	6	7
00	NUL	SOH	STX	ETX	EOT	ENQ	ACK	BEL
01	BS	HT	LF	VT	FF	CR	SO	SI
02	DLE	DC1	DC2	DC3	DC4	NAK	SYN	ETB
03	CAN	EM	SUB	ESC	FS	GS	RS	US
04	SP	!	"	#	\$			
05	(	)	*	+	,	-	.	/
06	0	1	2	3	4	5	6	7
07	8	9	:	;	<	=	>	?
10	\$	\$A\$B\$C\$D\$E\$F\$G						
11	H	I	J	K	L	M	N	O
12	P	Q	R	S	T	U	V	W
13	X	Y	Z	[	\	]		
14	'	a	b	c	d	e	f	g
15	h	i	j	k	l	m	n	o
16	p	q	r	s	t	u	v	w
17	x	y	z	{				DEL

Figure D-1. ASCII Communications Codes.

	0	1	2	3	4	5	6	7
00	NUL							BEL
01	BS	HT	NL	VT	FF	CR	RRS	BRS
02	RCP	RHT	HLF	RVT	HLR			
03								
04	SP	!	"	#	\$			
05	(	)	*	+	,	-	.	/
06	0	1	2	3	4	5	6	7
07	8	9	:	;	<	=	>	?
10	ABCDEFGH							
11	H	I	J	K	L	M	N	O
12	P	Q	R	S	T	U	V	W
13	X	Y	Z	[	\	]		
14	'	a	b	c	d	e	f	g
15	h	i	j	k	l	m	n	o
16	p	q	r	s	t	u	v	w
17	x	y	z	{				

Figure D-2. Internal ASCII Codes.

Table D-1. Control Characters.

## CONTROL CHARACTERS FOR COMMUNICATIONS

<u>Code</u>	<u>Name</u>	<u>Use</u>
201	SOH	Used at the beginning of a sequence of characters (a heading) containing address, routing, and possibly other information.
202	STX	Precedes a sequence of characters to be treated as an entity (a message or a message block) and passed to the destination station. STX terminates the heading, if any is present.
203	ETX	Terminates a sequence of characters (a message) begun with STX.
204	EOT	Terminates transmission.
205	ENQ	A request for a response from a remote station. It may be used to request station identification or status.
206	ACK	An affirmative acknowledgement returned to the sender from the receiver.
220	DLE	An "escape" character which changes the meaning of an immediately following string of characters. DLE was provided so that new control functions could be added using this extension character. Several two-character extension sequences have already been added.
225	NAK	A negative response returned to the sender by the receiver.
226	SYN	Used in synchronous transmission systems to provide a signal pattern from which synchronism may be attained or maintained. It is placed at the beginning of all transmitted character sequences and inserted in a sequence of characters in the absence of a data character to be transmitted.
227	ETB	Terminates a transmission block (heading or text) which is not the last block of message.

Basically, an internal message ("file") is composed of a number of

ASCII lines terminated by a New Line character .NL. (012). The .NL. character is printed as a .CR. (Carriage Return) followed by a .LF. (Line Feed) followed (possibly) by a number of .NUL. (null) characters for timing.

Within each ASCII line, carriage motion is defined by the following characters.

<u>Name</u>	<u>Code</u>	<u>Meaning</u>
.SP.	(240)	Space Forward One Position
.BS.	(210)	Space Backward One Position
.HT.	(211)	Physical Horizontal Tab
.VT.	(213)	Physical Vertical Tab
.FF.	(214)	Form Feed (Top of Form)
.CR.	(215)	Carriage Return
.RHT.	(221)	Relative Horizontal Tab, following byte Defines a number of .SP. to insert
.HLF.	(222)	Half Line Feed Forward
.RVT.	(223)	Relative Vertical Tab, following byte Defines a number of .LF. to insert
.HLR.	(224)	Half Line Feed Reverse

In addition, the following characters are used internally for specific device action.

.BEL.	(207)	Audible Alarm
.RRS.	(216)	Red Ribbon Shift
.BRS.	(217)	Black Ribbon Shift

The following characteres are used for packing and compression:

<u>Name</u>	<u>Code</u>	<u>Meaning</u>
.NUL.	(200)	Allowed and Ignored in any Position
.RCP.	(220)	Relative Copy - following byte specifies number of characters to copy from corresponding positions of preceding line



## VISIBLE STANDARDS

Several standards have been adopted for keyboard interfaces with standard software. Specifically:

<u>Name</u>	<u>Code</u>	<u>Meaning</u>
"	(242)	Erase (ignore last character typed on the current line)
?	(277)	Kill (restart current line)
(shift L)	(234)	Logical Tab (for editor, ED)
^	(236)	Logical Escape (visual escape for limited graphic devices)
.CR.	(215)	Interpreted as .NL. on Keyboard
.LF.	(212)	Ignored input

The logical escape conventions at present include:

^ ddd	Three octal digit representation of unprintable character such as ^007 (BEL)
^-	Backspace
^U	All subsequent letters are upper case
L	all subsequent letters are lower case until end of line

Table D-3

Keypunch Codes, Card Codes, &amp; Internal ASCII Codes for Prime Computers.

ASCII Code	Card Zone	Code No.	IBM 26 Char	IBM 29 CHAR
240	None		Space	Space
261	-	1	1	1
262	-	2	2	2
263	-	3	3	3
264	-	4	4	4
265	-	5	5	5
266	-	6	6	6
267	-	7	7	7
270	-	8	8	8
271	-	9	9	9
272	-	8-2		:
243	-	8-3	#	#
300	-	8-4		,
247	-	8-5		=
275	-	8-6		"
242	-	8-7		0
260	0	-	0	0
257	0	1	/	/
323	0	2	S	S
324	0	3	T	T
325	0	4	U	U
326	0	5	V	V
327	0	6	W	W
330	0	7	X	X
331	0	8	Y	Y
332	0	9	Z	Z
333	0	8-2		
254	0	8-3	,	,
245	0	8-4		
337	0	8-5		(underscore)
276	0	8-6		>
277	0	8-7		?
255	11	-	-	-
312	11	1	J	J
313	11	2	K	K
314	11	3	L	L
315	11	4	M	M
316	11	5	N	N
217	11	6	O	O
320	11	7	P	P
321	11	8	Q	Q
322	11	9	R	R
241	11	8-2		!

244	11	8-3	\$	\$
252	11	8-4	*	*
251	11	8-5		)
273	11	8-6		;
335	11	8-7		(not sign)
246	12	-		
301	12	1	A	A
302	12	2	B	B
303	12	3	C	C
304	12	4	D	D
305	12	5	E	E
306	12	6	F	F
307	12	7	G	G
310	12	8	H	H
311	12	9	I	I
336	12	8-2		(cent sign)
256	12	8-3	.	.
274	12	8-4		<
250	12	8-5		(
253	12	8-6		+
334	12	8-7	{	



## APPENDIX E

## I-O VIRTUALIZATION

## INTRODUCTION

All all user programs running under PRIMOS III are executed in restricted mode, all I/O instructions executed by a user program cause traps to the supervisor.

Because user I/O instructions in virtual memory operation cause a trap, a mechanism is provided for user programs to perform a supervised form of I/O. This is accomplished by defining functional means of allowing certain devices to operate via user I/O commands (I/O virtualization). These devices are listed in Table E-1 along with the implemented values of the Virtual Memory Systems Controller Board Control Word values for input and output, and the associated port to which the devices are connected.

Table E-1. System Controller Board Control Word,  
Device, and Port Relationships

---

Port No.	Control Word Values		Device
	Input (Bits 11,12)	Output (Bits 13-16)	
1	00	000 (or 10 (octal))	User Terminal
2	01	100 (4 (octal))	CENPR (J2)
3	10	010 (2 (octal))	CE2PR (J3)
4	11	001 (1 (octal))	CARDR (J4)

---

A subset of all possible I/O functions that can be performed with a given device is defined; the PRIMOS III and IV system provides a mechanism for calling the supervisor to perform these I/O functions.

The PRIMOS III operating system provides a functional interpretation of most I/O instructions relating to the Controller Option. These I/O instructions include the following:

OCP 4, OCP 104

INA 4, INA 1004, INA 1204, INA 1304

OTA 4, OTA 104

SKS XX04

#### SYSTEM CONTROLLER CONTROL WORD

For every user terminal connected (logged-in) to the PRIMOS III operating system, a register is maintained that stores a virtual-memory-systems- command. (The instructions to initialize this register are OCP 4, OCP 104). The control word is set equal to the A-register by the instruction OTA 104, and the control word can be read by executing the instruction INA 1204.

The control word may also be set by the ASRCWD command. Only the port select fields of the control word (bits 11-16) are used when the INA 4, OTA 4 instruction sequence is executed.

#### INPUT/OUTPUT BUFFERS

I/O with Port 1 selected is performed through the user terminal buffers maintained by the operating system supervisor. I/O is always full-duplex. Ports 2, 3, and 4 have three associated buffers. Access to these buffers is allowed only if the corresponding I/O device has been assigned to the user's process by means of the ASSIGN command.

#### DATA TRANSFERS

##### Input

Execution of the INA 4 (INA 1004) instruction causes a transfer of a character from the buffer associated with the assigned device to the A-register. If the buffer is empty, the user's process is placed in INPUT-WAIT state, and the supervisor cycles to service another user's process. The user process is rescheduled when the requested input arrives.

##### Output

Execution of the OTA 4 instruction causes a transfer of a character from the A-register to the buffer associated with the assigned device. If the buffer fills up, the user's process is placed in the OUTPUT-WAIT state. Users are removed from the OUTPUT-WAIT state once per second. At that time, the user process is rescheduled to the location following the OTA instruction; no skip occurs.

### Emptying and Filling Buffers

The device interface modules (interrupt routines) empty and fill their associated buffers. The physical device may be different from the logical I/O device. For example: when an INA 4, OTA 4 instruction sequence is executed in the virtual memory system, the system performs output on the AMLC.

### SKIPS

PRIMOS III, on encountering an SKS instruction, always skips (with the exception of SKS 704, skip if receiver ready, and SKS 604). The SKS 704 skips only if there is input available either in the buffer associated with the user terminal or in the port that is specified by the virtual control word. The virtual control word is initially set to the user terminal, either by the ASRCWD command or the OTA 104 instruction. SKS 704 skips only if there is room in either the output buffer associated with the user terminal, or in the port that is specified by the virtual control word.

A user program may SKS for terminal input and input a character if one is available, or perform other computation if no character is available. No existing IOCS routines or other teletype routines such as T1IN, T1OU1, TNOUA, etc. in the FORTRAN library performs an SKS 704 or SKS 604.

### Paper-Tape Reader

To interface a paper tape reader with virtual memory, interpretation of the following instructions is provided:

OCP XX01 (treated as NOP's)

SKS XX01 (always SKIP)

INA 1, INA 1001

Execution of the INA 1 (INA 1001) instruction causes a transfer of a character from the paper-tape reader buffer to the A-register, and the INA instruction skips. If the buffer is empty, the INA is handled as NOP. The reader must be ASSIGNED by the user. An interrupt routine (PTRDIM) maintains the buffer full by reading the paper tape as long as there is room in the buffer.

### Paper-Tape Punch

To interface a paper tape punch with virtual memory, interpretation of the following instructions is provided:

OCF XX02 (treated as a NOP)

SKS XX02 (always a SKIP)

OTA XX02 (output character)

Execution of the OTA XX02 instructions causes a transfer of a character from the A-register to the paper tape punch buffer, and the OTA instruction skips. If the buffer is full, a user process goes into OUTPUT- WAIT state for up to one second. A restart is then made to the location following the OTA (no SKIP). An interrupt routine (BRPDIM) punches characters from the punch buffer until the buffer is empty. The punch must be ASSIGNED by the user.

#### CPU Control Panel

To interface the CPU control panel with virtual memory, interpretation of the following instructions is provided:

INA 1620 (read sense switches)

OTA 1720 (output lights)

A virtual sense-switch-register and a lights-register are maintained for each user that is logged-in. The sense-switch-register is set by the VRTSSW command and read by the instruction INA 1620. The lights-register is set from the A-register by executing an OTA 1720 instruction. The lights-register is displayed on the control panel by entering the memory address on the panel sense switches and setting the ADDRESS/DATA switch to DATA. The memory address is computed by taking the sum of 12377 plus the terminal number (number typed on login).

#### SVC INTERFACE CONSIDERATIONS

##### Disk

The disk interfaces with virtual memory through a supervisor call (SVC) instruction to perform a READ or WRITE operation on a single physical record of a physical disk. The disk must be assigned to the terminal by the ASSIGN command. Refer to RREC and WREC in the File System User Guide (MAN 2604). For information about the SVC instruction, refer to the Systems Reference Manual and the PMA User Guide.

##### Magnetic Tape

Input/Output operations for magnetic tape are effected by PRIMOS III through SVC calls. Refer to T\$MT in the File System User Guide (MAN 2604).



MPC Line Printer

Output to the parallel interface line printer is accomplished through SVC calls. Refer to T\$LMPC in the File System User Guide (MAN 2604).

MPC Card Reader

Input from the parallel interface card reader is controlled through SVC calls. Refer to T\$CMPC in the File System User Guide (MAN 2604).

## SVC VIRTUALIZATION

To allow debugging or execution of other operating systems, PRIMOS III allows virtualization of all SVC calls except a class of SVC's considered exclusive to PRIMOS III. (Function codes XXX5XX). This capability is turned off on LOGIN and can be set by the following commands:

SVCSW 1 turn-on virtual SVC handling

SVCSW 0 turn-off virtual SVC handling

If the SVCSW is turned on, the SVC instruction executed by a user program having a word following the SVC that is not of the form XXX5XX, results in a virtual trap through location '65.

## Example:

Assume that a version of PRIMOS that performs disk I/O using the PRIMOS III RREC/WREC SVC calls is stored in the UFD CMDNC0 under the name, UDOS64. Thus, a user may ASSIGN a disk to a terminal, turn on SVC calls and run PRIMOS. The following sequence shows a typical operation.

<u>User Input</u>	<u>Effect</u>
ASSIGN DISK 50	Assigns physical disk 50 to user.
AS DISK 51	Assigns physical disk 51 to user.
SVC 1	Turns on SVC virtual memory interface.
UDOS64 GO	Bring PRIMOS II into virtual memory and start execution. PRIMOS II types its usual message and OK:
STARTUP 50 51	Informs PRIMOS II or III to use physical disks 2 and 3.
A LIB	Attaches to any desired UFD.
FIXRAT	Performs VIRTUAL FIXRAT(of user desires) .
COPY GO FROM-TO: 50 5	Copy physical 50 to physical 5.
SHUTDN	Directs PRIMOS II to perform normal clean up functions prior to shutting down.
Press 'QUIT'	Returns to PRIMOS II command level.
UNASSIGN DISK 50	Releases physical disk 50.
UNASSIGN DISK 51	Releases physical disk 51.
SVC	Turns off SVC virtual memory interface.

Table F-2 is a list of SVC codes used by PRIMOS III (SVC codes are not applicable to PRIMOS IV users).

#### OTHER VIRTUALIZATION

Unimplemented Instructions (UII) floating-point exceptions (FLEX), and Procedure Stack Underflow (PSU) are also virtualized (i.e., these cause interrupts that vector the trap location in the user's virtual address space). For optimal performance, the appropriate hardware configuration is recommended.

Table F-2. SVC's Numbers Used by PRIMOS III.

<u>SVC Number</u>	<u>Associated Call</u>
100	ATTACH (ufdnam, ldev, passwd, key, altrtn)
1	SEARCH (key, name, unit, altrtn)
2	SAVE (rvec, name)
3	RESTOR (rvec, name, altrtn)
4	RESUME (name)
5	EXIT
6	ERRTN (altrtn, a1, a2, a3)
7	UPDATE (1,0)
110	GETERR (buff, nw)
1	PRERR
2	GINFO (abuff, nw)
3	CNAME (oldnam, newnam, altrtn)
4	ERRSET (altval, altrtn, a1, a2, a3)
5	FORCEW (key, unit)
202	RDLIN (unit, line, nw, altrtn)
3	WTLIN (unit, line, nw, altrtn)
300	PRWFIL (key, unit, LOC(buff), nw, posv, altrtn)
500	RREC (pbav, nwv, nchn, ra, pdev, altrtn)
1	WREC (pbav, nwv, nchn, ra, pdev, altrtn)
2	TIMDAT (buff, nw)
3	-- reserved
4	-- reserved
5	RECYCL
6	D\$INIT (pdev)
7	BREAK\$ (onoff)
510	T\$MT (unit, LOC(buff), nw, inst, statv)
1	T\$LMPC (unit, LOC(buff), nw, inst, statv)
2	T\$CMPC (unit, LOC(buff), nw, inst, statv)
3	T\$AMLC (line, ba, charent, key, statv, altrtn)
4	T\$VG (Unit, ba, nw, inst, statv)
600	COMANL
1	CLIN (char)
2	CMREAD (buff)
3	COMINP (name, unit, altrtn)
4	CNIN\$ (buff, charent)
700	T1IN (char)
1	T1OU (char)
2	TNOU (msg, cnt)
3	TNOUA (msg, cnt)
4	TOOCT (num)
5	DUPLX\$ (argument)
1000	T\$MT See 510

1	T\$SLC	(key, 1in LOC(buff), nw)
1100	T\$LMPC	See 511
1200	T\$CMPC	See 512

# INDEX

\$E 3-27  
 \* [COMMENT] 3-108  
 0.5 BILLION BYTES 1-2  
 2.4 BILLION BYTES 1-2  
 32-BIT REGISTERS 3-63  
 60-MILLION BYTE DISK 1-1  
 64-WORD BLOCKS 3-103  
 64K PRIMOS II 3-101  
 7-BIT D-1  
 80-MILLION BYTE DISK 1-1

## A

A 3-13  
 A-REGISTER 3-36, 3-64, 3-74  
 ABBREVIATION 3-2  
 ABBREVIATION OF A COMMAND 3-6  
 ACCESS PROTECTION 2-4  
 ACCESS RIGHTS 3-13  
 ACCESS TO SYSTEM 2-4  
 ACCESSING FILES 3-13  
 ACTION COMMANDS 3-51  
 ACTION KEY 3-57  
 ADDISK 3-7, 4-3, A-1  
 ALTERNATIVES 3-2  
 AMLC A-1  
 AMLC COMMAND 3-7  
 AMOUNT OF SPACE ON DISK 3-16  
 ANOTHER USER 3-105  
 ANSI D-1  
 ARGUMENTS 3-6  
 ARGUMENTS TO COMMANDS 2-6  
 AS 3-9  
 ASCII 2-9, 3-3, 3-20, 3-33, D-1  
 ASCII COMMUNICATIONS CODES D-2  
 ASCII FILE 3-20, 3-26  
 ASCII SOURCE FILE 3-39  
 ASR 3-8  
 ASRCWD 3-8, A-1  
 ASSEMBLY 3-64  
 ASSIGN 3-9, 3-35, 3-45, 3-54, 3-6  
 9, 3-105, A-1  
 ASSIGN A DISK 3-11  
 ASSIGN CENPR 3-90  
 ASSIGN EXAMPLES 3-11  
 ASSIGN PTP 3-70  
 ASSIGN PTR 3-70  
 ASSIGN TABLE FULL 3-12  
 ASSIGNABLE DISKS TABLE 3-11  
 ASSIGNED 3-27  
 ATTACH 2-10, 2-24, 3-4, 3-13, A-1  
 ATTACH COMMAND 2-11, 3-13  
 ATTACH EXAMPLES 2-11, 3-14  
 ATTACH KEYS 2-17, 3-15

ATTACHING TO A UFD AND SUB UFD'S  
 3-15  
 ATTACHING TO A SUBDIRECTORY 2-13  
 ATTACHING TO MFD 2-12  
 ATTACHING TO UFD 2-11  
 ATTACHING TO UFD ON ANOTHER DISK  
 2-12  
 ATTRIBUTES OF FILES 3-41  
 AVAIL 3-16, A-1  
 AVAIL EXAMPLES 3-16  
 AVAILABLE RECORDS 3-16

## B

B-REGISTER 3-36, 3-7  
 BASIC 3-18, 3-40, 3-56, A-1  
 BASIC LANGUAGE 3-18  
 BASIC PROGRAM 3-56  
 BASINP 3-18, A-1  
 BATCH OPERATING SYSTEM 1-2  
 BAUD 1-2  
 BINARY 3-19, 3-36  
 BINARY EDITOR 3-32  
 BINARY EDITOR COMMANDS A-4  
 BINARY FILE 3-19, 3-64  
 BINARY FILE 3-36  
 BIPOLAR CACHE MEMORY 1-2  
 BOOT 2-12, 3-54  
 BR 3-86  
 BUFFERS 3-19  
 B FILE 3-89  
 B XXXX 3-19, 3-36, 3-64

## C

C 3-19  
 CARD IMAGE ASCII DATA 3-27  
 CARD READER 1-2, 3-60  
 CARD READING 3-27  
 CARD SPOOLER 3-91  
 CARDR 3-10, 3-27  
 CARRIAGE RETURN 2-6  
 CE2PR 3-10  
 CENPR 3-10  
 CENTRAL PROCESSOR 1-1  
 CENTRONICS PRINTER 3-12  
 CHAINING OF COMMAND FILES 3-22  
 CHANGING DISK PACKS 3-83  
 CHANGING NAME OF A FILE 2-18  
 CHANGING TEXT 2-9  
 CHARACTER PRINTER 1-2  
 CHARACTER SET 2-9  
 CHARACTERS 2-4  
 CHARACTERS PER WORD 2-7

# INDEX

CLOSE 3-4, 3-19, A-2  
 CLOSE ALL 3-19, 3-22  
 CMDNCØ 2-12, 3-54  
 CMPRES 3-20, 3-32, A-2  
 CMPRES EXAMPLE 3-20  
 CNAME 2-18, 3-21, A-2  
 CNVIMA 3-21, A-2  
 CO 3-22  
 CODE 3-36  
 COLLATING SEQUENCE D-1  
 COMINP 3-22  
 COMINPUT 2-18, 3-22, A-2  
 COMINPUT EXAMPLE 3-23  
 COMINPUT PAUSE 3-23  
 COMMAND COMPLETION 2-1  
 COMMAND DESCRIPTIONS 3-6  
 COMMAND DEVICE 3-96  
 COMMAND ERROR CORRECTION 3-3  
 COMMAND EXECUTION 2-6  
 COMMAND FILE 2-18, 3-1, 3-19, 3-2  
 2, 3-60, 3-86  
 COMMAND FILES 3-108  
 COMMAND FORMAT 2-6, 3-1  
 COMMAND FUNCTIONS 3-1  
 COMMAND LANGUAGE 3-1  
 COMMAND LEVEL 3-2, 3-5  
 COMMAND LINE 3-3  
 COMMAND LINE TERMINATION 2-6  
 COMMAND NAME 2-7, 3-1  
 COMMAND NAMES 3-6  
 COMMAND STRING 3-2  
 COMMAND STRUCTURE 3-1  
 COMMANDS 2-5, 2-12, 3-1, 3-22  
 COMMANDS ALLOWED IN PRIMOS III 3-3  
 COMMANDS UNIQUE TO PRIMOS III 3-6  
 COMMENT 3-108  
 COMMENT LINES IN COMMAND FILE 2-1  
 9  
 COMPATIBLE 1-2  
 COMPILE AND RUN EXAMPLE 2-24  
 COMPILING A PROGRAM 2-20  
 COMPUTE TIME 3-102  
 COMPUTER RESOURCES 1-1  
 CONCEPT OF ATTACHMENT 2-10  
 CONCEPT OF DIRECTORY 2-10  
 CONFIG 3-25, 3-60  
 CONFIGURATION 3-94  
 CONFIGURING THE NETWORKS 4-2  
 CONFLICTING FILE TYPES 3-48  
 CONNECT TIME 3-45, 3-102  
 CONSOLE DATA INDICATORS 3-87  
 CONTENT OF COMMAND FILE 2-19  
 CONTENTS OF COMMAND UFD 3-5

CONTINUE 3-22  
 CONTROL WORD E-1  
 CONTROL-P 3-2  
 COPY 3-25, A-2  
 COPYING A FILE 2-28  
 COPYING A UFD 2-28  
 CPU CONTROL PANEL E-4  
 CPU TIME 3-45  
 CR1 3-10, 3-27  
 CRDSPL 3-91  
 CREATE A FILE 2-9  
 CREATE COMMAND 3-26, A-2  
 CREATE EXAMPLE 3-26  
 CREATES A DISK 3-54  
 CREATING A FILE 2-16  
 CRMPC 3-27, A-2  
 CRSER 3-27, a-2  
 CRT 1-2  
 CURRENT MFD 3-26  
 CURRENT UFD 2-10, 2-11, 3-13, 3-2  
 6, 3-41, 3-73

## D

DAM 2-28  
~~DAM 3-57~~  
 DAM SEGMENT DIRECTORY 3-57  
 DATA TRANSFERS E-2  
 DATE 3-51  
 DBASIC 3-18, 3-27  
 DEBUGGING 2-28, 3-65, 3-69, 3-101  
 DEFAULT COMMANDS FOR RUNOFF 3-79  
 DEFECTIVE FILES 3-35  
 DELAY COMMAND 3-28  
 DELAY EXAMPLE 3-28  
 DELETE 2-17, 3-29  
 DELETE A DIRECTORY 3-29  
 DELETING A FILE 2-17  
 DELETING LINES OF TEXT 2-9  
 DEVICE 3-45, 3-105  
 DEVICE IN USE 3-11  
 DEVICE NAMES 3-10  
 DEVICE NOT ASSIGNED 3-11  
 DIAGNOSTICS 2-9  
 DIFFERENCE FILE 3-71  
 DIRECTORIES 2-28, 3-35  
 DIRECTORY 2-10, 2-11, 3-13  
 DIRECTORY FILE 3-35  
 DIRECTORY TREENAME 3-37  
 DISK 2-12, 2-12, 3-10, 3-11, 3-32,  
 3-54, 3-73, E-4  
 DISK CONTROLLER 1-1  
 DISK DRIVE 3-94  
 DISK ERRORS B-1

# INDEX

DISK FILE 3-47  
 DISK FORMAT 1-3  
 DISK I/O TIME 3-102  
 DISK NAME 3-96  
 DISK NO. 3-10  
 DISK OPERATING SYSTEM 1-1  
 DISK OPTIONS 1-1  
 DISK PACK 3-35  
 DISK RECORDS 3-16  
 DISK STATUS WORD B-1  
 DISK STORAGE SPACE 3-29  
 DISK VS. PRIMOS UNITS 3-4  
 DISKETTE 1-1  
 DISKETTE CONTROLLER 1-1  
 DISKS 1-1, 3-96, A-3  
 DISKS COMMAND 3-11  
 DISKS TABLE 3-11  
 DONSTP 4-3  
 DOS 2-12  
 DOUBLE PRECISION ARITHMETIC 3-27  
 DOUBLE PRECISION FLOATING POINT 3-18  
 DSKRAT 2-12, 3-54, 3-96  
 DSKRAT FILE 3-35  
 DSKRAT NAME 3-16

## E

EA 3-73  
 ED 2-9, 2-24, 3-30, A-3  
 EDB 3-32, A-4  
 EDIT MODE 2-10, 2-24  
 EDITING A PROGRAM 2-20  
 EDITOR 2-10  
 EDITOR COMMANDS A-3  
 EDLIN 3-30  
 EIGHT MILLION BYTES 1-2  
 ELM 3-80  
 EMPTY UFD 2-24  
 EMPTYING & FILLING BUFFERS E-3  
 ENDING ADDRESS 3-74  
 ENTER LOGICAL TAPE NUMBER: 3-51  
 ER! 3-3  
 ERASE CHARACTER 3-3  
 ERROR 3-3  
 ERROR CORRECTION 3-3  
 ERROR MESSAGE 2-8, 3-3, 3-22, 3-65, 3-90, 3-103  
 ERROR WHILE ATTACHING 2-17  
 ERRORS 2-6, 3-3, 3-49  
 ERRSET 3-65  
 ERRVEC CONTENTS E-1  
 ESSENTIAL FILES 2-12  
 EXAMPLE OF DELETE 2-17

EXAMPLE OF PM 3-64  
 EXECUTING COMMANDS FROM A FILE 2-18  
 EXIT SUBROUTINE 3-74  
 EXPAND 3-20, 3-22  
 EXTERNAL COMMANDS 3-4, 3-58

## F

FAM 4-1  
 FILE (EDITOR FUNCTION) 2-9  
 FILE 2-6, 3-27  
 FILE ACCESS MANAGER 4-2  
 FILE AND DIRECTORY MANIPULATIONS 2-17  
 FILE COMMAND 2-10  
 FILE COMPLETE 3-48, 3-103  
 FILE CONTENTS 2-6  
 FILE CREATION 2-9  
 FILE DIRECTORY 3-13  
 FILE INTEGRITY 3-35  
 FILE MARK 3-47  
 FILE NAME 1-3  
 FILE NOT FOUND 3-45  
 FILE RECORDS 3-47  
 FILE SHARING 4-1  
 FILE SYSTEM 1-3  
 FILE SYSTEM 2-11, 3-13, 3-19  
 FILE TYPE 3-57  
 FILE UNIT 2 3-69  
 FILE UNIT 3 3-19  
 FILE UNIT 3-23, 3-57, 3-68  
 FILE UNIT 6 3-22  
 FILE UNITS 3-19  
 FILE UTILITY 2-28  
 FILE UTILITY 3-37  
 FILENAME 2-18, 3-2, 3-22, 3-39  
 FILENAMES 3-41  
 FILES 2-5, 2-28, 3-13, 3-35, 3-88  
 FILMEM 2-24, 3-32, A-5  
 FILVER 3-33, A-5  
 FIRST CHARACTER 3-3  
 FIX DISK 3-35  
 FIXED HEAD DISK 1-1  
 FIXRAT 3-29, 3-35, A-5  
 FLEX E-6  
 FLOATING POINT EXCEPTIONS E-6  
 FLOPPY DISKS 1-1  
 FORCED LOGOUT 3-45  
 FORCED LOGOUT 3-45  
 FORTRAN IV 3-36  
 FORTRAN PROGRAM 2-20, 3-80  
 FTN 2-24, 3-19, 3-36, 3-39, A-5  
 FULL DISK ERROR 3-89

# INDEX

FUNIT 3-4  
FUTIL 2-28, 3-29, 3-37, A-5

## G

GOULD 3-91  
GRAPHIC D-1

## H

HARDWARE D-1  
HEADER RECORD 3-47  
HIERARCHY 2-11  
HIGH SPEED READER-PUNCH 3-70  
HIGH-SPEED FIXED-HEAD DISKS 1-1  
HILOAD 3-39, 3-43, A-6  
HOME UFD 3-13, 3-51  
HYBRID COMMANDS 3-4

## I

I/O INSTRUCTIONS E-1  
I/O VIRTUALIZATION E-1  
ILL REMOTE REF 4-1  
INDEX 3-51  
INDEX A TAPE 3-48  
INDEX REGISTER 3-74  
INPUT 2-9, 3-4, 3-39, A-6  
INPUT A COMMAND 2-2  
INPUT E-2  
INPUT FILE ERROR 3-89  
INPUT LINES OF TEXT 2-9  
INPUT MODE 2-10, 2-24  
INPUT PARITY D-1  
INPUT/OUTPUT BUFFERS E-2  
INSERT 3-72  
INSERTING LINES OF TEXT 2-9  
INTERACTING COMMAND FILES 3-23  
INTERACTING WITH THE SUPERVISOR 3-2  
INTERACTIVE DEBUGGER 3-69  
INTERACTIVE DEBUGGING 3-69  
INTERACTIVE SESSION EXAMPLE 2-1  
INTERLEAVING MAIN MEMORY 1-2  
INTERNAL ASCII CODES D-2  
INTERNAL COMMANDS 3-4  
INTERNAL PARITY D-1  
INTERNAL VECTOR 3-65  
IOCS 1-2  
ISSUES COMMANDS 2-2

## K

KEY 3-57

KEY PARAMETER IN ATTACH 2-17  
KEYBOARD 2-2  
KEYS 3-74  
KILL CHARACTER 2-8, 3-3

## L

LARGE DISKS 1-1  
LATE 3-40, A-6  
LB 3-63  
LBASIC 3-18, 3-40, A-6  
LDEV 2-13  
LDISK 3-4  
LED 3-30  
LEVELS OF COMMUNICATION 3-2  
LIBRARIES 3-24  
LINE FEED 3-2, 3-28  
LINE PRINTER 1-2, 3-60, 3-88  
LINE SIZE 3-20  
LINE SIZE ERROR 3-89  
LINE SIZES 3-32  
LINK BASE REGISTER 3-63  
LINKING LOADER 3-43  
LISTF 2-10, 2-24, 3-41, A-6  
LISTF EXAMPLES 2-11  
LISTING 3-4, 3-36, 3-64, A-6  
LISTING FILE 3-36  
LISTING FILES IN UFD 2-11  
LO 3-45  
LOAD 2-24, 3-43, A-6  
LOAD MAP 2-24  
• LOAD20 3-43  
LOADER 2-24  
LOADER SAVE COMMAND 2-25, 3-80  
LOCATE 2-9  
LOCATIONS 20000 - 24777 3-69  
LOG IN 2-4  
LOGICAL DEVICE 3-41  
LOGICAL DEVICE NUMBERS 3-44  
LOGICAL DISK 3-16  
LOGICAL DISK NUMBER 2-12  
LOGICAL DISK UNIT NUMBERS 3-4, 3-94  
LOGICAL DISK UNITS 3-4  
LOGICAL DRIVES 3-94  
LOGICAL TAPE 3-48  
LOGICAL TAPES 3-47  
LOGICAL UNIT NUMBERS 3-4  
LOGICAL-TO-PHYSICAL ASSIGNMENT 3-94  
LOGIN 2-4, 2-10, 3-44, 3-58, A-7  
LOGIN COMMAND 3-101  
LOGIN SPOOL SPOUT 3-90  
LOGIN UFD 2-10



# INDEX

LOGOUT 3-11, 3-45, 3-58, 3-61, 3-105, A-7  
 LONG PROGRAMS 3-60  
 LOW BOUNDARY OF PRIMOS II 3-95  
 LOW MEMORY 3-4  
 LOWER-CASE CHARACTERS 3-106  
 L FILE 3-89  
 L XXXX 3-36  
 L XXXX 3-64

## M

MACHINE CHECK MODE 3-46  
 MACHK 3-46, A-7  
 MACRO ASSEMBLER 3-64  
 MAGNETIC TAPE - FILE UTILITIES 3-47  
 MAGNETIC TAPE 1-2, E-2  
 MAGRST 3-47, A-7  
 MAGSAV 3-51, A-7  
 MAINTENANCE PROGRAM 3-35  
 MAKE 3-54, A-7  
 MASS STORAGE 1-1  
 MASTER DISK 3-18  
 MASTER DISK 3-36  
 MASTER FILE DIRECTORY 2-12  
 MAT FUNCTIONS 3-40  
 MAXIMUM NUMBER OF DISK DRIVES 3-12  
 MCG 3-54, A-7  
 MDL 3-54, A-7  
 ME 3-86  
 MEMORY 1-2  
 MEMORY 3-73  
 MEMORY LOCATIONS 3-32  
 MEMORY-RESIDENT OPERATING SYSTEM 1-1  
 MERGED FILES 3-86  
 MESSAGE 3-65, A-8  
 MESSAGE COMMAND 3-55  
 MESSAGE EXAMPLE 3-55  
 MFD 2-12, 3-13, 3-15, 3-16, 3-35, 3-52, 3-54  
 MICROCODE ASSEMBLY 3-54  
 MILESTONE MESSAGE 3-103  
 MINIMUM CONFIGURATION 1-2  
 MOUNTING A NEW DISK PACK 3-96  
 MOVING HEAD CARTRIDGE DISKS 1-1  
 MPC LINE PRINTER E-5  
 MT1 ... MT4 3-10  
 MTDISK A-8

## N

NAME OF FILE 3-21  
 NAME OF UFD 2-12  
 NAME: 3-51  
 NAMED FILES 3-19  
 NAMES 3-3  
 NETWORK 3-55, 4-1  
 NETWORK SOFTWARE 4-1  
 NEW FILE 3-57  
 NEW UFD 3-26  
 NINE-TRACK MAGNETIC TAPE 3-47  
 NONINTERACTIVE PROGRAMS 3-60  
 NONOWNER 3-26  
 NONOWNER STATUS 3-13  
 NONOWNER-PASSWORD 3-59  
 NORMAL LISTING DETAIL 3-64  
 NORMAL MODE 3-101  
 NUMBER 3-56, A-8  
 NUMBER OF PASSES 3-87  
 NUMBER OF PHANTOMS 3-60  
 NUMBER OF WORDS PER COMMAND 2-7

## O

OCTAL PARAMETER 3-2  
 OFF-LINE PRINTING 3-88  
 OK 2-6  
 OK, 3-44  
 OK: 3-1  
 OPEN 3-4, 3-19, 3-57, A-8  
 OPEN FILE UNITS 3-95  
 OPEN FOR READING 3-57  
 OPEN FOR READING AND WRITING 3-57  
 OPEN FOR WRITING 3-57  
 OPERATOR 3-3, 3-91  
 OPERATOR PRIORITY 3-58  
 OPRPRI 3-58, A-8  
 OPTIONAL 3-2  
 OPTIONS 3-35  
 OUTPUT 3-19, E-2  
 OUTPUT FILENAME 3-32  
 OUTPUT PARITY D-1  
 OVERWRITE 3-48  
 OWNER 3-26  
 OWNER STATUS 3-13, 3-21  
 OWNER-PASSWORD 3-59

## P

PACKNAME 3-16, 3-96  
 PAGING DEVICE 3-96  
 PAGING DISK 3-11  
 PAGING TIME 3-45  
 PANEL LOAD 3-54  
 PAPER TAPE 1-2, 3-12, 3-32

# INDEX

PAPER TAPE READER E-3  
 PAPER TAPE READER/PUNCH 1-2  
 PAPER TAPES 3-54, 3-70  
 PAPER-TAPE PUNCH E-3  
 PARALLEL INTERFACE CARD READER 3-27  
 PARAMETERS 3-2  
 PARITY BIT D-1  
 PARTITION OF DISK 2-12  
 PASSWD 3-41, 3-59, 3-66, A-8  
 PASSWORD 2-12, 3-13, 3-16, 3-44  
 PASSWORDS 3-26  
 PAUSE 3-22  
 PB 3-63  
 PC 3-74  
 PDEV 2-13  
 PDISK 3-94  
 PERIPHERALS 1-2  
 PHANTOM 3-60, 4-2, A-8  
 PHANTOM TTY REQUEST 3-45, 3-61  
 PHANTOM USER 3-45, 3-60  
 PHYSICAL DISK DRIVES 3-4  
 PHYSICAL DISK DRIVE = 3-35  
 PHYSICAL END OF TAPE 3-49  
 PHYSICAL UNIT NUMBER 3-51  
 PHYSICAL-TO-LOGICAL DEVICE 3-95  
 PM 2-25, 3-63, 3-74, A-8  
 PM EXAMPLE 3-63  
 PM FOR PRIMOS IV 3-63  
 PMA 3-19, 3-39, 3-64, A-8  
 POSITIONS 3-51  
 POST MORTEM 3-63  
 PRERR 3-65, A-8  
 PRIME 100 1-1  
 PRIME 100/200/300 PROCESSOR MEMORY LOCATION 3-74  
 PRIME 200 1-1  
 PRIME 300 1-1  
 PRIME 400 1-1  
 PRIME 400 3-36, 3-63  
 PRIMOS 3-4  
 PRIMOS ERROR MESSAGES B-1  
 PRIMOS II 1-1  
 PRIMOS II 3-1  
 PRIMOS II 3-2  
 PRIMOS II 3-3, 3-32, 3-36  
 PRIMOS II 3-54  
 PRIMOS II 3-54  
 PRIMOS II 3-59  
 PRIMOS II 3-63  
 PRIMOS II 3-69  
 PRIMOS II 3-70  
 PRIMOS II 3-83  
 PRIMOS II 3-94  
 PRIMOS II FEATURES 1-2  
 PRIMOS II FEATURES 1-2  
 PRIMOS II FILE UNITS 3-4  
 PRIMOS III 1-1  
 PRIMOS III 1-3  
 PRIMOS III 3-1  
 PRIMOS III 3-101  
 PRIMOS III 3-102  
 PRIMOS III 3-105  
 PRIMOS III 3-106  
 PRIMOS III 3-101  
 PRIMOS III 3-103  
 PRIMOS III 3-105  
 PRIMOS III 3-108  
 PRIMOS III 3-2  
 PRIMOS III 3-21  
 PRIMOS III 3-28  
 PRIMOS III 3-3  
 PRIMOS III 3-40  
 PRIMOS III 3-44  
 PRIMOS III 3-45  
 PRIMOS III 3-47  
 PRIMOS III 3-54  
 PRIMOS III 3-59  
 PRIMOS III 3-60  
 PRIMOS III 3-63  
 PRIMOS III 3-69  
 PRIMOS III 3-70  
 PRIMOS III 3-83  
 PRIMOS III 3-95  
 PRIMOS III FEATURES 1-3  
 PRIMOS III FILE UNITS 3-4  
 PRIMOS III MINIMUM REQUIREMENTS 1-2  
 PRIMOS III ON THE PRIME 400 1-2  
 PRIMOS III STARTUP 3-60  
 PRIMOS IN NETWORKS 4-1  
 PRIMOS IV 1-1  
 PRIMOS IV 3-105  
 PRIMOS IV 3-108  
 PRIMOS IV 3-21  
 PRIMOS IV 3-3  
 PRIMOS IV 3-47  
 PRIMOS IV 3-70  
 PRIMOS IV FEATURES 1-3  
 PRINT USING 3-40  
 PRINTER/PLOTTER 3-69, 3-92  
 PRINTING A FILE 2-19  
 PRINTING A FILE OFF-LINE 2-20  
 PRINTING FILE AT TERMINAL 2-19  
 PRINTING OF ^ CHARACTER 3-28  
 PRINTOUT OPTIONS 3-35  
 PRIMOSL RUN FILES 3-101  
 PRIMOSU RUN FILES 3-101

# INDEX

PRMPC 2-20, 3-20, 3-69, A-8  
 PROCEDURE BASE REGISTER 3-63  
 PROCEDURE STACK UNDERFLOW E-6  
 PROCESSOR'S REGISTERS 3-93  
 PROGRAM 3-2  
 PROGRAM COUNTER 3-74  
 PROGRAM DEVELOPMENT 1-1, 2-20  
 PROGRAMMING UNDER PRIMOS 2-20  
 PROGRAMS 1-33  
 PROMPT 3-2, 33-44  
 PROTECT 3-41, 3-66, A-9  
 PROTECTION 3-41  
 PROTOCOL 3-103  
 PRSER 2-20, 3-36, A-8  
 PRVER 3-69, A 9  
 PRVER EXAMPLE 3-69  
 PRWFIL 3-92  
 PRWFIL UNIT NOT OPEN 3-22  
 PSD 3-69, A-9  
 PSD20 3-69  
 PSU E-6  
 PTBOOT A-9  
 PTCPY 3-70  
 PTP 3-10  
 PTR 3-10, 3-32  
 PUNCH 3-10  
 PUSS 3-71, A-9

## Q

QUE A FILE FOR PRINT OR PUNCH 2-2  
 0  
 QUESTION MARK 2-8  
 QUIT 3-2, 3-93  
 QUOTE 2-8

## R

R 3-76  
 R 30000 3-69  
 R 50000 3-69  
 R 70000 3-69  
 READ CARDS 3-27  
 REAL MEMORY 1-2  
 REAL TIME OPERATING SYSTEM 1-2  
 RECEIVE 3-103  
 RECEIVER PROGRAM 3-103  
 RECORD 0 OF THE DISK 3-54  
 RECORDS IN FILE 3-84  
 RECYCL 3-40  
 REGISTERS 3-63  
 RELATIVE COPY CHARACTER 3-20  
 RENAMING A FILE 2-18  
 REPLACE 3-72

RESERVED WORDS 3-22  
 REST 3-73  
 RESTART A PROGRAM 3-93  
 RESTOR 3-58, 3-73, A-9  
 RESTORE COMPLETE 3-48  
 RESTORING SEQUENTIAL LOGICAL TAPE  
 S 3-48  
 RESTRICTED MODE E-1  
 RESUME 2-25, 3-76, A-9  
 RESUME COMMAND 3-58  
 RESUME SPLCEN 3-90  
 RESUME SPLMPC 3-90  
 RETURN 3 5  
 REV NO: 3-51  
 REVERSE SORTING 3-86  
 REVISION NUMBER 3-47  
 REWINDS 3-51  
 RING 3-64  
 ROOT 2-12  
 RTOS 1-2  
 RUNNING A PROGRAM 2-20  
 RUNNING SPOOL 3-88  
 RUNOFF 3-77, A-9  
 RUNOFF COMMANDS 3-77  
 RUNOFF DEFAULT COMMANDS 3-79  
 RVEC 3-80, 3-93  
 RVEC PARAMETERS 3-73  
 RVEC VECTOR 3-63

## S

S 3-27  
 SA 3-73  
 SAM 2-28, 3-57  
 SAM SEGMENT DIRECTORY 3-57  
 SAMPLE INTERACTIVE SESSION 2-1  
 SAVE 3-73, 3-80, A-9  
 SAVE AN ENTIRE DISK 3-52  
 SAVE COMMAND 3-73  
 SAVING SPACE 3-20  
 SB 3-63  
 SED 3-30  
 SEG 3-81, A-9  
 SEGMENT 3-64  
 SEGMENTED PROGRAMS 3-81  
 SEGMENTED RUN FILES 3-81  
 SELF-LOADING FORMAT 3-54  
 SEMICOLON 2-10  
 SENSE SWITCH 3-70  
 SEQUENTIAL DIRECTED FILE 3-57  
 SEQUENTIAL THREADED FILE 3-57  
 SERIAL INTERFACE 1-2  
 SERIAL INTERFACE CARD READER 3-27  
 SETIME 3-83, A-10

# INDEX

SFIN 3-36  
 SHUT DOWN 1-2  
 SHUTDN 3-3, 3-11, 3-83, 3-105  
 SHUTTING DOWN DISK 3-96  
 SHUTTING DOWN PRIMOS III 3-96  
 SIZE 3-41, 3-84, A-10  
 SKIPS E-33  
 SKS E-33  
 SLIST 2-19, 3-85, A-10  
 SOFTWARE 1-33, D-1  
 SOFTWARE DEVELOPMENT 1-1  
 SORT 3-86, A-10  
 SORT BRIEF 3-86  
 SORT EXAMPLE 3-86  
 SORT MERGE 3-86  
 SORT SPACE 3-86  
 SORTED FILE 3-86  
 SOURCE 3-19  
 SOURCE COMPARE 3-71  
 SOURCE FILE 3-36, 3-64  
 SOURCE PROGRAM 2-20  
 SP 3-86  
 SPACE CHARACTER 2-6  
 SPACES 2-8  
 SPACES IN COMMANDS 2-6  
 SPLCEN 3-88  
 SPLCRD 3-60  
 SPLMPC 3-60  
 SPLVER 3-91  
 SPOOL 2-20, 3-36, 3-88, A-10  
 SPOOL ARGUMENT 3-89  
 SPOOL DIRECTORY 3-89  
 SPOOLER 3-60  
 SPOOLER FOR PRINTER/PLOTTERS 3-91  
 STACK BASE REGISTER 3-63  
 STANDARD CHARACTER SET D-1  
 START 3-27, 3-64, 3-74, 3-93, A-10  
 START COMMAND 3-58  
 START UP 1-2  
 STARTING ADDRESS 3-73  
 STARTING FAM 4-2  
 STARTING REMOTE DISKS 4-3  
 STARTUP 3-3, 3-11, 3-105, 4-3, A-10  
 STARTUP OF PHANTOM 3-60  
 STATUS 3-19, 3-95, 4-3, A-11  
 STATUS COMMAND 2-13  
 STATUS KEYS 3-74  
 STOPPING A FILE PRINT BY SPOOL 3-91  
 STORAGE MODULES 1-1  
 SUBDIRECTORIES 2-11  
 SUBROUTINES 2-24  
 SUBSYSTEMS 2-9  
 SUBUFD 2-10, 3-15  
 SUPERVISOR 3-2  
 SUPERVISOR E-1  
 SUPERVISOR TERMINAL 3-11, 3-45, 3-58  
 SUPERVISOR TERMINAL 1-2  
 SVC 0 3-101  
 SVC 3-101  
 SVC E-4  
 SVC SWITCH 3-101  
 SVC VIRTUALIZATION E-5  
 SVC'S NUMBERS E-7  
 SVCSW 3-101  
 SVCSW A-11  
 SVCSW E-5  
 SYMBIONT 4-1  
 SYNTAX OF A COMMAND 3-1  
 SYSTEM CONTROLLER CONTROL WORD E-2  
 SYSTEM EDITOR 2-9  
 SYSTEM GENERATION 1-2  
 SYSTEM RESPONSE 2-6  
 SYSTEM SUBROUTINE 3-65  
 SYSTEM TERMINAL COMMANDS 3-3  
 SYSTEMS CONFIGURATIONS 1-1  
 SYSTEMS DEFINITIONS 1-1  
 SYSTEMS OPTION CONTROLLER 1-2  
  
 T  
 TAP 3-101  
 TAPE NAME 3-47  
 TAPE NAME: 3-51  
 TAPE READ OR WRITE ERRORS 3-49  
 TAPE RECORDS 3-47  
 TAPE UNIT 3-51  
 TARGET UFD 2-12  
 TELETYPE 1-2  
 TEMPORARY BASE REGISTER 3-63  
 TERMINAL 1-3, 2-2  
 TERMINAL KEYBOARD 2-2  
 TERMINALS 3-88  
 TERMINATE A LOGICAL TAPE 3-51  
 TEXT 2-9  
 TEXT EDITOR 3-22  
 TEXT FORMATTING 3-77  
 TIME 3-40, 3-102, A-11  
 TIME ACCOUNTING REGISTERS 3-102  
 TIME FUNCTION 3-28  
 TIMOUT 3-45  
 TRAMLC 3-103, A-11  
 TRANSMIT 3-103  
 TRANSMITTER PROGRAM 3-103