

PRIME

THE SYSTEM ADMINISTRATORS GUIDE IDR3109



P/N MAN3251-001

PRIME SOFTWARE DOCUMENTATION

| HIGH LEVEL LANGUAGE PROGRAMMER'S GUIDES | ASSEMBLY LANGUAGE REFERENCE GUIDES | OPERATING SYSTEM REFERENCE GUIDES | SOFTWARE SUBSYSTEM REFERENCE GUIDES | COMMUNICATIONS SUBSYSTEM REFERENCE GUIDES |
|--|---|--|--|---|
| <ul style="list-style-type: none">• FORTRAN IV PDR3057 PTU47• COBOL PDR3056 PTU48• RPGII IDR3031 FDR3275* PTU49• BASIC/VM IDR3058• INTERPRETIVE BASIC IDR1813 | <ul style="list-style-type: none">• SYSTEM ARCHITECTURE INSTRUCTIONS IDR3060 MAN1812*• PRIME MACRO ASSEMBLER PDR3059 PTU50 | <ul style="list-style-type: none">• PRIMOS COMMANDS PDR3108 FDR3250*• SYSTEM ADMINISTRATOR GUIDE IDR3109• FILE SYSTEM PDR3110 PTU51• SOFTWARE LIBRARY PDR3106 PTU52 | <ul style="list-style-type: none">• DATA BASE MANAGEMENT IDR3043 IDR3044 IDR3045 IDR3046 PTU55• EDITOR & RUNOFF FDR3104• MIDAS PDR3061 PTU54• SPSS PDR3173• FORMS IDR3040 PTU45 PTU53 | <ul style="list-style-type: none">• RJE/2780 PDR3067• HASP IDR3107 |

PRIME DOCUMENTATION TYPES

IDR Initial Documentation Release: provides usable, accurate advanced information.

PDR Preliminary Documentation Release: provides more complete and accurate information about the product, but is not in final format.

FDR Final Documentation Release: a complete product description; edited, formatted and presented in Prime's highest standards. **The Programmer's Companion*** is another type of FDR; a series of pocket-size, quick reference guides on Prime software products.

PTU Prime Technical Update: interim updates to existing documents.

THE SYSTEM ADMINISTRATOR'S GUIDE

This guide documents the operation of the Prime computer and its supporting systems and utilities, as implemented at Master Disk Revision Level 15. It is organized to make life easier for both the Operator and System Administrator.

Operator

The operator of a Prime system is the person authorized to use the Supervisor Terminal. The Supervisor Terminal accepts commands which configure and change system operation; these commands cannot be invoked from ordinary user terminals. One of the main functions of this guide is to define these commands and describe their effects on the system and user community. The Prime computer and its operating system, PRIMOS, are designed to simplify your tasks. We have extracted the information necessary for you to perform your job and presented it in the first 10 sections of this guide:

- Immediately getting you started on your primary function - bringing up the computer and making it available to the user community.
- Providing a detailed listing of tasks supporting and supplementing your primary function; referring you to the parts of this guide where their operations are detailed.
- Discussing the usual operating conditions and providing complete reference sections covering all cases.
- Including all the details on the essential tools
- Summarizing optional, convenience and advanced features
- Leaving out what is irrelevant.

The result is a single document containing everything needed for normal and for most exceptional situations (for other cases, references to other documents are made at the appropriate places in this guide).

Administrator

This IDR version concentrates primarily on the Operator's role. However, much information of use to the Administrator is included in the reference sections and the appendices. Information to be added in the next version includes discussions of strategies and philosophies to aid you in selecting system options and allocating system resources. This information will be based upon the experience of our own in-house systems as well as those of our users. This information has not been previously documented. It is currently obtained from Prime system analysts and customer training courses.

We hope you will find this to be a helpful guide to the operation and administration of a Prime computer system. We invite comments on the organization and philosophy of this guide, as well as its contents, accuracy, and clarity.

All correspondence on suggested changes to this document should be directed to:

Anthony R. Lewis, Technical Writer
Technical Publications Department
Prime Computer, Inc.
3 Newton Executive Park
Newton, MA 02162

Acknowledgements:

We wish to thank the members of the SYSTEM ADMINISTRATOR'S GUIDE team who contributed to and reviewed this IDR.

PRIME DOCUMENTATION TYPES

- IDR Initial Documentation Release: provides usable, accurate advance information.
- PDR Preliminary Documentation Release: provides more complete and accurate information about the product, but is not in final format.
- FDR Final Documentation Release: a complete product description; edited, formatted and presented in Prime's highest graphic standards.
- MAN Manual: early reference documents to be phased out by PDR's and FDR's.
- PTU Prime Technical Update: interim updates to existing documents.

Copyright 1978 by
Prime Computer, Incorporated
3 Newton Executive Park
Newton, Massachusetts 02162

The information in this document is subject to change without notice and should not be construed as a commitment by Prime Computer Corporation. Prime Computer Corporation assumes no responsibility for any errors that may appear in this document.

First Printing May 1978

TABLE OF CONTENTS

PART I - OVERVIEW

SECTION 1 OVERVIEW

| | |
|-----------------------------|-----|
| INTRODUCTION | 1-1 |
| CONVENTIONS AND DEFINITIONS | 1-5 |

PART II - THE OPERATOR

SECTION 2 SYSTEM STARTUP AND SHUTDOWN

| | |
|-------------------------|------|
| INTRODUCTION | 2-1 |
| STARTUP | 2-1 |
| SYSTEM SHUTDOWN | 2-11 |
| CONFIGURATION DATA FILE | 2-12 |
| STARTUP UP COMMAND FILE | 2-13 |

SECTION 3 THE OPERATOR AS USER

| | |
|----------------|-----|
| OVERVIEW | 3-1 |
| INTRODUCTION | 3-1 |
| FILE STRUCTURE | 3-1 |
| COMMANDS | 3-2 |

SECTION 4 OPERATOR TASKS

| | |
|--|-----|
| INTRODUCTION | 4-1 |
| ALLOCATION AND MONITORING OF RESOURCES | 4-1 |
| SUBSYSTEM USAGE | 4-4 |
| SYSTEMS HALTS: PREVENTION AND RECOVERY | 4-7 |
| DIRECT USER INTERACTIONS | 4-9 |

SECTION 5 MONITORING THE SYSTEM

| | |
|-------------------------------|------|
| OVERVIEW | 5-1 |
| LOOKING AT THE SYSTEM | 5-1 |
| SUPERVISOR TERMINALS MESSAGES | 5-7 |
| EVENT LOGGING IN PRIMOS | 5-9 |
| EVENT MESSAGE FORMAT | 5-11 |
| THE SYSTEM LOGBOOK | 5-15 |

SECTION 6 FIXRAT

| | |
|-----------------------|------|
| INTRODUCTION | 6-1 |
| FILE STRUCTURE | 6-2 |
| RUNNING FIXRAT | 6-5 |
| FIXRAT ERROR MESSAGES | 6-12 |

SECTION 7 MAKE

| | |
|------------------|-----|
| OVERVIEW | 7-1 |
| PRELIMINARY | 7-1 |
| RUNNING MAKE | 7-2 |
| FINAL OPERATIONS | 7-6 |

SECTION 8 BACKING UP

| | |
|---|-----|
| INTRODUCTION | 8-1 |
| BACKING UP ON DISK OR MAGNETIC TAPE | 8-2 |
| MAGSAV/MAGRST (DISK-TO-TAPE/TAPE-TO-DISK) | 8-8 |

SECTION 9 OPERATING THE SPOOLER

| | |
|------------------------------|-----|
| INTRODUCTION | 9-1 |
| ADMINISTRATOR AREAS | 9-2 |
| OPERATOR COMMANDS | 9-2 |
| SPOOLER MESSAGES TO OPERATOR | 9-6 |
| USER COMMANDS | 9-8 |

SECTION 10 SYSTEM CRASHES

| | |
|-----------------------------------|------|
| GENERAL | 10-1 |
| PRIMOS II | 10-1 |
| PRIMOS III | 10-2 |
| PRIMOS IV AND V | 10-3 |
| ACTIONS FOR PRIMOS III, IV, AND V | 10-4 |

PART III - THE ADMINISTRATOR

SECTION 11 BUILDING AND EXTENDING THE SYSTEM

| | |
|------------------|------|
| SHARED LIBRARIES | 11-1 |
|------------------|------|

SECTION 12 SYSTEM RESOURCE ALLOCATION

SECTION 13 DEFAULTS AND PARAMETERS

SECTION 14 USAGE ENHANCEMENTS

PART IV - REFERENCE

SECTION 15 PRIMOS COMMANDS FOR THE OPERATOR

| | |
|---|------|
| PRIMOS III, IV, and V INTERNAL COMMANDS | 15-1 |
|---|------|

SECTION 16 USING THE CONFIG COMMAND

| | |
|---|-------|
| INTRODUCTION | 16-1 |
| SINGLE-LINE CONFIG COMMAND | 16-1 |
| CONFIG EXTENSIONS (PRIMOS IV AND V) | |
| PRIMOS IV AND V INITIALIZATION ERROR MESSAGES | 16-18 |

SECTION 17 PRIMOS II OPERATION

| | |
|-------------------------------------|------|
| OVERVIEW OF PRIMOS II OPERATION | 17-1 |
| STARTUP OF PRIMOS III, IV, or V | 17-3 |
| INITIAL PRIMOS II OPERATING SESSION | 17-3 |
| PROGRAM DEVELOPMENT USING PRIMOS II | 17-4 |
| SHUTDOWN | 17-5 |
| RESTARTING PRIMOS II | 17-6 |

SECTION 18 PRIMOS III

| | |
|----------------------------------|------|
| PRIMOS III SYSTEM OVERVIEW | 18-1 |
| PRIMOS III VERSIONS | 18-1 |
| SYSTEM CONFIGURATION | 18-4 |
| PRIMOS III SYSTEM INITIALIZATION | 18-9 |

SECTION 19 PRIMOS IV-V

| | |
|---------------------------------|------|
| PRIMOS V | 19-1 |
| OVERVIEW OF PRIMOS IV | 19-1 |
| BUILDING PRIMOS IV | 19-2 |
| SYSTEM CONFIGURATION | 19-5 |
| PRIMOS IV SYSTEM INITIALIZATION | 19-7 |

APPENDICES

| | | |
|------------|---------------------------------------|-----|
| APPENDIX A | BOOTSTRAPS | A-1 |
| APPENDIX B | THE SPOOLER | B-1 |
| APPENDIX C | PRIMENET | C-1 |
| APPENDIX D | OPERATION OF THE EVENT LOGGING SYSTEM | D-1 |
| APPENDIX E | DISK ERRORS | E-1 |
| APPENDIX F | CONTROL PANEL | F-1 |
| APPENDIX G | PHYSICAL DEVICE NUMBERS | G-1 |

TABLES

| <u>Table No.</u> | <u>Title</u> | <u>Page</u> |
|------------------|--|-------------|
| 17-1 | PRIMOS II Versions | 17-2 |
| 18-1 | Disk Space Required for 64K Per User | 18-8 |
| F-1 | Control Panel Functions | F-5 |
| F-2 | Register and Memory in Prime 350-400-500 | F-8 |
| F-3 | Reserved Memory Locations (Prime 100-200-300) | F-11 |
| F-4 | Prime CPU Register Correspondence | F-14 |

ILLUSTRATIONS

| <u>Figure No.</u> | <u>Title</u> | <u>Page</u> |
|-------------------|--|-------------|
| 6-1 | Sample File Structure | 6-3 |
| 6-2 | Typical FIXRAT Traverse of File Structure | 6-4 |
| 15-1 | Setup of Line Configuration | 15-5 |
| 15-2 | Information Printed by STATUS Command At Operator Terminal | 15-21 |
| B-1 | Queue File Format | B-4 |
| C-1 | Prime Computers with IPC Connection | C-3 |
| C-2 | Remote File Access | C-5 |
| C-3 | Resource Sharing | C-6 |
| F-1 | Prime CPU Controls and Indicators | F-2 |
| F-2 | APL Sense Switch Settings | F-18 |
| F-3 | Panel Setup for Automatic Restart or Reload after Power Failure | F-19 |
| G-1 | Construction of Physical Disk/Disk Partition Number | G-5 |
| G-2 | Example of Storage Module Partitions | G-8 |

PART I - OVERVIEW

SECTION 1

OVERVIEW

INTRODUCTION

This document is a comprehensive guide for the System Administrator and Operators of a Prime computer installation. It contains everything normally necessary for operation of a Prime computer. The entire guide is aimed at the System Administrator; part of the guide is written specifically for the Computer Operator. At some installations, these two roles may be performed by the same person. The guide is designed to:

- Enable the System Operator to run the computer efficiently and optimally with a minimum of supervision by the Administrator.
- Provide the Operator with sufficient information to cope with certain emergency situations and to be aware of those which require the aid of the Administrator.
- Provide the Administrator with sufficient information to make choices about system building, option selection, resource allocation and system expansion.
- Serve as an archival repository for operator-oriented system information: PRIMOS, control panel, commands, bootstraps, etc.

This Version

This is an Initial Documentation Release, documenting the operation of Prime computer systems and supporting utilities at software revision level 15. It is more complete than the previous documentation on system operation, and replaces the following documents:

PRIMOS Computer Room User Guide, MAN2603
PRIMOS Computer Room Features (Rev 13), PTU32
PRIMOS Computer Room, PTU43
Control Panel Operation, PTU40

This document is not yet in its final form. Certain sections, less central to the major purposes of this guide, have not been included; they are represented here by detailed outlines. These sections, dealing mainly with strategy and decision making for the Administrator, will be included in the next version of this guide, which will be a Preliminary Documentation Release.

Roles of the Operator and Administrator

The roles as defined by the tasks below are suggested for guidance only. At some installations, some of the operator tasks may be assigned to the Administrator, and vice versa.

Operator: The following are the general duties of the System Operator.

- Be familiar with the operation of the system, its utilities, and features.
- Make the system available to the user community. This includes: bringing up the system, starting any special subsystems.
- Monitor the system status.
- Monitor the supervisor terminal and respond appropriately to user requests (e.g., mount tape on unit 2, etc.)
- Monitor the equipment and environment.
- Make sure that hardware preventive maintenance, filter changes and disk cleaning are performed as scheduled by the Administrator.
- Modify the system under the supervision or control of the Administrator.
- Perform normal housekeeping functions: file integrity checking (FIXRAT), backing up.
- Perform tasks for users: printing on special forms, disk formatting.
- Maintain the system logbook.

Administrator: The following are general tasks of the System Administrator.

- Build the system, possibly with the aid of a Prime field analyst.
- Select version of PRIMOS to be used, optional software systems, etc.
- Establish/change system and utility defaults, if so desired.
- Allocate the resources of the systems among users: disk space, shared segments, number of users, terminal locations.

- Arrange for and schedule preventive maintenance for hardware, peripherals, disk cleaning.
- Arrange for instruction of operators.

Organization

This guide is composed of four major parts:

- Part 1. An introductory section that defines the roles of the System Administrator and Operator, and refers to other sections containing information supporting these roles.

This section also covers typographic and Prime conventions used in this guide (Section 1).

- Part 2. The Operator. This part covers the normal functions of the operator as we have defined the role. It starts with the basic operator function of making the system available to users and continues with operator tasks supporting this function. The order of information presentation is:

- System Startup/Shutdown (Section 2)
- Accessing the System as a User (Section 3)
- Operator Tasks (Section 4)
- Monitoring System Status (Section 5)
- Checking File System Integrity (Section 6)
- Formatting Disks (Section 7)
- Backing Up (Section 8)
- Using the Spooler (Section 9)
- Handling System Crashes (Section 10)

Prime system utilities and all PRIMOS-level commands necessary to perform these tasks are presented in this Part, either in complete detail or with references to the appropriate section in Part 4 or the Appendices. In most cases, the operator will not need to refer to any software document other than this one.

Part 3. The Administrator. In general, these sections discuss options available on the system and strategies for Administrator decision-making. Some of the generic tasks covered are:

Building and Upgrading the System (Section 11)
Allocation of System Resources (Section 12)
System Default and Parameter Values (Section 13)

A final section covers some more specific topics which historically have been of interest to Prime System Administrators (Section 14).

Part 4. Reference Sections 15 and 16 form a complete reference for operator-oriented PRIMOS commands. Many of these commands may be used only at the supervisor terminal.

Section 17 is a brief view of PRIMOS II, Prime's single user operating system, which is also used to start up the multi-user operating systems. This section also includes a discussion of single-user operations. Section 18 is an overview of PRIMOS III, Prime's operating system for the Prime 300 computer. Section 19 covers PRIMOS IV and V operating systems for the Prime 350, 400, and 500 computers. These systems utilize the advanced architecture and segmentation available in the Prime 350 and higher computers.

Related Documents

The following documents contain detailed reference information useful to the Prime computer system Operator and Administrator.

REFERENCE GUIDE, PRIMOS COMMANDS, PDR3108
REFERENCE GUIDE, FILE MANAGEMENT SYSTEM (FMS), PDR3110
FILE MANAGEMENT SYSTEM (Rev 15), PTU51

Manuals and guides supplied with peripherals (line printers, tape drives, etc.), should be maintained in the computer room for use by the operators.

CONVENTIONS AND DEFINITIONS

Terminal Functions

| | |
|------------|---|
| (CR) or CR | Carriage return. |
| " | Character erase; deletes preceding character. |
| ? | Line kill; deletes all characters in current line. |
| ^xxx | symbol used in text editor to enter octal codes of non-printing characters, as in ^207 for CONTROL-G (or Bell). |

Command Line Conventions

| | |
|-----|--|
| [] | Brackets enclose optional item(s). |
| { } | Braces enclose a number of items, one of which must be selected. |

ALLCAPS A literal item that is to be entered exactly as shown. Acceptable abbreviations are underlined (see below).

lower-case A parameter whose legal value is to be selected by the user.

 Underlining. There are three major uses;

- Indicate acceptable abbreviations in a command line, e.g.,

ASSIGN device -WAIT

is equivalent to

AS device -WAIT

- Indicate user input in an interactive session, e.g.,

OK, COPY
GO

COPY, REV 15.0

FROM PHYS DISK= 50

etc.

- Emphasize command line parameters being described in text following command lines, e.g.,

SPOOL filename [-option-1] [-option-2]...

places a file in the queue to be printed by the line printer.

filename is the file to be placed in the line printer spool queue; option-1 ... (etc.).

CR or (CR) A carriage return; terminates an input line at the user terminal.

2048 A decimal number.

'4000 An octal number. The second form is that used in FORTRAN programs.
or
:4000

K 1024 (decimal); '2000 (octal).

Example: 64K words of memory is

64x1024 = 65536 words (decimal)

or

'100x'2000 = '200000 words (octal)

Basic Concepts

binary file A translation of a source file generated by a language translator (PMA, COBOL, FTN, RPG). Such files are in the format required as input to the linking loader or segmented loader.

byte 8 bits; 1 ASCII character.

device number An octal number identifying a disk type, volume size, drive unit and, if a partition, head offset. Also called: physical device number, disk number, physical disk number; called pdev or pdisk in commands.

directory A special type of file containing a list of filenames or other directories, along with information on their characteristics and location. A file whose name is on this list is said to be in this directory. A directory may be the MFD, a UFD or a sub-UFD. Directories with names in the MFD are UFDs; all other directories are sub-UFDs.

| | |
|--|---|
| directory-chain | Part of a pathname: a series of directories and optional passwords separated by the > symbol, as in: directory[password] [>subdirectory [password]]... |
| external command | A command that executes in user address space. All external commands cause GO to print when starting. External commands cannot be abbreviated. |
| file | An organized collection of information stored on a disk (or a peripheral storage medium such as tape). Each disk file has an identifying label called a filename. |
| file-unit | A number between 1 and 17 ('21) assigned as a pseudonym to an active file by PRIMOS. This number may be given in place of a filename in certain commands, such as CLOSE. PRIMOS-level internal commands require octal values. |
| filename | The name of a file or directory. Filenames may be up to 32 characters long. The first character must not be numeric (0-9). Filenames can be composed only of the following characters: A-Z, 0-9, _ # \$ & - * . and /. |
| <u>Note</u> | |
| On some devices, underline (_) may print as back arrow (+). | |
| identity | The addressing mode plus its associated repertoire of computer instructions. Programs compiled in 64R mode execute the R-identity; programs compiled in 64V mode execute the V-identity. |
| internal command | A command that executes in PRIMOS space. Does not overwrite user memory. Internal commands can be abbreviated. |
| ldev | logical device (disk). |
| ldisk | logical disk (device). |
| logical disk | The octal number assigned to a disk volume by the system when the disk is brought up by a STARTUP or ADDISK command; called ldev or ldisk in commands. |
| MFD | The Master File Directory. A special directory which contains the name of the UFDs on a particular disk or partition. There is one MFD for each logical disk. |

| | |
|---------------|---|
| mode | An addressing scheme. The mode used determines the construction of the computer instructions by the compiler. Modes available are 16S, 32S, 32R, 64R, 64V, and 32I. |
| object file | See binary file. |
| output stream | Output from the computer that would usually be printed at a terminal during command execution, but which is written to a file if COMOUTPUT command was given. |
| packname | A name assigned by the MAKE operation when a disk volume is created. Same as volume-name. |
| page | A block of 1024 16-bit words within a segment (512 words on Prime 300). |
| partition | A portion (or all) of a multihead disk pack. Each partition is treated by PRIMOS as a separate physical device. Partitions are an integral number of heads in size, offset an even number of heads from head 0 of the disk pack. A disk partition is also a volume. |
| pathname | A chain of directories starting with the disk volume name and ending with the name of the directory containing the files to be accessed. Its format is: |

$$\text{pathname} = \left. \begin{array}{l} \langle \text{volume} \rangle \\ \langle \text{ldisk} \rangle \\ \langle * \rangle \end{array} \right\} \text{directory-chain}$$

volume is the name of the disk ldisk is a logical disk number and * represents the current volume.

Note

The angle brackets are required.

| | |
|--------------|---|
| pdev | A physical device (disk). |
| pdisk | A physical disk (device). |
| phantom user | A user process running independently of a terminal under control of a PHANTOM command file. |
| PRIMOS | Prime's family of single-user and multi-user disk operating systems. |

| | |
|-------------|--|
| runfile | The executable version of a program consisting of the binary file, subroutines and library entries used by the program, COMMON areas, initial settings, etc. This file is created using LOAD or SEG. (Prime's segmentation utility.) |
| segment | A 65,536-word block of addressing space. |
| source file | A program file consisting of text, program statement, comments, etc. |
| sub-UFD | A directory which is in a UFD or another sub-UFD. |
| treename | The complete description of a directory tree, starting with a specified disk volume or partition and ending with a filename. The general format of a treename is: pathname>filename |
| UFD | A User File Directory, one of the directories listed in the MFD of a disk or partition. It may be used as a LOGIN name. |
| volume | A self-sufficient unit of disk storage, including an MFD, a disk record availability table, and associated files and directories. A volume may occupy a complete disk pack or be a partition within a multi-head disk pack. |
| volume-name | Same as packname. |
| word | 16 bits; 2 bytes; 2 ASCII characters. |

Filename conventions

| | |
|------------|---|
| filename | Source file |
| B_filename | Binary (object) file; compiler convention |
| L_filename | Listing file; compiler convention |
| M_filename | Map file |
| *filename | Saved executable memory image (R-identity) |
| #filename | Saved executable segmented runfile (V-identity) |
| C_filename | Command input file |
| O_filename | Command output file |

PART II - THE OPERATOR

SECTION 2

SYSTEM STARTUP AND SHUTDOWN

INTRODUCTION

The major function of the operator is to make the computer system available to the user community. The Prime system is designed to allow the operator to perform this function simply, easily, and reliably.

This section takes the operator through an example of startup and actual usage on a Prime computer. This example has many more operations and subsystems than most users would have. If a subsystem is not implemented on your system, you may ignore it. We have shown operator answers to system prompts that are specific to the example; space is provided to write in the appropriate response for your own system.

START UP

The general order of start up procedure is:

- Turn on power to equipment.
- Bootstrap in PRIMOS II.
- Start up command device.
- Attach to appropriate directory and run PRIMOS III, IV or V (depending upon version installed).
- If the file C_PRMO is in UFD=CMDNC0, it will specify system configuration and set any other parameters and conditions and startup any subsystems. (The file C_PRMO is built by the Administrator according to system needs.)
- If the file C_PRMO has not been installed, the operator must manually enter the configuration and subsystem startup commands.
- Set system time and date.

Turning on Equipment

Turn on power to equipment in this order:

- Turn on power to CPU by turning keyswitch on control panel to ON.
- Turn on all disk drives; wait for ready lights to go on.

- Turn on magnetic tape drives which are to be used.
- Turn on all dataphone sets; dataphones should be set to automatic answer.
- Other peripheral devices, such as line printers, may also be turned on at this time.

An attempt to boot in PRIMOS II before the disk drives are ready will fail (no message will be printed at the supervisor terminal). If this happens, wait for the drives to become ready (ready light on), and repeat the boot procedure.

Booting in PRIMOS II

After the equipment has been turned on and is ready, bootstrap the PRIMOS II operating system.

- Turn the CPU front panel rotary selector switch to STOP/STEP.
- Press MASTER CLEAR switch.
- Turn the ADDRESS/DATA switch to ADDRESS.
- If booting from a storage module disk pack, set '000114 in the panel lights (depress sense switches 10,13,14).
- If booting from a cartridge disk, set '000014 in the panel lights (depress sense switches 13 and 14).
- Turn the selector switch to LOAD.
- Press the START switch.
- Make sure all sense switches are down.

Notes

See Appendix A for details of booting PRIMOS II from devices other than disks. See Appendix F for details of control panel operation.

If the bootstrap is successful, the system will respond with:

PHYSICAL DEV= 460 (your command device= _____)

Enter the physical device number of the command device (Appendix G explains the construction of physical device numbers). The command device is the one whose command UFD is searched when a PRIMOS command is given.

The device number given here is for a 2-head partition of a storage module; the partition is the first one on the module. If yours is different, write it in the space provided.

After the device number is entered, the system responds with

PRIMOS II REV 15 02/28/78 (AT 170000)

OK:

The date is the date of the revision, not the current date. The number in parentheses is the starting location in memory of PRIMOS II.

Note

At this point, the operator can use PRIMOS II for program development. See Section 17 for complete details.

Starting up Command Device

Preparatory to bringing up PRIMOS III, IV, or V, the command device must be established as logical device 0. The command is:

OK: STARTUP 460 (your command device= _____)
OK:

The physical device number is the same as that used for the boot. (See Section 15 for STARTUP details.)

Selecting Version of PRIMOS

Attach to the appropriate directory for the version of PRIMOS which is to be used (see Sections 18 and 19 for lists of these directories). In this example, the 64-user version of PRIMOS IV with PRIMENET network support will be used. Network support is a separately priced software package.

OK: ATTACH PRINET (your PRIMOS directory= _____)
OK: ATTACH PR4.64 0 2
OK: R PRIMOS

Note

The second ATTACH command is necessary only if the network-supporting versions of PRIMOS IV or V are used. These versions are in sub-UFDs; all other versions are in UFDs. (See Section 3 for details of the ATTACH command.)

Configurations

After the command R PRIMOS, the operating system begins running; it attaches to UFD=CMDNC0 and looks for the file C_PRMO. If this file exists, commands will be read from it; if it does not exist, the system will print PLEASE ENTER CONFIG on the supervisor terminal, and commands must be entered from the supervisor terminal. In either case, the first command (either in C_PRMO or entered at the supervisor terminal) must be a CONFIG command. (See Section 16 for details on the CONFIG command.)

Example of Configuration Using C_PRMO

The following example shows the dialog ensuing when the C_PRMO command file is present. (Use of this file is strongly recommended.) Explanatory comments follow their respective actual terminal printout. The contents of the startup command file C_PRMO (and the configuration data file, CONFIG, which it invokes) appear at the end of this section.

After the command R PRIMOS, the system responds with:

```
GO
CONFIG -DATA CONFIG
PRIMOS IV <REV 15.0X-63>
```

```
512.0K MEMORY IN USE
```

```
CONFIG 60 100460 460 1000 6/6 7/10
```

Set system parameters:

```
60 Number of terminal users (decimal 48)
100460 Paging device - a storage module on drive unit
      0, partition is 2 heads, offset by 16 heads from start
      of pack
460 Command device - a storage module on drive unit 0,
      partition is 2 heads at start of pack
1000 Number of pages of physical memory to validate at cold
      start
6/6 Number of phantom users allowed
7/10 Number of remote users allowed (decimal 8)
```

Total number of configured users: $60+6+10=76$ (62 decimal)

The 6/ and 7/ prefixes are a feature of the command line which allows the omission of parameters. In this case, parameters 4 and 5 have been omitted (the first parameter, 60, is parameter 0).

```
PAGDEV 100460 34736
```

Sets size of paging space on paging device to '34736 records

SMLC ON

Enables SMLC in default configuration

MYNAME SYSE

Specifies local node-name to be SYSE

NET SYSE SMLC
 NET RSRCH1 SMLC 1 363 10401
 NET SYSB IPC 2
 NET SYSE IPC 3
 NET SYSD IPC 4

Sets network configuration parameters. SYSB, SYSE, and SYSD are connected via the IPC with the associated lines 2, 3, and 4 respectively. SYSE is connected to RSRCH1 via the SMLC. SYSE has SMLC line 0 with default parameters; RSRCH1 has SMLC line 1 - the other parameters are specified as described in Section 16.

FAM SYSB IPC
 FAM SYSE IPC
 FAM RSRCH1 SMLC
 FAM SYSD IPC

Specifies the configuration of the File Access Manager (FAM) network, which allows remote file access. All nodes which are to be made available to FAM are listed along with the type of network path. The local node must be included.

RLOGIN SYSB IPC
 RLOGIN RSRCH1 SMLC
 RLOGIN SYSE IPC
 RLOGIN SYSD IPC

Specifies the nodes on which local users may login remotely. The network paths must also be specified. The remote nodes must be configured for remote users in order for this feature to work.

LOGREC 0

Disables quota checking in Event Logger. See Section 5 for the operation of the Event Logger, and Appendix D for details of installation.

LOUTQM 74

Set inactivity logout time to '74 minutes (60 decimal=1 hour)

GO

Indicates end of configuration data file

OK, ADDISK 10460 21460 50460 61060

Connects additional physical devices to the system. The command device is logical device 0; these disks are assigned the next available logical device numbers, 1 - 4.

OK, AMLC TTY 0 2213

OK, AMLC TTY 1 2213

OK, AMLC TTY 2 2213

OK, AMLC TTY 3 2213

OK, AMLC TTY 4 2213

OK, AMLC TTY 5 2213

Configures AMLC lines 0-5. The configuration word, 2213, specifies 300 Baud, parity disabled, and 8-bit character length. See Section 15 for full details of the AMLC command.

OK, /* START 'CX'

OK, /*

OK, A CX*** 1/177777

OK, PH PH_GO

Run CX processor phantom command file

PHANTOM IS USER 57

PRIMOS returns phantom user number

OK, /* START.FAM.FOR.SYSE

OK, A FAM 1/177777

OK, PH PH_FAM

Run File Access Manager (FAM) phantom command file

PHANTOM IS USER 58

PRIMOS returns phantom user number

OK, CHAP -58 2

Use the CHAP command (Section 15) to change the priority of the FAM to 2. If the FAM user number, here 58, is known from previous startups, this command may be incorporated in C PRMO. If this is the first startup for a configuration, the user number will not be known in advance. Note the FAM user number in the logbook and, after startup is completed, change the priority with the CHAP command.

FAM is a separately priced software package.

OK, /* START SPOOLER
OK, A SPOOLQ 1/177777
OK, PH PH_PR0

Run spooler phantom command file

PHANTOM IS USER 59

PRIMOS returns phantom user number

OK, /* SHARED SUBSYSTEMS (PRIMOS IV/V ONLY)
OK, OPR 1

Set operator priority to allow sharing

OK, A SYSTEM 1/777777
OK, SHARE ED2000 2000

Shared Editor

OK, SHARE UI2000 2000

Shared UII subroutines

OK, /* SHARE SPSS

Shared version of Statistical Package for the Social Sciences (SPSS). SPSS is a separately priced software package.

OK, SHARE SP2004 2004
OK, SHARE SP2005 2005
OK, SHARE SP2006 2006
OK, SHARE SP2007 2007
OK, SHARE SP2010 2010
OK, SHARE SP2011 2011

OK, /* SHARE BASIC/VM

OK, SHARE BA2013 2013

This is the shared version of BASIC/VM. BASIC/VM is a separately priced software package.

OK, /* BRING UP SHARED LIBRARIES

The shared libraries are: FORMS, MIDAS, COBOL, and FORTRAN. If any of these are installed, then all must be installed. (See Section 4 on Shared Segments and Section 11 on Shared Libraries).

```
OK, SHARE S2014A 2014 700
OK, SHARE S2014B 2014 700
OK, R S4000
GO
THIS IS PACKAGE #1.
```

```
OK, SHARE K2014A 2014 700
OK, SHARE K2014B 2014 700
OK, R K4000
GO
THIS IS PACKAGE #2.
```

```
OK, SHARE C2014A 2014 700
OK, SHARE C2014B 2014 700
OK, R C4000
GO
THIS IS PACKAGE #3.
```

```
OK, SHARE F2014A 2014 700
OK, SHARE F2014B 2014 700
OK, R F4000
GO
THIS IS PACKAGE #4.
```

```
OK, SHARE 2014
```

Sets shared library access rights to read and execute

```
OK, /* SHARE DBMS
OK, CO OC_SHR DBMSLB 7
```

Open the command input file OC_SHR in UFD=DBMSLB for input on .
file unit 7. The unit must be specified to avoid conflict as
C_PRMO has been opened on the default unit number 6.

```
OK, A DBMSLB 1/177777
OK, OPR1
OK, SHARE DB2001 2001 700
OK, SHARE DB2002 2002 700
OK, SHARE DB2003 2003 700
OK, OPR
OK, A CMDNC0
OK, CO -CONTINUE 6
```

Return to command file C_PRMO

```
OK, CLOSE 7
OK, A CMDNC0
```

External commands will not execute from user terminals unless
the supervisor terminal is attached to UFD=CMDNC0.


```
OK, /* SET.THE.DATE.AND.TIME. *****
OK, /* TYPE.CO REMOTE SYSTEM...TO.BRING.UP.THE.NETWORK.
OK, CO -TTY
```

Accept input from the terminal

Note

This completes the processing of commands from C PRMO. Other startup functions proceeding from this point are entered by the operator directly at the supervisor terminal.

Set System Time and Date

```
OK, SETIME -052478 -0640
```

Operator sets date and time (See Section 15)

Bring Up Remote Disks

```
OK, CO REMOTE SYSTEM
```

Run command input file REMOTE in UFD=SYSTEM

```
OK, /* ADD SYSB DISKS OVER THE IPC NET
OK, ADDISK SYSB 460 3462 71063 464 10464
```

Logical devices 5, 6, 7, 10, 11

```
OK, /* ADD SYSD DISKS OVER THE NET
OK, ADDISK 460 21460 51060 71061 2062
```

Logical devices 12, 13, 14, 15, 16

```
OK, CO -TTY
```

Accept input from terminal

PRIMOS Messages

During the start-up procedure, PRIMOS prints messages at the supervisor terminal when certain requested operations have been performed. These messages do not occur immediately after the requested operation. For clarity of presentation, these messages have been gathered together below:

FAM (58) LOGGED IN AT 0'02

The FAM phantom is logged in and is running. The time is 2 minutes after system startup began.

SYSTEM (57) LOGGED IN AT 0'03

The CX phantom program has logged in and is running at 3 minutes after system startup.

**** 0'03 SYSE FAM <F15.0 > OPERATIONAL**

The local FAM program is in operation.

**** FAM NODES ENABLED: SYSB (IPC) SYSD (IPC)

Direct file access over the IPC net to remote nodes SYSB and SYSD has been established.

**** 0'03 SYSB FAM <F15.0 > OPERATIONAL **

The FAM running on SYSB is operational

*** 57 0'03

#CX# CX MONITOR, REV 15.0

SYSTEM (60) LOGGED IN AT 0'03

Both CX Monitor and slave phantoms are operational.

**** 6'41 SYSD FAM <F15.0 > OPERATIONAL**

The FAM running on SYSD is operational.

SYSTEM SHUTDOWN

Before shutting down the system, it is good operational procedure to inform users and operators at other PRIMENET nodes (if any) using MESSAGE (See Section 15). The shut down procedure is:

Send shut down messages

OK, MESSAGE ALL NOW
SYSTEM GOING DOWN IN 5 MINS. PLEASE LOGOUT
OK, MESSAGE RSRCH1
SYSE GOING DOWN IN 5 MINS.
OK, MESSAGE SYSB
SYSE GOING DOWN IN 5 MINS.
OK, MESSAGE SYSD
SYSE GOING DOWN IN 5 MINS.
 OK,

Messages sent regularly may be run from command input files, for convenience.

System Shutdown

OK, SHUTDN ALL
REALLY? YES any answer but YES aborts shutdown
 WAIT,
 LDEV 0 DETACHED, YOUR FILES CLOSED.
 PRIMOS NOT IN OPERATION

Equipment Shutdown

Equipment power may now be turned off, in the reverse order as for start up:

- Turn off peripheral devices.
- Turn off dataphone sets.
- Turn off all magnetic tape drives.
- Turn off disk drives; wait until all disks have stopped.
- Turn off the CPU by turning the keyswitch on the control panel to OFF.

CONFIGURATION DATA FILE

The configuration data file used in this example is listed below. The initial comment lines and the ASRATE command were not printed out during startup as they precede the TYP0UT YES command. The normal condition is not to type out configuration commands as they are processed. (The ASRATE command in this file sets the supervisor terminal to 300 Baud.)

```
/* CONFIG DATA FOR SYSE, 5/21/78
ASRATE 1010
TYP0UT YES
CONFIG 60 100460 460 1000 6/6 7/10
PAGDEV 100460 34736
SMLC ON
MYNAME SYSE
NET SYSE SMLC
NET RSRCH1 SMLC 1 363 10401
NET SYSB IPC 2
NET SYSE IPC 3
NET SYSD IPC 4
FAM SYSB IPC
FAM SYSE IPC
FAM RSRCH1 SMLC
FAM SYSD IPC
RLOGIN SYSB IPC
RLOGIN RSRCH1 SMLC
RLOGIN SYSE IPC
RLOGIN SYSD IPC
LOGREC 0
LOUTQM 74
GO
```

START UP COMMAND FILE

Below is the command file C_PRMO used as an example in this section.

```
CONFIG -DATA CONFIG
ADDISK 10460 21460 50460 61060
AMLC TTY 0 2213
AMLC TTY 1 2213
AMLC TTY 2 2213
AMLC TTY 3 2213
AMLC TTY 4 2213
AMLC TTY 5 2213
/* START 'CX'
/*
A CX*** 1/177777
PH PH GO
/* START.FAM.FOR.SYSE
A FAM 1/177777
PH PH FAM
CHAP -58 2
/* START SPOOLER
A SPOOLQ 1/177777
PH PH PR0
/* SHARED SUBSYSTEMS (PRIMOS IV/V ONLY)
OPR 1
A SYSTEM 1/777777
SHARE ED2000 2000
SHARE UI2000 2000
/* SHARE SPSS
SHARE SP2004 2004
SHARE SP2005 2005
SHARE SP2006 2006
SHARE SP2007 2007
SHARE SP2010 2010
SHARE SP2011 2011
*/ SHARE BASIC/VM
SHARE BA2013 2013
/* BRING UP SHARED LIBRARIES
SHARE S2014A 2014 700
SHARE S2014B 2014 700
R S4000
SHARE K2014A 2014 700
SHARE K2014B 2014 700
R K4000
SHARE C2014A 2014 700
SHARE C2014B 2014 700
R C4000
SHARE F2014A 2014 700
SHARE F2014B 2014 700
R F4000
SHARE 2014
/* SHARE DBMS
```

```
CO OC_SHR DBMSLB 7
CLOSE 7
A CMDNCØ
/* SET THE DATE AND TIME *****
/* TYPE CO.REMOTE.SYSTEM...TO.BRING.UP.THE.NETWORK.
CO -TTY
```

SECTION 3

THE OPERATOR AS USER

OVERVIEW

This section contains a summary of the major commands needed by the operator to log in and utilize the non-supervisor (user) terminals of the system. The operator should already have read the basic sections of THE NEW USER'S GUIDE TO EDITOR AND RUNOFF (FDR3104) and REFERENCE GUIDE, FILE MANAGEMENT SYSTEM (PDR3110) and have for reference REFERENCE GUIDE, PRIMOS COMMANDS (PDR3108).

INTRODUCTION

A number of operator tasks are normally performed elsewhere than at the supervisor terminal. They include:

- Creating/altering command files (ED)
- Altering source code (ED)
- Checking file system integrity (FIXRAT)
- Formatting disks (MAKE)
- Backing up the system (COPY, MAGSAV, MAGRST)
- Operating the line printer (SPOOL)
- Creating directories for applications users (CREATE)

FILE STRUCTURE

PRIMOS supports a tree structured file system. Each disk or partition has a Master File Directory (MFD) which contains User File Directories (UFDs), segment directories, and files. The UFDs can contain subUFDs, segment directories, and files; segment directories contain files. Directories and files can be nested as deeply as needed by the user.

Note

Certain PRIMOS utilities have a limit on the number of levels they will process. For example, MAGSAV and MAGRST have a limit of 13 levels of directories.

COMMANDS

Formats of the major PRIMOS commands used to access the system, together with some important comments are given below. They are:

LOGIN
ATTACH
CREATE
PASSWD
PROTEC
DELETE
TERM
DELAY
LOGOUT

Logging In

Under PRIMOS III, IV, or V a user gains access to the system by logging in at a user terminal. The command format is:

LOGIN ufd-name [password] [logical-device] [-ON node-name]

ufd-name is the name of a UFD (a directory in a MFD) on a device accessible to the system.

password is the owner or non-owner password of the UFD. If the UFD is password protected, the password must be specified.

logical-device is the device to be searched for ufd-name. If not specified, all started up devices will be searched starting with device 0.

node-name is the network name of a remote system on which ufd-name resides. The network must have been properly configured, specifically node-name must have been configured to allow remote users.

After a successful login, the user is attached to ufd-name, time-accounting registers are cleared, and a login message is printed at the terminal and the supervisor terminal:

ufd-name (user-number) LOGGED IN AT hours minutes month day year

Attaching

The attaching process traverses the tree-structure of the file system. After a successful attach, the name, location, and status of the UFD attached to is referred to as the current UFD. Optionally, this information may be stored elsewhere in PRIMOS and referred to as the home UFD. ATTACH is an internal command; the format is:

ATTACH ufd-name [password] [logical-device] [key]

ufd-name Is the UFD to be attached to.

password is the owner or non-owner password of ufd-name, if any. If ufd-name is password-protected, the password must be specified.

logical-device Is the started-up device which is to be searched for ufd-name. If omitted or 100000, all started-up devices are searched (default). If specified as 17777, the current device is searched. If specified as 0-n, the specified device is searched.

key Specifies whether ufd-name is a UFD or sub-UFD and whether ufd-name is to be set to home.

| <u>key</u> | <u>meaning</u> |
|------------|---|
| 17777 | Attach to <u>ufd-name</u> in MFD on <u>logical-device</u> ; do not set as home. |
| 0 | Attach to <u>ufd-name</u> in MFD on <u>logical-device</u> ; set as home. This is the default value. |
| 1 | Attach to <u>sub-UFD</u> in current UFD; do not set as home. |
| 2 | Attach to <u>sub-UFD</u> in current UFD; set as home. |

Creating Directories

UFDs and sub-UFDs are created by the internal command CREATE. The directory is located in the current UFD. (UFDs are created by first attaching to the MFD.) Users must have owner rights in the current UFD to invoke this command. The format is:

CREATE directory

directory is the name of the newly created UFD or sub-UFD.

The new directory has an owner password of blank (6 spaces) and a non-owner password of 0. The owner and non-owner protection keys are set to 7 and 0.

Note

Under PRIMOS II only an owner password is created.
There are no file protection keys under PRIMOS II.

Securing Directories - PASSWD

The password(s) for the current UFD are replaced by the PASSWD command.
The user must have owner rights in the current UFD to use this command.
The format is:

PASSWD owner-password [nonowner-password]

owner-password is the password giving the user owner-rights in the current UFD. Under PRIMOS II this is the only password of the UFD.

nonowner-password is the password giving the user nonowner-rights in the current UFD. If not specified it is set to blanks (not \emptyset).

Securing Files - PROTEC

Under PRIMOS III, IV, or V a user having owner-rights can open/restrict access rights to files in the current UFD with PROTEC. The format is:

PROTEC filename key-1 key-2

filename is the file for which protection is to be specified.

key-1 specifies owner's access rights to filename.

key-2 specifies nonowner's access rights to filename.

| <u>key-1/key-2</u> | <u>meaning</u> |
|--------------------|-----------------------------|
| \emptyset | No access |
| 1 | Read only |
| 2 | Write only |
| 3 | Read and write |
| 4 | Delete and truncate |
| 5 | Delete, truncate, and read |
| 6 | Delete, truncate, and write |
| 7 | All access |

Files created by subsystems such as ED, compiler listings, etc. are assigned default protection keys of 7 \emptyset (owner - all rights, non-owner - no rights).

Deletion

Files in the current UFD may be deleted by the PRIMOS DELETE command. The command format is:

DELETE filename

filename is the file to be deleted (entry in the current UFD flagged for deletion).

A directory can be deleted if all files and subdirectories within it have been deleted. An attempt to delete a non-empty directory will result in an error message.

Note

FUTIL has more powerful deletion commands: DELETE, TREDEL, UFDDEL. See REFERENCE GUIDE, FILE MANAGEMENT SYSTEM (PDR3110) for details.

Setting Terminal Characteristics

TERM: allows the following terminal characteristic to be set.

| | |
|-----------------------|--|
| TERM -ERASE character | sets the user erase character to <u>character</u> (default is "). |
| TERM -KILL character | sets the user kill character to <u>character</u> (default is ?). |
| TERM -BREAK ON | enables BREAK (CONTROL-P) process interruption (default). |
| TERM -BREAK OFF | inhibits BREAK (CONTROL-P) process interruption. |
| TERM -FULL | treats user terminal as full-duplex (default). |
| TERM -HALF | treats user terminal as half-duplex. This is <u>not</u> reset to full by login; reset must be done explicitly if desired. |
| TERM -HALF -LF | treats user terminal as half-duplex; also echo a line feed after a carriage return. |
| TERM -HALF -NOLF | treats user terminal as half duplex; do not echo a line feed after a carriage return. |
| TERM -XOFF | enables X-OFF/X-ON feature (see below). |
| TERM -NOXOFF | disables X-OFF/X-ON feature (default). |
| TERM -DISPLAY | prints terminal kill and erase characters, full- or half-duplex, X-OFF/X-ON enabled or disabled, and X-OFF received or not-received. |

The X-OFF/X-ON feature allows the halting of a process without returning to PRIMOS; the process can then be restarted by the user. This is especially useful for stopping terminal printing (by SLIST, user program, etc.) so that the text may be read.

X-OFF halts process: type CONTROL-S ('223)

X-ON restarts process: type CONTROL-Q ('221)

If this feature is disabled, the control character will have no effect on the current process.

DELAY: defines time functions to allow terminals with non-standard line speeds to function in a Prime computer configuration. This should not normally be needed if Prime-specified terminals are used. The format is:

DELAY minimum maximum right-margin

minimum is the number of character-times (time for a character to be output on a line) to delay when the carriage-return (CR) has been output at the right end of the line (default is 6).

maximum is the number of character-times to delay when CR has been output at the left end of the line (default is 12).

right-margin is the number of characters required for the cursor to move to the right margin (default is 72).

Notes

The DELAY command may be issued prior to logging in.

DELAY may be issued from the supervisor terminal if it is designated as User 1. (See USRASR, Section 15.)

Logging Out

The user gives up access to the system (under PRIMOS III, IV, or V) by the command:

LOGOUT

Files are closed, devices deassigned, the UFD is detached, assigned segments returned to the supervisor (PRIMOS IV, V), and the following message printed at both the user and supervisor terminal:

ufd-name (user-number) LOGGED OUT AT time date
TIME USED = connect-time (hm) cpu-time (ms) disk-I/O-time (ms)

Phantom users (with the same login name) may be logged out by the command:

LOGOUT -user-number

user-number is the user number of the phantom to be logged out.

SECTION 4

OPERATOR TASKS

INTRODUCTION

In addition to starting up the system and shutting it down, there are a number of tasks routinely performed by the operator. These may be initiated by the operator, by a user or administrator request, or by the system. The task groups are:

- Allocation and Monitoring of Resources
- Subsystem Usage
- System Halts - prevention and recovery
- Direct user interactions

ALLOCATION AND MONITORING OF RESOURCES

Changing the Assignable Disks Table

This is not often done by the operator. Before a disk or partition can be assigned, its physical device number must be added to the Assignable Disks Table by the DISKS command (section 15). Once this is done, the disk can be assigned by:

ASSIGN DISK physical-device-number

Under PRIMOS III, IV, or V, devices should always be assigned to the user prior to a MAKE, FIXRAT, or COPY operation (among others). Devices should be unassigned (using UNASSIGN) after completion of the operation.

Note

The Assignable Disks table has a space for a maximum of 10 devices. Devices may be removed from the Table by the DISKS command (Section 15).

Changing Priority/Timeslice

To increase efficiency and/or system performance, priorities or timeslices can be changed. Normal process priority is 1. One example of this is the File Access Manager (FAM) which is set up to run at priority 2. Priority and timeslice are changed by the CHAP command which is described in Section 15.

Incorporating Shared Segments

Normally, shared subsystems will be incorporated into PRIMOS at system startup time. At times, experimental subsystems may need to be incorporated for test purposes. The command sequence for this, from the supervisor terminal is:

```
OPRPRI 1
A ufd-name 1/177777
SHARE filename segment-number [access-rights]
A CMDNC0
OPRPRI 0
```

ufd-name UFD in which file to be shared is located

filename is the file to be restored into segment-number

segment-number is the segment to be shared. Segments '2030 to '2037 are specifically reserved for user shared subsystems.

access-rights specifies user access to the segment. Default is '600 - read and execute rights.

See Section 15 for complete details. The Administrator will assign and coordinate shared segment usage.

CAUTION

It is possible to overwrite the operating system and the shared utilities with this command. Do not share into segment 5. PRIMOS occupies segments 0 to '12. Other segments which may contain system utilities are:

| | |
|-------------|---------------------------|
| '2000 | Editor and XISUII Library |
| '2001-'2003 | DBMS |
| '2004-'2011 | SPSS |
| '2013 | BASIC/VM |
| '2014 | Shared Libraries |

Monitoring System UFDs

The directories below are under the control of the operator.

| | |
|------------|--|
| CMDNC0 | Contains all external commands available on the system. (External commands run in the user's address space). |
| LIB | Contains all libraries available on the system. |
| PRINET>FAM | Contains the file access manager (FAM) used for remote file access (if purchased). |

SYSTEM Contains the following system files:

- ED2000 - shared part of editor
- UI2000 - shared XISUII package
- SP2004 - Shared SPSS (if purchased)
- SP2005 - Shared SPSS (if purchased)
- SP2006 - Shared SPSS (if purchased)
- SP2007 - Shared SPSS (if purchased)
- SP2010 - Shared SPSS (if purchased)
- SP2011 - Shared SPSS (if purchased)
- BA2013 - Shared BASIC/VM (if purchased)
- S2014A - Shared FORTRAN library
- S2014B - Shared FORTRAN library
- K2014A - Shared MIDAS library
- K2014B - Shared MIDAS library
- C2014A - Shared COBOL library (if purchased)
- C2014B - Shared COBOL library (if purchased)
- F2014A - Shared FORMS library (if purchased)
- F2014B - Shared FORMS library (if purchased)
- DISCS - list of partitions in systems. Used by the AVAIL * command. This file is operator generated.

FORMS* Contains files needed to run the Forms Management System (FORMS).

SYSOVL Required by COBOL

CX*** Contains CX monitor and job queue.

SPOOLQ Contains spool queue and files to be spooled.

SYSKOM Contains parameter insert files.

Additions to these directories should be done only by the operator. Periodically (about once per month), these directories should be checked to see if they are in order. The contents of the directories are obtained with the LISTF command and may be written into a file using the COMOUTPUT command. The current contents of the system directory should be compared to the proper contents (this list should be maintained in the system logbook).

Example:

OK, ATTACH SYSKOM owner-password
 OK, COMOUTPUT FILES.SYSKOM
 OK, LISTF

UFD=SYSKOM 0 0

FILD.F FILD.P PARM.K KEYS.F KEYS.P A\$KEYS ERRD.F ERRD.P

```

OK, COMOUTPUT -END
OK, SPOOL FILES.SYSTEM
GO
YOUR SPOOL FILE IS PRT052 (SHORT) REV 15.00
OK,

```

Monitoring Disk Space

UFDs have no specific quotas limiting their size. When a partition does fill up, users often cannot perform tasks until some files are deleted.

FIXRAT may be used to check disk usage. When disk partitions become very full (98% and up), a list of UFDs, and the amount of space used by each, may be generated using FIXRAT (Section 6). This list should be distributed and posted. Ask the people using the most disk space to reduce the number of records they are using. Use MESSAGE to tell all users to delete unnecessary files.

SUBSYSTEM USAGE

Disk Formatting

Before a disk pack can be used on the system, it must be formatted by the MAKE utility. This process is described in detail in Section 7. Under PRIMOS III, IV, or V the new disk or partition number must first be added to the table of assignable devices with the DISKS command. The new device can then be assigned during the MAKE procedure.

Note

The disk pack or partition to be formatted must not be connected to the system via a STARTUP or ADDISK command.

FAM Operation

The FAM is a phantom that assists the operating system in performing file access across the network. FAM runs under login name FAM and is set up to run at priority 2. If the FAM has logged itself out for some reason, it must be logged in again. (FAM should not log itself out. If this occurs, note it in the system logbook and report it to the System Administrator). To log FAM in again, attach to UFD=PRINET>FAM and type PH FAM. Then set the FAM priority to 2 from the supervisor terminal with the CHAP command. The sequence of operations is:

Assuming the operator is logged in at a user terminal (not the supervisor terminal)

```
OK, ATTACH PRINET
OK, ATTACH FAM 0 2
OK, PH PH FAM
GO
PHANTOM IS USER nn
OK,
```

Here, nn is the user number of the FAM phantom.

The operator now types, from the supervisor terminal:

```
OK, CHAP -nn 2
OK,
```

Where nn is the user number of the FAM phantom, the value printed by the phantom user message.

CX Operation

CX is a utility that allows users to queue jobs to execute sequentially on a first in, first out basis.

CX consists of a phantom monitor program which monitors the progress of another phantom, the CX slave. Normally, both phantoms are running under the login name SYSTEM. Which is the monitor and which the slave can be determined by examining the supervisor terminal output printed when the system was brought up. The CX monitor sends a message to the supervisor terminal announcing its initialization.

Note

Normally, three phantoms are logged in under login name SYSTEM, two CX phantoms and the spooler. The spooler is identified by having the line printer assigned to it.

Only in rare cases will the CX monitor need operator intervention. The CX monitor will log itself out if it encounters an unrecovered disk error (Appendix E). To restart properly:

1. Log in to UFD=SYSTEM with the owner password at user terminal.
2. From the supervisor terminal, force logout the CX slave phantom, (see Forcing logouts below).
3. Attach to UFD=CX***.
4. Type the command PH PH_GO.

If another user starts up a phantom during this small time window, it may cause the CX monitor to abort with the error message NO FREE PHANTOMS. This may also prevent the restart of the CX monitor. In this case, the operator must force-logout a phantom user to free enough phantoms for CX operation.

At the user's request or for other reasons (such as a phantom program looping infinitely) the operator may cancel a currently running job in the CX queue. To do this: login to UFD=SYSTEM and force logout the CX slave (not the CX monitor). See below for forced logout procedure.

Limiting Logins

The maximum number of users allowed to be logged in to the system can be decreased from the configured value by the command: MAXUSR n. Here, n is an octal integer (less than the configured number of users) specifying the maximum allowed number of logged in users. If the number of users logged in exceeds n, no users will be logged out but no new users may log in and no phantoms may be started. This feature is useful when:

- Modifications are to be made to PRIMOS at a cold start. Before setting the system time and date, set MAXUSR to 1 or 2, preventing users from logging in until the operator resets MAXUSR.
- The system is to be shut down. Set MAXUSR to 0 to prevent any new logins just prior to shut down. (also see Forced Logouts in this section).

Forcing Logouts

From the supervisor terminal, the operator may forcibly logout any user by the command:

```
LOGOUT -user-number
```

user-number is the decimal number of the user being logged out.

The connection of a process to a terminal over the network can be forcibly broken by the command:

```
DISCON -user-number
```

user-number is the decimal number of the terminal being disconnected.

If the user is a local terminal using a remote process, the terminal is reconnected to its local process. If the user is a remote terminal using a local process, the process is logged out and returned to the pool of free remote login processes.

All users - remote, local, and phantom, can be forcibly logged out by the command: LOGOUT ALL. In addition to logging out all users, this automatically sets MAXUSR to 1, preventing any subsequent logins untils MAXUSR is reset. LOGOUT ALL can be issued just prior to a SHUTDN ALL command to allow a more orderly shutdown of PRIMOS.

Note

When logging out or disconnecting a user, the operator, as a matter of courtesy, should use MESSAGE to notify the user(s) in advance allowing them to reach a reasonable stopping place in their work.

Using the Event Logger

The PRIMOS Event Logger automatically records major system events (such as warm starts, cold starts, disk startups, etc) and writes them to a file, LOGREC, in UFD=CMDNC0. The contents of this file are examined with the LOGPRT command. Details of the event logging system usage are in Section 5; details of its structure are in Appendix D.

The LOGREC file should be printed out and then purged once per day, either just after startup or just prior to shutdown. The default LOGREC size is 4096 words. If this is too small and events are being lost, either dump the file more often or increase the size of LOGREC. (Under PRIMOS IV or V, use the CONFIG LOGREC command; under PRIMOS III change the variable QUOTA in the LOGEV2 program. See Appendix D for details.) The printouts of the Event Logging file should be stored for such a length of time as the System Administrator has determined to be useful.

SYSTEM HALTS: PREVENTION AND RECOVERY

Backing Up

Backing up is the process of copying information from system disks onto other storage media (disks or magnetic tape). Backups are performed to provide copies of files or programs for use if the versions active on the system are damaged or lost. Such damage or loss may be from user or operator error or by system crash (from power failure, etc.). The backup details are in Section 8.

Backing Up on Magnetic Tape:

1. Send MESSAGE to users that they will be logged out (Section 15)
2. Logout users with the LOGOUT ALL command (Section 4)
3. Mount Tape and ASSIGN tape drive unit
4. Backup with MAGSAV (Section 8)
5. Attach to CMDNC0 (Section 3)
6. Reset maximum number of users with MAXUSR (Section 15)

Backing Up on Disk:

1. Send MESSAGE to users that system is being shut down (Section 15)
2. Use SHUTDOWN ALL to logout any user (Section 15)
3. Boot in PRIMOS II (Section 2)
4. STARTUP any necessary devices (Section 15)
5. Perform backup with COPY (Section 8)
6. Bring up PRIMOS III, IV, or V (Section 2,18,19)
7. Startup additional system devices with STARTUP or ADDISK (Section 15)

Recovering Backed Up Files

This is normally performed after bringing up the system but before allowing users to log in. The recovery procedures from disk (FUTIL) and magnetic tape (MAGRST) are described in Section 8.

Checking File System Integrity

File system and directory integrity are checked and repaired, if necessary, by the FIXRAT utility. A complete description of FIXRAT usage is in Section 6. In addition to checking the integrity, FIXRAT performs general file housekeeping by compressing directories. FIXRAT should be run on a regular basis by the operator. It has been found to be convenient to perform the FIXRAT operations in conjunction with system backups. (see Section 8 for a sample schedule).

Handling System Crashes

The procedures for handling a system crash vary with the reason for the crash. This reason can usually be determined by obtaining the halt location from the control panel. The operator should become familiar with the control panel operations described in Appendix F. The handling of crashes, halt location meanings, restart procedures, and information to be entered in the logbook is detailed in Section 10.

Keeping a Logbook

The logbook is used to record significant system conditions or changes. These include hardware configuration, disk partitions, system crashes, backups, etc. A major purpose of the logbook is to allow reconstruction of the system history. Complete logbook procedure is detailed in Section 5. In logging events:

- All entries should have time and date (including year).
- All entries should be signed. The administrator or system analyst will then know whom to ask for clarification or additional information.

DIRECT USER INTERACTIONS

Line Printer Operations

Details of the operator's Spooler commands are given in Section 9. In addition to generally overseeing the line printer(s), the operator is responsible for:

- Removing listings from the printer, separating them by user (banner name before each file), and placing them in a user-pickup area.
- Reloading paper and ribbons in the line printer(s) as required.
- Changing paper to print special forms requests. (Use SPOOL -LIST to see if any such requests are outstanding). It is good procedure to schedule special forms printing for a specific time of day, for example, directly after performing backups at the start of a shift.

If there are problems with the line printer spooling output. The order of corrective operations is:

1. Check that the printer is powered up and ready to print.
2. Check that the printer is not jammed or out of paper.
3. Check that the spool phantom is running by seeing if user SYSTEM has the line printer assigned (use STATUS USER command). If the Spooler is not running, login to UFD=SYSTEM (use the owner password) and then attach to the UFD SPOOLQ. Restart the spooler phantom by typing: PH PH_PR0. (This is the procedure for the first line printer, PR0. For the second line printer use the command PH PH_PRL, for the serial printer use PH PH_CEN, for the printer/plotter use PH PH_PLT).
4. Make sure the spooler is not set up to print on special forms paper.
5. Make sure that user FAM has had its priority set to 2. Check this with STATUS USERS. If FAM is not running at priority 2, set it using the CHAP command (Section 15).
6. See the Systems Administrator if the printer still does not function.

Sending Messages

From the supervisor terminal the operator can send messages to:

- All users on the local node of the network
- A specified user on the local node of the network
- The supervisor terminal of a different network node

This facility is useful for giving users general information (system to be shut down, new utility available), communicating with a single user (answering questions, requesting action), or for passing information between nodes (remote disk availability, etc). The command format is given in the MESSAGE command (Section 15).

Adding New UFDs to the System

New UFDs in an MFD can only be added by the operator as passwords to MFDs are not usually distributed to the users. When a request is made to add a new UFD to the system, the operator should first determine from the user: its name, and the partition on which it is to reside. The new UFD name should not duplicate any existing UFD names.

The operator should log in as a user, attach to the MFD on the appropriate partition, and generate the new UFD with the CREATE command (Section 3).

Since the MFD is password-protected, the operator must also perform all requests for UFD name changes (use the PRIMOS CNAME command).

Adding/Replacing Software in CMDNC0 and LIB

These UFDs are password-protected directories under operator control. New software is copied into these directories with the COPY command of FUTIL. All new or changed software should be debugged before installation, insofar as is feasible.

All changes to these directories should be noted in the system logbook. No new or changed software should be installed without complete details of operation being obtained from the software implementer. For commands, this should include command line options and keywords and answers to any queries asked by the program. Libraries should have indicated their proper position in loading sequences. This information should be entered in the system logbook and distributed to interested users.

SECTION 5

MONITORING THE SYSTEM

OVERVIEW

This section contains the following:

- A discussion of commands which give the operator information about system status.
- Messages which are printed at the supervisor terminal due to actions of users or of the system. These actions are not initiated by the operator.
- Use of the Event Logging mechanism in PRIMOS.
- Helpful hints on how to keep a system logbook: contents, format, and uses.

LOOKING AT THE SYSTEM

Introduction

The major tool available to monitor the condition of the system is the STATUS command. It prints information on the status of users, devices, and the network (if any) to which the local system is attached. A complete description of STATUS, its options, and details of the information it prints at the supervisor terminal is in Section 15. Certain other commands are available to monitor specific aspects of the system (spooler, CX, etc.). They are described fully in REFERENCE GUIDE, PRIMOS COMMANDS, PDR3108. Brief descriptions as they pertain to system monitoring are given below.

Disk Space Utilization

The AVAIL command prints, for a specified disk or partition, the number of records used, number of records still available and percent of disk or partition full. The format is:

$$\text{AVAIL} \left\{ \begin{array}{l} \text{packname} \\ \text{logical-device-number} \\ * \end{array} \right\}$$

packname is the name of the disk or partition (the volume name).

logical-device-number is the logical-device-number spelled out in English (e.g., FOUR, SIX, etc.).

If AVAIL is given without arguments, information will be printed for the device currently attached to.

Examples:

OK, AVAIL DOCUMN

GO

VOLUME DOCUMN
 34476 TOTAL RECORDS (NORMALIZED)
 4891 RECORDS AVAILABLE (NORMALIZED)
 85.8% FULL

OK, AVAIL ONE

GO

VOLUME SPOOLA
 34476 TOTAL RECORDS (NORMALIZED)
 16151 RECORDS AVAILABLE (NORMALIZED)
 53.2% FULL

OK, AVAIL

GO

VOLUME DOCUMN
 34476 TOTAL RECORDS (NORMALIZED)
 4891 RECORDS AVAILABLE (NORMALIZED)
 85.8% FULL

If the command AVAIL * is given, a table of record utilization on all started up devices will be printed.

Example:

OK, AVAIL *

GO

| VOLUME ID | TOTAL RECS | FREE RECS | % FULL | COMMENTS | | |
|--------------|---------------|--------------|-----------|----------|-----|--------------------|
| SOFIWR | 241333 | 29453 | 87.8 | 3462 | NEW | SYSB |
| SPOOLA | 34476 | 16151 | 53.2 | 10460 | NEW | NOT BACKED UP SYSB |
| PERIPH | 34476 | 11792 | 65.8 | 460 | NEW | SYSB |
| CPUGRP | 103428 | 3693 | 96.4 | 21460 | NEW | SYSB |
| DOCUMN | 34476 | 4891 | 85.8 | 50460 | NEW | SYSB |
| DBTEST | 86190 | 80537 | 6.6 | 71063 | NEW | NOT BACKED UP SYSB |
| M142A1 | 34476 | 17335 | 49.7 | 464 | NEW | READ ONLY SYSB |
| M142B1 | 51714 | 23524 | 54.5 | 10465 | NEW | READ ONLY SYSB |
| TRANS | 103428 | 5275 | 94.9 | 21460 | NEW | SYSD |
| DBGPR | 68952 | 7652 | 88.9 | 51060 | NEW | SYSD |
| DTEST | 137904 | 87463 | 36.6 | 2062 | NEW | NOT BACKED UP SYSD |
| SPOOLD | 34476 | 12025 | 65.1 | 460 | NEW | NOT BACKED UP SYSD |
| TEST | 86190 | 54118 | 37.2 | 71061 | NEW | NOT BACKED UP SYSD |

Note

AVAIL * will not work unless the file DISCS has been built in UFD=SYSTEM. This file contains a list of Volume IDs for all devices which might be connected to the system and comments pertaining to these devices. Record utilization data are determined from the system, all other information is from SYSTEM>DISCS. See Section 11 for details on building DISCS.

AVAIL requires non-owner MFD passwords to be XXXXXX and the packname protection set to 7 1.

Spool Queue

The contents of the spool queue may be printed by the command:

SPOOL -LIST option

| <u>option</u> | <u>action</u> |
|---------------|--|
| omitted | Lists all files. |
| ALL | Lists all files. |
| DEFER | Lists deferred files. |
| FORM type | Lists files queue with FORM specified by <u>type</u> . (For default type specify ' '). |
| OWN | Lists files spooled under user's login name. |
| PLOT | Lists files in plot queue. |
| PRINT | Lists files in print queue. |

Each listing contains the following information:

user name, spool file name, date/time of spool request, options, file size, file name, form (if any), defer time (if any). See section 9 for more details on the spooler.

Sequential Job Processor (CX) Queue

The contents of the local CX job file are printed at the terminal by:

CX -ALL

Example:

OK, CX -ALL
GO

CX JOB FILE LISTING 78/04/13 2:28 PM

| FILE | ID | OWNER | STATE | DATE/TIME |
|--------|--------|--------|-----------|------------------|
| CX##06 | MASTER | DOUGLA | EXECUTING | 78/04/13 2:03 PM |
| CX##05 | MASTER | DOUGLA | ABORTED | 78/04/13 1:56 PM |
| CX##04 | THIS | DOUGLA | COMPLETED | 78/04/13 1:51 PM |
| CX##02 | THIS | DOUGLA | ABORTED | 78/04/13 1:45 PM |
| CX##01 | C_ALL | STUMPF | COMPLETED | 78/04/13 0:33 AM |

[CX, REV 15.0]

- The command CX -PRINT prints only CX jobs requested by the user (OWNER column).
- The command CX -QUEUE prints only those jobs executing or awaiting execution. Completed and aborted jobs are not printed.
- Information about a specific job is obtained by

CX -Sjob#

Example:

OK, CX -S01
GO

OWNER: STUMPF JOB ID: C_ALL STATE: COMPLETED 78/04/13 0:33 AM

[CX, REV 15.0]

Networks: The commands

CX -QUEUE -ON logical-device

and

CX -Sjob# -ON logical-device

give the same information as above but for the CX queue on another logical device.

logical-device is the local logical device number of a remote disk containing the CX queue.

The local logical device number is obtained from the STATUS DISKS commands. Only CX queues on those remote disks brought up locally by a STARTUP nodename or ADDISK nodename command can be accessed.

Number of Users

The total number of system users (terminal, phantom, remote) can be obtained with the internal command USERS. The dialog is:

OK, USERS

USERS= 23

System Parameters

Below is a list of system parameters and values whose status the operator may wish to know. The command which will print that information (and usually more) is given. The operator should check the individual command for specific details.

| <u>Status Item</u> | <u>PRIMOS Command</u> |
|-------------------------|----------------------------|
| Active CX jobs | CX -QUEUE |
| Assigned devices, user | STATUS USERS |
| Available records | AVAIL |
| Command device | STATUS or STATUS DISKS |
| Complete list, CX jobs | CX -ALL |
| CX jobs, active | CX -QUEUE |
| CX jobs, complete list | CX -ALL |
| CX jobs, specific | CX -Sjob# |
| CX jobs, users | CX -PRINT |
| Deferred spool files | SPOOL -LIST DEFER |
| Device, command | STATUS or STATUS DISKS |
| Device, paging | STATUS |
| Devices mounted | STATUS DISKS |
| Devices, assigned, user | STATUS USERS |
| Devices, logical | STATUS DISKS |
| Devices, physical | STATUS DISKS |
| Devices, physical, user | STATUS USERS |
| Devices, remote | STATUS DISKS |
| Disks mounted | STATUS DISKS |
| Disks, remote | STATUS DISKS |
| File units in use | STATUS UNITS |
| Free records | AVAIL |
| Line, user | STATUS USERS |
| Local nodename | STATUS NET or STATUS UNITS |
| Logical devices | STATUS DISKS |
| Logins, remote | STATUS USERS |
| Mounted devices | STATUS DISKS |
| Mounted disks | STATUS DISKS |
| Network | STATUS NET |
| Network, type | STATUS NET |
| Node condition | STATUS NET |
| Nodename, local | STATUS NET or STATUS UNITS |
| Number of users | STATUS USERS or USERS |
| Number, user | STATUS USERS |
| Packnames | STATUS DISKS |

| | |
|----------------------------|------------------------------|
| Paging device | STATUS |
| phantom users | STATUS USERS |
| Physical devices | STATUS DISKS |
| Physical devices, user | STATUS USERS |
| Plot files, spool | SPOOL -LIST PLOT |
| Print files, spool | SPOOL -LIST PRINT |
| Priority, user | STATUS USERS |
| Records available | AVAIL |
| Records used | AVAIL |
| Remote devices | STATUS DISKS |
| Remote disks | STATUS DISKS |
| Remote logins | STATUS USERS |
| Remote systems, logins to | STATUS USERS |
| Remote users | STATUS USERS |
| Special form spool files | SPOOL -LIST FORM type |
| Specific CX jobs | CX -Sjob# |
| Spool files | SPOOL -LIST [ALL] |
| Spool files, deferred | SPOOL -LIST DEFER |
| Spool files, special form | SPOOL -LIST FORM type |
| Spool files, users own | SPOOL -LIST OWN |
| Spool plot files | SPOOL -LIST PLOT |
| Spool | print filesSPOOL -LIST PRINT |
| Type of network | STATUS NET |
| Units, file, in use | STATUS UNITS |
| User assigned devices | STATUS USERS |
| User line | STATUS USERS |
| User logins to other nodes | STATUS USERS |
| User number | STATUS USERS |
| User physical devices | STATUS USERS |
| User priority | STATUS USERS |
| Users CX jobs | CX -PRINT |
| Users own spool files | SPOOL -LIST OWN |
| Users, number of | STATUS USERS or USERS |
| Users, phantom | STATUS USERS |
| Users, remote | STATUS USERS |
| Volume names | STATUS DISKS |

SUPERVISOR TERMINAL MESSAGES

In addition to messages printed at the supervisor terminal in response to operator-initiated actions (e.g., bringing up the spooler, etc.), other messages are printed to inform the operator of changes in system status.

Terminal (Local and Remote) Users

- ufd-name (user-number) LOGGED IN AT hh:mm mmddyy

A terminal user has logged in

TEKMAN (35) LOGGED IN AT 12'17 040378

- ufd-name (user-number) LOGGED OUT AT hh'mm mmdyy
TIME USED = hh'mm mm'ss mm'ss
TIMED OUT (prints if PRIMOS logs out inactive user)

A terminal user has logged out. Time used is printed.

Times used are: connect time, CPU time, disk I/O time

```
TEKMAN (35) LOGGED OUT AT 13'43 040378
TIME USED = 1'26 67'20 0'42
```

Phantom Users

- USER user-number: PHANTOM TTY REQUEST

A phantom has requested terminal input and has been logged out

```
USER 35: PHANTOM TTY REQUEST
```

- PHANTOM phantom-user-number: error-text

A phantom has encountered an error and has been logged out.

```
PHANTOM 52: BAD PARAMETER
```

CX Processor

- ***cx-phan-user-no hh'mm
#CX# EXECUTING CX##job FOR USER ufd-name

The CX processor, running as user number cx-phan-user-no has started execution at hh'mm. The job, requested by ufd-name, is number job in the CX queue.

```
***49 13'05
#CX# EXECUTING CX##05 FOR USER TEKMAN
```

Disk Errors

Under PRIMOS III, IV, and V disk error messages are of the form:

DISK xx ER - other information.

xx is RD for a read error and WT for a write error

Under PRIMOS II, disk error messages are of the form:

DISK xx ERROR - other information.

xx is RD for a read error and WT for a write error.

See Appendix E for a full explanation of the other information.

Networks

- **** hh'mm node-name FAM<F15.0 > NOT OPERATIONAL**

The FAM on system node-name on the network has shut down.

**** 15'17 SYSD FAM<F15.0 > NOT OPERATIONAL**

- **** hh'mm node-name FAM<F15.0 > INITIALIZED**

**** hh'mm node-name FAM<F15.0 > OPERATIONAL**

The FAM on system node-name on the network has been brought up and is functional.

**** 15'30 SYSD FAM<F15.0 > INITIALIZED**

**** 15'31 SYSD FAM<F15.0 > OPERATIONAL**

- **** hh'mm node-name DISK device-number SHUTDN**

A remote device at system node-name on the network has been shutdown at the system where it is physically mounted. This message prints only if device-number has been declared a remote device on the local node by a STARTUP node-name or ADDISK node-names command.

**** 15'46 SYSD DISK 2062 SHUTDN**

EVENT LOGGING IN PRIMOS

Overview

An event logging mechanism is provided as part of the operating system to record information about significant events (cold/warm starts, machine checks, disk errors, etc.) in an internal buffer, periodically dump this buffer to a disk file, and format and print the disk recording file.

The following paragraphs describe how the logging mechanism works, necessary operator actions, and steps for the programmer or administrator to take to modify the mechanism.

Interrogating the Event Logging File

LOGPRT is a program that writes the contents of the event logging file (LOGREC) to a disk file or a user terminal. The command line to invoke LOGPRT under PRIMOS III, IV, or V is as follows.

```
A SYSTEM owner-password
R *LOGPRT [destination] [opt1 opt2 optn ...]
```

destination Is the destination for LOGPRT's output. If TTY is specified, the output will be to the user's terminal. If destination is omitted, output will be to the file LOGLST in the current UFD. Any other name will be considered a file name to which the output is directed.

opt1
opt2 The values and meaning of these options are as follows:

| <u>Option</u> | <u>Meaning</u> |
|------------------|--|
| - <u>H</u> ELP | Prints a list of LOGPRT options. The LOGPRT command must be reissued after the list of options is printed. |
| - <u>S</u> POOL | (PRIMOS III, IV, and V only) automatically spools the resulting output file. LOGPRT prints the name of spool file and indicates whether the spool file is long or short. This option is ignored if TTY is specified for <u>destination</u> . |
| - <u>D</u> ELETE | deletes the output file (after spooling). This option is ignored if TTY is specified for <u>destination</u> . |
| - <u>P</u> URGE | Empties LOGREC after LOGPRT has finished processing. The default is to leave LOGREC unmodified. |

- INPUT treename Specify the treename of the LOGREC file to be processed as treename. If this option is omitted, LOGPRT asks the user to specify a treename (INPUT TREENAME:).
- FROM mmddy Only LOGREC entries from the specified date mmddy to the latest entry are printed.
- TYPE t1 t2...tn Process entries only of the indicated type. The types, t1, t2...tn can be any of the following:

| <u>No.</u> | <u>Type</u> | <u>Meaning</u> |
|------------|-----------------|---|
| 0 | COLD | Cold starts |
| 1 | WARM | Warm starts |
| 2 | TIMDAT | Time/date entries (see Note) |
| 3 | CHECKS | Machine checks (including memory parity). |
| 4 | DISK | Disk errors |
| 5 | OVERFL | Record LOGREC overflow entries |
| 6 | SHUTDN | Operator shutdowns |
| 7 | CHK300 | Prime 300 machine checks |
| 8 | PAR300 | Prime 300 memory parity checks |
| 9 | MOD300 | Prime 300 missing memory module checks |
| 10-15 | TYPE10...TYPE15 | Entries for types 10 to 15 |
| 16 | DSKNAM | Either ADDISK or STARTUP entries |

Note

The time/date stamps associated with the selected entries will not be processed unless TIMDAT is explicitly selected. For example: `-TYPE DISK TIMDAT` will process all disk errors and their associated time/date stamps. If TIMDAT alone is specified in conjunction with one or more other types, only the time/dates of the selected types will be processed. If the `-TYPE` option is not specified, all entries will be processed.

Under PRIMOS II, the operation of LOGPRT is identical, with the exception of the `-SPOOL` option which is not supported by PRIMOS II.

LOGPRT Messages

If LOGPRT finds that the output file already exists, it prints:

OK TO DELETE OLD destination (Y or N):

The reply should be Y to delete the file or N to enter a new destination. If N is entered, the user is asked:

NEW SPECIFICATION:

All parameters following the LOGPRT command line may be re-entered.

Under PRIMOS II, LOGPRT then prints the prompt:

REPLY PU TO PURGE WHEN DONE:

Any reply but PU causes LOGPRT to leave LOGREC unmodified.

Finally, LOGPRT prints the message:

INPUT TREENAME:

The treename of the LOGREC file to be printed should be entered. If a blank line is entered, <0>CMDNC0>LOGREC is assumed.

EVENT MESSAGE FORMAT

Overview

LOGPRT writes a header line followed by formatted entries, one or more lines per entry. (All numbers are octal except where noted.)

The specific format for currently defined event types, with a brief explanation of the format, is given below. The messages are listed by approximate frequency of occurrence.

Information Printed by LOGPRT

The first line of an entry is the time and date (underlined) in the format:

hh:mm:ss day dd xxx yyyy

This is the date/time record entered when the event logging buffer (LOGBUF) was written to the event logging file on the disk (LOGREC). All events following this entry and before the next date time shown happened during the minute immediately before the time shown. An example of a date time record is:

09:01:20 WED 16 FEB 1977

The following entries define specific type of events which may be listed after a date/time entry.

COLD START

indicates a cold start of PRIMOS was performed.

WARM START

indicates a warm start of PRIMOS was performed.

MACHINE CHECK (xxx) DSWSTAT= ssssss ssssss DSWRMA= yyyyyy rrrrrr
rrrrrr DSWPB= pppppp pppppp

A machine check occurred. DSWSTAT, DSWRMA, and DSWPB constitute the DSW (descriptor segment word) at the time of the check. If the RMA INVALID bit is set (bit 9 of DSWSTATL), yyyyy is (INV), otherwise yyyyy is absent. xxx is an encoding of the machine check code in DSWSTAT as follows:

xxx Meaning

BMA Memory address

BMD Memory data output

BPAI Peripheral address input

BPAO Peripheral address output

BPD Peripheral data output

RCD Cache data

RCM RCM parity error (XCS only)

RDXI RDX-BPD input

RF Register file

MISSING MEMORY DSWSTAT= ...

A missing memory module check occurred. Information is the same as for a machine check, except that the machine check code xxx does not appear.

MEMORY PARITY (xxxx) DSWSTAT= ... PPN,WN= pppppp wwwwww

A memory parity error occurred. xxxx is either ECCC (corrected) or ECCU (uncorrected). The DSW is followed by PPN,WN=pppppp wwwwww, which identifies the physical page and word number of the error. For an ECCC error the PPN is followed by Bit = xx. (xx = bit in error.)

| <u>xx</u> | <u>Meaning</u> |
|-----------|-----------------|
| 1-15 | bit 1-15 |
| RP | Right parity |
| LP | Left parity |
| C2,C4,C5 | Other check bit |
| MB | Multiple bit |
| NE | No error |

This is followed by DP=x, x=0 or 1, setting of DSWSTATL bit 6 (overall parity).

DISK xx ERROR DVNO= dddddd (typecode) CRA= rrrrrr rrrrrr CYL= ccc
 HEAD= hh RECORD= rr RCRA= aaaaaa aaaaaa STATUS (ACCUM)=
 ssssss STATUS (LAST) = llllll RETRIES= tt mmmmmm

A disk error occurred during an xx operation, where xx is RD for read or WT for write. DVNO is followed by dddddd, the device number. typecode gives the controller number and device type (MHD for moving head disk, SM for storage module). CRA rrrrrr rrrrrr gives the record address, which is broken up into CYL (cylinder), HEAD, and RECORD address (all decimal). For a read operation, RCRA gives the CRA read on a CRA error. STATUS (ACCUM) is the OR of all status bits obtained during retries. STATUS (LAST) is the status of the last operation.

RETRIES gives the number of retries attempted. If RETRIES is less than 10, the operation was completed successfully, the value of string mmmmm will be (RECOVERED).

If RETRIES = 10, mmmmm will be (UNCORRECTABLE) if the error could not be corrected. If an ECC error has been successfully corrected by the software, mmmmm is WORDNO= and CORRECTION=, which gives the word number in the record and the 32-bit correction pattern used.

DISK MOUNT: packname ON dvno

An ADDISK or STARTUP command was issued. The indicated packname was mounted on the disk identified as dvno (which is a physical disk or partition number).

MACHINE CHECK USER= nn PC= pppppp

The format of a machine check message on a Prime 300. USER gives the user number, nn (decimal), PC gives the user's PC at the time of the check.

MEMORY PARITY

A Prime 300 memory parity error occurred (see next entry).

MEMORY PARITY PPN= pppppp WN= wwwwww CONTENTS= cccccc

The format of an entry for a Prime 300 memory parity error encountered during a WARM start memory scan. The physical page number (PPN), word number offset in the page (WN), and the incorrect contents are given.

MISSING MEMORY

A Prime 300 missing memory check entry.

LOGBUF OVERFLOW -- nnnnn ENTRIES LOST

Indicates nnnnn (decimal) event entries were lost due to overflow of the event logging buffet (LOGBUF).

SHUTDOWN BY OPERATOR

The operator issued a SHUTDN ALL command. (This automatically dumps LOGBUF.)

Messages

*** LOGREC EMPTY ***

This message is printed is LOGPRT finds no entries in LOGREC.

*** END OF LOGREC -- nnnnn ENTRIES, ppppp PROCESSED ***

This message is printed when LOGPRT reaches the end of LOGREC. nnnnn (decimal) gives the number of entries processed not including date/time and LOGBUF overflow entries, ppppp is the number of entries processed.

When all the entries in LOGREC (or other input file) have been processed, LOGPRT normally closes the file and exits. If the PURGE option has been specified (or 'PU' was the reply to the PURGE prompt under PRIMOS II), LOGPRT positions to the beginning of the input file before closing, in effect emptying the file.

Finally, if the -SPOOL option is in effect, LOGPRT sends the output file to the spool program and prints the name of the resulting spool file. If the DELETE option is in effect, the output file is then deleted. (These options are in effect under PRIMOS III, IV and V.

THE SYSTEM LOGBOOK

As a matter of good operating procedure, the System Administrator should establish a logbook for recording system status and operation. The Administrator should see, that as part of their training, operators are instructed in the details of maintaining a logbook.

Purpose

A logbook should contain sufficient information about system operation and history to allow unusual or undesirable occurrences to be analyzed. Logbook records should be adequately detailed so that the Administrator or operator can restore all or part of the system to normal status. Unusual operational events include: system crashes, operator errors, or external happenings such as power failure.

Format

How logbooks should be kept and what information should be recorded is at the discretion of the Administrator. However, we would like to pass on some of the experience accumulated by Prime's system operators. Certain standards will facilitate logbook usage; the following procedures are recommended:

- Logbooks should be bound. Looseleaf pages are easily detached and lost. This is especially true of pages frequently referred to.
- The logbook should stay open and flat for ease of writing.
- Exact page size is not important. However, select a convenient size that will allow printouts and listings to be pasted in.
- All logbook entries should be labelled with time and date. This gives a historical record, useful in reconstructing how a crash or other unusual event occurred. It also enables these events to be correlated with external events (power failure, etc.).
- Logbook entries should be signed (or initialed) by the person making the entry. The Administrator (or an analyst) then knows whom to ask for more information about a specific event.
- All logbook entries should be made in ink, not pencil. Incorrect entries should be crossed out neatly.

Contents

The following types of information and events should be recorded in the system logbook. Certain of these items apply only to installation using PRIMOS IV and V.

Hardware:

- Equipment configuration - disks, peripherals, etc. (including model number and serial number)
- Changes to equipment configuration: additions, removals, replacements, alterations
- Changes in operating status of equipment (component failures, etc.)

Software:

- Listing of the startup command file C_PRMO (listing of other command files, if alternative configurations are used)
- Copy of the CONFIG command line or CONFIG data file (PRIMOS IV or V)
- Listing of the loadmap, M_PRMO, for each version of PRIMOS used on the system
- Allocation of shared segments, both Prime-supplied and user-installed.
- Contents of the command directory, CMDNC0, and the library directory, LIB.

Any changes - additions, deletions, replacements - in the above, should be entered in the logbook.

Operations:

- System startup. Include any special conditions: subsystems (SPOOL, CX, FAM, etc.) not brought up
- Any FIXRATs - disk partition, result
- Disk formatting - new partition names created
- Backups performed - partitions copied: to tape or to disk
- Files restored to the system
- Time, date, and place of storage, of Event Logger Dumps
- New UFDs created or names changed
- Replacement or other changes of commands in CMDNC0, or libraries in LIB
- Shutdowns - total or partial

In all the above, any anomalous conditions or responses should be noted.

Crashes: See Section 10 for complete details on information to be recorded.

- System status - register values
- WARM or COLD start - did system continue to function properly immediately after start? after a half-hour?, etc.

Environment:

- Abnormal temperature/humidity conditions
- Other adverse conditions - dust, chemicals, etc.
- Unauthorized access to computer area
- Any theft or damage to computer or components
- Unauthorized computer usage
- Any other unusual events

SECTION 6

FIXRAT

INTRODUCTION

The external command FIXRAT loads and starts a maintenance program that checks the PRIMOS file integrity on any disk pack. FIXRAT fully supports nested UFDs and nested segment directories. FIXRAT may be run from a command file. This section assumes the reader is familiar with the file structure described in the REFERENCE GUIDE, FILE MANAGEMENT SYSTEM, PDR3110. This guide should also be consulted for a description of segment directories and nested directories.

FIXRAT reads every record in every file, UFD, and segment directory, and checks that the information in each record header is consistent with record headers in the rest of the file and with the file directory that contains the record.

Any inconsistencies generate an error message. FIXRAT also builds a record availability table (RAT) from the existing file structure and compares it to the DSKRAT file for agreement. If discrepancies are found, FIXRAT prints an error message.

Note

The packname is the name of the file containing the disk record availability table. This file will be called the DSKRAT file (or just DSKRAT) in this discussion. Users can assign other names to DSKRAT when the disk is formatted by MAKE.

If requested, FIXRAT will not only check the file structure but also repair pointers (if possible), truncate or delete defective files, and generate a corrected DSKRAT file. Up to two repetitions of FIXRAT may be necessary to repair a damaged file structure. The recommended procedure is to repeat FIXRAT until an error-free printout is obtained.

If a program being debugged runs wild and writes over part of PRIMOS II (when running under PRIMOS II) FIXRAT should be run to repair the file system.

One suggested procedure for maintaining a disk pack is to run FIXRAT every morning and, if no errors occur, to then copy the pack onto a daily backup pack. If any files are truncated or deleted from the pack, they are copied from the existing daily backup disk, to the disk pack. The owners of the bad files must be notified that those files have been copied from the backup and any later modifications made to those files may have been lost.

FILE STRUCTURE

Figure 6-1 shows a sample PRIMOS file structure.

The file structure on any disk pack is a tree structure where the MFD is the root or trunk of the tree, the links between directories and files or subdirectories are branches, and the directories and files are nodes.

A directory tree consists of all files and sub-directories that have their root in that directory. In Figure 6-1, the directory tree for UFD1 is circled. The level of a file is the depth of that file in the tree. For example, as shown in Figure 6-1, the MFD is at level 1 in the tree, UFD1 is at level 2 in the tree, and FILEC is at level 4.

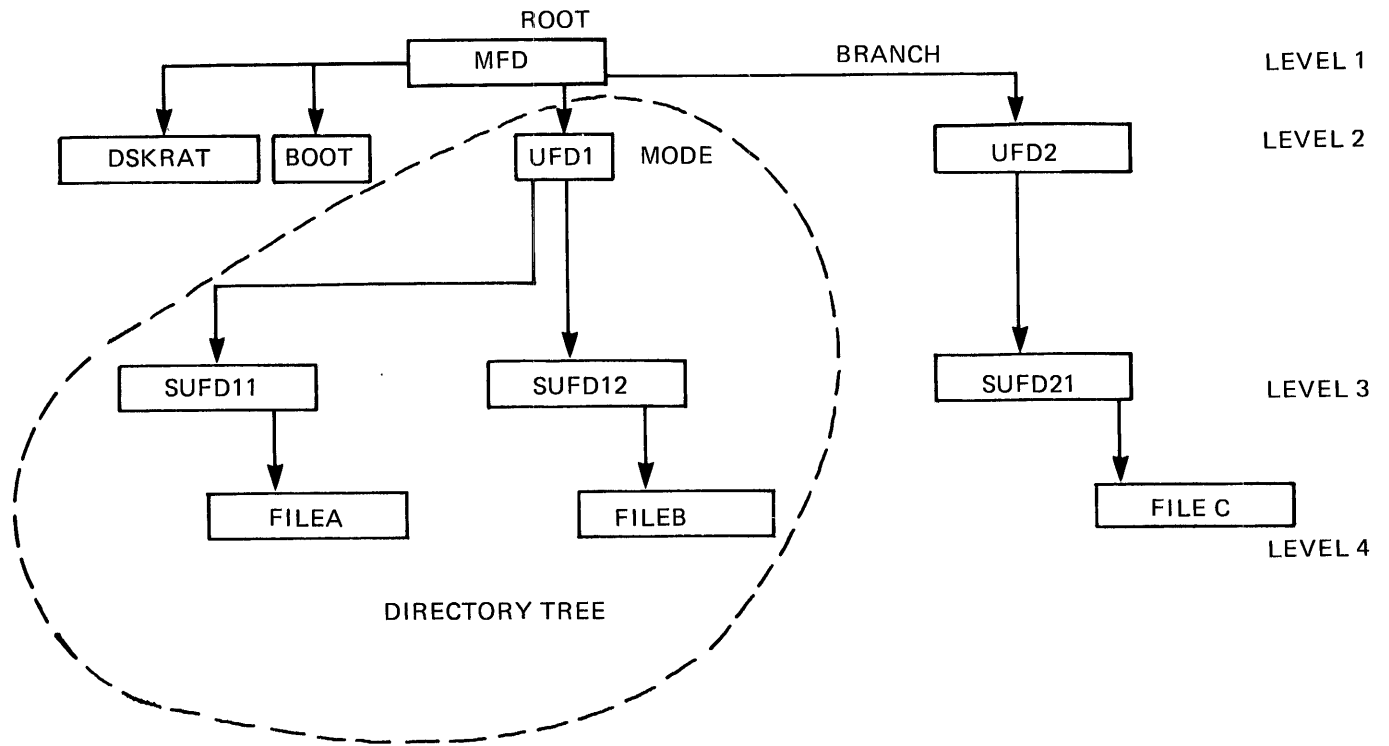
FIXRAT traverses the file structure, as shown in Figure 6-2, generating terminal output.

FIXRAT prints BEGIN directory-name when beginning processing of a directory tree. On leaving a directory tree, FIXRAT prints END directory-name followed by the number of physical records (in decimal) used by all files and directories in the directory tree. FIXRAT indents the printed output one space for each level down in the tree in which the directory is located. This indented format makes it easy to understand the relationship of each directory to the other directories in the tree. To prevent excessive output, FIXRAT as a default prints out only directory names at levels 1 and 2 in the tree. Unless OPTIONS is specified, FIXRAT processing of the tree shown in Figure 6-1 generates the following default output:

```

DISK  PACK  ID  IS  DSKRAT

BEGIN MFD
  BEGIN UFD1
  END   UFD1      21
  BEGIN UFD2
  END   UFD2      11
END   MFD         35
RECORDS USED (DECIMAL)=      35
RECORDS LEFT=                6223
DSKRAT OK
OK,
```



FIXRAT terminal output for the above file structure would be:

```

DISK PACK ID IS DSKRAT

BEGIN MFD
  BEGIN UFD1
    BEGIN SUFD11
      END SUFD11      10
    BEGIN SUFD12
      END SUFD12      10
    END UFD1          21
  BEGIN UFD2
    BEGIN SUFD21
      END SUFD21      10
    END UFD2          11
  END MFD             35
RECORDS USED (DECIMAL)=      35
RECORDS LEFT=                6223
DSKRAT OK
OK,
    
```

Figure 6-1. Sample File Structure

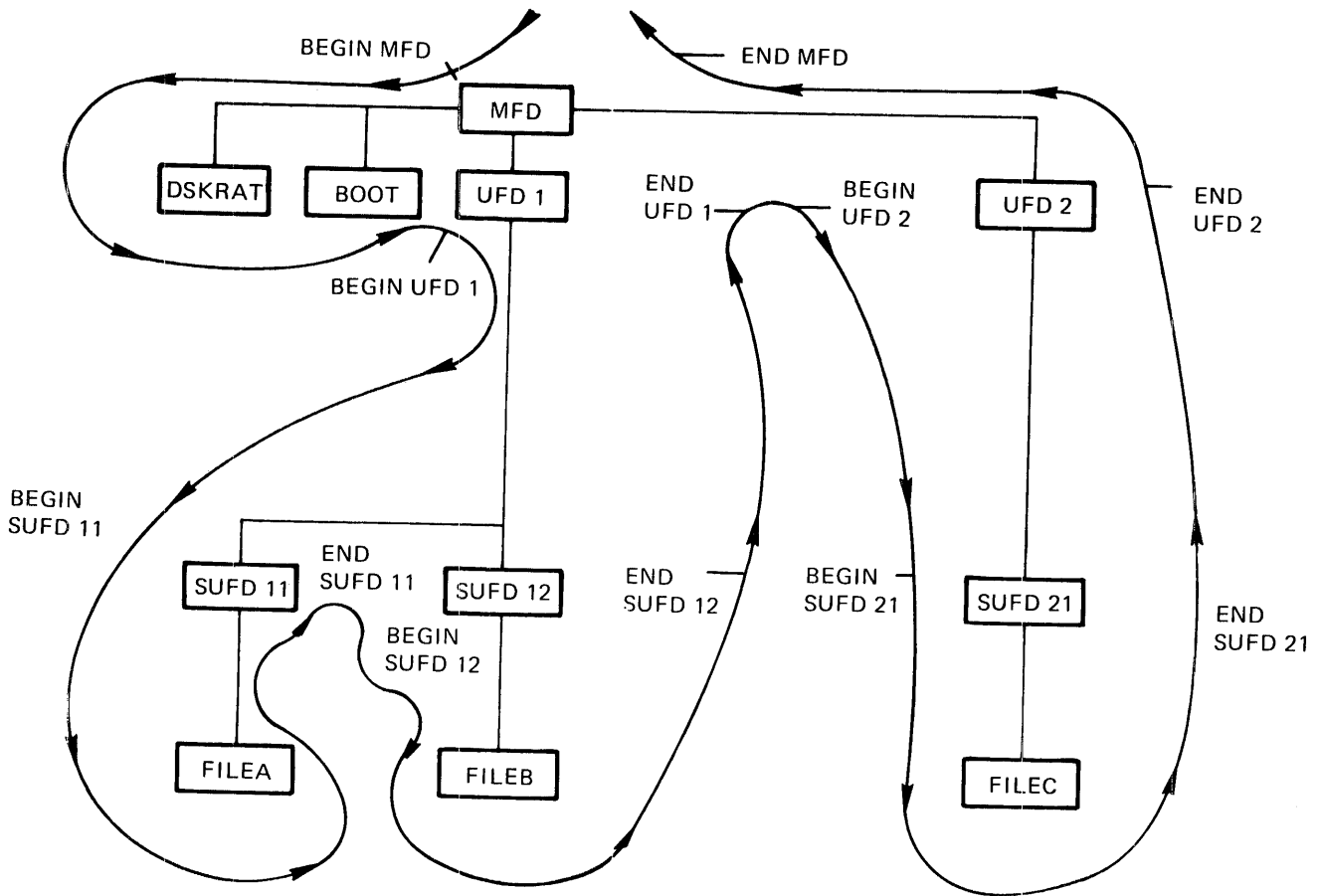


Figure 6-2. Typical FIXRAT Traverse of File Structure

RUNNING FIXRAT

When invoked, FIXRAT asks a series of questions; all answers are followed by a carriage return (CR). The command format is:

FIXRAT [OPTIONS]

If the optional argument OPTIONS is included in the command line, FIXRAT requests answers to printout option questions after the device to be checked is specified:

1. Level to which directory names are to be printed (Default is 2.)
2. Are files names to be typed? (Default is NO.)
3. Are file chains to be typed? (Default is NO.)

If the OPTIONS argument is omitted, FIXRAT uses the default answers, printing only the name and number of records used (in decimal) in the MFD and in each directory in the MFD.

FIXRAT first asks:

FIX DISK?

Note

We recommend the answer be NO for the first time. If any problems occur, first make sure these problems are not caused by hardware or the operating system, then, rerun FIXRAT and answer YES to this question.

If the answer is YES, FIXRAT compresses UFDs, truncates or deletes defective files and generates a corrected DSKRAT file, in addition to checking the file structure and repairing all file structure errors. If the answer is NO, FIXRAT will ask:

UFD COMPRESSION ?

YES causes FIXRAT to compress UFDs, eliminating entries flagged as being for deleted files or directories. A NO answer means that FIXRAT will not perform any disk modifications. This allows FIXRAT to be run on write-locked disks.

In all cases, FIXRAT tests the integrity of the file structure and prints error messages, whether or not it modifies the disk.

After the FIXRAT operations have been selected, the next question is:

PHYSICAL DISK DRIVE =

Enter the number of the physical device (or partition) on which FIXRAT is to be run; FIXRAT then prints the disk packname (which is the name of the DSKRAT) and begins processing the file structures. The DSKRAT is always the first file in the MFD.

If the disk is partitioned, disk numbers include head-offset and number-of-heads information. If the user gives an incorrect disk number, one of the following messages is printed at the terminal:

```

DEVICE, DSKRAT DIFFER IN HEAD COUNT. ABORT?

DISK READ ERROR with status of 177777      ...FIXRAT aborts

WRONG RECORD SIZE IN RAT HEADER           ...FIXRAT aborts

RAT HEADER WRONG LENGTH                   ...FIXRAT aborts

```

If the command was invoked as FIXRAT OPTIONS, three print option questions will be asked. If invoked as FIXRAT, the questions will not be asked and FIXRAT will use the default answers.

The first question is:

TYPE DIRECTORIES TO LEVEL =

Enter an octal number corresponding to the lowest level in the tree structure in which directory names are to be printed. The following table describes the output:

| <u>Level</u> | <u>Output</u> |
|--------------|--|
| blank | All directories |
| 1 | MFD only (level 1 directory) |
| 2 | MFD and all directories in MFD file (level 2 directories) |
| 3 | All output for level 2 and all directories at level 3 (level 3 directories) |
| . | . |
| . | . |
| . | . |

Note

FIXRAT will trace the nesting of directories to a depth of 100 levels only.

FIXRAT will then ask:

TYPE FILE NAMES?

If the answer is YES, FIXRAT prints all filenames in all directories, indented appropriately. This option is useful for listing the contents of a disk. Unless the user requests suppression of directory name output by answering the TYPE DIRECTORIES TO LEVEL = question with 1, directories are printed three times; twice as directories and once as files. If only a listing is desired use the LISTF or LISTSAVE command of FUTIL.

FIXRAT will then ask:

TYPE FILE CHAINS?

If the answer is YES, FIXRAT prints the disk address of all records in all files on the disk. All files consist of one or more records chained together by pointers. This option is useful to see how files are scattered on a disk. FIXRAT begins processing the disk after this point, as it would if the OPTIONS parameter was not specified.

If utility was invoked as FIXRAT rather than FIXRAT OPTIONS, the three print option questions will not be asked and operations will proceed with the default answers of 2, NO, NO. That is, information will be printed about the MFD and directories in the MFD and no filenames and no file chains will be printed.

Following the file structure analysis printout, FIXRAT prints the number of records used and the number of records left on the pack for file system use. Finally, FIXRAT compares a record availability table (built from the existing file structure) against the DSKRAT. If they match, FIXRAT types DSKRAT OK and exits to PRIMOS. If they do not match, FIXRAT types DSKRAT FILE DIRECTORIES MISMATCH.

FIXRAT Output Example

The following is sample FIXRAT output generated after all questions have been answered:

```

DISK  PACK  ID  IS  DSKRAT

BEGIN MFD
  BEGIN CMDNCØ
  END   CMDNCØ      21
  BEGIN DOS
  END   DOS         11
  END   MFD         35
RECORDS USED (DECIMAL)=      35
RECORDS LEFT=                6223
DSKRAT OK
OK:

```

The first line is the disk packname, the name of the DSKRAT file.

The next section of output concerns FIXRAT examining the file structure on the disk for consistency. This sample output is generated from a disk that contains only two directories, CMDNC0 and DOS, in the MFD. If either of these directories contains subfile directories, FIXRAT traces the nested directory structure but does not print the name of the subfile directories. Each directory is printed twice; following the word BEGIN when FIXRAT enters the directory, and following the word END when FIXRAT is finished processing the directory and any subfile directories nested within it. Directories that are files in the MFD are indented one space when typed, to show the nested structure.

Following the directory name, FIXRAT prints the number of records used in the directory plus all files nested within that directory. Since all files on a pack are nested within the MFD, the number of records used in the MFD always matches the number of records used on the disk pack.

After the file structure analysis, FIXRAT prints the number of records used on the pack and the number of records left on the pack for file system use.

Finally, FIXRAT compares a record availability table (RAT) built from the existing file structure against the DSKRAT. In the preceding example, they match and FIXRAT prints:

```
DSKRAT OK
```

and exits to PRIMOS.

If the RAT and DSKRAT totals do not match, FIXRAT prints:

```
DSKRAT, FILE DIRECTORIES MISMATCH
```

If the user typed YES to the question FIX DISK ?, FIXRAT repairs the DSKRAT and types:

```
DSKRAT FIXED
```

then exits to PRIMOS. Otherwise, FIXRAT asks the user: FIX DISK? If the user answers YES, the DSKRAT is repaired. This option is useful if there are no file structure errors but there is a bad DSKRAT.

Broken File Structure Messages

When FIXRAT detects a problem in the file structure, it prints an error in the following format (all numeric values are octal except index level which is decimal):

reason for error

FILE = filename TYPE= filetype

BRA= bra FATHER= fra INDEX LEVEL= index-level

BAD RECORD= cra TRACK=track HEAD=head

DIRECTORY PATH= directory-path (or MFD if file is MFD)

FILE DELETED, FILE TRUNCATED or blank

filetype is the type of file: SAMFIL, DAMFIL, SAMSEG, DAMSEG, SAMUFD, or ILLEGAL.

bra is the beginning record address of the file.

fra is the record address of the father-directory.

index-level is current index level (0 except for DAM files).

cra is the current record address.

track is the cylinder number at which the error occurred. The outside rim of the disk is track 0. Track numbers increase inwards up to 822.

head is the surface of the disk. The bottom of the disk pack is head 0. (Storage modules have up to 19 heads).

The directory-path is the list of nested file directories needed to get from the MFD to the bad file. For example, if FILEC in Figure 6-2 was broken, the directory path would be MFD, UFD2, SUFD21. Because all treenames have the MFD as a root, "MFD" is not printed as part of the path.

After printing the directory path, FIXRAT prints the disposition of the bad files. If the FIX DISK question was answered NO, FIXRAT does nothing to the file, and therefore prints nothing. Otherwise, FIXRAT either truncates the file before the bad record and prints FILE TRUNCATED, or, if no part of the file can be saved, removes the file directory entry from a UFD, or zeroes the entry in a segment directory and prints FILE DELETED.

If FIXRAT is requested to FIX DISK and detects a bad file, it either truncates or deletes the file, depending on where in the file a problem is detected. If FIXRAT deletes a file, the action taken depends on the type of directory into which the file is entered. If the directory is a UFD, FIXRAT removes the entry from the directory in an action similar to that of the DELETE command. If the directory is a segment directory, FIXRAT sets the entry to a null entry.

Segment Directories

A segment directory may contain references to files and other segment directories. The distinction between a UFD and a segment directory is that entries in a UFD are referenced by name and those in a segment directory by position in the directory. Each entry in a UFD consists of a disk address that is the beginning record address of the file, followed by a name (REFERENCE GUIDE, FILE MANAGEMENT SYSTEM, PDR3110). In a segment directory, FIXRAT prints the absolute position of the file in the segment directory as a decimal number. As with user file directories, identically named files in different segment directories represent unique files.

Disks With Badspots

FIXRAT handles disks with badspots (Refer to MAKE, Section 7). In checking integrity of the DSKRAT, the file BADSPT in the MFD is also examined by FIXRAT. The file BADSPT contains information about badspots on the disk, i.e., the record addresses of badspots on the disk. Disk records that have badspots are not available for file system use, and FIXRAT must take this factor into account when fixing a disk or checking file integrity.

The file BADSPT may not be present on the disk, or it may be empty. In both cases, FIXRAT performs no badspot handling.

The format of the file BADSPT in the MFD is described in Section 8. If the format of the file BADSPT is incorrect, FIXRAT prints the message:

```
BAD BADSPT FILE, IGNORED
```

Otherwise, FIXRAT types:

```
BADSPT FILE HAS ENTRIES = number-of-badspots
```

When FIXRAT is finished processing the disk, if there are any badspots that affect the current disk or partition, FIXRAT prints:

```
BADSPT RECORDS LOST = number-of-records-lost
```

If a badspot is found on the disk in records 0 to 15 (BOOT, MFD, etc.), FIXRAT prints the following message and aborts:

```
CANNOT PROCESS BADSPOT FOR RECORDS .LT. 16
```

Bad BOOT

If the BOOT file in the MFD is accidentally deleted or broken, PRIMOS will allocate record number 0 to the next new file. FIXRAT will complain if any file except the BOOT in the MFD contains record 0. The message given is:

BAD DISK ADDRESS BAD RECORD = 0

If this occurs, RESTORE the BOOT from a good MFD on another disk and SAVE it into the MFD before doing anything else.

FIXRAT ERROR MESSAGES

This section lists all error messages generated by FIXRAT and gives an expanded explanation of them. The user should be familiar with the details of the file structure. Error messages are of the form:

reason for error

FILE = filename TYPE= filetype

BRA= bra FATHER= fra INDEX LEVEL= index-level

BAD RECORD= cra TRACK=track HEAD=head

DIRECTORY PATH= treename (or MFD)

Description of Messages

Bad Structure Messages

FILE = MFD BAD RECORD = 1
 DIRECTORY PATH = MFD
 FIXRAT ABORTED

A MFD has been altered and damaged.
 The best action to take is to copy the
 backup disk onto the "daily user disk"
 and continue.

DSKRAT NOT IN MFD
 FIXRAT ABORTED

The DSKRAT has been accidentally deleted
 from the MFD. Suggested action
 is same as for a damaged MFD.

RECORD READ OK NOW CHECKS BAD
 POSSIBLE DRIVER ERROR, FIXRAT ABORTED

Suggested action is to run the disk
 diagnostic on a scratch disk pack.

DIRECTORY RECORD READ OK NOW CHECKS BAD
 POSSIBLE DRIVE ERROR, FIXRAT ABORTED

Suggested action is to run the disk
 diagnostic on a scratch disk pack.

Check For MFD

FIXRAT checks that the first three entries in the MFD are DSKRAT, MFD, and BOOT. The DSKRAT may have any name and the name is used on the disk pack ID (identification). The error messages that may arise as a result of one of these entries being missing are:

DSKRAT NOT IN MFD, REPLACE IT?

MFD NOT IN MFD, REPLACE IT?

BOOT NOT IN MFD, REPLACE IT?

MFD HAS BAD NAME, REPLACE?

If there is a YES (followed by CR) response to each of these questions, the specified action asked in the message is performed. The user must neither delete nor alter the DSKRAT, MFD, or BOOT, since these are system files used by PRIMOS.

2 FILES POINT TO SAME RECORD

Two files point to the same first record; FIXRAT prints the name of the second file only. This error may occur if the DSKRAT is changed by a user overwriting PRIMOS II, or if the BADSPT file is changed after first running FIXRAT. Records already have been erroneously made available to new files.

BACK POINTER MISMATCH SHOULD BE good IS bad

The back pointer of a record does not point to the previous record of the file. If the current record is the first record of a file, the back pointer is not 0.

BAD DAM POINTER

A DAM data file or DAM segment directory has a bad index in the first record of the file, and the nth index of the file does not point to the nth record of the file for all records of the file. This error is repaired by FIXRAT.

BAD DISK ADDRESS

A pointer to a disk record is out of range. Acceptable range is between 1 and NRECS-1, where NRECS is the number of records available for file system use. NRECS is stored in the DSKRAT data header. A record address of 0 is acceptable only for the disk bootstrap loader file BOOT in the MFD.

BAD FILE TYPE file-type-number

The file type, in the first record of the file is not between 0 and 4.

BAD FORWARD POINTER forward-record-address

The forward pointer address is not in the current physical disk or disk partition.

BAD INDEX LEVEL SHOULD BE good IS bad

The index has an incorrect level indicator.

BAD RECORD ID

The first word of a record contains a number unequal to its record address. This message may be preceded by 10 disk error messages because this problem could indicate a disk drive problem.

FIXRAT has difficulty determining whether the error is a disk drive error or a broken file. The disk driver retries 10 times, producing 10 disk error messages and then returns to FIXRAT, which prints the message BAD RECORD ID. Be sure to allow FIXRAT 10 disk error messages before assuming there is disk drive trouble. Refer to the description of an Unrecovered Read error in this section.

BAD UFD HEADER

The directory header contains bad data other than that covered by other error messages.

BAD WORD COUNT word-count

The data word count of a record is not reasonable. A word count of 0 indicates an empty record.

BRA POINTER MISMATCH SHOULD BE good IS bad

The beginning-record word of the second record (or greater) of a file does not point to the beginning record of the file.

CANNOT DELETE BOOT, RAT, OR MFD

An error, which would normally cause deletion of a file, has been found in one of the abovenamed files. FIXRAT aborts.

DAM INDEX TOO LONG

The index is too long to represent the file. FIXRAT truncates the index.

DAM INDEX TOO SHORT

The index is too short to represent the file. FIXRAT truncates the file.

DAM POINTER MISMATCH SHOULD BE good IS bad

The record pointers in the index do not match the record pointers in the file.

DIRECTORIES NESTED TOO DEEPLY

Directories may be nested to a depth of 100 levels. FIXRAT cannot follow the directory tree because the user has nested directories to more than 100 levels. FIXRAT aborted.

DISK ERROR, FIXRAT ABORTED

An error occurred in reading the MFD or DSKRAT file.

DSKRAT BAD

This message is obtained if the DSKRAT file contains any bad record pointers, or contains inconsistent information. If the DSKRAT is BAD, FIXRAT reconstructs it, using parameters typed by the user in response to the following questions. If the user types CARRIAGE RETURN to any of the questions, default values are used. The questions are:

```

INPUT DECIMAL RECORD SIZE =
INPUT DECIMAL FILE SPACE RECORD COUNT =
INPUT DECIMAL CYLINDERS =
INPUT DECIMAL HEADS =

```

The default values for the particular disk model are used by FIXRAT if the user does not specify them and type (CR) instead. FIXRAT prints the default or the specified values back to the user for verification then asks "OK"?. If the answer is YES, FIXRAT repairs the DSKRAT and continues, or else it requests the parameters again. Refer to Appendix G for tables of disks and values.

FATHER POINTER MISMATCH SHOULD BE good IS bad

The father-record word of the first record of a file does not point to the beginning record address of its file directory.

FILE TYPE MISMATCH

The file type in the file header does not match the file type in the UFD entry for this file.

INCONSISTENT ENTRY IN UFD: RECORD=record, WORD=word.
CHANGED TO VACANT

Information in a file entry in a UFD is not self-consistent and cannot be reconciled. The entry for this file is flagged as being deleted. If UFDs are compressed, this entry will be eliminated.

NOT ENOUGH MEMORY

There is insufficient space to read the DSKRAT file into memory.

RAT MISSING

The disk record availability table is not in the MFD. FIXRAT aborts.

UNRECOVERED DISK READ ERROR

On an unrecovered disk read error, the track and head of the bad record are reported. If desired, this badspot record location may be manually added to the BADSPT file using SAVE and RESTOR (see section 7). Normally, unrecovered read errors indicate a malfunctioning or misaligned disk drive, a head crash, or a bad disk pack. An unrecovered read error may also indicate a pack that has badspots that were not entered into the BADSPT file during the last MAKE operation. When an unrecovered read error occurs, FIXRAT must be rerun after the BADSPT file is modified. When FIXRAT is rerun as a result of the unrecovered read error recovery procedure described here, multiple errors of the form:

2 FILES POINT TO SAME RECORD

may occur. These errors result because a badspot may affect up to eight records belonging to multiple files.

EXAMPLE OF FIXRAT

```

OK, FIXRAT OPTIONS
GO
REV. 14.00
FIX DISK? YES
PHYSICAL DISK = 70460
TYPE DIRECTORIES TO LEVEL = 3
TYPE FILE NAMES? YES
TYPE FILE CHAINS? NO

DISK PACK ID IS DTNEW
BEGIN MFD
  DTNEW
  MFD
  BOOT
  CMDNC0
  BEGIN CMDNC0
    AVAIL
    SIZE
    FILVER
    PSD
    MAGSAV
    MAGRST
    SLIST
    FILMEM
    SAVER
    FUTIL
    FIXRAT
    HILOAD
    EDB
    MAKE
    TAP
    UPCASE
    TRAMLC
    COPY
    ED
    FTN
    SFRWLK
    LOAD
    PMA
    TA
    VPSD
    LS
    DL
    PSD120
    CMPF
    VLOAD
    PRUFD
  END  CMDNC0          229
END
DOS
BEGIN DOS
END  DOS              1

```

Partition of a storage module (type 6). The partition has 2 heads and a head-offset of 14 heads. The device is mounted on unit 0 and connected to controller address 27. See Appendix G for details.

```

LIB
BEGIN LIB
  FTNLIB
  APPLIB
  UII
  B_PSD
  B_LOAD
END LIB      84
SYSCOM
BEGIN SYSCOM
  FILERR
  KEYCOM
  FILD.F
  OFFCOM
  FILD.P
  PARM.K
  KEYS.F
  KEYS.P
  A$KEYS
  ERRD.F.OLD
  ERRD.F
  ERRD.P.OLD
  ERRD.P
  DEFI
  CBLK.I
END SYSCOM  26
MORRIS
BEGIN MORRIS
  RDENDIR
  BEGIN RDENDIR
    QDOS64
    C_RDEN$$
    C_LOAD
    B_RDEN$$
    L_RDEN$$
    L_LOAD
    QMAP64
    TDOS64
    PRUFD
    *PRUFD
    L_PRUFD
    B_PRUFD
    M_PRUFD
    RDEN$$ .IV
    RDEN$$ .III
    *LDUT
    RDEN$$ .CMP
    RDEN$$ .TEST
    RDEN$$
  END RDENDIR      88
END MORRIS      89
MAMNT
BEGIN MAMNT

```

```

*TEST
*PRUFD
*TDUMP
LSDIR
BEGIN LSDIR
  CRCVŞA.FTN
  LS.FTN
  SPANŞA.FTN
  IGCHŞA.FTN
  INDXŞA.FTN
  SBSTŞA.FTN
  ASSNŞA.FTN
  COMPŞA.FTN
  BRKŞA.FTN
  APNDŞA.FTN
  CCATŞA.FTN
  CTTVŞA.FTN
  XLATŞA.FTN
  SCHRŞA.FTN
  DL.FTN
  CEQNŞA.FTN
  STARŞA.FTN
  OLS.FTN
  SPL.FTN
  FPRTŞA.FTN
  CVDTŞA.FTN
  CFNMŞA.FTN
  FTN.C
  MOST.FTN
  MORE.FTN
  B MORE.FTN
  APNDŞA.B
  TLS
  TLS.MAP
END LSDIR          57
SEGDIR
BEGIN SEGDIR
  ( 0)
  ( 1)
  ( 2)
  ( 3)
  ( 4)
  ( 33)
END SEGDIR        15
END MAMNT         93
END MFD           525
RECORDS USED (DECIMAL) =      525
RECORDS LEFT =      14298
DSKRAT OK

```


SECTION 7

MAKE

OVERVIEW

MAKE is the system utility for the creation and formatting of disks and disk partitions, including both user and paging disks. MAKE will create and structure any disk or storage module supported by PRIMOS. The disk or disk partition is formatted with the following PRIMOS features:

- Disk record availability table (user-specified name)
- MFD
- BOOT
- CMDNCØ
- DOS
- BADSPT (if there are badspots on the disk)

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

MAKE may be run from a command file.

PRELIMINARY

The operator must first construct the number of the physical disk (or partition of the disk) from the tables in Appendix G. This number tells the system the type of storage device, the unit on which it is mounted, and the offset and size (if a partition). This physical device number must be added to the table of assignable disks with the DISKS command at the supervisor terminal.

The operator logs in on a user terminal to a UFD on another physical device. The disk to be formatted by MAKE is assigned to this terminal by the ASSIGN DISKS command, using the same physical device number as for DISKS. To prevent accidental erasure of data on a disk because a physical device number was mistyped, the following procedure is recommended.

- Under PRIMOS III, IV, or V - only the disk to be created by MAKE should be assigned to the terminal.
- Under PRIMOS II - all running disks should be write-protected except the disk to be created by MAKE. (Disk drives have switches labelled WRITE PROTECT.)

The disk to be created must not have been connected to PRIMOS by the STARTUP command.

RUNNING MAKE

To run MAKE, type the command: MAKE. The response is:

```
MAKE, REV 15.1.  
BUILDING NEW PARTITION.  
PHYSICAL DISK:
```

The user must then type the number of the physical disk to be created. An impossible number causes the message ILLEGAL DEVICE NUMBER (MAKE) to be typed out and the request to be repeated. A complete guide to the construction of physical device numbers (including partitions) is given in Appendix G.

MAKE computes the number of records on the disk pack from the disk number. In case of ambiguity, MAKE asks resolving questions, answerable by YES or NO, such as: 40 MB STORAGE MOD?

When the ambiguity, if any, is resolved MAKE asks:

```
SPLIT DISK?
```

If part of the disk is to be used for paging (see Sections 18 and 19), answer YES, otherwise NO. This feature is rarely used and the answer is normally NO. If answered YES, MAKE then asks:

```
PAGING RECORDS (DECIMAL)
```

The user responds by typing the number of records to be used for paging. This number is determined from the number of users to be run under PRIMOS III, IV or V. (See Sections 18 and 19 for details).

Notes

It is not recommended that partitions on storage modules be split. Instead, create a 1-head paging partition on the module.

If the user desires to make a split disk, the recommended procedure is to first run MAKE with the SPLIT DISK question answered NO, then rerun MAKE with the SPLIT DISK? question answered YES. The reason for this action is to write correct checksums on the entire disk initially, not just on the file system part of the disk. This action is also important if it is desired to copy the split disk at a later time using the COPY command.

MAKE then prints the disk number, file records, and paging records at the terminal.

| DISK | FILE-RECORDS | PAGE-RECORDS (DECIMAL) |
|--------|--------------|------------------------|
| number | number | number |

and asks:

PARAMETERS OK?

If the number is correct, type YES in response to the OK? query. If not, type NO, and MAKE will return to the PHYSICAL DISK: question. MAKE then asks the question:

PACK NAME?

Enter a valid filename which will be the name of the disk record availability file (DSKRAT).

Note

The name of the DSKRAT file (packname) may be changed later, for example, by the CNAME command.

For all devices except diskettes (floppy disks), MAKE asks:

BADSPOTS ON DISK?

Badspots are part of a disk pack that cannot hold data. On some disk packs, this information is recorded on the inside bottom cover in the form of a list of pairs of numbers. These pairs (track number, head number) represent parts of the disk determined by the disk manufacturer to be probable badspots. If the list exists, the user must type YES response to the BADSPOT question. MAKE then prints:

TRACK=

The user responds by typing the track (decimal) of the first badspot. MAKE then prints:

HEAD=

The user responds by typing the head (decimal) of the first badspot. This dialogue between MAKE and the user continues allowing the user to record the track and head of all badspots on a given disk. The TRACK, HEAD pair 0, 0 terminates input of badspots. MAKE then prints a list of the badspot HEAD and TRACK numbers at the terminal and asks:

PARAMETERS OK?

If they are OK, the user types YES. At this point, MAKE writes a file containing the badspot information in the MFD with the filename BADSPT. The file named BADSPT is used by the utility commands FIXRAT and COPY. MAKE also initializes the DSKRAT file to ensure that badspot records are not available for file system use. A NO answer returns to the TRACK= question.

If there are no badspots on the disk, the user responds by typing NO to the query BADSPOTS ON DISK?, and no BADSPT file is created.

MAKE then asks:

VIRGIN DISK?
VERIFY DISK?

Virgin Disks

If the user answers YES to the first question, MAKE writes the records of the disk or partition. The first word of each record is set to the record address. This action also writes a valid hardware checksum for each record.

If the user answers NO, MAKE does not initialize the records but writes BOOT, DSKRAT file, and MFD and proceeds to verification. The records need not be initialized if all the records have been initialized by a previous run of MAKE. However, it is strongly recommended that the user answer YES to the VIRGIN DISK ? question at each invocation of MAKE.

Unless the disk is a storage module MAKE immediately begins disk initialization. For a storage module MAKE asks:

FORMAT DISK?

A NO answer begins disk initialization. A YES answer causes the disk tracks to be correctly formatted. During this process the current track number is displayed on the control panel lights. Upon completion MAKE prints the following message and begins disk initialization:

FORMAT COMPLETED

Disk Initialization

MAKE creates and writes to the disk the bootstrap (BOOT), disk record availability table, and the MFD. If requested by the user, MAKE initializes all remaining records on the disk. At the start of this process it prints:

BEGINNING WRITE

and upon completion:

WRITE COMPLETE

If the disk is not virgin, this step is omitted by the user.

Verification

If the verification question was answered YES, MAKE reads every record in the file system part of the disk or partition to verify that each record can be read. If the answer was NO, MAKE is complete.

During the reading and writing of all records, MAKE displays the track number it is processing in the DATA lights. When done, MAKE prints:

DISK CREATED

and returns to the operating system, which types OK,. If any of the badspots affected the current partition or platter, the message: LOST RECORDS is printed. If any disk write errors occur, MAKE retries nine times. Each error results in an error message of the form:

DISK WT ERROR device # PRIMOS record # status-word

If the write is not successful after 10 tries, MAKE aborts, prints the message UNRECOVERED ERROR, and returns to the operating system. If a DISK-NOT-READY status is detected, a single disk error message is typed with a status of 177776. The software then retries to write, waiting for the disk to become ready. If a read is not successful, MAKE gives up and prints:

READ ERROR, RECORD = record-number

MAKE then asks:

ADD TO BADSPOT FILE?

If the user responds by typing YES, MAKE prints the track and head of the badspot and modifies the BADSPT file and DSKRAT appropriately. If a badspot is found on record 0 to 15 MAKE prints the error message:

CANT ADD RECORDS<=16 TO BADSPOT FILE. (MAKE)

and returns to PRIMOS level. Records 0 to 15 contain the bootstrap and badspot file and cannot contain any badspots. A disk with flaws in these locations is not usable by PRIMOS.

If the user runs MAKE, then answers YES to VIRGIN DISK? and VERIFY DISK?, it is possible to find out immediately if there is any problem in the file structure part of the disk pack!

FINAL OPERATIONS

After the disk has been created, the user should unassign it with the UNASSIGN DISK command. Afterwards the operator should use the operator terminal to remove this disk from the table of assignable disks with the DISKS command. The physical device number to be used in these commands is the one constructed for the disk just created.

If PRIMOS is to be bootstrapped from this disk, use FUTIL to copy *DOS32, *DOS64, and RDOS64 from UFD DOS on a master disk to UFD DOS on the newly created disk. The BOOT file in the MFD that is read from the disk by the control panel boot expects these files to be in UFD DOS in order to bootload PRIMOS using the newly created disk pack. If the disk is to be used only as a user partition, it is not necessary to copy these files

It may be desirable to use part of a disk pack for the file structure and part for the paging device under PRIMOS III, IV, and V. The user must follow the directions given in Section 12.

DESCRIPTION OF THE BADSPT FILE

The BADSPT file is a saved memory image. The file may be examined and modified by RESTORing it and referencing the file with the debugging command PSD. BADSPT is restored into consecutive memory locations starting at location '1000. A BADSPT file is created only if badspots exist; this can be verified by issuing the PM command following the command line:

RESTOR BADSPT

Pairs of numbers, starting at '1000, constitute the track and head number of each badspot on the disk. The file BADSPT can be no longer than one record. Thus, BADSPT records a maximum of 215 badspots. To add a badspot to the BADSPT file, place the track and head data in locations given symbolically by EA+1 and EA+2 (EA is the second parameter printed by the PM command); then, give the command:

SAVE BADSPT 1000 new-end-address

where new-end-address has the value EA+2.

The BADSPT file may be used in conjunction with all disks except fixed head disks and diskettes.

CAUTION

The BADSPT file must not be deleted, or overwritten, or copied TO. This file is used by FIXRAT and COPY to process badspots.

SECTION 8

BACKING UP

INTRODUCTION

Backing up is a procedure for making copies of current data files and programs. These copies are then available to be restored, in part or in full, in the event the files on the system are lost or broken.

Backing up:

- Aids in maintaining the integrity of user data.
- Provides recovery in case of loss of on-line data.
- Gives the user the ability to restore earlier versions of files or programs.
- Allows inactive files to be removed from the system.

File losses may be considered major (many files or entire disk of data lost) or minor (one or a few files lost).

Causes of major losses include:

- Physical damage to disks by fire, being dropped, etc.
- Operator error (running MAKE on a disk which contains current data, using a FUTIL UFDDEL from the MFD, etc.).

Causes of minor losses include:

- Power failure during write operations which may cause pointer mismatch, discrepancies with DSKRAT file, etc.
- User deletion or truncation of the wrong file by mistake (most common cause).

Scheduling

The exact scheduling of backups depends upon the installation, how often files are changed, and how important current information is. Example: One development system at Prime is a Prime 500 with two 300MB storage modules and one 80MB storage module. On Monday, Wednesday, and Friday mornings the active partitions on one of the 300MB modules are backed up onto a storage module with COPY; on Tuesdays and Thursdays the other 300MB storage module is COPYed. FIXRAT is run on the 300MB module not being backed up that day. The 80MB module is not as active and is not backed up during the week.

Two sets of backup disks are used to provide coverage between the weekly magnetic tape backups.

Each weekend the entire system is copied onto magnetic tape with the MAGSAV utility. Weekly tapes are kept for 2 months; the first tape of each month is kept for two years. This degree of backup protection is probably well in excess of that needed by the average system.

Note

To protect the system data in the event of a fire, some level of backup should be stored away from the computer room, preferably in another building or in a fire-proof vault.

BACKING UP ON DISK OR MAGNETIC TAPE

Backups may be disk-to-disk (using COPY) or disk-to-tape (using MAGSAV); both procedures are described in this section. Each method has particular advantages, which are made use of in the scheduling example above.

Disk-to-Disk Advantages

- Faster data transfer rate. Typically a fully-used 300MB module can be COPYed in about 1 hour. (Times may be scaled for smaller mass storage devices.) It takes 10-15 minutes to completely fill one 2400 foot reel of magnetic tape at the nominal speed of 45 inches per second; backing up on magnetic tape will take about 6 hours.
- Random access of data. Files in a storage module or disk can be accessed rapidly by PRIMOS using the directory tree structure. To locate a file on a magnetic tape backup, first the correct tape must be located and then this tape searched by MAGRST to find the desired file. The random access feature is even more convenient when more than one file is to be restored from the backup.

Disk-to-Tape Advantages

- Lower relative cost. Although it takes about 25 magnetic tapes (at 800 bpi) to backup a 300MB module, the cost of these tapes is less than the cost of another module.
- Handling. Magnetic tapes can be stored under a wider range of environments than storage modules. Storage modules must be handled more carefully than tapes since the mechanical tolerances of the module surfaces, with respect to head alignment, are exacting.

COPY (DISK-TO-DISK)

COPY is an external command that copies and verifies a disk. COPY copies between physical devices (disks or partitions) under PRIMOS. Under PRIMOS III, IV, or V, both disks must be ASSIGNED before invoking COPY. Since COPY duplicates a disk track-by-track (with badspot consideration) it is considerably faster than a directory-driven copying procedure. However, COPY cannot perform UFD compression or pointer cleanup. Use the FIXRAT utility for these procedures.

Running COPY

After the user types COPY at the terminal, the COPY command responds by printing a series of questions to which the user replies at the terminal. All user responses are terminated by a carriage return (CR).

Device Specification

When asked, the user must specify the device to be copied from (FROM), and the device to be copied to (TO). Depending upon the device number entered, ambiguity-resolving questions as: 1.5M WORD PACK? or 40MB STORAGE MOD? may be asked. The size of the FROM device and the TO device must be equal; the user must be sure to specify disks or partitions of the same size when replying to the FROM and TO queries. If not, COPY prints the message: REC LENGTH AND NR RECS MUST BE = FOR BOTH DEVS and repeats the FROM question. The FROM and TO parameters are physical device numbers (see Appendix G).

If FROM or TO is not a valid physical disk number, or if the user replies NO to the query PARAMETERS OK?, COPY repeats the series of questions and waits for reply from the user. If the new parameters are acceptable, COPY initiates the copy operation.

Note

COPY does not allow rewriting of the same disk. For example, specification of the same physical disk number in reply to the FROM and TO queries is an illegal specification of COPY parameters.

WARNING

A TO disk number must not be a disk connected to PRIMOS by the STARTUP command. It is good practice, when running COPY under PRIMOS II, to place all active disks in WRITE PROTECT before initiating the COPY command, except for the disk to be written to (TO disk). It is good practice when running under PRIMOS III, IV, or V, to place all disks assigned to the user terminal at which the COPY command is to be initiated to WRITE PROTECT, except the TO disk, before initiating the COPY.

Copying Method

COPY copies disk records from the FROM disk to the TO disk and, when done, verifies the copy by reading each record from both disks and performing a word-by-word comparison in memory. During this process, COPY displays the track number it is processing in the DATA lights on the processor control panel, bits 2-16. Bit 1 is off during the copy operation and on during the verify operation. When done, COPY prints DONE and returns to PRIMOS, which prints OK. If any disk read errors occur during the copy, the read is retried nine times. Each error results in an error message of the form (all numbers in the error messages are octal):

```
DISK RD ERROR device-number PRIMOS-record-number status
```

If the read operation is not successful after ten tries, PRIMOS ignores that record and prints the message:

```
DISK RD ERR, DISK=device-number RECORD record-number
```

```
ERROR IGNORED, COPY CONTINUED
```

Then, PRIMOS continues the copy operation. If any disk write errors occur, COPY retries nine times. Each error results in an error message of the form:

```
DISK WT ERROR device-number PRIMOS-record-number status
```

If the write operation is not successful after ten tries, COPY aborts, prints the error message UNRECOVERED ERROR, and returns to PRIMOS. If on either read or write a DISK-NOT-READY status is detected, a single disk error message is printed with the status 177776. The software then retries the read or write, waiting for the disk to become ready. If while verifying the copy, a discrepancy is detected, COPY prints VERIFY ERROR, the record number of the track that contained the error, and the word number within the track that did not compare correctly; verification then continues. (See Appendix E for complete list of status words.)

When the COPY is finished, both writing and verifying, the message

```
DONE
```

is printed at the terminal.

Badspots

COPY can copy FROM a disk or partition that has badspots. COPY looks for the file BADSPT in the MFD (see Section 7). If the COPY command finds the file BADSPT in the MFD, COPY interprets the BADSPT file and skips reading bad tracks on the disk to be copied FROM. The corresponding tracks on the disk to be copied TO are written with data, in order to write good checksums.

COPY skips writing tracks containing badspots on the TO disk. Those users who desire to use a disk containing badspots as the TO disk during a COPY operation should perform the following: create the FROM disk with the MAKE command and specify to MAKE the badspots on both the FROM and the TO disks. If the FROM disk already has something useful on it, the information must be backed up before invoking the MAKE command. Then, the information must be retrieved following the MAKE of the FROM disk, using either the FUTIL or the MAGRST command.

If the BADSPT file exists on the FROM disk, but has a bad format (e.g., from a mechanical flaw occurring where the BADSPT file is located), the message:

BAD BADSPT FILE, IGNORED

is printed; and COPY attempts to copy the disk as if the file BADSPT did not exist. It is worthwhile to try to copy a pack with badspots, for the purpose of backup. COPY skips writing tracks containing badspots on disks it copies TO.

Example

In this example COPY is used under PRIMOS II. Operation under PRIMOS III, IV, or V is similar.

OK: COPY
GO

COPY 15.0
FROM PHYS DISK= 50
1.5M WORD PACK? NO
TO PHYS DISK= 51
1.5M WORD PACK? NO
FROM, TO, RECORDS = 000050, 000051, 6496
PARAMETERS OK? YES

DONE
OK:

Checksums

When copying between a disk written with a 4000 controller (obsolete) and a disk written with a 4002 controller, the controllers will generate different checksums. The command:

COPY NOCHECKSUM

copies the disk without checking checksums, thereby permitting the disk to be copied. During this procedure, there is a small risk that bad information will be copied without detection.

Restoring Data From Disk - FUTIL

Files, trees, and UFDs may be restored to an active disk from a backup disk with the FUTIL utility. The user must be familiar with the TO, FROM, COPY (and possibly TRECPY and UFDCPY) commands of FUTIL (see REFERENCE GUIDE, PRIMOS COMMANDS, PDR3108).

The backup disk must be physically mounted on one of the system drive units. This disk should be write-protected. After the disk is ready, the desired partition should be connected to the system by the STARTUP command at the supervisor terminal. (Use the tables in Appendix G to construct the physical device number for the partition.)

The operator should then log in as a user. The backup, having been created by COPY, will have the same packname as the system disk to which the restoration will be done. Therefore, the user cannot specify the TO and FROM directories in FUTIL with the pathname as: <packname>UFD>etc. Instead the form of the pathname <logical-device-number>UFD>etc. must be used.

The logical device numbers may be obtained by giving the PRIMOS command STATUS DISKS. A table will be printed on the terminal whose first three columns are DISK, LDEV, and PDEV -- the packname, logical device number, and physical device number respectively. This table gives the logical device numbers corresponding to the known physical device numbers for the backup (FROM) and system (TO) disks.

Example: The system disk, containing two partitions SOFTWR and DBTEST, is mounted on unit 1. It is necessary to restore the tree structure of the UFD=DBADM in the MFD on the DBTEST partition. The backup disk is mounted on unit 0. To determine the logical device numbers, the STATUS DISKS command is given:

OK, STATUS DISKS

| DISK | LDEV | PDEV | SYSN |
|--------|------|-------|------|
| SOFTWR | 0 | 3462 | |
| DBTEST | 1 | 71061 | |
| DBTEST | 2 | 71063 | |

OK,

Invoke FUTIL, define the FROM and TO directories and then restore the desired files and or directories with COPY, TRECOPY, or UFDCPY as appropriate. Exit to PRIMOS when restoration is completed.

```

OK, FUTIL
GO
> FROM <1>MFD>DBADM
> TO <2>MFD>DBADM
> UFDCPY
> QUIT

```

OK,

The backup disk is then disconnected from PRIMOS by the SHUTDN command at the supervisor terminal, dismounted, and returned to storage.

MAGSAV/MAGRST (DISK-TO-TAPE/TAPE-TO-DISK)

Complete descriptions of these commands will be found in REFERENCE GUIDE, PRIMOS COMMANDS, PDR3108.

General Information For Magnetic Tape - File Utilities

MAGSAV and MAGRST are utility programs that move files on any disk including storage modules, to a magnetic tape and vice versa. The files may be SAM, DAM, segment directories, UFDs, or an entire disk. Whenever a directory is specified, the directory and all components (the subtree) are transferred. Under PRIMOS III, MAGSAV and MAGRST must be run from a user terminal. Under PRIMOS II, IV, or V, MAGSAV and MAGRST may be run from the supervisor terminal.

Logical Tapes

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.

A disk file appears on tape as a record containing a treename, followed by as many data records as are required for the file. The treename contains the path from the file specified by the user to the current file. When an entire disk is saved, all treenames begin in the MFD.

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 before being considered unrecoverable. Both recovered and unrecovered errors are logged. The first record on a tape is not retried. The total number of recoverable errors, if any occur, will be printed when the end of the tape is reached.

Assigning Tapes

When running MAGSAV under PRIMOS III, IV, or V, the magnetic tape drive must be ASSIGNED. Files or directories 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.

Backing up on Magnetic Tape - MAGSAV

MAGSAV saves files and directories on disks and partitions onto magnetic tape for backup and storage.

MAGSAV is invoked by:

MAGSAV [-7TRK] [-LONG] [-UPDT] [-INC]

- 7TRK Use 7-track tape format; default is 9-track.
- LONG Use 1024-word record size; default is 512-word record size. This is useful for large files as it decreases the amount of space on the tape used for record headers.
- UPDT Update. Set the DUMPED switch in the UFD entry for files and directories that are MAGSAVED; the default is not to set the switch.
- INC Incremental dump. Only those files and directories with a reset (=0) DUMPED switch are MAGSAVED. The default is to MAGSAV all files and directories.

Note

When a file is modified, its DUMPED switch is reset (=0). When the update option (-UPDT) is used with MAGSAV, the DUMPED switch is set (=1) for each file or directory that is MAGSAVED. If the MAGSAV program is run with the incremental save option (-INC), only files that have a zero DUMPED switch will be MAGSAVED. (i.e., only files that have been modified since the last time the MAGSAV program was run, will be MAGSAVED.

MAGSAV requests information in the following order:

- | | |
|-------------------------------|---|
| TAPE UNIT: | The proper response is the physical unit number of the tape (0-7). |
| 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 (MM DD YY): 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. A (CR) answer enters the current system date (PRIMOS III, IV, V).

REV NO: An arbitrary number.

NAME OR COMMAND: Enter a filename to be saved or an action command. filename is the name of the file to be saved on tape.

\$A changes the home UFD (e.g., \$A USER3 PASWD5).

\$Q ends logical tape and exits to PRIMOS.

\$R ends logical tape, rewinds physical tape, and exits to PRIMOS.

\$I causes index to be printed.

\$I level generates an index to the level specified (e.g., \$I 3 prints an index of the MFD, any UFDs and any entries in the UFDs).

\$UPDT ON turns on update. The DUMPED switch will be set for all files and directories MAGSAVed. (Same as option -UPDT).

\$UPDT OFF turns off update.

\$INC ON turns on incremented dump (same as option -INC).

\$INC OFF turns off incremented dumps.

The prompt NAME OR COMMAND: is printed when writing or other action has been completed to allow further action or termination of the logical tape. An incorrect entry causes the prompt to be repeated.

To save an entire disk, attach to the MFD, enter MFD in response to the query NAME OR COMMAND. To save a UFD, attach to the MFD and give the name of the UFD that is to be saved. To save a file in a UFD, attach to the UFD and give the name of the file. MAGSAV also saves a disk that contains nested segment directories.

MAGSAV operates on all disks and storage modules supported by PRIMOS.

Example:

```

OK, AS MT1
OK, MAGSAV
GO
REV. 14.3
TAPE UNIT (9 TRK): 1
ENTER LOGICAL TAPE NUMBER: 1
TAPE NAME: PDR311
DATE (MM DD YY):
REV NO: Ø
NAME OR COMMAND: S1
NAME OR COMMAND: S2
NAME OR COMMAND: S3
NAME OR COMMAND: S4
NAME OR COMMAND: S5
NAME OR COMMAND: S6
NAME OR COMMAND: S7
NAME OR COMMAND: S8
NAME OR COMMAND: S9
NAME OR COMMAND: APPA
NAME OR COMMAND: APPB
NAME OR COMMAND: SR

```

OK,

Restoring Data From Magnetic Tape - MAGRST

MAGRST restores information from a magnetic tape into the PRIMOS file system. The format is:

```
MAGRST [-7TRK]
```

-7TRK Use 7-track tape format; default is 9-track.

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: Enter the action to be taken.

YES restores the entire logical tape.

NO requests another tape unit/logical tape combination.

PARTIAL allows restoration of part of a tape (MAGRST will ask for files to be restored).

\$I [filename] level creates an index to the level specified and writes it to a the terminal or file on the disk, if filename is specified. The files and directories are restored.

NW [filename] level performs the same function as \$I... but files and directories are not restored. The option is useful to determine what is on the tape.

TREE NAME: This prompt is typed when a partial restore is requested. The treename specified is restored from the tape to the disk. The prompt is repeated after each restoration until a null line (carriage return) is entered. The null line signals the end of restoration.

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 treenames 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 need not search all logical tapes when restoring sequential logical tapes. After MAGRST has exited to PRIMOS, the magnetic tape is not rewind. Instead, it is positioned at the location before the beginning of the next logical tape in sequence. In the case of sequential logical tapes, run MAGRST again and specify 0 to the prompt LOGICAL TAPE NO:. Then, the next logical tape is restored without rewinding and reading through the preceding logical tapes.

Index

MAGRST can 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 the \$I command with a filename, and then with the number of index levels.

Example of index with MAGRST:

READY TO RESTORE: \$I TAPE#1 5

In this case, MAGRST restores the tape and writes a 5-level index to the file TAPE#1.

```
OK, MAGRST
GO
REV. 14.1
YOU ARE NOT ATTACHED TO AN MFD
TAPE UNIT (9 TRK): 1
ENTER LOGICAL TAPE NUMBER: 1
NAME: PDR311
DATE (MM DD YY): 02-27-78
REV NO:      0
REEL NO:     1
READY TO RESTORE: NW 1
*** STARTING INDEX ***
S1
S2
S3
S4
S5
S6
S7
S8
S9
APPA
APPB
*** END LOGICAL TAPE ***
*** INDEX COMPLETE ***

OK,
```


SECTION 9

OPERATING THE SPOOLER

INTRODUCTION

The SPOOL subsystem allows any time-sharing user to submit disk files to be printed or plotted on the system line-printer(s) and/or plotter. The subsystem consists of a user interface program (SPOOL), a set of phantom programs to print queued files on the various peripherals, and a queue management facility.

The SPOOL program:

- Allows users to submit files to the queue mechanism, interrogate the queue status, and to cancel queued requests.
- Permits the system operator to communicate with the spooler phantom(s) to request utility operations such as form change, file printing restart/abort, default form length change, etc.

The spool phantom(s) normally run from UFD=SPOOLQ and are run as part of the normal system start-up procedure. For example, the commands:

```
ATTACH SPOOLQ 1/177777  
PHANTOM PH_PR0  
ATTACH
```

from the supervisor terminal (or from the start-up command file), attach the operator to UFD=SPOOLQ and start the phantom spooler program for MPC Printer 0, and attach back to UFD=CMDNC0.

Note

The login name under which the spooler phantom is run (usually SYSTEM) is the name under which the operator must log in to use the operator commands of the spooler. the operator, logged in under a different name, attempts to perform operator functions, the spool program will return an error message and ignore the command.

ADMINISTRATOR AREAS

The following spooler features are discussed as part of the functions of the System Administrator.

- Build phantom file(s) for printer(s), plotter(s) on system
- Build and install spool runfiles
- Select print and/or plot files as default
- Enable and set value for (or disable) short/long file threshold
- Set default paper length (in lines)

OPERATOR COMMANDS

A list of all operator options is given below. These options are detailed later in this section. The command format is:

SPOOL option

| <u>Option</u> | <u>Function</u> |
|---------------|--|
| -ABORT | Stops printing of current file; leaves files in queue |
| -BACK | Restarts printing 128-256 lines prior to current line |
| -DROP | Stop printing of current file; removes file from queue |
| -FINISH | Finishes current file and then halts spooler |
| -GO | Restarts spooler after FINISH, HANG, or PAPER |
| -HANG | Stops printing and halts spooler |
| -LENGTH | Changes length of page |
| -LOGOUT | Logs out spooler phantom |
| -PAPER | Specifies form type |
| -RESTART | Restarts printing from beginning of file |
| -TIME | Sets acknowledge time |
| -USER | Specifies spooler if more than one running |

Specifying the Spooler

If more than one spooler is running on the system, the operator must specify, by user number, the spooler to which the subsequent commands are directed. The format is:

SPOOL -USER user-number

If the user-number is not known it may be determined with the USER option of the STATUS command. The right-most column in the list generated by this command will have PR0 (or PR1, etc.) for the user-number under which the spooler is running.

If more than one spooler is running and the spooler is not specified, the error message:

MULTIPLE SPOOLERS ACTIVE; USER # MUST BE SPECIFIED

will result if any commands are given.

Setting the Acknowledge Time

A command given by the operator may not be implemented immediately if the spooler is still processing or printing (for example, a -PAPER command given during the printing of a long file). If the spooler is unable to acknowledge a command request within a specified time-out period, the message TIMED OUT is printed on the user terminal. The command will be processed but its status will not be available to the operator. This time period may be changed by the command:

SPOOL -TIME time-out

where time-out is the waiting time in seconds. The default value is 120 seconds (2 minutes).

Stopping Printing Operations

Options to terminate printing and/or other spooler operations are summarized below:

| SPOOL option | Stop printing immediately | Stop scanning SPOOL Queue | Drop Current File from Queue |
|--------------|---------------------------|---------------------------|------------------------------|
| -HANG | Yes | Yes | No |
| -FINISH | No | Yes | Yes* |
| -ABORT | Yes | No | No |
| -DROP | Yes | No | Yes |

*after completion of printing

-HANG causes the spooler to stop printing and/or scanning the queue. After this option is used other options such as USER, etc. may be given. Printing and/or scanning is resumed by giving the -GO option (see below).

-FINISH the spooler continues printing the current file. Upon completion of printing, the spooler prints a message on the supervisor terminal informing the operator that printing is completed. The spooler then goes into a "HANG" as above. The spooler is restarted with the -GO option.

- ABORT causes the spooler to stop printing the current file; the file is left in the spool queue. This command allows the operator to force the printing of all short files followed immediately by the restart of the aborted file. This option also forces a previously issued paper change (see below) to occur immediately.
- DROP causes the spooler to stop printing the current file and to delete this file from the spool queue. The user option -CANCEL will delete a file from the spool queue if it has not started printing.

Changing Paper

Paper type is changed by the following actions:

1. Specify the type of paper by the command

```
SPOOL -PAPER form-name
```

where form-name is a 1-6 character paper type; if specified ' ', the paper type is set to the default value.

The spooler will finish printing the current file, if any, and print the message:

```
ACKNOWLEDGED - SPOOLER BUSY - A MESSAGE WILL BE SENT TO THE
SYSTEM CONSOLE WHEN THE CURRENT JOB IS DONE
```

The message sent to the supervisor terminal is:

```
*** spool-phantom-user-number time
MOUNT FORM 'form-name', TYPE 'GO' TO CONTINUE
```

or

```
*** spool-phantom-user-number time
MOUNT DEFAULT FORM, TYPE 'GO' TO CONTINUE
```

The second message prints if the default paper type is requested.

Note

Once this command has been given the spooler will search the queue for files submitted with that paper type specified, whether or not the paper is actually changed. It is not necessary to change the paper in the printer unless so desired.

2. Following the instructions in the vendor-supplied printer manual, change the paper, and ready the line printer.
3. Set the new paper length, if necessary, by the command:

```
SPOOL -LENGTH number-of-lines
```

where number-of-lines is a positive integer specifying the number of printable lines per page. If the file does not contain printer format control codes (the usual case), this is the page length used.

4. Restart the spooler with the command:

```
SPOOL -GO
```

The spooler will respond with:

```
WAIT... ACKNOWLEDGED
```

The queue will now be scanned for files entered into the queue with the specified paper type (form-name) and only those files will be printed.

Restarting the Spooler

If the spooler has been halted by the operator (by a HANG, FINISH, or PAPER command), the spooler is restarted by:

```
SPOOL -GO
```

If the spooler has been halted by a paper jam, running out of paper, or a similar cause, the operator should first get the printer ready to work and, with the printer off-line, give the command:

```
SPOOL -RESTART
```

The system will respond with

```
WAIT...
```

Place the printer on-line; printing of the current job restarts from the beginning of the file. If it is not reasonable to start at the beginning of the file due to length, etc. the command:

```
SPOOL -BACK
```

instead of SPOOL -RESTART will restart printing at 128-256 lines prior to the current line. The spooler will not necessarily restart on an even page boundary.

Logging Out and Restarting the Spool Phantom

The spool phantom is immediately logged out by the command:

```
SPOOL -LOGOUT
```

The user number must have previously been specified (with the -USER option) if more than one printer is running. Any printing is immediately stopped, but the file is not dropped from the queue.

To restart the spooler:

- Log in under the normal operator name (supplied as SYSTEM)
- Attach to the UFD in which the spooler is to be run (supplied as SPOOLQ)
- Run as a phantom the appropriate command file

```
PHANTOM command-file
```

| <u>device</u> | <u>command file</u> |
|-----------------------------------|---------------------|
| MPC Printer 0 (PR0) | PH_PR0 |
| MPC Printer 1 (PR1) | PH_PR1 |
| Centronics printer | PH_CEN |
| Versatec/Gould printer/plotter | PH_PLT |

Each phantom command file assigns the appropriate device and then runs the spool print program with the proper register settings.

SPOOLER MESSAGES TO OPERATOR

The following messages may appear as a response to a command issued by the operator:

```
ACKNOWLEDGED
```

The command was received and acknowledged by the spool phantom.

```
ACKNOWLEDGED - MOUNT NEW PAPER, TYPE 'GO' TO CONTINUE
```

The spooler has entered "hang" mode, waiting for the operator to mount the form type just specified. To start printing files with the given type, type "SPOOL -GO" at the user terminal.

ACKNOWLEDGED - SPOOLER BUSY - A MESSAGE WILL BE SENT TO THE
SYSTEM CONSOLE WHEN THE CURRENT JOB IS DONE

The PAPER command just issued was acknowledged, but the printer is currently printing a file with the previous form type. When the spool phantom sends the message to the supervisor terminal, the operator should mount the specified form and type "SPOOL -GO" to continue from the user terminal.

ANOTHER REQUEST PENDING

This command could not be honored because there is another command waiting for action by the spool phantom.

MULTIPLE SPOOLERS ACTIVE; USER # MUST BE SPECIFIED

In a system with multiple spool phantoms active, the user issued a command without specifying a user number.

NO SPOOLER

Either no spool phantom is active on the system, or no spooler is active on the specified user number.

PRIVILEGED COMMAND

The user issuing the command was not logged in under the same user name as the spooler phantom to which the command was directed.

SPOOLER IDLE

An ABORT, DROP, RESTART, or FINISH command was issued and the spool phantom was not printing a file.

TIMED OUT

The spool phantom did not acknowledge the command within the time limit. The command remains pending, but the acknowledge or negative acknowledge status is lost.

Examples:

1. Stop printing and drop the current print file from the queue:

```
OK, SPOOL -DROP
GO
WAIT ... ACKNOWLEDGED.
```

```
OK,
```

2. Change the form type to "SZ.132":

```
OK, SPOOL -PAPER SZ.132 -LENGTH 51
GO
ACKNOWLEDGED - SPOOLER BUSY -- A MESSAGE WILL BE SENT TO THE
SYSTEM CONSOLE WHEN THE CURRENT JOB IS DONE
```

```
***49 03'56
MOUNT FORM 'SZ.132', TYPE 'GO' TO CONTINUE
```

```
OK, SPOOL -GO
GO
WAIT ... ACKNOWLEDGED.
```

```
OK,
```

3. Change to default form type:

```
OK, SPOOL -PAPER ' ' -LENGTH 38
GO
WAIT ... ACKNOWLEDGED, SYSTEM BUSY - A MESSAGE WILL BE SENT TO THE
SYSTEM CONSOLE WHEN THE CURRENT JOB IS DONE
```

USER COMMANDS

The SPOOL command format for users is:

```
SPOOL [filename] [option-1] [option-2] ...
```

If a filename is specified, the named file in the home UFD is copied into the spool queue. If filename is omitted (or the -OPEN option is used), a file is created in the spool queue with null contents, and left open on disk file unit 2 (unless otherwise specified by the TUNIT option). The latter case allows users to open a print file and then execute a program which outputs information directly into this file (UPCASE for example). This eliminates the need for a superfluous copy of the spool file to exist in the user's directory.

The user may request that the spooler provide certain internal services when printing the spool file, for example, inserting line numbers in the left margin, using FORTRAN output conventions (column 1 represents carriage control information), etc. Other requests may be specified, for example that printing be deferred (held in the queue without printing) until a certain time of the day, that the file be printed or plotted on a special form type, and/or that it is not to be printed on a remote printer (when a PRIMENET system is configured). These attributes are called print options, and are specified on the command line by one or more mnemonics, each preceded by a dash.

These options are available to any user logged in under any name. Consequently, they are also available to the operator. These user options, summarized here, are completely described in REFERENCE GUIDE, PRIMOS COMMANDS, PDR3108.

| <u>Option</u> | <u>Argument</u> | <u>Function</u> |
|---------------|-----------------|--|
| -CANCEL | printfile | Allows the user to remove a print or plot request from the queue. The name of the print file (PRInnn) is specified on the command line following the option. |
| -DEFER | 'time' | Defers printing of the file until the specified <u>time</u> . The file remains in the spool queue but is ignored by the spool printer phantoms until the specified time. The <u>time</u> may be entered in either 24 hour format (0000 = midnight) or in 12 hour format, with an AM or PM (12:00AM = midnight). The defer print time is requested if not specified on the command line. The <u>time</u> must be enclosed in single quotes. |
| -EXPAND | none | Expands a compressed file before printing. This option is used when the file to be printed has been compressed by the PRIMOS CMPRES command. Output is meaningless if this option is used on an uncompressed file. |
| -FORM | formname | Specifies a special paper type. This file is not printed until the system operator indicates that the requested paper is mounted on the printer. The <u>formname</u> argument must be a 1-6 character name; it does not have to start with an alphabetic character. The default form name is 6 spaces. The SPOOL program prompts for the form type if not supplied on the command line. |
| -FTN | none | Uses FORTRAN output conventions when printing spooled file. Characters in column 1 of each line have the following special significance: <p style="margin-left: 40px;"> l eject to top of page before printing 0 skip 2 lines Space skip 1 line + overprint (on top of last line) available on MPC printers only </p> <p>The spooler will not generate an initial page eject after the banner page is output when this option is specified. This option may not be used in conjunction with -LNUM.</p> |

- FUNIT unit Specifies from-unit number. The user can override the default file unit of 1 (performing I/O operations) while copying or opening a file in the spool queue. This is only useful if the user wishes to avoid disturbing any file previously open on unit 1. The unit argument must be an integer in the range 1-16 and must not conflict with any default or explicit TUNIT specification.
- HOME none Prevents the file from being printed on a remote printer. This option is only useful when the computer is connected to others via PRIMENET.
- LIST option Lists all or selected queue entries.

The information printed includes the login name of the user who submitted the print request, print file name, date and time spooled, print attributes (long/short file, 'home' printer only, etc.), file size if known, file (banner) name if specified, special form type and deferred print time, if applicable.

Should the user wish to restrict the entries printed to those in a certain class (for example, user's print requests only), this may be done by specifying options to the LIST option.

| <u>Option</u> | <u>Lists</u> |
|---------------|--|
| OWN | Files spooled under user login name |
| DEFER | Deferred files |
| PLOT | Files in the plot queue |
| PRINT | File in print queue |
| FORM type | Files for form <u>type</u> ; for default, specify ' ' |
| ALL | Lists all files in queue (default) |

The 'OPTS' column in the queue listing may contain a string of characters, each representing a different attribute of the print file. Following is a list of the characters that may appear in this column:

| <u>Character</u> | <u>Meaning</u> |
|------------------|--|
| H | File will be printed on home printer |
| L | Long file (if implemented at this installation. See Appendix B) |
| P | Plot file |
| S | Short file (if implemented at this installation. See Appendix B) |

| | | |
|--------|--------|--|
| -LNUM | none | Generates line numbers. The spooler will prefix each line of output with a 4 to 5 digit line number enclosed within parentheses and followed by a space. This option may not be used with the FORTRAN print convention option, -FTN. |
| -NOFMT | none | No format control. The spooler supplies no format control (pagination and header generation) as the file is printed. It is the user's responsibility to output the necessary line-printer control codes in the text of the file. |
| -OPEN | none | Specifies open-only operation. Instead of copying the named file to the spool queue, SPOOL creates a new queue file and opens it on unit 2 (exception: see TUNIT). The name, if specified, appears in the queue listing and on the banner page when the file is printed. |
| -PLOT | nwords | Specifies a plot file; <u>nwords</u> is the decimal number of words to be read and output per raster scan. If <u>nwords</u> is unspecified, the default is 128 (for a 200 raster/inch plotter). This spool file is ignored by the line-printer spooler. |
| -TUNIT | unit | Defines to-unit number. This file unit is opened on an open-only operation and used for the output of a copy operation. The notes regarding unit conflict and argument specification described under FUNIT also apply here. |

SECTION 10
SYSTEM CRASHES

GENERAL

A system halt or crash is indicated by the red light on the control panel. This light is directly above the rotary switch. It will light up when the system processor stops.

PRIMOS II

PRIMOS II may be halted by the operator to abort a long listing or to recover from a bad STARTUP command. Such a halt could also be caused by a hardware condition or equipment failure. If the fault is in the hardware, the operator should not attempt to restart the system until the hardware problem has been corrected.

To restart from a halt:

1. Turn the rotary switch to STOP/STEP.
2. Press the MASTER CLEAR switch.
3. Set '170000 in the sense switches (sense switches 1, 2, 3, 4 up).
4. Turn the rotary switch to LOAD.
5. Press the START switch.
6. Reset the sense switches to 0.

Note

If restarting 32K S-DOS or 32K R-DOS, set sense switches to '70000 (sense switches 2, 3, 4 up).

PRIMOS II will restart and respond with:

OK:

PRIMOS III

After the machine halts find the halt address:

1. Turn the rotary switch to STOP/STEP.
2. Place the ADDRESS/DATA switch to ADDRESS. Write the address display in the lights in the logbook.

Use this address to find the symbol corresponding to this halt address in the PRIMOS III loadmap. Then perform the action(s) listed below for that halt. Record the location and the results of the action(s) in the logbook

Note

If PRIMOS III fails to print its introductory message and the machine halts during a cold start, it usually means a memory parity error, BDMEM_.

| <u>Halt Location</u> | <u>Halt Reason</u> | <u>Action(s)</u> |
|----------------------|--------------------------------|------------------|
| AMLCI_ | Spurious AMLC interrupt | R3, W |
| BDMEM_ | Parity error during cold start | S |
| BOOT0_ | Halt after SHUTDN ALL | C |
| MCHK_ | Machine check | R3, W |
| MEMPA_ | Memory parity halt | R3, U, M |
| All others | | R3, C |

The action codes are described later in this section.

PRIMOS IV AND V

After the machine halts find the halt address:

1. Turn the rotary switch to STOP/STEP.
2. Place the ADDRESS/DATA switch to ADDRESS. Write the address displayed in the lights in the logbook.

Determine the segment number of the halt:

1. Turn the rotary switch to FETCH Y.
2. Push the DATA CLEAR switch.
3. Enter '14 in the lights.
4. Place sense switches 1 and 4 up.
5. Place the ADDRESS/DATA switch to DATA.
6. Push START. Write the segment number displayed in in the lights in the logbook.

Using the segment number and address, find the symbolic address corresponding to the halt loction in the PRIMOS IV or V loadmap. Then perform the action(s) listed below for that halt. Record the location and the results of the action(s) in the logbook.

| <u>Halt Location</u> | <u>Halt Reason</u> | <u>Action(s)</u> |
|----------------------|--|------------------|
| AMLCI_ | Bad AMLC Interrupt | D, W |
| BDMEM_ | Parity Error during Cold Start | S |
| BOOT0_ | Halt after SHUTDN ALL | C |
| MCHK_ | Machine check | D, R4, W |
| MEMH2_ | Halt after automatic mapout of bad page | W |
| MEMPA_ | Uncorrected Memory Parity | X |
| All others | | D, C |

The action codes are described later in this section.

ACTIONS FOR PRIMOS III, IV, V

- C Cold Start. Bring the system up as if it had been shut down normally as described in Section 2.
- D Crash magnetic tape dump. Mount a scratch tape on drive unit 1. Push the MASTER CLEAR switch. Restart at address '776 (See Appendix F). The tape will write and then rewind. Label the tape with date and time of crash (and machine, if more than one). Record this information in the logbook. The tape should be turned over to the person whom the System Administrator has designated to examine unusual crashes.
- M Map bad pages out of memory.

1. Perform a warm start (Action W). This runs the memory parity finder program. If bad pages are found, the page number(s) are printed out and are followed by:

PARITY ERROR AT ABOVE PAGES
STOP, FIX MMAP AND PAGEMAPS, WARM START

For each bad page:

2. Record the page number in the logbook.
3. Set the rotary switch to FETCH Y.
4. Press the DATA CLEAR switch.
5. Set the ADDRESS/DATA switch to ADDRESS.
6. Set the value MMAP+badpagenumber in the display lights. The value of MMAP is found in the PRIMOS III load map.
7. Push the START switch.
8. Set the ADDRESS/DATA switch to DATA. The number displayed should be entered in the logbook. This number is a memory location; if it is less than '2400, the system cannot be warm started after locking out the bad pages, but must be cold started. This number will be set in the display lights in step 15 below.
9. Set the rotary switch to STORE Y.
10. Set '177777 in the display lights.
11. Press the START switch.
12. Set the rotary switch to FETCH Y.

13. Set the ADDRESS/DATA switch to ADDRESS.
14. Press the DATA CLEAR switch.
15. Set the memory address value (found in step 8) in the display lights.
16. Set the ADDRESS/DATA switch to DATA.
17. Set the rotary switch to STORE Y.
18. Set 0 in the display lights.
19. Press the START switch.

The bad page is locked out of memory. Repeat this procedure for all listed bad pages.

20. If CUSR=1 or SUPSW<>0, perform a cold start (action C). If CUSR<>1 and SUPSW=0, perform a warm start (action W).
21. Logout the user whose page was in memory when the machine halted with the supervisor command:

LOGOUT -cusr

R3 Record the contents of registers 0 - '37 in the logbook. The procedure is:

1. Set rotary switch to FETCH Y.
2. Set ADDRESS/DATA switch to ADDRESS.
3. Press the DATA CLEAR switch.
4. Press the START switch.
5. Set the ADDRESS/DATA switch to DATA. The contents of Register 0 are now displayed in the lights.
6. Set the rotary switch to FETCH Y+1.
7. Press the START switch. The contents of register 1 are displayed in the lights.
8. Each time the START switch is pressed, the register address is automatically incremented by 1 and the contents of that register displayed. Continue pressing START and recording the register contents until register '37 is reached.

Note

If you lost track of the register address, turn the ADDRESS/DATA switch to ADDRESS. The address of the register being accessed will be displayed in the lights. Be sure to turn the switch back to DATA before proceeding.

- R4 Record the contents of registers '34, '35, and '36 in the logbook. (See Appendix F.) For each register:
1. Turn the rotary switch to FETCH Y.
 2. Push the DATA CLEAR switch.
 3. Enter the register ('34, '35, or '36) in the lights.
 4. Place sense switches 1 and 4 up.
 5. Place ADDRESS/DATA switch to DATA.
 6. Push START. The number in the lights is the high half of the 32-bit register. To display the low half, put sense switch 4 down. The contents of the registers should be recorded in the logbook.
- S Push the START key. If the halt is a parity error during cold start, this action will automatically map out the bad page.
- U User number information. Obtain and record the contents of addresses CUSR and SUPSW, the user number and the supervisor switch at the time of machine halt. The procedure is:
1. Find the numerical address of CUSR and SUPSW from the PRIMOS III load map.
 2. Set the rotary switch to FETCH Y.
 3. Set the ADDRESS/DATA switch to ADDRESS.
 4. Press DATA CLEAR switch.
 5. Set the numerical address for CUSR (or SUPSW) in the display lights.
 6. Press the START switch.
 7. Set the ADDRESS/DATA switch to DATA. Record the contents of CUSR (and SUPSW) in the logbook. These values are needed when locking bad pages out of memory.

W Warm Start.

1. Turn the rotary switch to STOP/STEP.
2. Press the MASTER CLEAR switch.
3. Turn the rotary switch to RUN.
4. Press the START switch twice.

The system should restart and type WARM START at all terminals except the supervisor terminal.

Note

Warm start runs the memory parity finder before restarting PRIMOS. Thus, the warm start may run for twenty seconds or more before the WARM START message appears at the user terminals; no message appears at the supervisor terminal. Do not assume a warm start has failed without waiting twenty seconds and checking the user terminals for the WARM START message.

- X Uncorrected memory parity. Access locations 0, 1, and 2 (with no sense switches set) to obtain the contents of the X-, A-, and B-registers. These are 16-bit registers giving the following information:

| <u>location</u> | <u>register</u> | <u>contents</u> |
|-----------------|-----------------|---------------------------|
| 0 | X | user getting parity error |
| 1 | A | page number |
| 2 | B | offset within page |

Note

If user number is 1 (supervisor) then a Cold Start must be done.

If user number is not 1, press the start key. This will automatically map out the bad page. Following mapout, the system will halt at MEMH2_ and must be warm started (see W).

PART III - THE ADMINISTRATOR

SECTION 11

BUILDING AND EXTENDING THE SYSTEM

This section is an outline of the contents planned for future versions of this IDR. Where available, references have been made to other sections of this guide or to other documents where the information is available. The information on shared libraries appears nowhere else at present and so is presented here in its entirety.

PRIMOS

Versions of PRIMOS

PRIMOS II - see section 17

PRIMOS III - see section 18

PRIMOS IV and V - see section 19

Paging Space

See the sections on PRIMOS III, IV, and V for discussions on calculating paging space requirements. Section 7 on disk formatting explains how to create split disks.

SOFTWARE COMPONENTS

UtilitiesLibrariesShared Libraries

Shared V-mode libraries are available to the user community. The libraries, which are all or partially sharable, are the FORTRAN library (the I/O routines only) and MIDAS for all users; and COBOL and/or FORMS for those users who have purchased these packages.

Use of the shared libraries is indicated when the majority of users on the system are normally using the library routines. Small systems with few users and only one MIDAS user or one COBOL user or where the FORTRAN formatted I/O routines are seldom used, may see no benefit from shared libraries.

Features of Shared Libraries: Each user of shared library routines uses private segment '6001 in addition to the segments otherwise required by programs. Segment '6001 is used for the impure portion of

the shared libraries and represents a reduction in the size of the user's load file but not in the size of the single user working set at run time. This additional segment may be compensated for by a corresponding reduction in the number of segments in the run file. (The MI option of SEG's Loader is used for this. See PTU50 for details of operation).

Several benefits result from using the shared libraries.

- User run files will be smaller, reducing the time required to restore the SEG runfile. User interaction with the program will begin sooner.
- Reduction of system load with respect to private segments and private memory image sizes. If properly used, paging may be reduced. This is of importance for users with many large V-mode programs making extensive use of the shared library routines.
- Installation of a new revision of the library does not necessitate program reloading. Installation of a rebuilt shared library is all that is required to make the modified library available to all users of the shared library.

Installation: Shared libraries occupy segment '2014 and must be installed each time the system is cold started. The runfiles are resident in UFD=SYSTEM on the B1 (or chargeable) partition of the Master disk. Copy these runfiles to UFD=SYSTEM on the system disk. These runfiles are installed at startup time by including the command file C_SHLB in the startup command file. (Section 2) The format is:

```
CO C_SHLB SYSTEM
```

Running this command file installs 8 memory image files in segment '2014 and runs the programs required to inform the operating system that shared libraries are activated. Once the libraries are installed, users with programs loaded using the special shared library object files may run V-mode programs accessing these shared libraries. If the shared libraries are not installed, programs expecting the shared libraries to be resident will get a NOT FOUND message from the operating system whenever an attempt is made to access a shared library routine.

Shared Library Usage: Using the shared libraries requires that programs be reloaded using the special shared library object files in UFD=LIB.

Note

FORMS users must also make a source change to their main program. Except for FORMS there are no other special requirements (see PTU53).

If one of the shared libraries is to be used, all appropriate shared libraries must also be used. If the user wishes to use the shared FORTRAN library and also requires MIDAS or COBOL, the shared MIDAS and COBOL libraries must also be used. After the new V-mode run file has been created, and the shared libraries installed, the user's programs may be run as before.

Administration: The shared libraries files, in UFD=LIB, are SFTNLB, SKDALB and, for those purchasing COBOL and/or FORMS, SCOBLB and SFORMS. If the shared libraries are to be used system-wide, the Administrator should rename these modules as follows:

SFTNLB to PFTNLB
SKDALB to VKDALB
SCOBLB to VCOBLB - if available on the system
SFORMS to VFORMS - if available on the system

If the shared libraries are not to be used system-wide, then those users planning to use them must modify their command files to use the special library files. In particular SEG's Loader command:

LI SFTNLB

must be included immediately prior to the IL command.

Rebuilding and Re-installation: Each of the shared libraries is represented by a set of runfiles and an installation program. If one of the libraries must be replaced it is necessary to rebuild that library only. These command files put all the necessary files into UFD=SYSTEM so that installation is easily accomplished by running the command file C_SHLB in that UFD.

CAUTION

A library should not be replaced while being used. As programs using the shared libraries execute, links are made to the appropriate shared library routines in such a way that altering the memory image in use by the program can cause random and unpredictable behavior. Changing a shared library (replacing its memory image in segment '2014) has the effect of making such an alteration to the user's memory image. Install new shared libraries only when bringing up the system with a cold start.

It is perfectly safe to replace the memory image files in UFD=SYSTEM at any time as these are only loaded into memory when the explicit SHARE commands are given.

MIDAS

V-mode features

Spooler

See section 9 for operation, and appendix B for installation options.

Phantom Users

Versions of SEG

SYSTEM SECURITY

Adding to UFD=CMDNCØ

See section 4

Adding new directories to the MFD

See section 4

Protecting the packname file (DSKRAIT)

SECTION 12

SYSTEM RESOURCE ALLOCATION

This section is an outline of the contents planned for future versions of this IDR. Where available, references have been made to other sections of this guide or to other documents where the information is available.

NUMBER OF USERS

Terminal

Remote

Phantom

USER SEGMENTS

SHARED SUBSYSTEMS

COMMAND UFD (CMDNC0)

PARTITIONING DISKS

SECTION 13

DEFAULTS AND PARAMETERS

This section is an outline of the contents planned for future versions of this IDR. Where available, references have been made to other sections of this guide or to other documents where the information is available.

TRANSLATORS

FORTRAN - FTN

COBOL

ASSEMBLER - PMA

RPGII

LOADERS

R-Identity - LOAD

V-Identity - SEG

SYSTEM AND NETWORK PARAMETERS

Inactivity Timeout

Users logged in at a terminal but inactive are automatically logged out after N inactive minutes, where N is a system configuration parameter. A user is considered inactive if the system is waiting for a PRIMOS command, or if a user program is waiting for either terminal input or card reader input. The specified number N is an installation parameter. If an inactivity logout occurs, the logout message is preceded by the word TIMEOUT.

The parameter N is set by changing LOUTQM, which is the first variable in the common area FIGCOM in subroutine TMAIN.

Under PRIMOS III, this change is made using the TAP utility. Under PRIMOS IV or V, the inactivity logout time is changed with the configuration data file LOGOUT command.

SECTION 14

USAGE ENHANCEMENTS

This section is an outline of the contents planned for future versions of this IDR. Where available, references have been made to other sections of this guide or to other documents where the information is available.

EXTERNAL LOGIN PROGRAMOverview

When a user of the system LOGS IN or LOGS OUT, any existing program LOGIN in UFD=CMDNCØ is RESUMEd. This program may be custom-written by a given installation to perform special LOGIN/LOGOUT functions, such as accounting or restricting system access.

The program is RESUMEd with .BREAK. inhibited so as to prevent the user from exiting LOGIN via CONTROL-P or .BREAK.. The LOGIN program should perform a CALL BREAK\$(.TRUE.) before exiting.

Note

The BREAK\$ subroutine maintains a BREAK inhibit counter, rather than a single flag. External login programs (which are entered with BREAKs inhibited) should enable breaks before calling EXIT. That is, they should always enable BREAKs one time more than the number of inhibits; otherwise the user will be logged in with BREAKs inhibited. (If this happens, causing any error will re-enable BREAKs.)

The command line that called LOGIN is available via the library subroutines RDTK\$\$ and CMREAD. Accounting information is available via the library subroutine TIMDAT. The LOGIN program exits via a call to the library subroutine EXIT and must not encounter any uncontrolled errors that result in EXIT being bypassed.

Writing an External Login ProgramSample External Login Program

MANAGING DISK STORAGE

Monitoring for Charges with FIXRAT, FUTIL

FIXRAT - see Section 6 for operation

FUTIL - see REFERENCE GUIDE, PRIMOS COMMANDS, PDR3108

What to do When Partition is Full

Looking at partitions - Section 5

Deleting files - Section 4

Creating new partitions - Section 7

CHANGING CONFIGURATION WHEN ADDING MEMORY

RUNNING SPOOLER ACROSS THE NETWORK

PART IV - REFERENCE

SECTION 15

PRIMOS COMMANDS FOR THE OPERATOR

This section describes commands that are normally used by the operator at the PRIMOS III, IV, or V supervisor terminal. Some of these commands may also be available for use under PRIMOS II; if so, this is indicated in the command description.

PRIMOS III, IV and V INTERNAL COMMANDS

After PRIMOS III, IV, or V is started, the PRIMOS II supervisor terminal becomes the PRIMOS III, IV, or V supervisor terminal. Unless the USRASR or OPRPRI command is given, the supervisor terminal can be used only for a restricted set of operator commands (labelled as supervisor in this section). Under PRIMOS III, attempts to invoke external commands and/or RESTORE, RESUME, and START will generate an error message. External commands and RESTOR, RESUME, and START may be invoked from the supervisor terminal under PRIMOS IV or V. Normally, the supervisor terminal is used only to STARTUP and SHUTDN disks, configure devices, check status, and to collect a record of LOGIN, LOGOUT, and other messages.

CAUTION

The supervisor terminal must remain attached to CMDNC0 in order that external commands may be invoked from user terminals.

 * ADDISK *

(PRIMOS III, IV, V - supervisor)

The command format is:

ADDISK pdev-1 [pdev-2] ... [pdev-n]

ADDISK searches the table of logical devices for an available table entry location; if one is found, ADDISK starts up the device specified by pdev. ADDISK repeats this procedure for each argument specified. If a device specified by pdev is already started, no action is taken. If there are no free logical devices, the error message:

NO ROOM

is printed at the supervisor terminal. Logical device numbers may be freed by using SHUTDOWN to remove devices from the table. If a non-existent or not-ready device is specified the message:

DISK pdev NOT READY

will be printed at the supervisor terminal or, if the device is ASSIGNED, the terminal of the user to whom assigned.

PRIMOS III, IV, and V support 18 logical devices; since STARTUP will start up at most 8 logical devices, ADDISK must be used to start up devices in excess of 8.

Networks

Devices on other systems on the network may be made available to users on the local system by the command:

ADDISK node-name pdev-1 [pdev-2]...

node-name is the network name of the system on which the physical devices are located.

pdev-n are the physical device numbers of the device on the remote system which are to be made available to local users. These devices must be connected to their local system by STARTUP or ADDISK before being made available to any other system.

 * AMLC *

(PRIMOS III, IV, V - supervisor)

The command format is:

AMLC [protocol] line [configuration] [lword]

protocol

The following protocols are available:

TTY - normal terminal protocol
 TTYHS - terminal with per-character interrupt
 TRAN - transparent (no character conversion)
 TRANHS - TRAN with per-character interrupt
 TTYNOP - ignores all traffic

Terminal protocol is used by lines controlling interactive terminals. With terminal protocol, all input from the terminal is echoed if the line is set for full duplex, and a CARRIAGE RETURN - LINE FEED is echoed following CARRIAGE RETURN. Bit 8 of each character (the ASCII code parity bit) is forced on. CONTROL-P or BREAK are interpreted as a QUIT if the terminal is connected to PRIMOS as a user terminal. If connected to PRIMOS as an assigned AMLC line, CONTROL-P or BREAK are not interpreted. A CARRIAGE RETURN input by the terminal is transmitted as a NEW LINE to the program requesting input. Input is no longer echoed if the line input buffer becomes full. Terminal protocol is identical to the protocol used to control PRIMOS III, IV, and V user terminals.

Transparent protocol is used by lines connected to peripheral devices or other computers. With transparent protocol, no input is echoed, no response is made to the input of a line feed or carriage return, and there is no transformation of carriage return to line feed.

The high-speed protocols (TTYHS and TRANHS) are used by lines connected to peripheral devices that can run at greater than standard terminal speeds. These protocols are the same as those described above with one exception: for output only, the line's character time interrupt flag is set when the output buffer contains more than 40 characters, and it remains set until the output buffer contains fewer than 40 characters. The protocols have a burst mode effect on the output device.

With a line using the high-speed protocols, a drastic increase in system overhead can result depending upon the Baud rate and the number of lines in the group. The user must be careful not to assign protocols to lines that normally have their character-time-interrupt flag always set; as, for example, the last line in each group of lines. If the protocol is not given, the transparent protocol is assigned by the operating system. The line number is an octal integer less than or equal to the number of terminal users (ntusr) specified by the CONFIG command. (See Figure 15-1).

line

The AMLC line number, is a number from 0 to the maximum allowed value.

config

The config argument which sets the line configuration, is an octal number that corresponds to the bit pattern illustrated in Figure 15-1. For example, '2213 would specify a data set at 300 Baud with parity disabled and 8-bit character length.

lword

This optional parameter is an 16-bit octal integer constructed as follows:

Bit

| | | |
|------|-------|---|
| 1 | Set | Line is half-duplex |
| | Reset | Line is full-duplex |
| 2 | Set | Don't echo LINE FEED for RETURN |
| | Reset | Do echo LINE FEED for RETURN |
| 3 | Set | Recognize X-OFF (CTRL S or '223) |
| | Reset | Don't recognize X-OFF (CTRL S or '233) |
| 9-16 | --- | Number of user to which AMLC is connected |

The user number is the number printed at the terminal upon LOGIN or LOGOUT, or printed by the STATUS command indicating user number. STATUS prints this number as a decimal value but its octal equivalent must be used in the AMLC command. If the rightmost eight bits (9-16) of lword are zero, the AMLC line is not associated with any user space and is available to be assigned.

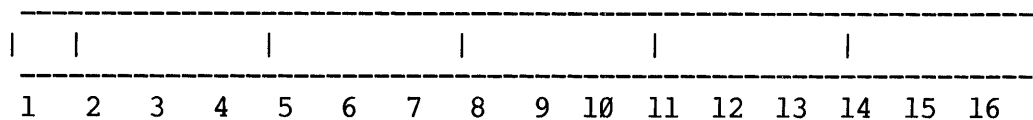
Terminal characteristics such as FULL DUPLEX/HALF DUPLEX, XOFF, etc. may be set at the user terminal with the PRIMOS-level TERM command.

The AMLC command may be used to configure ASSIGNED AMLC lines as well as terminal AMLC lines. After the system is running, users may assign/unassign the AMLC lines through the following commands:

```
ASSIGN AMLC [protocol] line [configuration]
```

```
UNASSIGN AMLC line
```

Where the parameters are the same as for the AMLC command.



| <u>Bits</u> | <u>Assignment</u> | |
|-------------|-----------------------------|-------------------------------|
| 1-4 | Line number (bit 4 is LSB) | |
| 5 | Not used | |
| 6 | Data Set Control Bit (1=on) | |
| 7 | 1=Loop line, 0=do not loop | |
| 8-10 | <u>Octal Value</u> | <u>Line Speed (Data Rate)</u> |
| | 0 | 110 Baud |
| | 1 | 134.5 Baud |
| | 2 | 300 Baud |
| | 3 | 1200 Baud |
| | 4 | Not Assigned* |
| | 5 | 75 Baud* |
| | 6 | 150 Baud* |
| | 7 | 1800 Baud* |

Note

Speeds marked with * are assignable by hardware jumpers. The speeds shown are the default values. Other choices are 75, 150, 600, 1800, 2400, 4800, 9600 or 19200 Baud.

| | |
|-------|-----------------------------------|
| 11 | Not used |
| 12 | 0= 1 stop bit, 1=2 stop bits |
| 13 | 0=enable parity, 1=disable parity |
| 14 | 0=odd parity, 1=even parity |
| 15-16 | Character length: |
| | 0 0 5 bits |
| | 1 0 6 bits |
| | 0 1 7 bits |
| | 1 1 8 bits |

Figure 15-1. Setup of Line Configuration

 * CHAP *

(PRIMOS III, IV, V -- supervisor)

CHAP is an internal operator command that changes a user's timeslice and, under PRIMOS IV or V, the priority level in the ready list. The format of the command is:

$$\text{CHAP} \left\{ \begin{array}{l} \text{-userno} \\ \text{ALL} \end{array} \right\} [\text{priority}] [\text{timeslice}]$$

where:

userno is the number of the user to be modified in the form -nn

priority is the priority level to be assigned to the user in the form of an integer from 3 (highest priority) to 0 (lowest priority). If omitted, the priority is left unchanged. (Under PRIMOS III, this parameter is ignored.)

timeslice specifies a new timeslice value in tenths of a second

If specified as 0, the timeslice is reset to the standard value. If omitted, the timeslice is left unchanged. If both priority and timeslice are omitted, the user's priority is reset to 1 (the level on which users normally run), and the timeslice is reset to the standard value. If ALL is specified instead of -userno, the specified changes are made to all users. The default value is '24 (20 decimal)- a two-second timeslice.

Note

The level and timeslice for user 1 cannot be modified.

 * CONFIG *

(PRIMOS III, IV, V -- supervisor)

The CONFIG command defines system parameters and defaults that are specified once per system session. The CONFIG command is disabled after its first used during a session. The CONFIG command in all its forms is fully described in Section 16.

* COPY *

(PRIMOS II, III, IV, V)

COPY is an external command that copies one disk to another and verifies the copy. It is fully described in Section 8.

* DISCON *

(PRIMOS III, IV, V - supervisor)

DISCON is an external command which disconnects a terminal from a remote process on the network. The format is:

DISCON user-number

user-number is the user number of the terminal. It may be determined by the STATUS USERS command.

If user-number corresponds to a local terminal using a remote process, the terminal will be reconnected to its presently dormant process. If user-number corresponds to a local process in use by a remote terminal, the process will be logged out and returned to the pool of free remote login processes.

 * DISKS *

(PRIMOS III, IV, V - supervisor)

The command format is:

DISKS [NOT] pdev-0 [pdev-1] ... [pdev-7]

The DISKS command may be given only from the supervisor terminal. The DISKS command adds or removes the specified physical disk(s) to/from the Assignable Disks Table. pdev-0 ... pdev-7 are physical device numbers. No more than ten disks may be entered into the Assignable Disks Table. A physical disk number must be specified in this table before a user can invoke the ASSIGN command to assign that disk.

When the optional argument NOT is specified in the DISKS command line, the subsequently specified physical disks are removed from the Assignable Disks Table. Removing a physical disk number from the table does not cause the disk to be unassigned; the operator must give the UNASSIGN command to unassign a disk from a user.

Example:

DISKS 1 20250 50250 60250 70250 10020 110250 20252

adds the specified physical disks (disk 1) and partitions (20250, etc.) to the Assignable Disks Table. These disk and partitions may now be ASSIGNED by the users or operators. The command sequence:

DISKS NOT 20250
 UNASSIGN 20250

removes the physical disk partition 020250 from the Assignable Disks Table and unassigns that partition.

 * ELIGTS *

(Supervisor - PRIMOS IV, V)

ELIGTS is an internal command which modifies the eligibility time slice for system users. The format is:

ELIGTS tenths

where tenths is the time, in tenths of a second (in octal), that a user will run before being placed on the eligibility scheduler queue. The default value is 3 (about 1/3 second).

The default user time-slice is 2 seconds, but a user will not remain on the ready list for this interval. When the ready list slice (default=1/3 second) of CPU time has been used, a user is moved from the ready list to another scheduler queue, the eligibility queue, and the user time slice is decremented by the ready time slice. This queue is checked by the scheduler after checking for interactive users (on the high priority queue) and before checking the low priority queues. Users cycle between the ready list and the eligibility queue until user time slice is exhausted, at which time the user is entered in the low priority queues.

 * FIXRAT *

(PRIMOS II, III, IV, V)

FIXRAT is an external command that checks the PRIMOS file integrity on any physical device or partition. It is described fully in Section 6.

 * LOGPRT *

(PRIMOS II, III, IV, V)

LOGPRT is an external command that prints the contents of the event logging file. It is described in detail in Section 4.

 * LOOK *

(PRIMOS IV, V - supervisor)

LOOK is an internal operator command that provides access to any segment in the system. The format of the command, which must be preceded by an OPR 1, is:

LOOK [-userno] [segno] [access] [mapseg]

userno is the number of the user owning the segment and defaults to user 1

segno is the number of the segment to be examined. It defaults to '6000 (the Ring 0 stack segment for the user)

access specifies the access rights to be granted as in the SHARE command and defaults to '200 (read-only)

mapseg is the segment of user 1's address space into which the specified segment is to be mapped. The default is '4001.

This command is intended mainly for the use of systems engineers and field analysts as a debugging tool. The operator and administrator will normally have no use for it.

 * MAKE *

(PRIMOS II, III, IV, V)

MAKE creates a structure for any PRIMOS-supported disk pack or partition. It is described fully in Section 7.

 * MAXSCH *

(PRIMOS IV, V - supervisor)

MAXSCH is an internal command controlling the amount of overlapped processing performed by the system. It controls the number of processes on the ready queue and fast I/O wait queues. The format is:

MAXSCH n

where n is the value of the variable MAXSCH in SUPCOM; the default value is 3.

 * MAXUSR *

(PRIMOS IV, V - supervisor)

MAXUSR is an internal command that controls the number of users that are allowed to be logged into PRIMOS IV or V at any time. Normally, this variable is set to the number of configured users. When it is necessary to limit the number of users on the system, the command

MAXUSR n

can be used. Thereafter, no more than 'n' users will be allowed to login. If the number of users is already above 'n', no user will be forcibly logged out, but no new users will be allowed to login and no phantoms will be started. Users attempting to login or start a phantom will receive the error message:

MAX NO. USERS EXCEEDED

There are two situations in which this facility will be useful:

- o At cold start, if there are modifications to be made to PRIMOS, setting MAXUSR to 1 or 2 will prevent users from logging in until the operator decides it is appropriate. (The MAXUSR command should be entered before setting the date and time.)
- o When the system is to be shut down, setting MAXUSR to zero will prevent new logins just before the shutdown.

 * MESSAGE *

(PRIMOS III, IV, V)

MESSAGE is an internal command with the following facilities:

From any user terminal -

- Send a message to the supervisor terminal.

From the supervisor terminal -

- Send a message to all users on the local node.
- Send a message to a specified user on the local node.
- Send a message to a supervisor terminal at a different network node.

The format for user-to-operator message is:

```
MESSAGE  

text-of-message
```

text-of-message is a one-line message. Two lines are printed at the supervisor terminal:

```
*** uu hh'mm  

text-of-message
```

where: uu is the user number; hh'mm is the time of day in hours and minutes.

The format of an operator-to-user message is:

```
MESSAGE ALL [NOW]  

text-of-message
```

```
MESSAGE -uu [NOW]  

text-of-message
```

```
MESSAGE node-name  

text-of-message
```

The operator can send messages to all users or to a single user. When the parameter ALL is specified, the message is sent to all users. The parameter -uu is a dash followed by the user number. When the parameter -uu is specified, a message is sent to the user specified.

If the optional argument NOW is not specified, the message is stored in a Broadcast Buffer (ALL) or a Single User Buffer (-uu). The message is printed at the user's terminal when that user returns to PRIMOS III,

IV, or V command level. A message that is in the Broadcast Buffer is also printed after a user issues the LOGIN command.

If the argument NOW is specified, the message is printed immediately. This is undesirable if the user is in the middle of a sensitive operation. When NOW is specified, stored messages are not affected. Also, when NOW is specified, the format of the message at the user terminal is:

*** BULLETIN ***

If the operator attempts to send a message to a single user before the previous message to a single user has been received, the error line:

MESSAGE NOT SENT

is printed at the supervisor terminal.

If node-name is included after MESSAGE, the message will be printed at the supervisor terminal of that node of the computer network. If the node-name given is not connected to the local node then the error message ILLEGAL NAME will be printed after the operator enters the message to be sent.

Note

The node name of the originating system is not printed at the receiving system. The sending operator should include the originating node-name as part of the message text.

To cancel a stored message, a null line must be entered as the text of message.

 * OPRPRI *

(PRIMOS III, IV, V - supervisor)

OPRPRI is an internal command which allows the supervisor terminal to issue certain commands. The format is:

$$\text{OPRPRI} \left\{ \begin{array}{l} 1 \\ \emptyset \end{array} \right\}$$

- 1 Removes protection and allows external commands (and START, RESTOR, RESUME, LOGIN, LOGOUT) to be issued at supervisor terminal (PRIMOS III). Under PRIMOS IV or V, OPRPRI 1 allows the SHARE command to be given.
- \emptyset Resets protection against issuing commands; this is the default.

 * SETIME *

(PRIMOS III, IV, V - supervisor)

The SETIME command sets date and time. It can be entered at any time during system operation. The format is:

SETIME -mmddy -hhmm

where mmddy are digits that represent the month, day, and year (last two digits), and hhmm are digits that represent the time in hours and minutes. The two arguments to SETIME must be separated by spaces and start with a dash as the first character.

Example:

SETIME -050178 -1630

sets the date and time: May 1, 1978, 4:30 PM.

 * SHARE *

(PRIMOS IV, V - supervisor)

SHARE is an internal command for the installation of a command or library into a supervisor segment. The SHARE command can be issued only by the system operator and must be preceded by an OPR 1 command and followed by an OPR 0 command. The principal use of the SHARE command is to make shared procedures available to all users on the system. Segments '2000 to '2037 are available to hold shared programs. Those shared programs can be executed by any user.

It is also possible to change the access rights of supervisor segments '1 to '12. By changing access rights of specific supervisor segments, it is possible to either monitor or patch the supervisor from a user terminal. This feature is primarily for PRIMOS development and debugging; for users, it is dangerous and not recommended.

The command format is:

SHARE [filename] segment-number [access-rights]

filename An optional parameter naming a runfile in the current UFD. If filename is specified, the named file is restored into segment-number. If omitted, the command is being used to change access-rights.

segment-number The number of the segment to be shared. Valid segment numbers are from '1 to '3777, however, only segments '1-'12 and '2000-'2037 should be specified. Specification of a segment number outside these ranges will cause unpredictable results.

access-rights The access rights to be given segment-number. Possible values are:

'0 No access.
 '200 Read access.
 '600 Read and execute access.
 '700 Read, write, and execute access.

The default, if no access rights are specified, is '600.

CAUTION

The gate segment, segment 5, should not be specified in a SHARE command. If it is given anything but '100 access rights, direct-entrance calls from user space will cause ACCESS VIOLATION messages.

 * SHUTDOWN *

(PRIMOS II, III, IV, V - supervisor)

The SHUTDOWN command performs tasks necessary to shutting down the PRIMOS III, IV, or V system in an orderly manner. The format is:

$$\text{SHUTDOWN} \left\{ \begin{array}{l} [\text{pdev-0} [\text{pdev-1} \dots \text{pdev-n}] \\ \text{ALL} \end{array} \right\}$$

SHUTDOWN ALL performs a complete PRIMOS III, IV, or V system shutdown. All user files are closed, physical disks are disconnected from the system, and the PRIMOS III, IV, and V systems shut down by inhibiting interrupts, exiting page mode, stopping the system clock, and halting. SHUTDOWN with no arguments specified performs the same functions for PRIMOS II. Under PRIMOS III, IV, or V, the SHUTDOWN ALL command from the supervisor terminal causes the prompt:

REALLY?

The operator must answer YES for the shutdown to occur.

If the SHUTDOWN command is issued with a list of physical devices (pdev-0 ... pdev-n), the listed devices are closed by closing all files opened in the listed devices and by detaching all users attached to the listed devices. The specified devices are no longer available for PRIMOS file I/O operations until they have been specified in a subsequent STARTUP command. The STATUS command can be used to list the devices currently started up. Unlike the STARTUP command, the pdev's do not have to be given in logical drive order.

CAUTION

Do not shut down the physical device associated with logical 0, or PRIMOS III, IV, and V will lose the command directory (from memory, not disk). To recover, STARTUP the disk and ATTACH CMDNC0.

Note

If the shutdown is a normal one, we recommend using the LOGOUT ALL command before SHUTDOWN ALL.

Example of Selective Shutdown

Assume the initial STARTUP command was:

```
STARTUP 51 50 56 54
```

To replace the pack on physical device 56 (logical device 2): give the command SHUTDOWN 56, stop the drive, replace the pack, and restart the drive. To start up physical device 56 as logical device 2 give the command:

```
STARTUP 2/56
```

Networks

Remote devices connected to the system can be disconnected by the command:

```
SHUTDOWN node-name pdev-1 [pdev-2]...
```

node-name is the network name of the system on which the devices are physically mounted.

pdev-n are the remote devices, made available to the local system, which are to be disconnected.

Notes

If a device is shut down at its local system it is no longer available for use on any other system in the network.

A SHUTDOWN ALL command disconnects all local disks and remotely available disks. It does not shut down the remote disks on the system on which they are physically mounted.

 * SPOOL *

(PRIMOS III, IV, V)

SPOOL is an external command invoking the spool queue management utility for the system printer. It is described fully in section 9.

 * STARTUP *

(PRIMOS II, III, IV, V - supervisor)

The STARTUP command defines a list of physical devices to be used by PRIMOS. A device is considered started if it has been mentioned in a previous STARTUP command. Additional devices may be started if the new list in a subsequent STARTUP command does not conflict with the list in a previous STARTUP, and if no user has assigned a disk specified in the list. The format is:

STARTUP comdev [pdev-1]...

where comdev and pdev-1... are items in a list of physical disk (device) numbers. The argument comdev must be specified in the initial STARTUP command; the remaining device numbers are specified optionally. The order of the list defines the logical number sequence of the devices (e.g., comdev is logical 0, pdev-1 is logical 1, etc.)

comdev must match the comdev specified in the CONFIG command.

Example:

```
STARTUP 52 53 54
```

defines that physical devices 52, 53, and 54 are to be used with PRIMOS III, IV, and V and associates the following logical device numbers with the physical device numbers specified: 52 is logical 0; 53 is logical 1; and 54 is logical 2. In PRIMOS III, IV, and V logical device numbers may also be specified as arguments to the STARTUP command. When used in this manner, they must be followed by a slash and the associated physical device number. Examples:

Examples:

```
STARTUP 0/52 1/53 2/54
```

```
STARTUP 4/100250
```

If a non-existent or not ready device is specified the message:

```
DISK pdev NOT READY
```

will be printed at the supervisor terminal or, if the device is assigned, at the terminal of the user to whom assigned.

Networks

Devices on other systems on the network may be made available to users on the local system by the command:

```
STARTUP node-name pdev-1 [pdev-2]...
```

node-name is the network name of the system on which the physical devices are located.

pdev-n are the physical device numbers of the devices on the remote system which are to be made available to local users. These device must be connected to their local system by STARTUP or ADDISK before being made available to any other system.

Note

Only 8 logical devices can be connected by STARTUP; additional devices must be connected with the ADDISK command. ADDISK is normally preferred to STARTUP.

 * STATUS *

(PRIMOS II, III, IV, V)

STATUS is an internal command for monitoring system usage. In its full form at the supervisor terminal, it prints network node name, PRIMOS file units open, physical devices connected to the system (packname, logical device number, physical device number, network node), status of network nodes, the paging and command devices, and logged in users (login name, user number, line number, devices and peripherals). Subsets of this information can be obtained by appending keywords (DISKS, USERS, etc) to the STATUS command (see below). If the STATUS command is given at a user terminal slightly different information will be obtained (see REFERENCE GUIDE, PRIMOS COMMANDS, PDR3108).

Some typical instances where the STATUS command must be used are:

- Prior to mounting a new disk pack to determine what physical disk assignments are available.
- After a request that all users release a given disk or disks, to determine that they have done so before shutting down that disk or disks.
- As a check that all users have logged out before shutting down PRIMOS III, IV, or V. (No harm to the system results if the users of a particular disk are still logged-in when the disk or the system is shut down. However, the user's files are closed and a message is printed at the terminal to that effect.)

The STATUS ALL command prints all the system information shown in Figure 15-2. (At the supervisor terminal the command STATUS will print out the same information as STATUS ALL.)

Description of Status Information

- I The user will always be SYSTEM followed by the local node-name, if any. (e.g., SYSB)
- II Prints a list of all PRIMOS file units currently open. In this example there are no open file units.
- III Column 1 (DISK) is the packname of the disk or partition, i.e., the name of DSKRAT file. Column 2 (LDEV) is the logical device number associated with the physical device by the STARTUP command. Logical device 0 must be the command device; the paging device or partition is not included in this list (see V) as it is not directly accessible by user or operator. Column 3 (PDEV) is the physical device number indicating type of device, drive unit, partition size, and offset (see Appendix G). The fourth column (SYSN) tells the node of the network on which the disk is physically mounted. A blank in this column means the

OK, STATUS ALL

USR=SYSTEM SYSB

I

} UNITS

FUNITS

II

| DISK | LDEV | PDEV | SYSN |
|--------|------|-------|------|
| SOFTWR | 0 | 3462 | |
| SPOOLA | 1 | 10460 | |
| PERIPH | 2 | 460 | |
| CPUGRP | 3 | 21460 | |
| DOCUMN | 4 | 50460 | |
| DBTEST | 5 | 71063 | |
| TRANS | 6 | 21460 | SYSD |
| M142A1 | 7 | 464 | |
| M142B1 | 10 | 10465 | |
| DBGPR | 11 | 51060 | SYSD |
| TEST | 12 | 71061 | SYSD |

III

} DISKS

IPC NETWORK

IV

| NODE | STATE |
|------|-------|
| SYSB | **** |
| SYSC | UP |
| SYSD | UP |

} NETWORK

PAGDEV = 100461 COMDEV = 3462

V

| USER | NO | LIN | PDEVS |
|--------|----|-----|------------------------|
| MASSE | 4 | 2 | 3462 |
| STEF | 14 | 14 | 3462 |
| VMS | 27 | 31 | 51060 (SYSD) |
| TEKMAN | 31 | 35 | 50460 |
| GOUDY | 32 | 36 | 50460 |
| CRABB | 34 | 40 | 21460 |
| JAWS | 35 | 41 | 3462 |
| BURLEY | 38 | 44 | 3462 |
| FLOWER | 44 | 75 | 3462 (FROM SYSC TTY#3) |
| SYSTEM | 46 | 77 | 10460 |
| FAM | 47 | 77 | 3462 10460 (2) |
| SYSTEM | 49 | 77 | 10460 PR0 |
| SYSTEM | 50 | 77 | 3462 10460 |
| SYSTEM | 51 | 77 | 10460 |

VI

} USERS

OK,

Figure 15-2. Information Printed by STATUS Command At Operator Terminal

disk is a local one; a node name in this column shows that this is a remote disk mounted at that node of the network.

- IV There are two types of networks: IPC and SMLC; this section indicates which type is in use. (Under PRIMOS IV and V more than one may be in operation simultaneously). The node name is given and the state of that node, either UP (in operation) or DOWN (not in operation). The local node is indicated by ****.
- V Specifies the paging device and command device physical device numbers. The command device is that started up as logical device 0 (see CONFIG and STARTUP).
- VI Lists users currently logged into the system. Column 1 (USER) is the login directory name of the user. Column 2 (NO) is the user number; this is a decimal number and is usually the line number plus 2. Column 3 (LIN) is the AMLC line number of the user terminal (octal). Specially assigned line numbers are:

| LIN | Meaning |
|-----|--|
| 75 | user logged in remotely from another node in the network |
| 76 | user is the supervisor terminal using USRASR command |
| 77 | all phantom users |

Column 4 (PDEVS) lists all disks and assigned devices in use by a particular terminal. A disk is considered to be in use (under PRIMOS III, IV, or V) if it contains the user's home UFD, current UFD, or if the user has opened any files on that disk. Currently assigned devices are indicated after disks used with the same device abbreviation used by the ASSIGN command (e.g., PR0, CR1, MT2, etc.) except that assigned disks are shown by DISK physical-device, and assigned AMLC lines by AL line-number. For example, user number 49 (SYSTEM) is running as a phantom and has the line printer (PR0) assigned; this is the spooler.

Other information shown in this column is:

- remote login to another system on the network (No example here).
- remote login from another system on the network (FLOWER is logged into SYSB from terminal 3 on SYSC).
- user priority (FAM is running at priority 2). Normal user priority is 1; a priority of 1 is not printed.

- use of a remote disk (VMS is using a disk on SYSD).
(See CHAP.)

Partial Information

Subsets of the total status information may be obtained by appending key words to the STATUS command.

| <u>Command</u> | <u>Sections printed</u> |
|----------------|-------------------------|
| STATUS DISKS | III |
| STATUS NETWORK | IV |
| STATUS UNITS | I, II |
| STATUS USERS | VI |

Note

Section V is printed only by the STATUS ALL (or STATUS) command.

* USRASR *

(PRIMOS III, IV, V - supervisor)

The USRASR command allows the supervisor terminal to act as a user terminal by associating it with a different address space. After initiating USRASR, it is still possible to invoke supervisor commands after associating with a user. The format is:

USRASR usrho

where usrno is a user number.

Example:

USRASR 4

Restrictions

The USRASR command works only if the associated communications line is not enabled on the AMLC. The communications line can be disabled with the AMLC command:

AMLC TTYNOP 2
USRASR 4

The AMLC command disconnected the user normally attached to AMLC line 2.

Return

To return to operations as a normal supervisor terminal, type:

USRASR 1

SECTION 16

USING THE CONFIG COMMAND

INTRODUCTION

System initialization, configuration, and system parameter definition are performed by the CONFIG command. This command is given once per system session and is then disabled. Under PRIMOS III, CONFIG is a single-line command capable of specifying up to 9 system parameters. Under PRIMOS IV and V, the CONFIG command optionally reads commands from a data file and can define many more system features. The single-line CONFIG command may also be used under PRIMOS IV or V but cannot specify as many features as the extended version of CONFIG.

SINGLE-LINE CONFIG COMMAND

The command format is:

```
CONFIG [myname] ntusr pagdev comdev [maxpag] [altdev] [namlc] [npusr]
      [nrusr] [smlc]
```

- | | |
|---------------|--|
| <u>myname</u> | A character (6 characters maximum) specifying the network node name of the local system. |
| <u>ntusr</u> | An octal integer defining the number of terminal users, including the supervisor (e.g., for a 4-user system, <u>ntusr</u> = 5, for a 7-user system, <u>ntusr</u> = 10). |
| <u>pagdev</u> | A physical device number specifying the device to be used for paging. (See Appendix G for possible values.) |
| <u>comdev</u> | A parameter that specifies the physical device number initially assigned as logical device 0. This is the device on which UFD=CMDNC0 is searched when a user invokes an external PRIMOS command. If <u>comdev</u> and <u>pagdev</u> are the same, the disk is considered to be split into a file system and a paging part. The boundary between the partitions is defined by the DSKRAT header, and it may be set by the MAKE program. |
| <u>maxpag</u> | An optional parameter defining available physical memory storage. It corresponds to the last sector number (octal) to be used. The values for <u>maxpag</u> and associated storage used are as follows: |

| | |
|------------|---|
| Blank or 0 | All of Memory up to 256K; at least 64K ('100K) |
| '200 | 64K ('100K) |
| ... | ... |
| '1000 | 256K ('400K) |

altdev CONFIG may specify either one or two disk devices on which paging is to take place. The alternate paging device cannot be a split disk.

namlc An optional parameter defining the number of assignable AMLC lines.

npusr An optional argument defining the number of phantom users.

nrusr An optional parameter specifying the number of remote users who can run on the system. (Not supported under PRIMOS III)

smlc An optional parameter enabling the SMLC. If used under PRIMOS III, specify as 10/smlc.

Example:

The one-line CONFIG command:

```
CONFIG 60 100460 460 1000 6/4
```

Specifies a system with 48 ('60) terminals and 4 (6/4) phantom users. The paging device (100460) and the command device (460) are partitions of a storage module on drive unit 0. The value of MAXPAG ('1000) specifies '400K (256K) of available physical memory storage.

The 6/ prefix is a feature of the command line which allows the omission of parameters. In this case, parameters 4 and 5 have been left at the default values (the first parameter, 60, is parameter 0).

Command File with CONFIG

PRIMOS III, IV, and V can be started up with a command file containing the CONFIG command. The user first creates a command file named C_PRMO and places this file in the UFD CMDNC0. The first line in the command file C_PRMO must be a CONFIG command. The command file C_PRMO may contain other PRIMOS III, IV, or V internal commands such as STARTUP and MESSAGE, and must terminate with the command line: CO -TTY

If CMDNC0 has a password, the nonowner password must be blanks and C_PRMO must allow read access rights for the nonowner.

Starting PRIMOS III, IV, or V from a Command File

To bring up PRIMOS III, IV, or V from the command file C_PRMO, the operator first starts up, under PRIMOS II, the physical disk that is to be logical disk 0 at the supervisor terminal. This logical disk 0 will be the command disk under PRIMOS III, IV, or V. The operator then attaches to the appropriate UFD and RESUMES PRIMOS (See Section 18 for PRIMOS III, Section 19 for PRIMOS IV and V).

The PRIMOS III, IV, or V operating system then starts running and checks if a command file C_PRMO exists in the UFD named CMDNC0 on logical disk 0. If C_PRMO does not exist, PRIMOS III, IV, or V will take commands from the supervisor terminal. If C_PRMO does exist, it will read commands from the command file until it encounters the command line, CO -TTY. Subsequently it will return to PRIMOS III, IV, or V and take commands from the terminal.

Errors

If the user types a detectable error while issuing the CONFIG command, the system requests that the CONFIG command be retried.

CONFIG EXTENSIONS (PRIMOS IV AND V)

Under PRIMOS IV or V, CONFIG parameters are also accepted in the form of a series of CONFIG subcommands. These subcommands are kept in a data file in CMDNC0 and are processed by the preloader to set up all the system parameters currently specified by the CONFIG command. In addition, the configuration subcommands can be used to set variables in FIGCOM and to override the default network configuration.

Overview of Preloader Actions

The preloader attaches to CMDNC0 and looks for the file C_PRMO. If the file exists, it is opened for command input; if it doesn't, the 'PLEASE ENTER CONFIG' prompt is issued. The first executable command is read (from the terminal or from C_PRMO), and a 'CO -TTY' is issued. The command is examined to ensure it is a CONFIG command.

Comment lines starting with /* may precede the CONFIG command in C_PRMO.

At this point, the preloader makes an additional check for the keyword '-DATA' as the first name on the CONFIG command. If this keyword is present, the second name following the command is taken as the name of a configuration data file. The file is opened for input, and configuration commands are processed as described below. The format is:

```
CONFIG -DATA configuration-data-filename
```

No restrictions are placed on configuration-data-filename, the name of the configuration data file; however, it is suggested that the name CONFIG be adopted as a de facto standard.

A one-line CONFIG command can be included anywhere in a configuration data file. It will not, however, be printed unless TYP0UT YES is in effect (see TYP0UT command). one-line CONFIG command is:

| | |
|-----------|-------------------------------------|
| myname | Network node name (6-char. max.) |
| 0/ntusr | Number of terminal users |
| 1/pagdev | Paging device |
| 2/comdev | Command device |
| 3/maxpag | Number pages physical memory to use |
| 4/altdev | Alternate paging device |
| 5/namlc | Number assignable AMLC lines |
| 6/npusr | Number phantom users |
| 7/nrusr | Number remote users |
| 10/smlcon | Non-zero value enables SMLC |

Note

All numerical vaues in command lines are octal.

CONFIG Subcommands

The subcommands for configuration data files are given below. Comments may be inserted in this file; comment lines should begin with the characters /*.

ALTDEV physical-device [records]

Specifies the alternate paging device and, optionally, its size.

physical-device is the physical device number of the paging disk or partition.

records is the size of the alternate paging device. If specified (and the equivalent parameter is specified in PAGDEV), then the total number of segments needed by PRIMOS will be automatically calculated. (records is a 16-bit non-zero positive integer.)

This specification is equivalent to the one-line CONFIG parameter 4/physical-device.

The alternate paging device is used only if the primary paging device (PAGDEV) has records specified.

AMLBUF line [in-buff-size] [out-buff-size] [dmq-size]]]

Sets terminal I/O buffer sizes.

line is the AMLC line number for which buffer sizes are to be set.

in-buff-size is the terminal input buffer size in words. If 0 is specified, buffer size is unchanged. The default value is '200 (128 decimal).

out-buff-size is the terminal output buffer size in words. If 0 is specified, buffer size is unchanged. The default value is '300 (192 decimal).

dmq-size is the size in words for the DMQ buffer (only meaningful if the system has DMQ AMLC controllers). If omitted or specified as 0, its value is not changed. The default value is '40 (32 decimal).

ASRATE control-word

Sets the supervisor terminal Baud rate.

control-word is an octal integer specifying supervisor terminal Baud rate (default value is '110).

| <u>control-word</u> | <u>BAUD rate (decimal)</u> |
|---------------------|----------------------------|
| 110 | 110 |
| 1010 | 300 |
| 2010 | 1200 |
| 3410 | 9600 |

If used, the ASRATE command must be the first one in the configuration data file. This ensures that any subsequent configuration error messages are printed at the appropriate speed.

The command is equivalent to, and overrides, the B-register setting of *COLDS, the cold start program.

ASRBUF line [in-buff-size [out-buff-size]]

Sets ASR terminal I/O buffer sizes.

line is the ASR line number. At present, only 0 is a valid line number.

in-buff-size is the ASR terminal input buffer size in words. If 0 is specified, buffer size is unchanged. The default value is '200 (128).

out-buff-size is the ASR terminal output buffer size in words. If omitted or specified as 0, buffer size is unchanged. The default value is '300 (192).

COMDEV physical-device

Specifies the system command device, i.e., the device on which the system UFD=CMDNC0 resides.

physical-device is the physical device number of the command disk or partition.

This command corresponds to the one-line CONFIG parameter 2/physical-device.

The command device must be specified with either the COMDEV or CONFIG command in the configuration data file.

CONFIG options

Specifies basic system configuration.

This command and its options are discussed at the beginning of this section.

DISLOG $\left\{ \begin{array}{l} \text{YES} \\ \text{NO} \end{array} \right\}$

Sets disconnect logout option.

YES user logged out if AMLC line is disconnected.

NO user is not logged out if AMLC line is disconnected (Default).

This command sets the variable DLOGOT in FIGCOM.

ERASE { character }
 { octal-value }

Specifies system default erase character; supplied default is ".

character is printing ASCII character, such as ".

octal-value is the octal value of any ASCII character.

Example:

ERASE 377 sets the system default
 erase character to DEL.

This command sets the variable DEFERA in FIGCOM.

FAM node-name net-type

Specifies configuration of FAM network.

node-name is the name of a network node (remote system) which
 is to be made available to FAM.

net-type is the path to be used to the remote system: IPC or
 SMLC.

FAM commands will be ignored unless the local node name has been
 specified. (See MYNAME).

GO

Marks end of configuration data file; any subsequent lines are
 ignored. The configuration data file must include a GO command.

KILL { character
 } { octal-value }

Specifies system default KILL character; supplied default is ?

character is a printing ASCII character, such as +, ?, *, etc.

octal-value is the octal value of any ASCII character.

Example

KILL + or KILL 253 sets the system default kill character to +.

This command sets the variable DEFKIL in FIGCOM.

LOGLOG { YES
 } { NO }

Allows LOGINs while logged in.

YES users are permitted to use the LOGIN command while logged in (Default).

NO the LOGIN command is inhibited for a logged-in user.

This command sets the variable LOGOVR in FIGCOM.

LOGMSG { YES
 } { NO }

Prints LOGIN/LOGOUT messages:

YES LOGIN and LOGOUT messages will be printed. (Default).

NO LOGIN and LOGOUT messages will be suppressed.

This command sets the variable NLGPRT in FIGCOM.

LOGREC file-size

Specifies maximum size of LOGREC file.

file-size is the size of the LOGREC file in words. The default value is '10000 (4096 decimal). Specifying a size of 0 disables the quota check. Specifying a negative value suppresses writing to the LOGREC file; this should be used when running a write-protected disk.

If size of LOGREC file is exceeded and quota check is not disabled, the message EXCEEDING QUOTA ON LOGREC will be printed for each additional entry added to the file.

This command sets the variable LRQUOT in FIGCOM.

LOUTQM minutes

Specifies inactivity time for automatic LOGOUT.

minutes is the number of minutes of inactivity allowed, at the terminal, before the user is automatically logged out. The default value is '1750 (1000 decimal). The value must be greater than 0.

This command sets the variable LOUTQM in FIGCOM.

MAXPAG number-of-pages

Specifies number of pages of memory to validate. Memory validation occurs at each cold start. This command must be included in order to use more than 256K words (512KB) of memory.

number-of-pages is the number of 2000 (1024 decimal) word pages of physical memory to validate for use.

This command corresponds to the one-line CONFIG parameter 3/number-of-pages.

MYNAME node-name

Specifies the network name of the local node.

node-name is the network node-name (6 characters maximum) of the local system.

The local node-name must be specified, either by a MYNAME or CONFIG command, to enable any network activity. If the local node-name is not specified, the commands: FAM, NET, and RLOGIN are ignored.

An attempt to run FAM with the network disabled yields the error BAD FAM SVC.

This command is equivalent to the optional first parameter (non-numeric) of the one-line CONFIG command line.

NAMLC number-of-lines

Specifies number of assignable AMLC lines in the system.

number-of-lines is the number of assignable AMLC lines in the system; the default is 0.

This command corresponds to the one-line CONFIG parameter 5/number-of-lines.

NET node-name { IPC line-number
 { SMLC line-number config data-set-control } }

Specifies network configuration parameters.

If any NET command is in the configuration file, the entire network configuration must be specified.

NET commands are ignored unless the local node name is specified by the MYNAME or CONFIG myname... commands. The proper FAM and RLOGIN commands must also be specified.

node-name is the name of a system which is part of the network. If node-name of NET is the same as node-name of the MYNAME (or CONFIG) command, the line-number, config, and data-set-control parameters are ignored.

line-number is the number of the line associated with that node. For an SMLC node line-number is the logical line number (0-3) over which connection to node-name is made. (See SMLC command for mapping of logical line number to controller number and physical line number.)

config is the SMLC configuration command word for line-number (SMLC only). Default is '363.

| <u>bits</u> | <u>meaning</u> |
|-------------|---|
| 1-7 | not used - must be 0 |
| 8 | Select Binary Synchronous Communications format 0 = CRC-16 (default) 1 = LRC - ASCII mode only |
| 9 | Select Formatter mode 0 = EBCDIC 1 = ASCII (default) |
| 10 | Formatter 0 = disabled 1 = enabled (default) |
| 11 | Data set change reporting 0 = disabled 1 = enabled (default) |
| 12-14 | Parity 000 no parity 001 odd-parity bit in addition to # bits/char 011 even-parity bit in addition to # bits/char 100 odd-parity bit included in # bits/char 110 even-parity bit included in # bits/char |

Default is 100

| | |
|-------|--------------------------|
| 15-16 | Number of bits/character |
| | 00 5 |
| | 01 6 |
| | 10 7 |
| | 11 8 (default) |

Note

See HSSMLC User Guide, MAN2362, for further details on these controllers.

data-set-control is a word specifying the data set control and data set status expected for line-number. (SMLC only). Default is 50401.

| <u>bits</u> | <u>meaning</u> |
|-------------|---|
| 1-4 | Received data set status expected 100000 - Ring indicator 040000 - Data carrier detect (default) 020000 - Clear to send 010000 - Data set ready (default) |
| 5-8 | Transmitter data set status expected 004000 - Ring indicator 002000 - Data carrier detect 001000 - Clear to send 000400 - Data set ready (default) |
| 9-16 | Data set control 000370 - Unused 000004 - Speed select/send new sync 000002 - Request to send 000001 - Data terminal ready (default) |

Bit values are added to form the complete data-set-control word.

The receiver data set status expected must contain at least the same bits as the transmitter set status expected.

NPUSR number

Specifies number of phantom users to be configured.

number is the number of phantom users for which the system is to be configured. (Non-negative octal integer.) Default is 0.

This command corresponds to the one-line CONFIG parameter 6/npusr.

NRUSR number

Specifies the number of processes to be reserved for remote logins. PRIMOS III does not support remote logins.

number is the number of remote users for which the system is to be configured. (Non-negative octal integer). Default is 0. The maximum is '40 (decimal 32).

This command corresponds to the one-line CONFIG command parameter 7/nrusr.

NSEG number

Specifies number of available segments.

number is the total number available to the system, PRIMOS IV or V, plus all users: local, phantom, and remote. Default values are:

| <u>PRIMOS IV or V</u> | <u>default value</u> |
|-----------------------|----------------------|
| 16 user | '220 (decimal 144) |
| 64 user | '300 (decimal 192) |
| 16 user, large | '500 (decimal 320) |

number must be greater than total users +9; number must be less than or equal to the default value.

This command sets the value of the variable SEG in segment 4.

If the paging space specified by PAGDEV (and ALTDEV) does not allow NSEG segments to be allocated, NSEG is reduced to conform to the paging space requirements.

NTUSR number

Specifies number of terminal users.

number is the number of terminal (local system) users for which the system is to be configured. (Positive octal integer less than or equal to the user or users supported by the version of PRIMOS IV or V used: 16 or 64 user.)

This command corresponds to the one-line CONFIG parameter 0/number.

The number of terminal users must be specified, either by NTUSR or the CONFIG command.

PAGDEV physical-device [records]

Specifies paging device and size.

physical-device is the physical device number of the disk or partition on which paging is to take place.

records is the size of the paging disk (a non-negative 16-bit integer). If specified, this value (with the ALTDEV value of records, if any) limits the total number of segments in the system. If records is specified and an alternate paging device has been specified by ALTDEV, then records defines the point at which page space allocation switches from the primary to the alternate paging device.

This command corresponds to the one-line CONFIG parameter 1/physical-device. The paging device must be specified by PAGDEV or CONFIG.

If records is small (e.g., 1), almost all paging is forced to occur on the alternate device. However, the primary device is always used to page the PRIMOS IV or V segments (0-'12) and user 1's segment '6000.

PREPAG pages

Specifies number of pages to prepage.

pages is a positive octal integer specifying the number of pages to prepage out when a page fault occurs. The default value is 3.

This command sets the variable PREPGK in PAGCOM.

RLOGIN node-name net-type

Specifies remote login network configuration.

node-name is the name of remote system on which remote logins are to be allowed.

net-type specifies the network path to be used. Legal arguments are: IPC or SMLC.

RLOGIN commands are ignored if the local node-name is not specified by MYNAME (or CONFIG myname...).

Remote logins are not possible from the local system unless the RLOGIN command is given.

RWLOCK value

Specifies file system read/write lock setting.

value is an octal integer specifying the system-wide file read/write lock. The default value is 1. Possible values are:

| <u>Value</u> | <u>Meaning</u> |
|--------------|---|
| 0 | 1 reader <u>or</u> 1 writer (writer has exclusive control) |
| 1 | N readers <u>or</u> 1 writer (writer has exclusive control) |
| 3 | N readers <u>and</u> 1 writer |
| 5 | N readers <u>and</u> N writer |

The command sets the RWLOCK variable in FIGCOM.

| | | |
|-------------|----------------------------------|-----|
| SMLC ON | | (1) |
| SMLC CNTRLR | controller-number device-address | (2) |
| SMLC SMLCnn | controller-number line-number | (3) |

Enables and configures SMLC lines. There are three formats:

- (1) Enables the SMLC in the default ON configuration; the normal OFF default is SMLC disabled. This command corresponds to the one-line CONFIG parameter 10/1.
- (2) Specifies physical device number(s) of the SMLC controllers.

controller-number is the controller, either 0 or 1.

device-address is the physical device address of the specified controller. The default value is address '50 for controller 0; default address for controller 1 is not defined.

(3) Maps logical line numbers onto physical controllers and line numbers.

nn is the logical line number; values are 00 to 07.

controller-number is the controller, either 0 or 1.

line-number is the physical line number of the specified controller onto which the logical line number is mapped. The default values map SMLC00 to SMLC03 onto controller 0, physical lines 0 to 3.

TYPOUT { YES }
 { NO }

Controls printing of configuration commands.

YES subsequent commands in configuration data file will be printed on the supervisor terminal as they are processed. Printing continues until a TYPOUT NO or GO command is encountered.

NO commands are not printed as they are processed. Printing is suppressed until a TYPOUT YES or GO command is encountered. TYPOUT NO is the default.

Any number of TYPOUT commands can be used in a configuration data file to print selected commands.

PRIMOS IV AND V INITIALIZATION ERROR MESSAGES

Errors encountered during PRIMOS IV or V initialization cause an error message to be printed. The system then halts at location BOOT0 in segment 6. PRIMOS II must be reloaded; then the offending command (or lack thereof) should be corrected in the configuration data before attempting to bring up PRIMOS IV or V. A complete list of error messages generated by the PRIMOS IV or V pre-loader (PRIMOS) and the PRIMOS IV or V initialization sequence are listed below.

Error Messages

file-system-error-message config-file (PRIMOS)

A file system error was encountered by the preloader while attempting to open the configuration file config-file.

file-system-error-message CAN'T ATTACH TO CMDNC0 (AINIT)

A file system error was encountered while attempting to attach to CMDNC0 for User 1.

file-system-error-message CMDNC0 (PRIMOS)

A file system error was encountered by the preloader while attempting to attach to CMDNC0.

file-system-error-message C_PRMO (PRIMOS)

A file system error (other than FILE NOT FOUND) was encountered by the preloader while attempting to open the file C_PRMO for command input.

file-system-error-message PRnnnn (PRIMOS)

A file system error was encountered by the preloader while attempting to open or read the indicated PRnnnn file.

BAD command PARAMETER (AINIT)

One or more of the parameters specified for the configuration command is invalid.

BAD CONFIG COMMAND: command (AINIT)

The command in the configuration command file is not a recognized configuration command.

BAD DMQ AMLC CONFIGURATION (AINIT)

A DMQ buffer size in an AMLBUF command is not equal to a power of 2, or total size of terminal input, output and DMQ buffers exceeds 64K words.

BAD LINE # IN AMLBUF CMND (AINIT)

An AMLBUF command specifies a line number less than 0 or greater than the number of lines configured for the system.

BAD LINE # IN ASRBUF CMND (AINIT)

An ASRBUF command specifies an invalid line number.

END OF FILE. MISSING 'GO' CMND (PRIMOS)

The configuration data file does not include the required GO command.

FIRST COMMAND MUST BE CONFIG

The command typed in response to the 'PLEASE ENTER CONFIG' prompt or the first executable command in C_PRMO is not a CONFIG command. (Pre-loader)

ILLEGAL ALTDEV

The device specified as the alternate paging device is not legal. (Pre-loader)

ILLEGAL COMDEV

The device specified for the command device is not legal. (Pre-loader)

ILLEGAL PAGDEV

The device specified for paging is not a legal paging device. (Pre-loader)

MISSING NTUSR, PAGDEV, OR COMDEV

The configuration data file did not specify all these required parameters. (Pre-loader).

NETWORK NOT CONFIGURED (AINIT)

A node name was specified on a system not configured for networks.

NRUSR INVALID (AINIT)

The number of remote users specified by an NRUSR command exceeds the

maximum number of configurable remote users (40, decimal 32).

NTUSR+NPUSR+NRUSR TOO BIG (AINIT)

The number of terminal plus phantom plus remote users exceeds the maximum number of configurable users (16 or 64).

RESTART PLEASE

This message appears following any error message printed by the PRIMOS IV or V initialization logic (AINIT). The system will halt at location `BOOT0` in segment 6. PRIMOS II must be reloaded. The offending command in the configuration data file must be corrected.

SEEK FAILURE ON ALTDEV (AINIT)

The initial seek to cylinder 0 on the alternate paging device failed.

SEEK FAILURE ON PAGDEV (AINIT)

The initial seek to cylinder 0 on the primary paging device failed.

SMLC CTRLR # OUT OF RANGE (AINIT)

An SMLC command specifies an invalid controller number.

SMLC LINE # OUT OF RANGE (AINIT)

An SMLC command specifies an invalid line number.

TERMINAL I/O BUFFERS TOO LARGE (AINIT)

The total size of the terminal input and output buffers exceeds 32K words. (AMLBUF and ASRBUF commands)

TPIOS ERROR

An I/O error occurred while preloading the paging device. (Pre-loader)

UNKNOWN NODE NAME (AINIT)

The network node name specified does not match any name in the network configuration tables.

USE physical-device FOR PAGING?

The disk physical-device has been specified as the paging device, but is formatted as a standard PRIMOS disk. A reply of YES is required to enable paging on physical-device. (Pre-loader)

SECTION 17

PRIMOS II OPERATION

This section describes how to load and start PRIMOS II and summarizes the essential operator tasks during a single-user operating session.

OVERVIEW OF PRIMOS II OPERATION

What is PRIMOS II

PRIMOS II is the single-user version of PRIMOS, intended primarily for single-user program development and execution: for example, dedicated (and unrestricted) control of real-time processes. It has essentially all the capabilities of other versions of PRIMOS except that multi-user environment commands (USERS, SHARE, SPOOL, CHAP, etc.) are not included. Ordinarily, PRIMOS II is used only momentarily during startup of PRIMOS III, IV, or V (the bootstrap brings a version of PRIMOS II into memory; See Section 2). However, in some situations it may be desirable to run the computer as a single-user system under PRIMOS II: for backups, file operations with absolute security, etc.

Certain operational aspects of PRIMOS II differ from the time-shared versions. For example:

- It is not possible to interrupt a program with BREAK (CONTROL-P). Instead use the control panel to HALT and then restart by STARTing at a specified starting location. (See Appendix F)
- PRIMOS II occupies the user's address space - the same physical memory as the user. (See Table 17-1.) This places some restrictions on program loading to ensure that PRIMOS II and the loader are not overwritten.

Versions of PRIMOS II

Three versions of PRIMOS II are supplied on the master disk.

Q-DOS is 50000 locations in size, including file-system buffers. It supports file-system partitions and calls, and long-name command lines. It is loaded in 64R addressing mode and requires a CPU with the high-speed arithmetic option.

Relative-mode PRIMOS II (R-DOS) is '14000 locations in size, excluding file-system buffers and hybrid commands. It is loaded in 64R addressing mode and can be loaded or called from anywhere in memory. It supports the storage modules but not the file-system partitions and calls or the long-name command lines. Its main function is to start up PRIMOS III on a system with only 32K of high speed memory.

Sectored-mode PRIMOS II (S-DOS) is '11000 locations in size, excluding file-system buffers and hybrid commands. It is loaded in 32S addressing mode and cannot be loaded or called from above the 32K boundary. It does not support the storage module, file-system partitions and calls, or the long-name command lines.

These versions of PRIMOS II are supplied loaded for various memory sizes in both real-memory and virtual-memory configurations. The real-memory configurations run stand-alone on any Prime CPU (except that Q-DOS requires high-speed arithmetic). The virtual-memory configurations run in user space under PRIMOS III, IV, or V on Prime 300 (and higher) CPUs.

The locations and names of the PRIMOS II configurations on the master disk as supplied in Table 17-1.

Table 17-1. PRIMOS II Versions

| <u>UFD</u> | <u>File</u> | <u>Contents</u> | <u>Low</u> | <u>Restart</u> | <u>High</u> |
|------------|-------------|---|------------|----------------|-------------|
| DOS | SDOS32 | 32K real-memory S-DOS | 067000 | 070000 | 077777 |
| DOS | *DOS32 | 32K real-memory R-DOS | 064000 | 070000 | 077777 |
| DOS | RDOS64 | 64K real-memory R-DOS | 164000 | 170000 | 177777 |
| DOS | *DOS64 | 64K real-memory Q-DOS | 130000 | 170000 | 177777 |
| DOS | VDOS32 | 32K virtual-memory S-DOS | 067000 | 070000 | 077777 |
| DOS | UDOS32 | 32K virtual-memory R-DOS | 064000 | 070000 | 077777 |
| DOS | UDOS64 | 64K virtual-memory R-DOS | 064000 | 170000 | 177777 |
| CMDNCO | TDOS64 | 64K virtual-memory Q-DOS | 140000 | 170000 | 177777 |
| CMDNCO | DOEXT | saved-file containing hybrid commands for S-DOS | | | |
| CMDNCO | DOSXT | saved-file containing hybrid commands for R-DOS | | | |

The hybrid commands, which are in files DOEXT and DOSXT for S-DOS and R-DOS, are internal to Q-DOS, so there is no hybrid command file in UFD CMDNCO for Q-DOS.

File-system buffers are internal to Q-DOS. The bottom of Q-DOS is fixed at location '130000, independent of the number of open file units. Buffers are external to R-DOS and S-DOS and are allocated downward from the bottom of the operating system toward the user program region.

The banner printed when Q-DOS is loaded is:

PRIMOS II REV 15 02/28/78 (AT nnnnnn)

where nnnnnn is the restart location.

Similar banners are printed for R-DOS and S-DOS.

Bootstrap Loading

PRIMOS II is started up from the control panel by the procedure described in Section 2. When a master disk is loaded from the processor control panel, the bootstrap assesses the size of main-memory physically present on the machine and loads file *DOS_{nn} from UFD DOS, where nn is 32 or 64, depending upon main-memory size. Q-DOS is loaded on 64K machines. An installation can change this by renaming the files in UFD DOS. For example, if an installation desires relative-mode PRIMOS II to load on 64K machines, it should rename *DOS64 to QDOS64 and RDOS64 to *DOS64. Setting sense switch one limits nn to 32 on 64K machines.

All versions of PRIMOS II automatically acquire their supervisor terminal typewriter control words directly from the bootstrap. (See Appendix A.)

On systems with a storage module disk drive, any version of PRIMOS II can be bootstrap-loaded from the storage module. The sense switch settings are xx0114. However, sectored-mode PRIMOS II does not support the storage module as a file-system device and can make no further reference to it.

See Appendix A for additional details on bootstraps.

STARTUP OF PRIMOS III, IV, or V

Once PRIMOS II is started and running, PRIMOS III (Section 18), IV (Section 19), or V (Section 19) can be started.

INITIAL PRIMOS II OPERATING SESSION

Attaching to UFD

After a STARTUP, the user must attach to a User File Directory to execute PRIMOS II commands and create or manipulate files. The MFD has a password to discourage casual or inadvertent use of this important directory. (At MAKE time, a default password of XXXXXX is assigned.)

CAUTION

Do not attach to the MFD for program development or normal file handling tasks. Be very careful in entering commands while attached to the MFD. FIXRAT can be used to fix the disk if the UFDs DOS, BOOT, or CMDNCØ are intact; but if any of the files in the MFD are damaged, the master disk is compromised.

PROGRAM DEVELOPMENT USING PRIMOS II

PRIMOS II and its supporting software allow the user to:

- Format disks and partitions using MAKE
- Check file system integrity with FIXRAT
- Backup disks with COPY
- Create programs (Editor)
- Compile or assemble programs (FIN, PMA)
- Load and save programs (LOAD)
- Execute programs (RESUME, RESTORE, START)
- Use the interpretive BASIC language for program development

The PRIMOS II commands are described in REFERENCE GUIDE, PRIMOS COMMANDS (PDR3108); other manuals to be consulted are those for the specific language users and the NEW USER'S GUIDE TO EDITOR AND RUNOFF (FDR3104).

Notes

Q-DOS and R-DOS refuse to restore a saved file if this operation would overwrite the operating system or its buffers. This check is not made in S-DOS, and unpredictable behavior can result if this error is made.

PRIMOS II cannot run any V-mode programs such as SEG or the shared editor.

Some commands use all of memory and will not run under PRIMOS II (e.g., RUNOFF).

Recovering from Errors

If an equipment failure or program error causes the CPU to leave PRIMOS II control, it is usually possible to restart the CPU from the control panel. The procedure is:

- Turn the selector switch to STOP/STEP
- MASTER CLEAR
- Set the sense switches to the restart address for the specific version of PRIMOS II being used. (See Table 17-1).
- Turn selector switch to LOAD
- START
- Reset sense switches to 0.

SHUTDOWN

Before terminating an operating session with PRIMOS II by loading another operating system or turning off power, enter the following command:

OK: SHUTDN

The SHUTDN command writes to disk PRIMOS II data that is buffered in memory.

Changing Disk Packs

To change removable disk packs, shut down PRIMOS II as above. Then power down the disk drive and replace the pack. If PRIMOS III or IV is the system in control, SHUT DOWN the physical disk with the SHUTDN command, then cycle down the disk. Restart the disk drive. When the unit is READY, give the STARTUP command appropriate to the operation with the new pack, and continue typing commands.

Turning Power Off

After a shutdown, the CPU can be used to run other software or power can be turned off. The following power-down order is recommended:

1. Disk drives
2. ASR, high-speed tape unit, and other peripheral devices
3. Prime CPU

RESTARTING PRIMOS II

A typical procedure to restart PRIMOS II after a shutdown is:

1. Turn on power and boot PRIMOS II into control.
2. Give the appropriate STARTUP command to bring up the command device.
3. ATTACH to a UFD and resume operation.

Pitfalls to Avoid

Some users are accustomed to keeping the PRIMOS II restart location in the sense switches so that it can be easily restarted. With 64K PRIMOS II (either Q-DOS or R-DOS), this is not recommended, as it requires Sense Switch 1 to be SET. Setting Sense Switch 1 may produce undesired results with the FTN, PMA, and MDL commands.

SECTION 18

PRIMOS III

PRIMOS III SYSTEM OVERVIEW

PRIMOS III enables a community of up to 31 simultaneous users to share computer resources in a virtual memory environment.

The resources shared are the central processor, high-speed memory, the file system, and the peripheral devices which are assignable to only one user. Each user is provided with a terminal to interact with PRIMOS III and a 64K word virtual memory space. Any user can access files on disks using the same commands and system subroutines that are available when running under PRIMOS II. Peripheral devices, such as the paper tape reader, may be used in the same manner as under PRIMOS II, provided they are first assigned to the user by means of the ASSIGN command. Under PRIMOS III, users are protected from interfering with each other, and user privacy is assured. No user can peek into another user's memory to find out what the other user is doing, and no user can alter another user's memory. Under both PRIMOS II and PRIMOS III, disk files can be protected by passwords on file directories.

Note

The number of users supported by a version of PRIMOS III is assumed to include the supervisor terminal.

PRIMOS III VERSIONS

PRIMOS III is distributed in three versions:

- 16-user version with IPC network support (standard)
- 32-user version without network support
- 5-user version with processor serial interface support

PRIMOS III

All versions of PRIMOS III consist of three runfiles: PRIMOS, PRIMOSL, and PRIMOSU. The associated load map is M PRMO. PRIMOS is a utility program that preloads the paging device with the file PRIMOSU (the upper part of PRIMOS III) and RESUMES PRIMOSL. Each version of PRIMOS III resides in its own UFD. Any of these versions may be copied into UFD=CMDNC0 where PRIMOS III may be invoked by the command:

PRIMOS

The version supplied in UFD=CMDNC0 is the 16-user version with IPC network support (it is also found in UFD=PR3.16).

Note

When copying PRIMOS III from its UFD to the command UFD=CMDNC0, be sure to copy all three runfiles: PRIMOS, PRMOSU, and PRMOSL to avoid unspecified and undesirable results. It is also useful to copy the load map M_PRMO into UFD=CMDNC0.

Any version of PRIMOS III may be brought up by attaching to the appropriate UFD and RESUMEing PRIMOS.

16-user version (IPC): This version supports 16 users on the Prime 300 (or higher) with either an 8- or 16-line AMLC. It also supports the SMLC and the Inter Processor Controller (IPC). File sharing between two Prime 300 (or higher) computers is supported with the aid of the File Access Manager (FAM), an optional extension to PRIMOS III. This version is supplied in UFD=PR3.16. It is also supplied as the standard version of PRIMOS III in UFD=CMDNC0.

32-user version: This version supports 32 users on the Prime 300 (or higher). It does not support networks or file sharing. The SMLC is supported. This version is supplied in UFD=PR3.32.

5-user version: This version supports 5 users on the Prime 300 (or higher). It runs the processor serial interface with the 330 cycle clock and does not require an AMLC. Networks and file sharing are not supported by this version. This version is supplied in UFD=PRM330.

Networks

In PRIMOS III, PRIMENET supports communications among PRIME processors over the IPC. When so configured, PRIMOS provides network services.

BUILDING PRIMOS III

To build the desired version of PRIMOS III, attach to the appropriate UFD (see table below) and run the command file C_ALL. This creates the 3 runfiles: PRIMOS, PRMOSU, and PRMOSL and a load map M_PRMO in the UFD attached to.

| <u>Version of PRIMOS III</u> | <u>UFD</u> |
|------------------------------|------------|
| 16-user (IPC) | PR3.16 |
| 32-user | PR3.32 |

| | |
|--------|---|
| C_FS | compiles file system sources |
| C_FTNK | compiles kernel Fortran sources |
| C_PMAK | compiles kernel PMA source |
| C_CS | compiles communications system sources |
| C_NS | compiles network sources |
| C_LOAD | loads PRIMOS object modules. Generates PRMOSL and PRMOSU. Incorporates file PAGMAP into PRMOSL. Generates M_PRMO, the load map. |

Source Files

The source files of PRIMOS III are in UFD=PRI300 on volume A6 of the Master Disks. This UFD contains the following subdirectories.

| | |
|-------|--|
| CS | contains sources for communications subroutines |
| FS | contains sources for file system subroutines |
| NS | contains sources for network subroutines |
| KS | contains sources for kernel subroutines |
| KS1 | contains additional kernel sources and interlude sources |
| CO.16 | contains objects for communications subroutines, 16-user |
| FO.16 | contains objects for file system subroutines, 16-user |
| NO.16 | contains objects for network subroutines, 16-user |
| KO.16 | contains objects for kernel subroutines, 16-user |
| IO.16 | contains objects for SVC interlude binaries, 16-user |

- CO.32 contains objects for communications subroutines, 32-user
- FO.32 contains objects for file system subroutines, 32-user
- NO.32 contains objects for network subroutines, 32-user
- KO.32 contains objects for kernel subroutines, 32-user
- IO.32 contains objects for SVC interlude binaries, 32-user

Modifying PRIMOS III

In addition to parameters set by the PRIMOS III CONFIG command, other system defaults and constants can be modified using HPSD. These include: kill and erase characters, inactivity logout time, file system read/write lock. The modification technique is discussed in Section 13.

To build a version of PRIMOS III to support the Gould Printer/Plotter, delete the file VERDIM, rename the file GLDDIM to VERDIM, and run the appropriate command file.

SYSTEM CONFIGURATION

Upon obtaining a master disk from Prime, the system administrator must install the desired version of a PRIMOS III for the installation into CMDNC0. PRIMOS III supports 1 to 15 or 31 users (plus the supervisor) and a variety of peripheral devices including the SMLC.

Memory Requirements

A minimum of 48K words of contiguous memory is required to bring up a 32-user version of PRIMOS III previously described. It is suggested that 8K words of memory be available per user. Therefore, a 31 terminal user configuration contains at least 256K words of memory. Some application-oriented systems may run with less memory. Of course, a system may be configured for less than 31 terminal users. For example, an installation to run 20 terminal users would require proportionally less memory.

AMLC Requirements

Two 16-line AMLCs are required to handle user terminals for 31 terminal users. For 21 terminal users, a 16-line AMLC along with an 8-line AMLC may be used.

Disk Space

Two platters of a 6.0-million word disk drive (or the equivalent partition) are required for paging, if each user is to have a 64K word virtual address space. Users 1 through 25 use one platter, and users 26 through 31 use the other. Only one head is needed for the storage module. The CONFIG command lines format to accomplish this must be given as follows:

```
CONFIG ntusr pagedev comdev [maxpeg] [altdev] [namlc] [npusr]
      [l0/smlcon]
```

For details, refer to the description of the CONFIG command in Section 16. If pagedev is not large enough, a failure to give the parameter altdev causes the message NO PAGEDEV to be printed repetitively at the supervisor terminal if a user having a system-assigned user number between 26 and 31 attempts to use the system.

Paging Space Requirements

PRIMOS III paging requirements may be calculated from:

$$\text{records} = (\text{users} + 1) * 128 * (\text{records}/\text{page})$$

users is the total number of configured users: terminal and phantom.

record/page is 1 for storage modules, and 2 for other disks.

Modification of Page Maps

A system administrator may wish to custom-modify the PRIMOS III page maps to:

Run some user with 64K address space and other users with 32K address space,

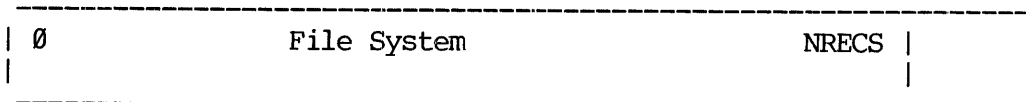
Run some users on the fixed head disk (obsolete) and others on a moving head disk. This is rarely done.

Running Split Disk

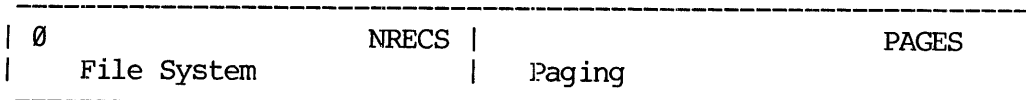
The following procedure is rarely done. Care should be taken to follow instructions exactly. The system administrator may wish to use part of a disk surface for paging and the rest for the file system.

On a normal disk, the file system uses space from record 0 to NRECS, where NRECS is the number of 448-word records that may be written on the disk. On a split disk, NRECS must be specified as something less than the maximum; the remainder of the disk space is used for paging (see the following schematic diagram):

Normal disk:



Split disk:



The program MAKE must be run to configure a split disk. For further information, refer to Section 7.

The amount of paging space required is calculated as follows:

Paging spaces on the moving-head disk is 256 records per user (64K address space) plus 352 records for the supervisor.

For an n-user PRIMOS III configuration, where n is the number of users, the amount of paging space needed is:

$$\text{PAGES} = 256 \times n + 352 \text{ records}$$

Example:

For a six-user system: $\text{PAGES} = 256 \times 6 + 352 = 1888$
 and $\text{NRECS} = 6496 - 2488 = 4008$. This assumes the disk is one platter of a 6.0M disk drive.

To make a split disk, mount a scratch pack on the drive and do the following:

```

MAKE
PHYSICAL DEVICE = 52
1.5M WORD PACK? NO
SPLIT DISK? YES
PAGING RECORDS (DECIMAL): 1888

DISK      FILE-RECORDS  PAGE RECORDS (DECIMAL)
      000031    1360           1888

PARAMETERS OK? YES
BAD SPOTS ON DISK? NO
VIRGIN DISK? YES
VERIFY DISK? YES
BEGINNING WRITE
BEGINNING VERIFY
DISK CREATED (after a while)

```

The system administrator must then use the UFDCPY feature of FUTIL to copy the UFDs CMDNC0 and DOS from the master disk.

Note

If a split disk is used for paging, it must be used also as the command device under PRIMOS III.

Table 18-1 provides a guide to disk addresses for system administrators. They show disk space required for the supervisor and up to 31 users.

Table 18-1. Disk Space Required for 64K Per User

| User | Disk Addresses Decimal | FHD Disk (obsolete) DecimalAddresses - decimal |
|------------------------------------|---------------------------|---|
| Supervisor | 0-512 | 0-175 |
| 1 | 513-768 | 176-303 |
| 2 | 769-1024 | 304-431 limit 256K |
| 3 | 1025-1280 | 432-559 |
| 4 | 1281-1536 | 560-687 |
| 5 | 1537-1792 | 688-815 limit 512K |
| 6 | 1793-2048 | 816-1044 |
| 7 | 2049-2304 | |
| 8 | 2305-2560 | |
| 9 | 2561-2816 | |
| 10 | 2817-3072 | |
| 11 | 3073-3328 | |
| 12 | 3329-3584 | |
| 13 | 3585-3840 | |
| 14 | 3841-4096 | |
| 15 | 4097-4352 | |
| 16 | 4353-4608 | |
| 17 | 4609-4864 | |
| 18 | 4865-5120 | |
| 19 | 5121-5376 | |
| 20 | 5377-5632 | |
| 21 | 5633-5888 | |
| 22 | 5889-6144 | |
| 23 limit 3.0M platter = 6495 | 6145-6400 | |
| 24 | 0-255 altdev | |
| 25 | 256-511 " | |
| 26 | 512-767 " | |
| 27 | 768-1023 " | |
| 28 | 1024-1279 " | |
| 29 | 1280-1535 " | |
| 30 | 1536-1791 " | |
| 31 | 1792-2047 " | |

PRIMOS III SYSTEM INITIALIZATION

PRIMOS III is started from PRIMOS II at the supervisor terminal.

The steps to get PRIMOS III running are:

1. Turn on processor, disk drive, and all needed peripheral devices, and boot load PRIMOS II as explained in Section 2.
2. To ensure that PRIMOS III is available, enter the command:

```
STARTUP command-device-number
```

if this has not already been done.

3. To initiate PRIMOS III, issue the command:

```
PRIMOS
```

Response is:

```
PRIMOS REV xx.xx  
yy.y K MEMORY IN USE  
PLEASE ENTER CONFIG
```

At this point, PRIMOS III is not running, but a utility program that will bring up PRIMOS III is running. The operator must enter the CONFIG command. If a mistake is made in the CONFIG command, the system does not ask for the command to be retyped; instead, it returns to PRIMOS II. The operator must then re-invoke the PRIMOS command.

If the CONFIG command is acceptable, PRIMOS III is brought up and after approximately one minute, the introductory message is followed by the message:

```
PLEASE ENTER DATE
```

After the operator inputs the date with the SETIME command, the message:

```
LOGIN PLEASE
```

is printed at user terminals, and users may log in.

CAUTION

User terminals will not respond to commands until the SETIME command is given by the operator.

PRIMOS III may be brought up from a command file. The command file to bring up PRIMOS III must be named C_PRMO and must be contained in the UFD named CMDNC0.

If the SMLC or networking (IPC) is to be used, additional parameters must be given to the CONFIG command. If the SMLC is to be used, the parameter 10/1 must be given.

Example:

```
CONFIG 10 51 50 10/1
```

If the IPC is to be used, the CONFIG command line must include the name of the system as SYSA or SYSB, depending on whether the slave address of the IPC is 1 or 2, respectively.

Example:

```
CONFIG SYSA 10 51 50
```

For complete information about CONFIG see section 16; SETIME and other operator commands are described in Section 15.

Bypassing Bad Memory

PRIMOS III includes features to detect and bypass bad memory chips. These features are categorized as follows:

- On a STARTUP, PRIMOS III performs a simple data and parity check of all memory locations above 32K.
- If memory chips are known to be bad, PRIMOS III can be modified (and SAVED) so as not to use these bad pages (chips).
- If the system crashes while running, a self-contained routine can be started that tests all available memory to locate any memory cell that contains incorrect parity.

Memory Parity

See Section 10 for a detailed discussion of system crashes. On a memory parity error, PRIMOS III halts. After determining that a memory parity failure has in fact occurred and writing down all the registers, the operator must perform a warm start. If any bad pages are found, the bad page numbers are printed in octal followed by the message:

```
PARITY ERROR AT ABOVE PAGES
STOP, FIX MMAP AND PAGEMAPS, WARM START
```


The operator must stop the machine, fix MMAP and PAGMAP if possible and do another warm start.

PRIMOS Halts

Locations at which PRIMOS III halts are defined in the load map M_PRMO. This map is in the UFD in which PRIMOS III was built. The word number of the halt is determined from the control panel lights and then that halt location in the load map is identified with a variable (all halt location definitions are of the form xxxxx_.)

| <u>Halt</u> | <u>Meaning</u> |
|-------------|--|
| AMLCI__ | Spurious AMLC controller interrupt |
| BDMEM__ | Memory parity during cold start |
| BOOT0__ | SHUTDN ALL halts here |
| CLCK__ | Clock trap when not allowed |
| F\$HT__ | Network halt |
| IPAGE__ | Bad page fault in interrupt process |
| MCHK__ | Machine check |
| MEMPA__ | Memory parity halt |
| MMOD__ | Missing memory |
| PWRFL__ | Power fail interrupt |
| REFL0__ | FLEX, UII, PSU when not allowed |
| RMCF0__ | Restricted mode fault when not allowed |
| SVCF0__ | SVC when not allowed |
| TRPRH__ | Too many returns for interrupt process |

SECTION 19

PRIMOS IV-V

PRIMOS V

PRIMOS V includes all the features of PRIMOS II, III, and IV. It is designed to take advantage of the Prime 500 hardware (32-bit general register architecture): 256K - 8M bytes of MOS memory, error correcting memory, bipolar cache memory, fast floating-point hardware and an expanded instruction set. Some of the hardware features of the Prime 500 may be emulated by software on a Prime 400 (described under system configuration).

The volume of the master disk that contained PRIMOS IV sources and command files for Prime 350 and Prime 400 computers will contain PRIMOS V sources and command for a Prime 500. For ease in building the master disks, the UFDs and filenames for PRIMOS V are the same as for PRIMOS IV. PRIMOS IV or PRIMOS V is supplied, not both, eliminating any chance of error with UFD or filenames.

OVERVIEW OF PRIMOS IV

PRIMOS IV is the operating system for the Prime 350 and 400 computers. (It will also run on the Prime 500 but PRIMOS V makes better use of the Prime 500 architecture). All versions of PRIMOS IV consist of the following files: PRIMOS, *COLDS, PR0000, PR0004, PR0005, PR0006, PR0011, PR0012, PR6000, C_COLD, *CMEM, and M_PRMO. The last is the load map for PRIMOS IV. The operator should print out a copy of the load map for any version of PRIMOS IV which will be run on the computer and save these maps in the logbook.

Versions of PRIMOS IV

PRIMOS IV is distributed in three versions with default characteristics specified below:

| <u>Version</u> | <u>UFD</u> | <u>Number of Segments</u> | <u>Maximum Segments Per User</u> |
|----------------------------------|------------|---------------------------|----------------------------------|
| 64-user | PR4.64 | '300 (192) | '40 (32) |
| 16-user | PR4.16 | '220 (144) | '10 (8) |
| 16-user with large address space | PR4L16 | '500 (320) | '400 (256) |

The versions above do not support networking. Attempts to configure networks in any of these versions will result in the message:

PRIMENET NOT AVAILABLE IN THIS VERSION

and a machine halt at the cold start address. Aside from this, the versions of PRIMOS IV supporting and not supporting networks are the same.

Network-supporting versions of PRIMOS: The three version of PRIMOS IV supporting networks are located in:

| <u>version</u> | <u>UFD</u> |
|-------------------------------------|---------------|
| 64-user | PRINET>PR4.64 |
| 16-user | PRINET>PR4.16 |
| 16-user with large address space | PRINET>PR4L16 |

PRIMENET included in these versions supports:

- The Inter Processor Controller (IPC). File sharing between two Prime 350 (or higher) computers is supported with the aid of the File Access Manager (FAM).
- Connection of full duplex high-speed synchronous multiple line controllers (HSSMLC). Speeds are up to 9600 band for a direct connection and up to 4800 band over a dedicated telephone line using full duplex modems.

BUILDING PRIMOS IV

To build PRIMOS IV attach to the appropriate directory and run the command file C_ALL to build the desired version. The runfiles are built in the directory attached.

The source files of PRIMOS are found in UFD=PRI400 on volume 6 of the master disk. PRI400 contains subdirectories as follows:

- CS-contains sources for communications subroutines
- FS-contains sources for file system subroutines
- NS-contains sources for network subroutines
- KS-contains sources for kernel subroutines

CO.16 or CO.64 - contains objects for communications subroutines
 FO.16 or FO.64 - contains objects for file system subroutines
 NO.16 or NO.64 - contains objects for network subroutines
 KO.16 or KO.64 - contains objects for kernel subroutines

C_PRML - Generates file PRIMOS. PRIMOS requests the CONFIG from the operator. Following the CONFIG, files PR0000 to PR0012 are prepaged to the paging device and *COLDS is resumed.

C_MAPG - Generates the pagemaps as file PAGMAP from the source file MAPGEN.

C_CMEM - Generates the run file *CMEM from the source CMEM. *CMEM is a utility used to create *COLDS, which is the initial memory image of PRIMOS.

C_ALL runs a series of command files as follows:

C_FTNF - compiles file system FORTRAN source

C_PMAF - assembles file system PMA source

C_FTNK - compiles kernel FORTRAN source

C_PMAK - assembles kernel PMA source

C_FTNC - compiles communications system FORTRAN source

C_PMAC - assembles communications system PMA source

C_FTNN - compiles network FORTRAN source

C_PMAN - assembles network PMA source

C_LOAD - Loads object of PRIMOS. Generates VPRMOS, PR0000, PR0004, PR0006, PR0011, PR0012. C_LOAD also incorporates file PAGMAP into file PR0004 where it belongs. Also generates M_PRMO the load map.

C_COLD - Runs *CMEM to generate *COLDS from PR0000 to PR0012. *COLDS is the initial memory image file of PRIMOS.

PRIMOS IV occupies segments number 0 - '12 and '6000. The layout of these segments is:

| <u>SEGMENT</u> | <u>CONTENTS</u> |
|----------------|---|
| 0 | LOC '61 (OPTION-A memory increment cell) DMC channels for AMLC, SMLC, MAG TAPE AMLC buffers DISK driver (DVDISK) Disk I/O windows (4 pages) Mag tape I/O windows (6 pages) Mag tape dump window (1 page) IPC I/O window (2 pages) SMLC I/O windows (12 pages) |

- 1 Associative buffers for file system (64 pges)
- 2,3 MOVU2U segment windows
- 4 Interrupt catchers (phantoms)
 - Check catchers
 - Semaphores
 - Ready PCB list (loc '600)
 - Configuration common (FIGCOM) (loc '700)
 - Crash 9 track magnetic tape dump program (loc '776)
 - Memory parity scanner (loc '777)
 - WARM restart routine (loc '1000)
 - COLD start routine (loc '1400)
 - Memory usage map (MMAP)
 - Page maps (HMAP, LMAP)
 - Segment tables
 - Process control blocks (PCBs)
 - Interrupt fault table
 - Interrupt stack
- 5 Gate segment for direct-entrance PCLs
- 6 TMAIN, including
 - Supervisor and user fault catchers
 - SVC front-ends
 - Supervisor locked data (SUPCOM)
 - Clock process
 - Kernel procedures
 - Linkage frames for all supervisor modules
- 7 User terminal buffers
- 10 Per-user unlocked data (USRCOM)
- 11 File system procedures
- 12 Network data and procedures
 - SMIC data and procedures
- 6000 Ring 0 stack segment (one per user)

Modifying PRIMOS IV

Almost all of the changes that would normally be made to PRIMOS IV operations can be done with the commands of the CONFIG configuration data file.

It is also possible to modify the runtime version of PRIMOS IV, but it is not generally recommended. If patching PRIMOS IV, be sure to make copies of the files to be modified before attempting any modification.

The running version of PRIMOS IV can be patched using VPSD. From the supervisor terminal type OPR 1 and VPSD, perform the patches, exit from VPSD, and type OPR 0. To use this, one must add '4000 to the supervisor segment number to be patched (i.e. 4006 to patch segment 6). This procedure can also be carried out at a user terminal by first allowing read/write access to the segment to be patched using the SHARE command.

PRIMOS IV can be patched from the disk by attaching to the appropriate UFD and restoring the file PR00xx, where xx is the segment number to be patched. Do a PM to determine the starting and ending addresses and run PSD (or any version of PSD that is loaded above the ending address of PR00xx). After patching the file, save it and run the command file C_COLD to build a new memory image. The patches will become effective on the next cold start of PRIMOS IV.

An alternative method of patching PRIMOS is to generate a run file which contains just those locations to be patched (using the PMA ABS and ORG pseudo-operations). This file can be restored over an existing segment by using the SHARE command.

SYSTEM CONFIGURATION

Upon obtaining a master disk from Prime, the system administrator must install the desired version(s) of PRIMOS IV. PRIMOS IV supports 1 to 15 or 63 users (plus) the supervisor and a large number of peripheral devices including:

- 8 magnetic tape drives (MT0-MT7)
- 4 line printers (PR0-PR3)
- 4 card reader/punches (CR0-CR3)
- 1 paper tape punch/reader (PTP/PTR)
- 4 SMLC lines (SMLC00-03)
- 8 disks (or storage modules)
- 8 diskettes

Paging Space

The number of segments required by PRIMOS IV is given by the equation:

$$nseg = nusers + 9 + usersegs$$

nseg is less than or equal to the number of segments for a given version of PRIMOS IV (192, 144, or 320)

nusers is the total number of configured users: terminal + remote + phantom.

usersegs is the total number of segments to be available to users.

The default conditions are:

| <u>Version</u> | Number of Segments (<u>nseg</u>) | Number of Users (<u>nusers</u>) | User-Available Segments (<u>usersegs</u>) |
|-------------------------------------|--|---|---|
| 64-user | 192 | 64 | 119 |
| 16-user | 144 | 16 | 119 |
| 16-user with large address space | 320 | 16 | 295 |

The number of user-available segments can be increased by decreasing the number of configured users.

To preserve paging space, nseg may be decreased with the NSEG, PAGDEV, and ALTDEV commands of CONFIG (see section 16).

Paging disk (or partition) space requirements are given by:

$$\text{records} = (64 * \text{usersegs} + 8 * \text{nusers} + 256) * (\text{records/page})$$

records is the number of records required for paging.

records/page is 1 for storage modules and 2 for other disks.

The default conditions are:

| <u>Version</u> | <u>records</u> (storage module) | <u>records</u> (other disks) |
|-------------------------------------|------------------------------------|---------------------------------|
| 64-user | 8384 | 1678 |
| 16-user | 8000 | 16000 |
| 16-user with large address space | 19264 | 38528 |

Note

The calculation of nseg can be performed automatically by the system with the records parameter of the PAGDEV and ALTDEV configuration commands (Section 16). This allows the system to be started up with a specified amount of primary and alternate paging space.

Configuring Extended Instruction Set UII Package: It is possible to use a subset of the PRIME 500 instruction set on the PRIME 400, in particular, the decimal arithmetic and character string manipulation instructions. These instructions are generated as object code in programs compiled by COBOL in V-mode. If the XISUII package is installed, these 500 instructions will trap to the PRIMOS IV operating system and will be simulated by PRIMOS IV subroutines. First copy the file UI2000 from UFD=SYSTEM on the master disk to UFD=SYSTEM on the system disks. To install the XISUII package, type the following commands at the supervisor terminal or incorporate them into the system C_PRMO command file:

```
ATTACH SYSTEM 1/177777
OPR 1
SHARE UI2000 2000
OPR
A
```

PRIMOS IV SYSTEM INITIALIZATION

PRIMOS IV is started from PRIMOS II at the supervisor terminal.

The steps to get PRIMOS IV running are:

1. Turn on processor, disk drive, and all needed peripheral devices, and boot load PRIMOS II as explained in Section 2.
2. To ensure that PRIMOS IV is available, enter the command:

```
STARTUP command-device-number
```

if this has not already been done.

3. To initiate PRIMOS IV, issue the commands:

```
A primos-ufd
R PRIMOS
```

primos-ufd is the directory containing the runfiles and command files necessary to bring up the desired version of PRIMOS IV; PR4.64, PR4.16, or PR4L16.

To start up a version of PRIMOS IV supporting network enter the command sequence:

```
A PRINET
A primos-subufd 0 1
R PRIMOS
```

primos-subufd is the directory containing the runfiles and command files necessary to bring up the desired network-supporting version of PRIMOS IV, PR4.64, PR4.16, or PR4L16.

At this point the preloader (PRIMOS) attaches to UFD=CMDNC0 and looks for the command file C_PRMO. If this file exists, it is opened and executable commands read from it, otherwise, the system prompts:

PLEASE ENTER CONFIG

The user may enter either a PRIMOS III type CONFIG command or the name of a configuration data file.

CONFIG SYSA 21 110061 11460 2000 511 6/5 7/2

or

CONFIG -DATA CONFIG

Note

No restrictions are placed on the configuration data filename. Prime recommends that the name CONFIG be used as a standard.

In the following example, the configuration data file command has been placed in the file C_PRMO. Further, the printing of the CONFIG file has been enabled by the TYP0UT YES command.

```

OK:  R PRIMOS
GO
CONFIG -DATA CONFIG
PRIMOS IV <REV 15.0X-63>
-----

```

```

512.0K MEMORY IN USE
PLEASE ENTER DATE
.
.
.
contents of configuration file
.
.
.
GO
other commands in C_PRMO
.
.
OK, SETIME -050178 -0805
OK,

```

At this point the message LOGIN PLEASE is printed at all user terminals and users may log in.

CAUTION

User terminals will not respond to commands until the SETIME command is given by the operator.

PRIMOS IV Page Maps and Memory Map

For every user of the system, PRIMOS IV maintains two segment descriptor tables. Another set of two segment descriptor tables is identical for all users. Currently, Descriptor Tables 0, 1, and 3 are identical for any therefore, PRIMOS maintains Descriptor Table 2 for each user. These tables can be found at label DTARL in program TMAIN and are locked to physical memory. Each of 31 tables contains room for 16 entries of 32 bits, allowing for 16 segments per user. The detailed format of segment descriptor entries and page map entries may be found in REFERENCE GUIDE, SYSTEMS ARCHITECTURE AND INSTRUCTIONS, PDR3060. Each segment descriptor word either has the fault bit set, indicating no page map allocated, or points to a page map. The page maps may be found at label HMAP in program TMAIN. These page maps are initially set by a utility program called MAKM64, which is not part of the PRIMOS memory image. Each page map consists of two parts, a 64 16-bit entry page table and a 64 16-bit entry disk table. Each page table entry either has the fault bit set, indicating page not in memory, or points to one of 256 physical pages. Each page is a block of 1024 words of memory. Each disk table entry consists of a 2-bit lock field indicating if the page is locked to memory, and a 14-bit number that indicates to PRIMOS where on the paging disk the corresponding page in the page table is to be stored when it is paged out.

PRIMOS contains 64 page tables that are not locked to memory and are themselves paged.

PRIMOS maintains two tables to keep track of physical memory usage. One table (MMAP in TMAIN) contains one word for each physical page, and is 377 octal words long. Each physical page may have one of the following three states:

| <u>State</u> | <u>Meaning of</u> |
|--------------|--|
| -1 | Page is not available. An entry in MMAP is set to this value if the memory does not exist. It may also be set to this value by hand, or it may be set automatically if the memory is bad. |
| 0 | Page is available but not in use (not owned). |
| >0 | Page is in use. In this case, the MMAP entry points to the page map entry that is using that physical page. From the entry address, one can determine the page map that is currently using the page. The algorithm is: |

$$\text{page map number} = (\text{MMAP entry} - 1400) / 200$$

All numbers are octal; the division is done in octal.

The second table maintained by PRIMOS contains the user number and segment number of each page map. This table, starting at label PTUSEG in program TMAIN, contains 66 32-bit entries. After determining which page map is used by a particular MMAP entry, one can determine what user number and segment number is using that page map by doubling the page map number, using that number as an offset into the PTUSEG table and reading out the user number and segment number from the table.

PRIMOS IV Memory Parity Tests

On a cold start, PRIMOS performs a simple memory test for all memory except that used by the memory tester. The test is nondestructive, and correct parity is stored in every cell upon completion. Each cell must be capable of reading two patterns 052525 and 125252.

System Crash (Memory Parity Halt)

A memory parity error can be recognized in two ways. First, in STOP/STEP mode the address displayed in the panel lights will point to the instruction following the entry point MEMPA in segment 4 (source file SEG4). Second, DSWSTAT, register '35 in the real register file, can be examined. (The contents of DSWSTAT are described in the REFERENCE GUIDE, SYSTEMS ARCHITECTURE AND INSTRUCTIONS, PDR3060.) DSWSTAT will have bit 3 set (=1). When it has been determined that a memory parity error has occurred, the following steps should be taken. (See Appendix F for details.)

- First determine the physical page number (PPN) in which the parity error occurred. When the machine halts, the PPN is in the A-register and the word number is in the B-register and the users whose page got the error is in the X-register. The A-register can be displayed by fetching location 1 (the "300 register set"). The B-register may be displayed by fetching location 2 and the X-register by fetching location 0.
- It is next necessary to determine the owner of the virtual page in which the parity error occurred. Add the PPN to the location of MMAP -- word number '2000 in segment 4 and display the contents of the resulting location. (For example, if the A-reg = '47, display location '2047 in segment 4.)

If the contents of this location in MMAP are greater than '5200, the virtual page belongs to a user and can be safely mapped out of physical memory (made unavailable for paging). If the contents are less than '5200, the page is a supervisor page and cannot be mapped out.

Note

The actual virtual page number is also contained in DSWRMAH -- register '34 in the real register file. DSWRMAH may, however, be invalid if the parity error occurred during DMX transfer. It is safer to check MMAP.)

- If the parity error occurred in a user page, the page can automatically mapped out by selecting RUN mode and pressing START. PRIMOS IV will map out the page in error and halt. A warm start can then be performed.

It is also advisable to inform the user whose page was mapped out that that user cannot continue after the warm start. The user whose page got the error is displayed in the X-register following a memory parity halt.

If the parity error occurred in a supervisor page or if it is suspected that other locations in memory may be bad, it may be desirable to run the 'memory scanner'. This is a program that scans memory for parity errors and displays the location of each bad word found. To run the memory scanner after a parity error, MASTER CLEAR and START at location '777 (load '777 into location 7) with the ADDRESS/DATA switch on DATA and all sense switches down. When the display in the lights changes, the scanner has found a parity error. The panel display with all switches down is the word number of the parity error. The physical page number (PPN) is displayed by raising switch 15. The contents of the location in error are displayed by raising switch 14 (switch 15 down). To continue the memory scan, depress any key on the supervisor terminal. The scanner will continue scanning for the next error. The scanner will halt when the end of memory is reached.

After the memory scanner has been run, bad page(s) cannot be automatically mapped out -- this must be done by hand as follows:

1. As described above, add the PPN (displayed by the scanner when switch 15 is raised) to the location of MMAP -- '2000 in segment 4.
2. Note the contents of this location and then set the location to -1 ('177777).
3. The previous contents of this location are a word-number of a pagemap entry, also in segment 4. The contents of this location should be set to 0 (meaning that the copy of the page in memory is no longer valid).
4. A warm start can then be performed.

Note

If the computer entry address is below '2000, the bad page belongs to the supervisor and PRIMOS cannot be warm started.

Warm Start

After a successful warm start, the message ***** WARM START ***** is sent to all terminals, and all users are automatically restarted (e.g., typing START is not necessary). Because of this procedure, a warm start will not always cure a hung system. In general, do not attempt a warm start until after the system has halted.

Permanent Bad Page

If a page is known to be permanently bad, PRIMOS must be modified and saved with the appropriate pages marked 'unavailable'. For example, if page number 177 is known to contain a bad cell:

$$\text{MMAP} + 177 = 22000 + 177 = 22177$$

From PRIMOS II:

```
OK: A CMDNC0
OK: REST PRIMOS
OK: PM
SA, EA, P, A, B, X, K:
60 64777 1001 0 0 0 2000
OK: S 56000
$ U 22177 77777
22177 000000 177777
$ R 7000
OK: SAVE PRIMOS 60 65777 1001 0 0 0 4000
OK:
```

PRIMOS Halts

Locations at which PRIMOS IV halts are defined at the load map M_PRMO. This map is in the UFD in which PRIMOS IV was built and from which it is normally RESUMED. In general, it is not possible to accurately determine the cause of a halt condition by observing only the address displayed in the panel lights in STOP/STEP mode. This is because the halt may have occurred in one of several different possible segments. To determine the segment in which the halt occurred, display the contents of PBH -- the high side of register '14 in the current register set. (See Section 10 and Appendix F.)

All locations at which PRIMOS can halt are defined in the load map. A user determines the segment number and word number of the halt and looks for a definition of that halt on the load map. All halt definitions are of the form xxxxx_. The following halts exist:

| <u>HALT</u> | <u>MEANING</u> |
|-------------|--|
| AMLCI_ | Spurious AMLC Controller Interrupt |
| BDMEM_ | Memory Parity During Cold Start |
| BOOT0_ | SHUTDN ALL Stops Here |
| IFLTB_ | Bad fault in interrupt process. Any halt that occurs between IFTLB_ and IFTLB_ + 64 is of this type. |
| INTRT_ | Too many returns in interrupt process. |
| IPAGF_ | Bad page fault in interrupt process. |
| MCHK_ | Machine Check |
| MEMH2_ | Halt after automatic mapping out of a page. |
| MEMPA_ | Memory Parity Halt. |
| MMOD_ | Missing Memory. |
| PAGFB_ | Page fault when not allowed. |
| REFL0_ | FLEX, UII, PSU when not allowed. |
| RMCF0_ | Restricted mode fault when not allowed. |
| SVCF4_ | SVC when not allowed. |
| XRNG0_ | Illegal ring number in supervisor. |

APPENDIX A

BOOTSTRAPS

INTRODUCTION

When power is first turned on, the semiconductor read/write memory of a Prime CPU is empty. To make the CPU useful, a program (preferably an operating system) must be loaded into memory from an external device such as a disk. This is done by Prime's Automatic Program Load (APL) feature which works when the control panel rotary switch is in the LOAD position. To start this process, a short bootstrap program contained in a Read-Only Memory (physically located on the control panel) takes control temporarily. It loads a more extensive bootstrap from the external device into main memory. Control then passes to this longer bootstrap, which completes the job of loading the memory image of the operating system from the external device.

This appendix describes Prime bootstraps: control panel bootstraps and second level bootstraps stored on paper tape, magnetic tape, disk, or storage module.

Note

Certain of the devices described here are no longer supplied. The information is included here for reference and archival purposes.

CONTROL PANEL BOOTS

Control Panel Micro-code

A control panel can have either 256 or 512 16-bit words of PROM from which bootstrap programs can be loaded into memory. After pressing MASTER CLEAR and dialing the selection switch to the LOAD position, press the START switch to cause the control panel micro-code to read PROM locations '0 to '50 into memory locations '6 to '56 and begin execution in 16S mode at the address loaded into Location '7. This initial program, the pre-boot, can then read succeeding PROM locations into memory.

Prime Pre-Boot

The Prime pre-boot saves the A-register in location '57 and then selects among three classes of bootstraps and stores the appropriate code from the PROM into memory. The three classes of bootstraps are auto-start, paper tape, and mass storage boots. The user selects the desired boot by setting Sense Switches 14, 15, and 16 as follows (set=1, reset=0):

| <u>SS</u> = | <u>14</u> | <u>15</u> | <u>16</u> | <u>Code</u> | |
|-------------|-----------|-----------|-----------|-------------|------------------------------------|
| | 0 | 0 | 0 | =0 | Auto-start |
| | 0 | 0 | 1 | =1 | ASR paper tape (MDL format) |
| | 0 | 1 | 0 | =2 | High speed paper tape (MDL format) |
| | 0 | 1 | 1 | =3 | Fixed head disk |
| | 1 | 0 | 0 | =4 | Moving head disk |
| | 1 | 0 | 1 | =5 | Magnetic tape |
| | 1 | 1 | 0 | =6 | Floppy disk (diskette) |
| | 1 | 1 | 1 | =7 | Spare |

Device Specific Boots

Auto-Start (0): enters 64R mode and jumps to the location specified in Sense Switches 1 to 10 ('100 to '177700). If no address is specified, a default of '1000 is used.

Paper Tape (1 and 2): Modifies itself for either ASR or high speed paper tape (by sense switches) and reads a second-level MDL boot into memory.

Mass Storage (3-7): performs further selection for fixed-head disk (FHD), moving head disk (MHD), magnetic tape, diskette, and spare, all of which are loaded by the pre-boot.

FHD (3): Sense Switch 13 is used to select between controller 4001 (SS 13 reset) and controller 4002 (SS 13 set). Sense Switch 12 is used to select between device addresses '21 (SS 12 reset) and '23 (SS 12 set). The boot reads record 0 (448 word PRIMOS record format) of the disk starting at location '770 and begins execution at '1000 (via a JST '777). This boot waits for the drive to come ready and retries on status errors.

| <u>SS</u> = | <u>12</u> | <u>13</u> | |
|-------------|-----------|-----------|-------------------|
| | x | 0 | 4001 |
| | 0 | 1 | 4002, address '21 |
| | 1 | 1 | 4002, address '23 |

MHD (4): Moving head disks come in three varieties: two platter drives (6M-obsolete or 12M bytes), 20-surface drives (obsolete) and 40MB, 80MB or 300MB storage modules. Sense Switches 10, 11, 12, and 13 are used as follows (x - don't care):

| SS = | <u>10</u> | <u>11</u> | <u>12</u> | <u>13</u> | <u>14</u> | |
|------|-----------|-----------|-----------|-----------|-----------|--|
| | x | x | 0 | x | 0 | Cartridge, upper surface |
| | x | x | 1 | x | 0 | Cartridge, lower surface |
| | x | 0 | 0 | 0 | 1 | 20-head disk pack, lower surface ('21) |
| | x | 0 | 1 | 0 | 1 | 20-head disk pack, upper surface ('21) |
| | x | 0 | 0 | 1 | 1 | 20-head disk pack, upper surface ('23) |
| | x | 0 | 1 | 1 | 1 | 20-head disk pack, lower surface ('23) |
| | 1 | x | x | 0 | 1 | Storage module ('26) |
| | 1 | x | x | 1 | 1 | Storage module ('27) |

In all cases except the storage module, record 0 (448 words PRIMOS record format) of the selected surface of physical drive 0 is read into memory starting at location '770 and execution is begun at '1000 (via a JST '777). In the storage module, a 1040-word PRIMOS II record is read into memory starting at location '760 (to allow for the 16-word header) and execution is begun at '1000, as for other disk bootstraps. The moving-head disk and storage module bootstraps wait for the disk drive to become ready, and the bootstraps retry on the occurrence of status errors.

MT (5): Sense Switch 12 is used to select between 9-track (SS 12 reset) and 7-track (SS 12 set) magnetic tape drives. The boot starts up the drive, ensures that the tape is set at a loadpoint (space forward, abort, and rewind), and reads one tape record into memory from location '220 through '777 (4K). Execution begins at '1000 (via a JST '777).

DISKETTE (Floppy) (6): Reads record 0 into memory, starting at location '770. To maintain IBM compatibility, the boot alternately tries to read a 448-word PRIMOS record and a 64-word IBM record. Execution then begins at '1000 (via a JST '777). This boot waits for the drive to come ready and retries on status errors.

SPARE (7): Intended as a user-supplied down-line loader. Currently, halts at location '57.

Key-In Substitutes for Control Panel Boots

If you ordered a Prime computer without the auto-load control panel PROM function, hand keyed-in programs are necessary. Because programs keyed in are likely to disappear after one use, these programs should be as short as possible.

SECOND-LEVEL DISK BOOTS (BOOT)

The bootstrap, placed on all devices by the MAKE utility, is the same for all devices. The bootstrap is '1402 words long and occupies 1 record on storage modules and 2 records on non-storage modules. The control panel loads only the first record of the bootstrap into memory, but the first record locates and reads the remainder of the bootstrap when it needs to. The bootstrap record is read in starting at location '770; execution begins at location '1000 in 16S mode. This second-level bootstrap reads PRIMOS II into memory from any PRIMOS II disk on the system. The source is BOOT, residing in UFD=FILAID on the master disk. The run file on record 0 (or 0 and 1) is also called BOOT and resides in the MFD.

BOOT OPERATION

BOOT performs the following instructions:

- Cleans up parity, non-destructively, throughout memory.
- Sizes available memory.
- Requests, from the operator, which device to boot from.
- Attaches to the MFD on that device.
- Attaches to UFD=DOS.
- Depending upon memory size and/or sense switches, reads *DOS16, *DOS24, *DOS32, or *DOS64 in memory and starts them.

There are three possible outcomes of a boot operation: a successful boot, in which case PRIMOS II takes control; a detected error, in which case the boot returns to request which device to boot from; or an undetected error, such as non-existent device, in which case the boot hangs.

When started, the boot types:

PHYSICAL DEV =

The operator response should be the physical device number constructed according to the tables in Appendix G. Which PRIMOS II (*DOS32, *DOS64) is read in is determined by either memory size or Sense Switches 1, 2, and 3 in the following manner:

If all sense switches are reset (0), the highest memory PRIMOS II that will fit in available memory is read.

If any of the sense switches are set (1), they are treated as the most significant bits of the high address of memory +1. *DOS64 cannot be forced by setting sense switches.

| <u>Sense Switch Address</u> | <u>PRIMOS II Version</u> |
|-----------------------------|--------------------------|
| 0 | highest that will fit |
| '20000 | error |
| '30000 | *DOS16 (obsolete) |
| '40000 | *DOS16 (obsolete) |
| '50000 | *DOS24 (obsolete) |
| '60000 | *DOS24 (obsolete) |
| '70000 | *DOS32 |
| '1x0000 (x=don't care) | *DOS32 |

Once the boot has been successfully brought into memory by the control panel boot, it can be re-executed by MASTER CLEARing and STARTing at '1000. If a status error is detected on the device, BOOT will restart automatically. All drivers will wait for the device to come ready.

Building BOOT

The BOOT program is stored as a normal PRIMOS II SAVE file on a normal PRIMOS II format record (=0). Consequently, physical record 0 contains first an 8-word PRIMOS II record header, second a 9-word PRIMOS II SAVE file header, and the BOOT program itself. On a non-storage module device the bootstrap occupies 2 records. The 8-word PRIMOS II record header is eliminated by reading the record starting at '770 but starting execution at '1000, the first word of the 9-word SAVE file header. For the storage module, the PRIMOS II record header is 16 words long, but the record is read starting at location '760.

The SAVE file header is as follows:

```

word 0 = start address (SA = '3011)
1 = end address (EA = '4401)
2 = program counter (PC = 0)
3 = A-register (0)
4 = B-register ('110)
5 = X-register ('27)
6 = Keys ('74006)
7 = unavailable
'10 = unavailable

```

Since execution starts at '1000, the start address must be '3011, which is also a JMP '1011 (since the boot is guaranteed to be executed in 16S mode either by MASTER CLEAR or the control panel boot). The boot actually executes in sector '1000 and so must be loaded there (at '1011), then later moved (PSD or TAP) to '3011 and SAVED there. Because BOOT can never be executed as a supervisor terminal command (it cannot execute in sector '3000), the PC, A-register, B-register, X-register, and Keys are available as constants to be used by BOOT. They have been defined as follows:

PC = 0 Not used
 A = 0 Not used
 B = '110 Option A master clear default control register
 X = '27 SOC master clear default control register 1
 Keys = '74006 SOC master clear default control register 2
 left byte ('170) = SOC control word 2
 right byte (6) = number of end-of-line delay
 characters. This is not used by
 the bootstrap but is passed by
 PRIMOS II to set the supervisor ter

minal

typewriter control words.

Once a BOOT has been placed on a disk, it can be copied to another disk with the following command sequence:

```

A MFD XXXXXX 0
RESTORE BOOT
A MFD XXXXXX 1
SAVE BOOT
  
```

Note

The bootstrap can be changed to settings for other terminal characteristics either permanently (by SAVEing it with new B, X, and Keys parameters) or temporarily (by patching locations '1004, '1005, and '1006 of the copy in memory, followed by a MASTER CLEAR and RUN - see Appendix F). When changing any of these values, care must be taken not to change any of the others.

The run-file of the boot will be properly placed on a disk by the MAKE program. Also, the file *BOOT will be produced as a result of running the command file C_MAKE. *BOOT is a copy of the boot exactly as it appears on record 0 of a storage module, and hence can be restored and saved, just as the file BOOT in an MFD.

APPENDIX B

THE SPOOLER

SPOOL PRINT PROGRAM

To start the spooler phantom, attach to the SPOOLQ directory and 'phantom' one of the following command files:

| <u>Phantom Command File</u> | <u>Device</u> | <u>Program Used by Phantom</u> |
|-------------------------------------|---------------------|--|
| PH_PR0 | MPC printer 0 (PR0) | *SPPR |
| PH_PR1 | MPC printer 1 (PR1) | *SPPR 3/1 |
| PH_CEN | Serial printer | *SPPR 2/1 |
| PH_PLT | Printer/plotter | *SPPR 2/2 |

Each phantom command file assigns the appropriate device and then starts the spool print program with the proper register settings.

The program attaches to the first available SPOOLQ directory and creates a file RUN.##, where ## represents the spooler user number. This file remains open for the duration of time the spooler is running and therefore provides a means for the spool program to determine if a spooler is running on a given user number.

Operator communication is handled by small message files that consist of the user name and number originating the request, the request code, and any other data that is required (paper type, for instance). The spooler, once having processed the request, returns to the originator an acknowledge file. This file contains a code informing the user as to whether or not the request was executed, spooler status, and/or current form type.

All communications files are temporary and are deleted by either the spool phantom or the originator. In addition, the spool phantom periodically scans the spool directory and deletes any stray RUN.## files.

INSTALLATION OPTIONS

The user may elect to modify the spooler to suit specific installation requirements. Following is a list of features that the user may wish to change:

Print/Plot Option

As supplied, the spool phantom queue scan algorithm only honors print files (those without the plot bit in the option word set). To modify this, edit the file SCANQ in the spool source directory and change the data statement defining the variable PRTPLT to one of the following values:

- 1 - recognize print-only files (default)
- 2 - recognize plot-only files
- 3 - recognize print/plot files

Short/Large File Threshold

The variable LGDEF defining the short/large file threshold, exists in both the spool program (SPOOL) and the spool print program (SPPR). To change this, modify the data statement defining this variable in both SPOOL and SPPR and set the variable to the record size at which a file is determined to be 'large'. To disable this feature, set the variable to -1.

Default Paper Size

To change the default paper size, modify the spooler start-up command file (PH_xxx) to set the A-register to the number of lines per page when the spooler phantom is resumed. For example:

```
R *SPPR 1/44 3/1
```

sets the default paper size to 44 lines/page for line printer 1.

After any modifications are made, the spool phantom and/or the spool program must be rebuilt as follows:

- SPOOL program - run command file C_SPOO.
- SPOOL phantom - run command file

```
C_PR0 for MPC printer 0
C_PR1 for MPC printer 1
C_CEN for Serial printer
C_PLT for Electrostatic printer/plotter
```

To install the new spooler, either use FUTIL to copy the new run files to their respective directories or run the command file C_INST.

QUEUE STRUCTURE

The spooler uses the UFD=SPOOLQ to store files in the print queue. If multiple systems are connected over a network, one of these directories should exist for each system. In addition to the temporary print files, a control file, Q.CTRL, used by the queuing mechanism is in

this UFD. This file contains information about each entry in the spool queue, queue top and bottom pointers, and a locking flag and counter to provide integrity when multiple readers and one writer are allowed to access the file simultaneously.

The queue structure permits a maximum of 200 entries. Any attempt to exceed this will yield an error message from the spool program.

The queue file is a fixed length, "circular" file (i.e., when the last physical entry is written, it wraps around to the first entry). There is one 6-word file header and 200 40-word data entries. Each data entry has a 1-word header, which is set to 1 if the data entry is valid and 0 if the entry is null (invalid). The head pointer points to the next entry to be popped from the queue and the tail pointer points to the next entry to be added. Because the head and tail pointers are equal when the file is empty as well as when it is full, an 'empty' flag was included in the header. If the flag is set to 1, there is one or more entries in the queue; if 0, the queue is empty.

The spooler is insensitive to the system read/write interlock value as long as multiple readers and writers are not permitted. The queue is opened for reading by the spool phantom to look for the next file and by the spool program for queue interrogation. When an entry must be added or updated, the file is temporarily re-opened for writing. Should this re-open fail, the file is closed for a fixed length of time (someone else may be attempting the same operation, in which case, not closing the file could cause a deadly embrace). If the file is not available after a fixed length of time (currently over 2 minutes to allow for remote (network) access), an error message is printed and return is made to PRIMOS command level.

To insure data integrity when multiple users are accessing the queue file, the queue subroutine package returns a flag on a close or lock function, informing the spooler proper whether or not someone has been updating the file while it has been open. Using this, it is possible for the spooler to determine if it should re-read a certain entry to verify accuracy.

Word Contents

| | | |
|--------------------|----|----------------------------|
| Queue File Header: | 0 | lock flag |
| | 1 | lock counter |
| | 2 | head pointer |
| | 3 | tail pointer |
| | 4 | entry size |
| | 5 | empty flag |
| Queue Entry Data: | 0 | . |
| | 1 | user name |
| | 2 | . |
| | 3 | . |
| | 4 | file name |
| | 17 | . |
| | 18 | . |
| | 19 | . |
| | 20 | form type |
| | 21 | . |
| | 22 | month |
| | 23 | day |
| | 24 | year |
| | 25 | time |
| | 26 | options |
| | 27 | defer time |
| | 28 | file size |
| | 29 | . |
| | 30 | pt file name |
| | 31 | . |
| | 32 | raster scan size (plot) |
| | 33 | . |
| | 34 | . |
| | | reserved |
| 98 | . | |
| 99 | . | |

Figure B-1. Queue File Format

APPENDIX C

PRIMENET

OVERVIEW

PRIMENET allows one Prime 300 (or higher) to communicate with others. A group of Prime systems can pool and share their files and other resources with each other. In addition, a user on one system has access to the capabilities of another, no matter where the other computer is located. Local networks are implemented with the Inter Processor Controller (IPC); data communications facilities can be used to access remote systems. This capability is available with no significant increase in complexity. The PRIMENET functionality allows users to access and use the resources of remote Prime systems as if they were available at the user's own system.

PRIMENET Size

In a system in which PRIMENET has been configured, approximately 15-20 K-bytes are required for the IPC and file sharing software. If synchronous communication is also desired, about 16 K-bytes of additional memory is required. The File Access Manager (FAM) operates in user space as a user (usually a phantom) and requires a working set of approximately 32 Kbytes.

PRIMENET Structure

PRIMENET structure has been designed and implemented so that the user interface is simple and transparent for most users. Except for user-to-user communication (which requires use of the network primitives), the operation of the network software is completely transparent to those using it. Running on a remote system from a local node of the network, or accessing remote files requires no reprogramming of user applications or extensive retraining of users. All the intricacies and communication protocols of the network are handled by the PRIMENET software and its attendant hardware; this is invisible to the user.

To provide this "invisible" interface and to allow for future enhancements to be added transparently, the PRIMENET software has been implemented in layers. Each layer supplies a specific set of functions to be performed on behalf of the user. To a large degree, these layers function independently.

Hardware Interface: The lowest layer provides the interface to the hardware component which interfaces to the external link: the IPC or high-speed SMIC. This software layer handles the specific characteristics associated with the hardware controllers.

Link Protocol: This layer is concerned with the link protocol, in a point-to-point configuration with the HSSMLC or up to 7 systems with the IPC.

The software supporting the HSSMLC is a modified Telenet host access protocol. This layer buffers the user from the particular protocol that is used on the link. It operates bi-directionally over full duplex common carrier facilities using IBM-compatible Binary Synchronous Communication (BSC).

Network Manager: This layer is responsible for managing the message formatting conventions and the flow of messages between systems. The user communicates with this layer when using the PRIMENET primitives. The File Access Manager (FAM) also communicates with this layer when operating on behalf of the user (e.g., in a remote file access).

The FAM provides the transparent PRIMENET link for most users. This software subsystem is invoked by PRIMOS when remote PRIMENET communication is needed. It uses the same primitives that are available to the user and operates on behalf of the user to provide the appropriate network function.

TYPES OF PRIMENET CONNECTIONS

The PRIMENET software is an optional enhancement to the PRIMOS III, IV, or V operating systems. It provides the direct link between the Prime local system that allows simultaneous communication via the IPC local link, or remotely via the HSSMLC and modems. The PRIMENET software allows the user to perform remote file sharing and remote log on with no programming changes as soon as network hardware and software are installed.

PRIMENET Interprocessor Controller (IPC)

Local Prime 300 or higher systems can communicate with each other by using Interprocessor Controller hardware and PRIMENET software. Up to 7 Prime systems (using PRIMOS III, IV, or V) can communicate with each other. Users on one system can communicate directly with each other or with users on the other systems. Users on one system can log on remotely to other systems and/or access files on those systems (if permitted by protection keys). Users may access files on disks that are physically connected to a system other than the one on which they are running. This file access is done with the standard file system calls.

The IPC hardware controller attaches to the Prime 300 (or higher) I/O bus and can interface up to 7 local Prime systems to form a network. The controller runs in 16-bit parallel transfer mode to achieve maximum data rates between systems. Each controller interfaces to a data bus operating in a contention manner. That is, the first IPC to request the bus, seizes it with ties resolved by a built-in hardware priority

scheme. Once the controller has the bus, it transmits a block of data (up to 512 bytes) to the target system using direct memory access (DMA). Once the block transfer is complete, another controller may seize the bus and begin data transmission. A parity bit is included on every byte transferred with parity checked automatically by the hardware. Each message is acknowledged by the receiver for proper reception. If not acknowledged (or negative acknowledged), the sender will retransmit the data.

Figure C-1 shows how several Prime IPC systems may be connected together.

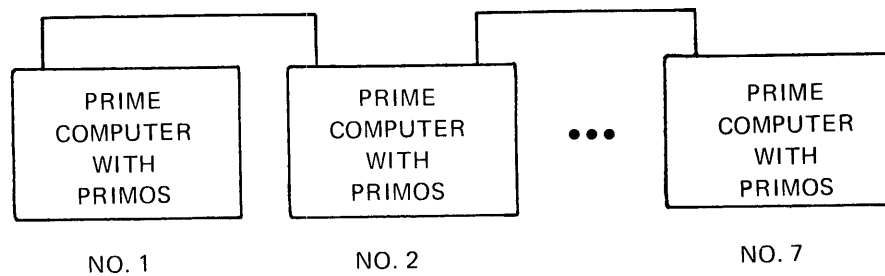


Figure C-1
Prime Computers with IPC Connection

The controller is designed so that the middle systems (2 to 6) may be powered down without affecting network operation. If either of the end points loses power or if the cable between any nodes is disconnected, the network ceases to function. The aggregate length of all cables connecting IPC controllers must not exceed 200 feet.

PRIMENET Synchronous Link (SLC)

The same user-visible functionality available with the IPC is also provided between remote Prime systems with a communications link. Telephone line communication is accomplished with the High-Speed Synchronous Multi-Line Controller (HSSMLC) equipped with binary synchronous protocol microcode option. The microcode on the HSSMLC assumes the communications protocol overhead. The CPU receives the data as if it were being read from a tape drive. The microcode provides and strips all the message-framing characters and generates and checks the 16-bit Cyclic Redundancy Check (CRC) polynomial on all communication transmissions. The user has all IPC functionality but at a lower speed; data transfer between systems being limited by the telephone line speeds. The HSSMLC supports speeds up to 9600 bits/second.

REMOTE OPERATIONS

Remote File Access

PRIMENET allows network system users to access files on remote systems. The standard PRIMOS file system calls (ATCH\$\$, SRCH\$\$, etc.) may be used to manipulate the files. This capability is provided by the FAM subsystem. With FAM, all PRIMOS language subsystems, MIDAS, and FORMS, operate on local and remote files identically as seen by the user.

When attempting to attach to another directory, the local PRIMOS file system is first searched. If not found, PRIMOS generates a request to its local FAM and suspends the user. The FAM immediately packages this request into a message to the appropriate remote FAM. The order of search of the remote system(s) is determined by the logical device numbers of the remote disks.

FAM operation is independent of the specific network hardware connecting the nodes, although this will affect speed of network file operation.

User dialog to access FILE.1 on System B by User on System A (See Figure C-2).

ATTACH TD User, from system A, wants to access FILE.1 located in UFD=TD on System B. System A searches for TD first, then FAM searches for TD on System B.

FIN FILE.1 User performs desired operation on FILE.1. All operations are performed independently of file location.

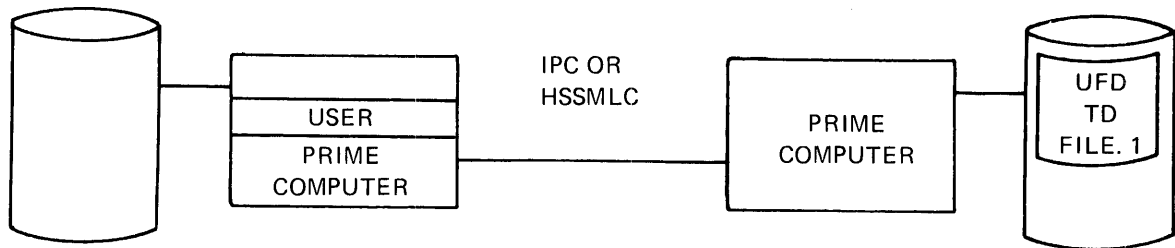


Figure C-2
Remote File Access

Remote Login

Users may log in remotely to a Prime 300 (or higher) while connected to the local system. The network software establishes a connection from the user to the remote system. To initiate a remote session, the user must supply a directory and, if necessary, a valid password on the remote system. The command format is:

```
LOGIN ufd-name [password] -ON nodename
```

ufd-name the directory on the remote system to which the user logs in

password the directory password, if necessary

nodename the name of the remote system on which the directory resides.

For example, a user at a terminal connected to System D wishes to log in to UFD=TEKMAN on System B. The command is:

```
LOGIN TEKMAN GREEN -ON SYSB
```

Remote login allows the remote use of specific processor functionality or the sharing of peripherals that may be attached to only one system. The user may log in to the only system in the network with magnetic tape drives and run a program requiring tape processing. Hardcopy output may be printed at a remote site or transferred back to the local system and then printed. Print output may be transferred automatically from the system on which the program was run to the system on which printer(s) reside with the spooler. Figure C-3 shows an example of such resource sharing.

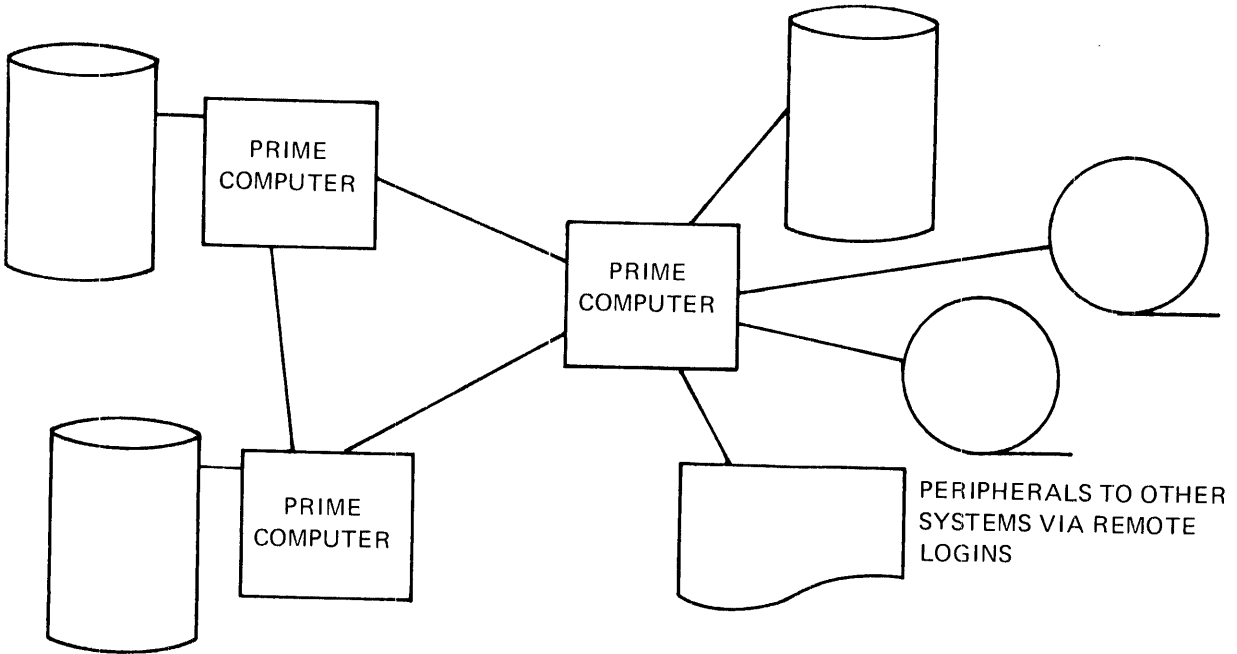


Figure C-3. Resource Sharing

OPERATOR FUNCTIONS

The operator has commands to configure the local system in PRIMENET and to perform operations across the network (starting up remote disks, sending messages, etc.)

System Configuration

The local system is configured with the CONFIG command (see Section 16 for a complete description). The following commands allow the operator to:

- Define local node-name: MYNAME. The local node-name can also be defined by the one-line CONFIG command.
- Define the type of linkage between the local node and all other nodes of the system: FAM.
- Specify the network configuration parameters for all lines from

the local system: NET.

- Specify which nodes local users may log in to remotely: RLOGIN
- Reserve the number of processes for remote users from other nodes: NRUSR. This can also be reserved by the one-line CONFIG command.
- Enable and configure the SMLC lines: SMLC. This can also be done, in part, by the one-line CONFIG command.

System Operations

Details of the commands below will be found in Section 15.

- Disks physically connected to remote systems may be made available to local users by the ADDISK or STARTUP commands. For a remote disk to be available, it must have already been started up at its local system. Any attempt to connect a non-existent or not-ready device causes an error message to print at the supervisor terminal. Remote devices may be disconnected from the local system with the SHUTDOWN command.
- Remote processes (file access, remote users, etc.) can be disconnected with DISCON.
- Information may be sent to the operator of a remote node with: MESSAGE.
- Network information including node status, remote logins to and from the local node, remote disks, etc., may be obtained with STATUS.

USER-TO-USER COMMUNICATION

PRIMENET allows communication between users on the same or on different systems of the network. Communication is independent of the network hardware and is implemented by a set of primitives (subroutine calls). These allow program-to-program communication to be implemented directly in FORTRAN and PMA. Interface routines allows this to be used by other languages. Communication over PRIMENET is independent of protocol, link type, and speed. See PTU52 for details on these subroutines.

PRIMENET Primitives

SUBROUTINE

FUNCTION

Establish Call

| | |
|--------|---|
| NETLNK | Establishes user as a network user so transmission and reception is possible. |
|--------|---|

CONNECT Performs NETLNK function, if not previously done. Requests connection to remote user/system supplied by argument. Two matching connect requests establish a connection.

GETCON Supplies user with user/system name of other users who wish to connect to this user.

RJCON Rejects the connection of user wishing to communicate with this user.

Move Data

TRNMIT Sends a block of data to another user. Multiple TRNMIT calls may be executed.

RECEIV Accepts the first/next block of data from another specified user. PRIMENET informs receiver how many words are received. Multiple RECEIV calls may be executed.

Utility

DISCON Disconnects communication with another user immediately.

NTSTAT Determines network status (number of nodes, node information, state of link, etc.)

UNLINK Terminates all operations of this user on the network.

NETWAT Puts user in inactive state and resumes operation when a significant event (operation complete, link down, etc.) has occurred.

APPENDIX D

THE EVENT LOGGER

OPERATION OF THE EVENT LOGGING SYSTEM

User Accessable Event Logger (LOGEV1)

Information about an event is entered into the event buffer, LOGBUF, by LOGEV1. This program is an internal PRIMOS subroutine that also may be called by the user to record user-specified events. Each entry in the buffer contains the type and length of the entry and a number of data words passed to LOGEV1 by the PRIMOS programmed to record the event. (The exact format of event entries is described below.) When LOGBUF fills up, LOGEV1 discards subsequent entries and increments LOGOVF, a counter of the number of events lost.

System Event Logger (LOGEV2)

The internal subroutine, LOGEV2, periodically examines LOGBUF and, if it is non-empty, dumps it to a disk file named LOGREC in the current UFD of the user known to PRIMOS as User 1. Normally, the UFD is CMDNC0 on the command device. LOGEV2 does not dump LOGREC until the time has been set by the system operator. LOGEV2 is called from two places in the PRIMOS supervisor: SCHED (COMXIT) when the one-minute alarm is set and DOSSUB when a SHUTDOWN ALL command is issued. LOGEV2 is not normally called by the user, but the user can monitor the output of LOGEV2 via the LOGPRT command.

LOGEV2 does not dump LOGBUF if the file LOGREC does not exist in CMDNC0. This allows operation with a write-protected disk.

Note

If the command device (disk) is write-protected and a LOGREC file exists in CMDNC0, a disk WRITE-PROTECT error message is printed on the supervisor terminal every minute.

The LOGREC file can be created with any sequence of PRIMOS commands equivalent to the following:

```
A CMDNC0 0 1
L LOGREC
C 2
A
```

The size of LOGREC is controlled by a parameter in LOGEV2 and is currently 4096 words. If LOGREC exceeds this size, LOGEV2 prints the message: EXCEEDING QUOTA ON LOGREC at the supervisor terminal. It

will, however, continue to log information to LOGREC and print the message every minute until LOGREC is emptied by the LOGPRT routine. (Alternatively, of course, LOGREC can be deleted.) Under PRIMOS IV or V, the size of LOGREC can be changed by the LOGREC command in the CONFIG data file. The quota checking can be disabled or writing to LOGREC can be suppressed.

Before dumping LOGREC, LOGEV2 writes an entry to LOGREC noting the current time and date. After LOGREC is dumped, if LOGOVF (the overflow counter) is non-zero, LOGEV2 writes an entry noting the number of LOGBUF overflows.

Note

To ensure that LOGEV2 gets a chance to dump the contents of LOGBUF, a warm start should be performed whenever possible before dumping LOGREC. wait one minute or issue a SHUTDN ALL.

MODIFYING THE EVENT LOGGING MECHANISM

The following paragraphs describe how to make modifications to the event logging mechanism. The relevant modules are found as follows: For PRIMOS IV and V: LOGEV1 and LOGBUF are in PRI400>KS>SEG4. LOGEV2 is PRI400>KS>LOGEV2. For PRIMOS III, IV, and V LOGPRT is supplied in SYSTEM>LOGUFD.

Increasing The Size Of LOGBUF

LOGBUF is defined in SEG4 (PRIMOS IV and V) or TMAIN (PRIMOS III). The first entry in the buffer (label LOGBUF) is a COLD START entry. The following BSZ defines the remaining size of LOGBUF (63). It can be redefined as desired.

Adding Event Types

To log a new event type, three actions are necessary:

1. An event message must be built that contains the event type, length of the message, and (optional) data words.

Event Message Format: An event message consists of a header word followed by up to 23 optional data words. The header word consists of the event type in bits 1-8 and the total message length in bits 9-16. In PMA, a message could be defined by:

```
MSG DATA (5.LS.8)+3,DATA1,DATA2
```

Currently Defined Event Types: Currently, the following event types are defined.

- 0 - COLD START
- 1 - WARM START
- 2 - DATE/TIME STAMP (LOGEV2)
- 3 - CHECKS (MACHINE, MEMORY PARITY, MISSING MEMORY)
- 4 - DISK ERRORS
- 5 - LOGBUF OVERFLOW (LOGEV2)
- 6 - SHUTDN ALL
- 7 - PRIME 300 MACHINE CHECK
- 8 - PRIME 300 MEMORY PARITY
- 9 - PRIME 300 MISSING MEMORY
- 16 - DISK MOUNT

2. LOGEV1 must be called to enter the message into LOGBUF.

Calling LOGEV1 -- PRIMOS III

In PMA:

```
CALL LOGEV1
DAC MESSAGE
```

In FORTRAN:

```
CALL LOGEV1 (MESSAGE)
```

Calling LOGEV1 -- PRIMOS IV or V

In PMA, code inside SEG4:

```
JSXB LOGEVL (NOTE DIFFERENT NAME)
IP MESSAGE
```

In PMA, code outside SEG4.

```
CALL LOGEV1
AP MESSAGE,SL
```

In FORTRAN:

```
CALL LOGEV1 (MESSAGE)
```

3. LOGPRT must be modified to recognize the new event type and appropriately format the data associated with the event. LOGEV1 and LOGEV2 do not examine the type field.

Modifying LOGPRT: Currently, LOGPRT recognizes and formats data for event types 0-9. Types 10-15 are accepted, but result in a printout of only

TYPE=type DATA=word-1 word-2 ... word-7

Only seven data words are allowed for these types. To add a new type, add a label to the computed GOTO following statement \$400. At the new label (between \$1950 and \$2000), call the STORE routine to perform the required formatting.

The calling sequence for STORE is as follows:

```
CALL STORE (TEXT, TXTLEN, ARRAY, NW, DEC)
```

The meaning of the parameters is as follows:

| | |
|--------|--|
| TEXT | A text string to be printed. |
| TXTLEN | The length in characters in TEXT. If zero, no text is printed. |
| ARRAY | An array of words to be translated and entered in the output line. ENTRY(1) is the first data word of the event message. ENTYP and ENLEN contain the type and length of the entry. |
| NW | The number of words in ARRAY. If zero, no words are translated. |
| DEC | Octal/decimal flag. If zero, translation is to octal with no leading zero suppression. If non-zero, translation is to decimal with leading zeroes suppressed. |

The total length of the text to be stored ($=\text{TXTLEN} + \text{NW} * 7$) should not exceed 67, the maximum length that can be printed at a terminal with an indent in effect. (All lines after the first for an entry are indented 5 spaces.) If the length of text is too long, TEXT TOO LONG (STORE) will be printed.

After formatting the entry, GOTO 2000. Code at that label finishes the formatting and obtains the next entry from LOGREC.

To rebuild LOGPRT, run the command File CLOG in SYSTEM>LOGUFD. This will create a run file called *LOG. *LOG is then moved to the UFD SYSTEM and renamed to LOGPRT.

Changing The Size Of LOGREC

The size of LOGREC (over which the EXCEEDING... message is printed) is defined in the source file LOGEV2 by the integer variable QUOTA (currently 4096 words). This can be modified as desired (or patched in PR0006).

Changing LOGPRT's Default Input/Output Filenames

The default input name, <0>CMDNC0>LOGREC, is in the array INPNAM. The size of INPNAM and length of the name, INNAML, should be set to the number of words and characters in INPNAM respectively. The default output name (LOGLST) is in the array OUTNAM, which is always 16 words long, blank padded.

APPENDIX E

DISK ERRORS

GENERAL

All record, device, and status numbers are octal.

STORAGE MODULES

Storage Module Error Detection

Under PRIMOS III, IV, or V, disk read/write errors on storage modules generate the error message:

DISK xx ER phys-dev rec-num(2) act-rec(2) status retries

xx is RD for a read error and WT for a write error

phys-dev is the module or partition on which the error occurred

rec-num(2) is the desired record number within the partition (2 words).

act-rec(2) is the actual record number (in the partition) read (2 words). On reads it should agree with the rec-num(2). On writes it has no validity.

status indicates reason for error. It may be one of the words in the table below or the sum of two or more.

| <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 |

retries the number of times the read or write operation was attempted unsuccessfully before printing the error message, normally 10 ('12).

Under PRIMOS II, the error message format is:

```
DISK xx ERROR phys-dev rec-num(2) act-rec(2) status
```

The definitions are the same as for the PRIMOS III, IV, V messages. There is no retry-count; PRIMOS II prints the message and tries ten ('12) times.

Error Correction

The storage module controller writes a two-word correcting code checksum on each record. It is generated by the following polynomial:

$$G(X) = X^{**} (1+X^{**2}+1)*(X^{**21}+1)$$

An error detection and correction scheme (ECC) is implemented for the storage module. The code, together with the correction logic in the storage module controller is capable of detecting any of the following: (1) up to two error bursts of combined length up to 22 bits; (2) a single error burst of up to length 32 bits; and (3) any odd number of errors. In addition, the ECC algorithm is capable of correcting any single error burst of up to 11 bits in length. Error correction is attempted only after ten attempts to read a record have failed.

The message printed when storage module error correction is attempted and fails is:

```
UNCORRECTABLE.
```

The message printed when error correction succeeds is:

```
wordno error error
```

wordno Is the offset relative to the beginning of the record at the beginning of the correction.

error error Is the 32-bit correction pattern (two-16-bit words) (of which at most, 11 consecutive bits will be non-zero).

Example

```
DISK RD ER 020063 000000 016357 000000 015477 100014 000012
UNCORRECTABLE.
DISK RD ER 020063 000000 016360 000000 016355 100014 000012
UNCORRECTABLE.
```

OTHER DISKS

Disk errors messages are printed in the same format as storage modules; the meaning of status-word is different.

DISK RD ERROR device-number record-address cra status-word

or

DISK WT ERROR device-number record-address cra status-word

On read request errors cra (2 octal words) is the actual record number read and should match the requested record record-address (2 octal words). It has no validity on a write request.

There is no way, in a program, to trap a detected disk error. Under PRIMOS II a message is printed and the operation is continually retried. Under PRIMOS III, IV, or V the operation is tried ten times.

Status Word

The meaning of the status word, typed as the right-most octal number of a disk error, depends on the type of controller as follows:

4000 Controller (obsolete)

| <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 (cartridge disk)

| <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 |

APPENDIX F

CONTROL PANEL

This appendix describes operation of the Prime central processors (CPU) from the control panel.

The control panel procedures of most importance to the operator in a multi-user environment are:

- Using autoloader to bootstrap the operating system during startup.
- Examining the contents of register and memory locations following an operating system crash.
- Setting up for automatic restart after power failure.

In addition, while running single-user PRIMOS II, the operator must start and stop program operation from the panel.

Other control panel capabilities such as program patching and single step operation are primarily useful to field service technicians, but are included in this Appendix for reference purposes.

CONTROL PANEL FEATURES

All Prime control panels have the same physical switch and indicator layout. (See Figure F-1.) In general, operation of all panels is identical, except that the Prime 350 and above have additional capabilities permitting:

- Accessing register sets.
- Addressing real memory using a full 22-bit address.
- Addressing mapped memory when segmentation is enabled.

These advanced features are enabled by sense switch settings that have no effect in the Prime 300 and below.

RUN Functions

RUN mode is the normal control panel state for execution of programs. In RUN mode, the control panel has no effect on processor operation, but the 16 sense switches and indicators act as programmed input/output devices. A hardware register in the panel controls the indicators; it can be loaded with data from the A register by an OTA instruction. The sense switches can be read into the A register by an INA instruction, or can be tested by SKS instructions.

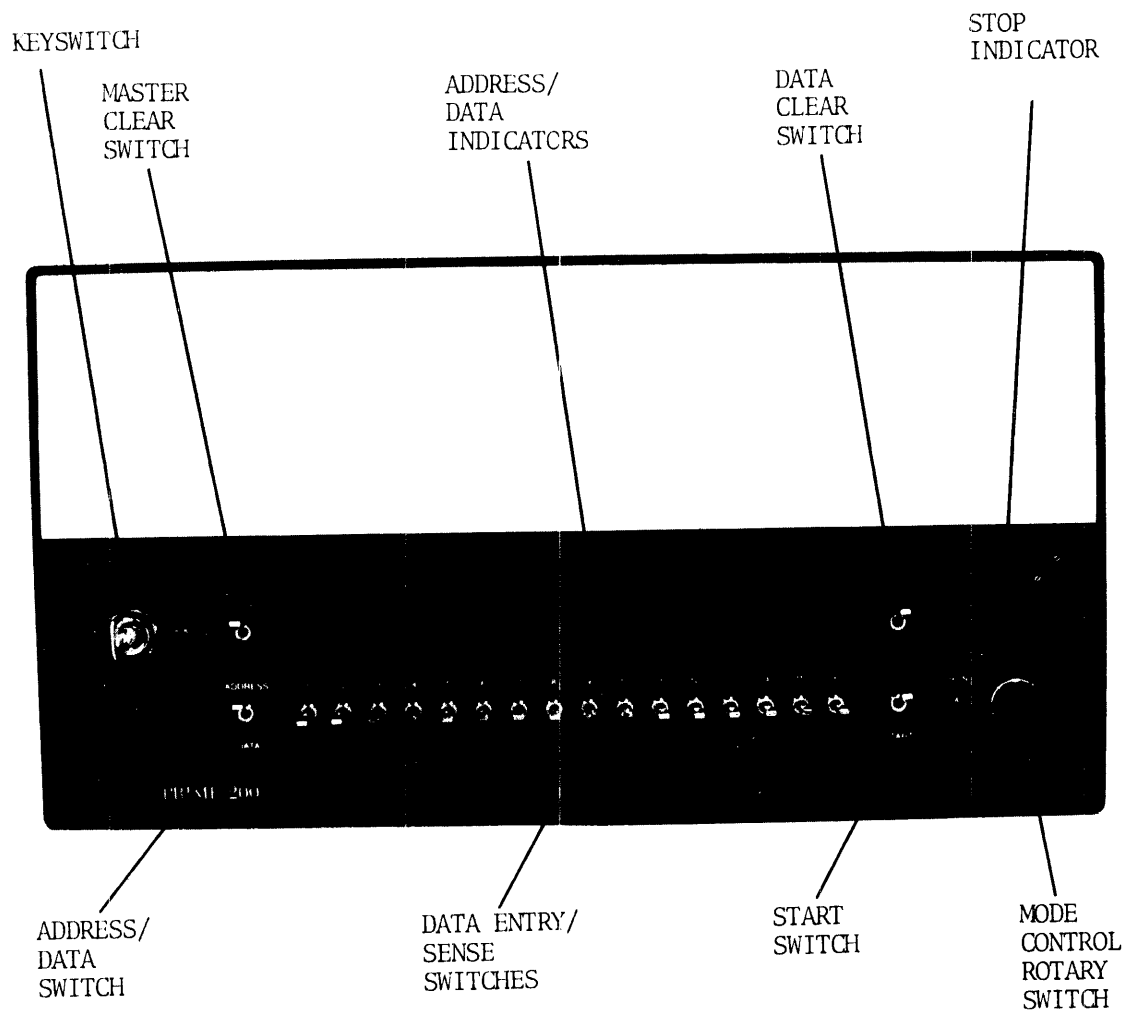


Figure F-1. Prime CPU Controls and Indicators.

Data transferred to the indicator register from the CPU is displayed when the ADDRESS/DATA switch is in the DATA position. In the ADDRESS position, the indicators present a dynamic display of whatever is present in the CPU's BMA (memory address) bus.

Single Step Functions

Immediately after power-up or whenever the rotary switch is turned to STOP/STEP, the CPU is halted (STOP light is on), and the control switches on the panel enable the operator to access, modify, or execute single 16-bit cells of high-speed memory (including the register files).

Access (FETCH): The memory cell specified by the indicator display when the ADDRESS/DATA switch is at ADDRESS is accessed by turning to FETCH and pressing START. The content of the cell is then displayed by the indicators when the ADDRESS/DATA switch is at DATA.

Modify (STORE): The contents of the addressed memory cell can be modified by setting up a new data display, turning to STORE, and pressing START.

Consecutive FETCH or STORE: The FETCH Y+1 and STORE Y+1 functions permit accessing or modifying of consecutive memory locations. The address is incremented by 1 automatically when the START switch is pressed before the operation is carried out.

Single-Step Execution: While the rotary switch is in STOP/STEP, each operation of the START switch executes the instruction designated by the content of the P register, memory location 7. All levels of indirect addressing and indexing are carried out, and the instruction is executed. The address of the next instruction to be executed is loaded into the P register (and displayed on the indicators if ADDRESS/DATA is at ADDRESS). After executing each instruction, the operator can FETCH or STORE other registers or memory locations. When the ADDRESS/DATA switch is at DATA, the indicators display the data or instruction in the last accessed location.

LOAD Functions

The LOAD position of the rotary switch enables Automatic Program Load (APL), which simplifies program loading and permits automatic restart (or reload) after a power failure.

Other Features

Master Clear: The MASTER CLEAR switch (operative in all rotary switch positions except LOAD and RUN) places the CPU and all peripheral controllers in a known starting condition.

STOP Lamp: The STOP lamp lights when the CPU is in a halt condition (either from a panel halt or execution of a HLT instruction).

Battery Backup Indication: The STOP lamp flashes on and off if there has been a power failure and the memory is being refreshed by the backup battery pack (optional).

Keyswitch: A rotary keyswitch provides line power on-off control and a lockout position that disables the control panel except for sense switch settings and indicator displays controlled by the running program.

CONTROL AND INDICATOR FUNCTIONS

Functions of the Prime control panel switches and indicators are described in Table F-1. Sense switch functions peculiar to the Prime 350, 400 and 500 are shown in Table F-2.

Table F-1. Control Panel Functions

| Item | Function |
|-------------------------|--|
| Keyswitch | Power control and panel locking rotary keyswitch. |
| | <u>Position</u> |
| OFF | Line power is removed from Prime CPU; the power supply and battery backup are inactive. |
| ON | Line power is applied to the Prime CPU power supply and control panel rotary switch determines CPU mode of operation. |
| LOCK | Same as ON, but control panel rotary switch has no effect on CPU run condition. (Data switches and indicators can be accessed by programmed I/O, however, and ADDRESS/DATA switch functions normally). |
| MASTER CLEAR Switch | Momentary-down switch sets CPU registers and status to known starting condition (see text entitled Master Clear). |
| Address/Data Indicators | What these indicators display depends on the the position of the rotary mode switch, the ADDRESS/DATA switch and the last operation that took place: |
| Rotary Switch | |
| | <u>Position</u> |
| | <u>ADDRESS/DATA Switch Position:</u> |
| | <u>ADDRESS</u> |
| | <u>DATA</u> |
| RUN or LOAD | Contents of CPU's BMA (address) bus |
| | Content of panel's indicator register (loaded by OTA '1720) |
| STOP/STEP | <u>After LOAD, RUN, or single step execution:</u> |
| | Address of next instruction to be executed same as contents of P register location 7) |
| | Contents of next instruction location |
| | <u>After FETCH or STORE operation:</u> |
| | Same as FETCH or STORE |
| | Same as FETCH or STORE |

| | | |
|----------------------------|--|--|
| FETCH or STORE | Address of next cell to be accessed | Data last set up by data switches, or FETCH or STORE |
| FETCH Y+1 STORE Y+1 | One less than address of next cell to be accessed (address is incremented before FETCH or STORE takes place). | Same as FETCH or STORE |
| DATA CLEAR Switch | Momentary-down switch clears address or data display (selected by ADDRESS/DATA switch). | |
| STOP Indicator | Lights when CPU is halted either from programmed HLT instruction or panel halt (rotary switch in any position except RUN or LOAD). | |
| Mode control rotary switch | Seven-position rotary switch controls Prime CPU mode of operation. Active only when keyswitch is at ON. | |

Position

| | |
|-----------|---|
| LOAD | Enables automatic program load sequence, a CPU option that takes the place of a key-in loader and permits automatic hardware read-in of programs from various media. Also implements automatic reload or restart after a power failure, and CPU restart at address in sense switches. |
| RUN | Normal operating state of CPU; after START switch is operated, clock runs and CPU executes sequential instructions. |
| STOP/STEP | Stops CPU and permits address/data changes, MASTER CLEAR, and single-step execution. Interrupt and DMX operations are disabled. |
| STORE Y | When START switch is depressed, content of data display is stored in register or memory location specified by address display. The address and data displays are unaltered. |
| FETCH Y | When START switch is depressed, content of register or memory location (specified by address display) is displayed on indicators. |

| | |
|---------------------|---|
| FETCH Y+1 | Same as FETCH, but address display is incremented by 1 before location is read. Each operation of START switch accesses the next sequential memory location for display. |
| STORE Y+1 | Same as STORE, but address display is incremented by 1 before location is loaded. To load sequential locations, leave ADDRESS/DATA switch at DATA, enter each new data value in data switches, and press START. Next memory address is set up automatically. |
| START Switch | Initiates operation set up by rotary switch (RUN, LOAD, FETCH, STORE). |
| Sense/Data Switches | Bank of 16 latching-up/momentary-down toggle switches. Momentary down position of any switch sets a 1 into corresponding position of indicator display (address or data, depending on position of ADDRESS/DATA switch).* Center and latching UP position provide sense switch data to INA '1620 and sense switch skip instructions (SNS, etc.). Also provide device identification for bootstrap LOAD operation. UP is a 1, center is a 0 for each switch. Switches 1 to 16 correspond to A register bits 1 to 16. For P350 and up, the sense switches also make it possible to examine locations in the general register sets and to specify 22-bit absolute addresses or segment addresses. (See Table F-2). |
| ADDRESS/DATA Switch | Selects information to be displayed on data/address indicators. (See description of indicators.) |

*In RUN mode, momentaries can be program tested by INA '1720.

Table F-2. Register and Memory Access in Prime 350-400-500

| Sense Switch | Register Access | Memory Access |
|--------------|--|--|
| SS1 | UP | DOWN |
| SS2 | UP: SS 10-16 address the four 32-bit register sets: '0-'37 microcode scratch (RF0) '40-'77 DMA (RF1) '100-'137 User Set (RF2) '140-'177 User Set (RF3) | Not used |
| | DOWN: SS12-16 address a location in the current register set (CRS) | |
| SS4 | UP: Address high-order half of 32-bit register | UP: Absolute address- ing of physical memory. |
| | DOWN: Address low-order half of 32-bit register | DOWN: Mapped addressing of segmented memory (if segmentation is enabled). Set segment number in SS5-16 |

SUMMARY OF OPERATING PROCEDURES

Note

The following procedures apply to all CPUs unless otherwise specified.

Turning Power On

1. If automatic program load is desired, turn rotary switch to LOAD. If not, select STOP/STEP.
2. Turn CPU keyswitch ON. The blowers must begin operating. A MASTER CLEAR sequence occurs, and an automatic program load takes place (if the option is present and the device is ready).
3. On the paper tape unit (if present), press the POWER switch; the internal indicator should light and the fans begin operating.
4. Turn the supervisor terminal LINE/OFF/LOCAL switch to LINE.
5. Refer to the appropriate vendor manuals for turn-on procedures of other peripheral devices.

Master Clear

The MASTER CLEAR switch places the Prime CPU in a known starting condition:

- Except for the P register, all registers in the register files (memory locations 0-'37 for the Prime 100-200-300) are cleared (set to all 0's).
- The P register is set '001000 (the starting address of most self-loading programs, and the restart address of most system programs).
- The CPU is set up in 16K sectored addressing mode and single precision arithmetic mode, with interrupt and machine check inhibited and standard interrupt mode in effect.
- All peripheral controllers are initialized.
- Micro-verification takes place if the CPU contains the microverify feature.

MASTER CLEAR operates in any position of the mode control rotary switch except LOAD or RUN.

Setting Address Display

Before a series of register or memory FETCH or STORE operations, the memory address must be set to a starting value using the panel data entry switches. The rotary switch can be in any position except RUN or LOAD.

1. Set the ADDRESS/DATA switch to ADDRESS.
2. Clear the address display by pressing the DATA CLEAR switch (the indicators go out). Set the desired address by pressing the momentary-action data entry keyswitches. (If you make a mistake, clear the display by pressing DATA CLEAR and try again.)

Setting Data Display

Data to be stored in a memory location is first set into the DATA display from the panel data entry switches. The rotary switch can be in any position except RUN or LOAD.

1. Set the ADDRESS/DATA switch to DATA.
2. Clear the display by pressing the DATA CLEAR switch. Set the desired value by pressing the momentary-action data entry switches. (If you make a mistake, clear the display and try again.)

Examining Registers and Memory Locations (P100-200-300)

1. Set the address display to the first location to be examined. Return ADDRESS/DATA switch to DATA position. Turn to FETCH and press START. The contents of the register (or location) will appear in the indicators. (See Table F-3 for a map of registers and reserved memory locations.)
2. To examine successive locations, turn to FETCH Y+1. Every time START is pressed, the address register is incremented before the location is read. The indicators display the contents of successive locations after each depression of the START switch.

Table F-3. Reserved Memory Locations (Prime 100-200-300)

| Address | Assignment |
|---------|---|
| 000000 | X Register - Index Register |
| 01 | A Register - Arithmetic, Shift, I/O |
| 02 | B Register - Ext. Arithmetic, Shift |
| 03 | Stack Pointer |
| 04 | FLPH - Floating Point High (Optional) |
| 05 | FLPL - Floating Point Low (Optional) |
| 06 | VSC - Visible Shift Counter |
| 07 | P Register - Program Counter |
| 10 | PMAR - Page Map Address Register (Optional) |
| 11 | Microcode Scratch Location |
| 12 | EAS - Effective Address Save (ILL, UII, Interrupts, etc.) |
| 13 | Microcode Scratch Location |
| 14 | Y Register Save for Control Panel DMA |
| 15 | M Register Save for Control Panel DMA |
| 16 | Microcode Scratch Location |
| 17 | Microcode Scratch Location |
| 20-37 | DMA Range/Start Address Pairs |
| 40-57 | Reserved for DMC Channel Pairs |
| 60 | PFI - Power Failure Interrupt |
| 61 | RTCI - Real Time Clock Increment (Optional) |
| 62 | REVI - Restricted Execution Violation (Optional) |
| 63 | INT - Standard Interrupt (Compatible Mode) |
| 64 | Page Fault - Addressed Page Not in Mem. (Optional) |
| 65 | SVC - Supervisor Call Trap |
| 66 | UII - Unimplemented Instruction Interrupt |
| 67 | PE - Memory Data Parity Error |
| 70 | Machine Check - Processor Detected Error |
| 71 | Missing Module - No memory at Accessed Location |
| 72 | ILL - Illegal Instruction Interrupt |
| 73 | PWV - Page Write Violation (Optional) |
| 74 | FLEX - Floating Point Exception (Optional) |
| 75 | PSU - Procedure Stack Underflow (PRIME 300 only) |
| 76-100 | Debugging Scratch Area |
| 101-177 | Interrupt Vectors |
| 200-777 | General Cross Sector Links |

Loading Registers or Memory Locations (P100-200-300)

1. Set the address display to the first location to be altered. Then, set the data display to the desired content and turn to STORE. Press START. The content of the data register will be loaded into the location. The indicators can display either data or the address, depending on the position of the DATA/ADDRESS switch.
2. To load sequential locations, turn to STORE Y+1. Every time START is pressed, the address display is incremented before the location is loaded. Therefore, by repeatedly loading a new value into the data register and pressing START, sequential locations are loaded.

Accessing Physical Memory (P350 and up)

When segmentation is not enabled, the control panel accesses real physical memory. Sense switches 11-16 provide the high-order bits of a full 22-bit physical address. Locations 0-'37 are the Prime 100-200-300 registers listed in Table F-3.

1. To do a STORE operation, set the 16-bit data value in the DATA display.
2. Set low-order 16 bits of address in ADDRESS display.
3. Set high-order 6 bits of address in SS11-16.
4. Set SS1 down and SS4 up.
5. Set rotary switch to desired function and press START for access.

Accessing Mapped Memory (P350 and up)

If segmentation is enabled, memory accesses are mapped. The segment number is set in SS5-16. Locations 0-'17 are specific Prime 350-400-500 register values as shown in the second column of Table F-4.

1. If this is to be a STORE operation, set the 16-bit data value in the DATA display.
2. Set address of word within segment in ADDRESS display.
3. Set segment number in SS5-16.
4. Set SS1 and SS4 down.
5. Set rotary switch to desired function and press START for access.

Note

If the addressed page is not currently in physical memory, the DATA indicator display is unpredictable.

Accessing Prime 350-400-500 Registers (Absolute Addressing)

With SS2 up, any location in any register set can be accessed by an absolute address:

| | |
|-------------|-------------------|
| '0 - '37 | microcode scratch |
| '40 - '77 | DMA |
| '100 - '137 | User Set 2 |
| '140 - '177 | User Set 3 |

The Y+1 functions increment the register address before the access. The address overflows from '177 to 0 as higher-order bits are ignored.

1. If this is to be a STORE operation, set the 16-bit data value in the DATA display.
2. Place SS2 up.
3. Set absolute address of desired register location in SS10-16.
4. Set SS4 to select the register half to be accessed: UP=high, DOWN=low.
5. Set rotary switch to desired function and press START for access.

Accessing Prime 350-400-500 Current Register Set (CRS)

With SS2 down, the current register set is accessed by addresses '0 through '37. The first column of Table F-4 shows the contents of the CRS. For Y+1 functions, the address is incremented before the address overflows from '37 to 0.

1. If this is to be a STORE operation, set the 16-bit data value in the DATA display.
2. Place SS2 down.
3. Set address of desired location in CRS in SS12-16.
4. Set SS4 to select the register half to be accessed: UP=high, DOWN=low.

Table F-4. Prime CPU Register Correspondence

| P350-400-500 Register Location | P100-200-300 Register Location | Register Contents |
|--------------------------------------|--------------------------------------|--|
| 2H | 7 | P (program counter) |
| 2L | 1 | A (accumulator) |
| | 2 | B (double-precision and long accumulator extension) |
| 3 H,L | - | EH,EL (accumulator extension for MPL,DVL) |
| 5 H | 3 | S (stack), Y (alternate index) |
| 7 H | 0 | X (index) |
| 10 H | 13 | - |
| 10,11 | - | FAR0,FLR0 (field address and length register 0) |
| 12,13 | - | FAR1,FLR1 (field address and length register 1) |
| 12 H | 4 | FAC (floating accumulator, mantissa high) |
| 12 L | 5 | FAC (mantissa middle) |
| 13 H | 6 | FAC (exponent) |
| 13 L | - | FAC (mantissa low, double-precision) |
| 14 H,L | - | PB (procedure base) |
| 15 H,L | 14,15 | SB (stack base) |
| 16 H,L | 16,17 | LB (linkage base) |
| 17 H,L | - | XB (temporary base) |
| 20 H | 10 | (high half of DTAR3) |
| 20 H,L | - | DTAR3 (descriptor table address segments 3072-4095) |
| 21 H,L | - | DTAR2 (segments 2048-3071) |
| 22 H,L | - | DTAR1 (segments 1024-2047) |
| 23 H,L | - | DTAR0 (segments 0-1023) |
| 24 H,L | - | keys, modals (see Table 2.9) |
| 25 H,L | - | OWNER (address of process control block of process owning register contents) |
| 26 H | 11 | FCODE (fault code) |
| 27 H,L | - | FADDR (fault address) |
| 27 L | 12 | (fault address word number) |
| 30 H | - | process 1024-microsecond c.p.u. timer |

Notes

1. In mapped access for P350-400-500, the P100-200-300 Register Contents are accessed as shown in the second column.
2. H means high-order half of 32-bit register. L means low-order half.

Single Step Operation

1. Turn to STOP/STEP. The address display shows the address of the next instruction to be executed. (The data display is the instruction itself.) To begin single stepping at another location, STORE a new address in the P register.
2. Press START. The current instruction is executed and the address of the next instruction is formed. (The next address is normally $P + 1$, but may be $P + 2$ after a skip or a two-word instruction, or the jump destination after a jump instruction.)
3. Registers or memory locations can be examined or altered using the FETCH and STORE functions. When the rotary switch is returned to STOP/STEP, the CPU is ready to execute at the current program count in the P register, even though the address display may contain the address used for the last FETCH or STORE cycle. After the next operation of the START switch, however, the address display is updated to show the program count again.
4. Interrupt and DMX operations are disabled in STOP/STEP mode.

Starting a Program

Note

Also see AUTOMATIC PROGRAM LOAD for alternate ways to start execution.

1. Turn the rotary switch to any position except RUN or LOAD. If the machine needs to be initialized, press MASTER CLEAR.
2. Load the P register with the desired starting address, obtained from the listing, the paper tape label, or other documentation. (A MASTER CLEAR leaves the P register set to '1000, the starting location of most Prime system software.)
3. Determine whether other registers or memory locations must also be preset to starting values. (See Table F-4.)
4. Turn to RUN and press START. The program should begin running (STOP indicator goes out). Panel switches and indicators are then accessible to programmed I/O instructions only.

Recovering from Errors

If an equipment failure or program error causes the CPU to leave PRIMOS II control, it is usually possible to restart the CPU from the control panel. The procedure is:

1. Turn the rotary switch to STOP/STEP; press MASTER CLEAR.
2. Set the sense switches to the restart address.
3. Turn rotary switch to LOAD; press START.
4. Reset sense switches to 0.

Panel Lockout

Turn the keyswitch to the LOCK position. The CPU is then insensitive to the position of the rotary switch, and the running program cannot be stopped from the panel. (The sense switches and indicators are still accessible to programmed I/O instructions, however.)

Stopping Execution

1. To stop program execution, turn panel lockout keyswitch to ON and turn rotary switch to STOP/STEP. (STOP indicator should light.) Memory and registers can be examined or altered, and a new starting address can be set.
2. To continue execution, turn rotary switch to RUN and press START. Lock panel if desired.

AUTOMATIC PROGRAM LOAD

The Automatic Program Load (APL) feature of Prime CPUs simplifies loading of programs and operating systems from external storage devices. Operating procedures are described below. See Appendix A for the details of internal operations of Prime's APL bootstraps.

1. Install the medium from which a program is to be loaded on the appropriate device. Make sure power is on and both the device and the supervisor terminal are ready to operate.
2. Select the autoload device (and disk surface, if applicable) in SS11-16 as shown in Figure F-2.
3. Set SS1-4 to select the desired version of PRIMOS II (*DOS16, etc.). If SS1-4 are left at zero, the system automatically selects a version that will fit the available memory.
4. Place SS10 down unless you want to prevent automatic startup when bootstrapping is complete.

5. Press MASTER CLEAR.
6. Turn rotary switch to LOAD and press START. If bootstrapping is successful, a message will appear on the supervisor terminal.

AUTOMATIC RESTART OR REBOOT AFTER POWER FAILURE

Once a program has been loaded and started, the CPU is capable of an automatic restart (or reload) after a power failure. There are four possible methods, depending on whether the CPU contains the APL or battery backup/PFI features. Figure F-3 summarizes the panel setup required to prepare the CPU for a restart/reboot sequence.

When a reboot is desired, the autoload device must be ready to operate. If paper tape is used, it must be repositioned to the beginning of tape. (The tape must have an autostart address.)

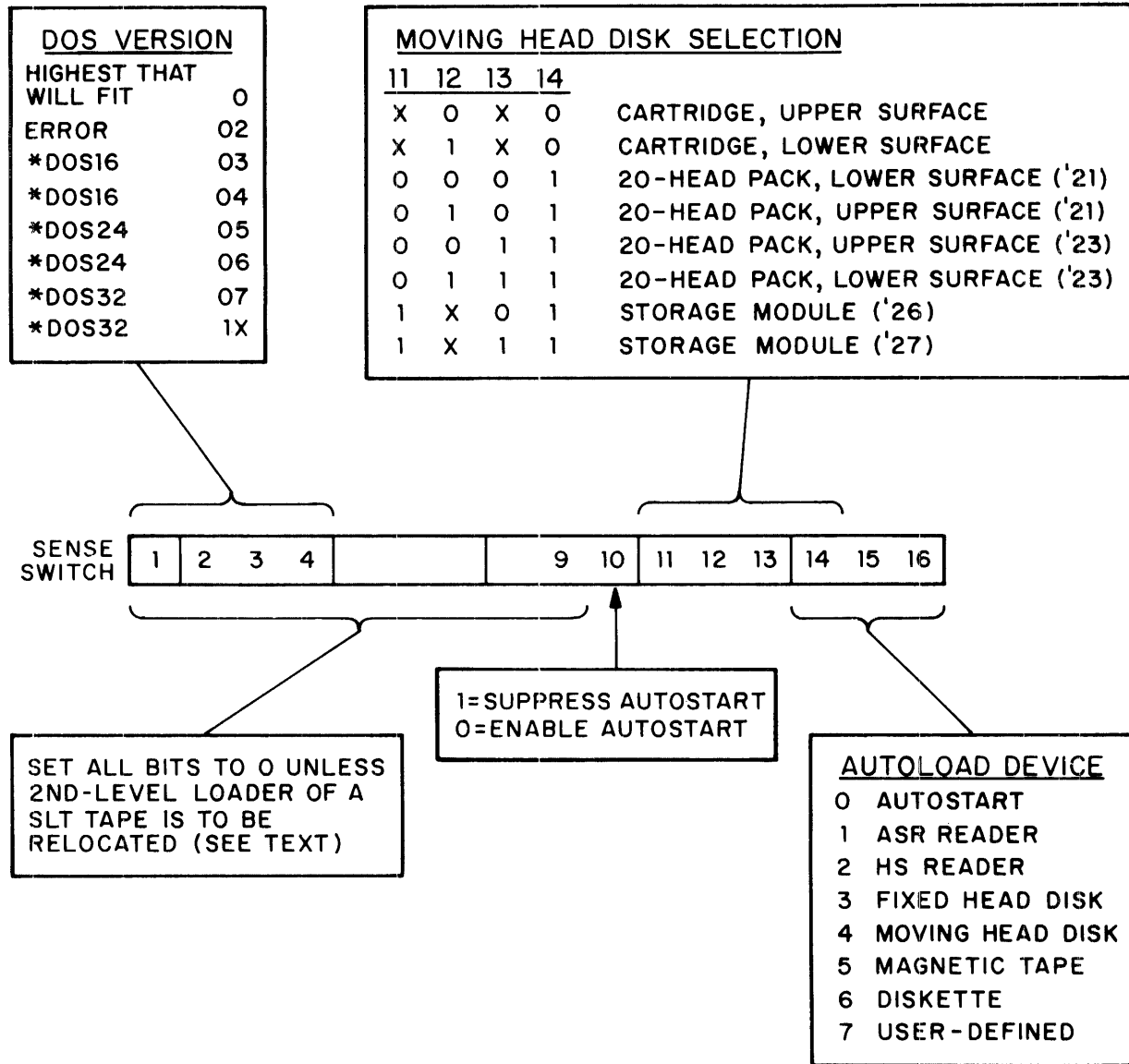


Figure F-2. APL Sense Switch Settings.

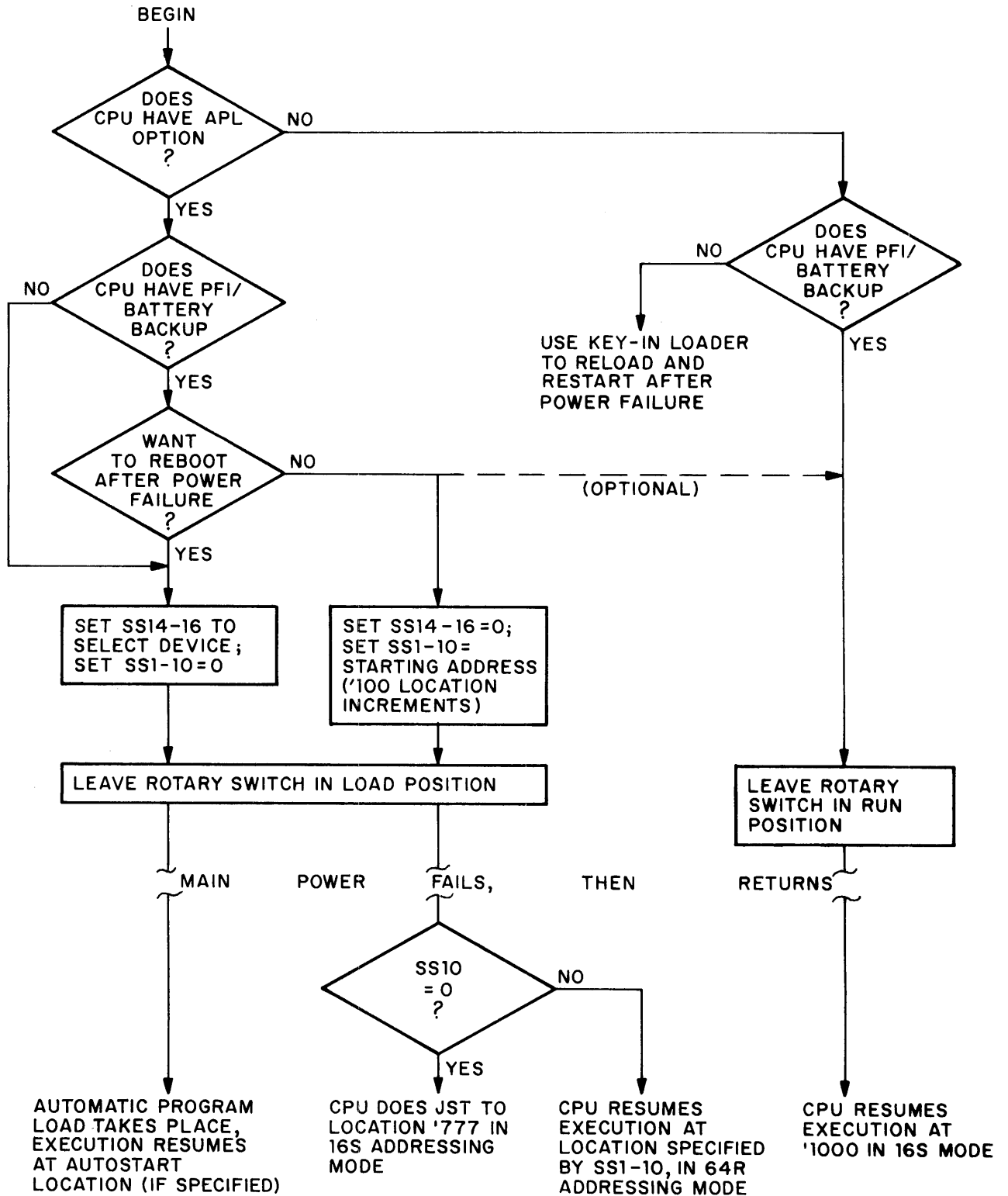


Figure F-3. Panel Setup for Automatic Restart or Reload after Power Failure.

APPENDIX G

PHYSICAL DEVICE NUMBERS

INTRODUCTION

Each physical disk or disk partition has a physical device number identifying the type of storage device, the drive unit on which it is mounted, and, for partitions, the size of the partition and its location on the disk pack. These physical device numbers are used in the commands: ASSIGN DISK, CONFIG, COPY, DISKS, FIXRAT, MAKE, SHUTDN, STARTUP, and UNASSIGN DISKS. Some of the devices listed in this section are no longer supplied; the information is included here for archival and reference purposes.

DRIVE UNIT NUMBERS

The drive unit number identifies the physical drive on which the disk is mounted. Diskette and cartridge disk drive unit numbers are set by a thumbwheel switch on the drive unit. Thumbwheel numbers must be different for all drives connected to the same system. Drive unit numbers for storage modules are set by the cable connections. The system installer should have labelled these units; in many cases the drive unit number will be on one of the push button switches on the front of the drive unit. It is important to keep a record in the system log book of drive unit numbers and of the physical device number(s) (including partitioning) for disks mounted on these drives.

CONTROLLER

The controller is an interface between the CPU and the drive unit. The system installer will indicate which controller address has been used. In almost all cases the first four storage modules (or 60 MByte disks) will be connected to the default address; drives in excess of 4 will be connected to the explicit address.

DISKS

Fixed Head Disks

Physical device numbers are:

| <u>Sectors/Track</u> | <u>Physical Device Number</u> |
|----------------------|-------------------------------|
| 8 | 10 |
| 64 | 40 |

Diskettes (Floppy Disk)

All diskettes have 4 sectors/track and a total of 304 records. The physical device number is: 2u

where u is the number of the diskette drive, (0 - 7) i.e., 24 is the fifth (unit 4) diskette drive.

Moving Head Disks - Cartridges

The physical device numbers are: ts

where t is the type of disk and s defines the surface of the disk and the drive unit on which it is mounted.

| <u>Sectors/Track</u> | <u>Type (t)</u> | <u>Records</u> | <u>Comments</u> |
|----------------------|-----------------|----------------|-------------------------------------|
| 8 | 0 | 3248 | 6MB cartridge (obsolete controller) |
| 8 | 3 | 6496 | 6MB |
| 32 | 5 | 6496 | 6MB cartridge |
| 32 | 5 | 12992 | 12MB cartridge |

s = 2*(drive unit number) for upper (removable) surface

s = 2*(drive unit number)+1 for lower (fixed) surface

Drive unit numbers are 0, 1, 2, 3.

Examples: A 12 MByte cartridge)

| <u>Drive Unit</u> | <u>Surface</u> | <u>Device Number</u> |
|-------------------|----------------|----------------------|
| 0 | upper | 50 |
| 0 | lower | 51 |
| 1 | upper | 52 |
| 1 | lower | 53 |
| 2 | upper | 54 |
| 2 | lower | 55 |
| 3 | upper | 56 |
| 3 | lower | 57 |

Moving Head Disks - 60 MByte (30-million words)

This disk has 32 sectors/track and 64960 records.

The physical device number is: 5c5u

where c is 0 if the drive is connected to controller address '21 (default) and 2 if connected to controller address '23 (explicit).

u is twice the value of the drive unit (0 - 3) which the disk is mounted.

Example:

A 60 MByte disk mounted on drive unit 2 connected to address '23 of the controller has physical device number 5254.

Partitioning: This disk may be sub-divided into partitions, each of which will be treated as if it were an actual physical device. Partitions must be an integral number of heads in size and must be offset an even number of heads from head 0 of the disk pack. Physical device numbers for partitions are given in the table below.

| -Partition Size- | | -Physical Device Number- | |
|------------------|---------|---------------------------|---------------------------|
| heads | records | Controller Address='21 | Controller Address='23 |
| 2 (default) | 6496 | xx005y | xx025y |
| 2(explicit) | 6496 | xx045y | xx065y |
| 4 | 12992 | xx105y | xx125y |
| 6 | 19488 | xx145y | xx165y |
| 8 | 25984 | xx205y | xx225y |
| 10 | 32480 | xx245y | xx265y |
| 12 | 38976 | xx305y | xx325y |
| 14 | 45472 | xx345y | xx365y |
| 16 | 51968 | xx405y | xx425y |
| 18 | 58464 | xx445y | xx465y |
| 20 | 64960 | xx505y | xx525y |

The head offset is the number of heads (disk surfaces) which lie between the start of the partition and the start of the disk pack (bottom).

xx is one-half the head-offset of the partition (octal value)

y is twice the value of the drive unit on which the disk is mounted

Examples:

A 6-head partition with an offset of 4 heads is mounted on drive unit number 0 connected to controller address '21.

xx=02, y=0

Physical device number is: 021450

A 20-head partition (entire disk) is mounted on drive unit number 2 connected to controller address '23

xx=00 (a 20-head partition takes up the entire disk and cannot have a non-zero offset)

y=4

Physical device number is 005254, the same as constructed above for non-partitioned disk.

Moving Head Disks - Storage Modules

Storage modules exist in three sizes as below:

| <u>Size of Module</u> | <u>Number of Heads</u> | <u>Type</u> |
|-----------------------|------------------------|-------------|
| 40 MBytes | 5 | 6 |
| 80 MBytes | 5 | 6 |
| 300 MBytes | 19 | 6 |

Storage modules are usually partitioned (sub-divided) with each partition being treated as if it were an actual physical device. Partitions must be an integral number of heads in size and must be offset an even number of heads from the start of the disk pack. However, the last partition on the disk may contain an odd number of heads.

The physical device number is constructed as a 16-bit number, converted to octal (see Figure G-1).

| <u>bits</u> | <u>Meaning</u> |
|-------------|---|
| 1,2,3,4 | one-half the head offset |
| 5,6,7,8,16 | number of heads in partition |
| 9 | Controller address indicator. 0 if '27 (default), 1 if address '26 (explicit) |
| 10,11,12,13 | device type. This is 6 (0110) for present storage modules |
| 14,15 | drive unit number (0 - 3) |

A table of these device numbers is given below.

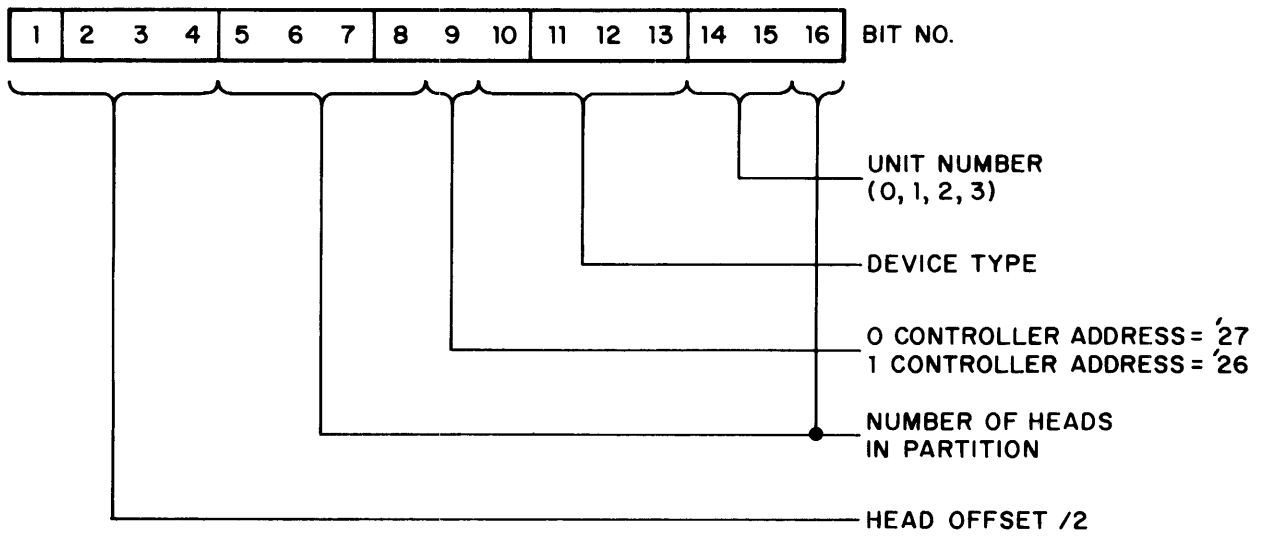


Figure G-1. Construction of Physical Disk/Disk Partition Number

| -Partition Size- | | -Physical Device Number- | |
|------------------|-------------------------|---------------------------|---------------------------|
| Heads | Records (Normalized) | Controller Address='27 | Controller Address='26 |
| 1 | 17238 | xx006z | xx026z |
| 2 | 34476 | xx046y | xx066y |
| 3 | 51714 | xx046z | xx066z |
| 4 | 68952 | xx106y | xx126y |
| 5 | 86190 | xx106z | xx126z |
| 6 | 103428 | xx146y | xx166y |
| 7 | 120666 | xx146z | xx166z |
| 8 | 137904 | xx206y | xx226y |
| 9 | 155142 | xx206z | xx226z |
| 10 | 172380 | xx246y | xx266y |
| 11 | 189618 | xx246z | xx266z |
| 12 | 206856 | xx306y | xx326y |
| 13 | 224094 | xx306z | xx326z |
| 14 | 241332 | xx346y | xx366y |
| 15 | 258570 | xx346z | xx366z |
| 16 | 275808 | xx406y | xx426y |
| 17 | 293046 | xx406z | xx426z |
| 18 | 310284 | xx446y | xx466y |
| 19 | 327522 | xx446z | xx466z |

- xx is one-half the head-offset (octal)
- y is twice the drive unit number on which the disk is mounted
- z is twice the drive unit number plus one

Notes

1. Only the last partition on the disk may have an odd number of heads.
2. All head-offsets must be even.
3. 40- and 80-MByte modules have 5 heads; 300-MByte modules have 19 heads.
4. There are 7407 1040-word records on a storage module head. This is normalized, in the table, to 17238 448-word record equivalents. The 448-word record is standard for all disk packs except storage modules.

Example:

A system contains 3 drive units; drives 0 and 1 have 300 MByte storage modules, drive 2 has an 80 MByte storage module (See Figure G-2). For program development, operations, testing, etc., the modules are to be partitioned as follows:

drive 0: partitions of 2,2,6,2,2,2, and 3 heads
 drive 1: partitions of 14 and 5 heads
 drive 2: partitions of 2 and 3 heads

The physical device numbers are:

| <u>0</u> | <u>1</u> | <u>2</u> |
|----------|----------|----------|
| 000460 | 003462 | 000464 |
| 010460 | 071063 | 010465 |
| 021460 | | |
| 050460 | | |
| 060460 | | |
| 070460 | | |
| 100461 | | |

In all cases the drives are connected to the default controller address of '27. Each partition is treated by PRIMOS as if it were a separate physical device.

DRIVE UNIT
DEVICE
CAPACITY

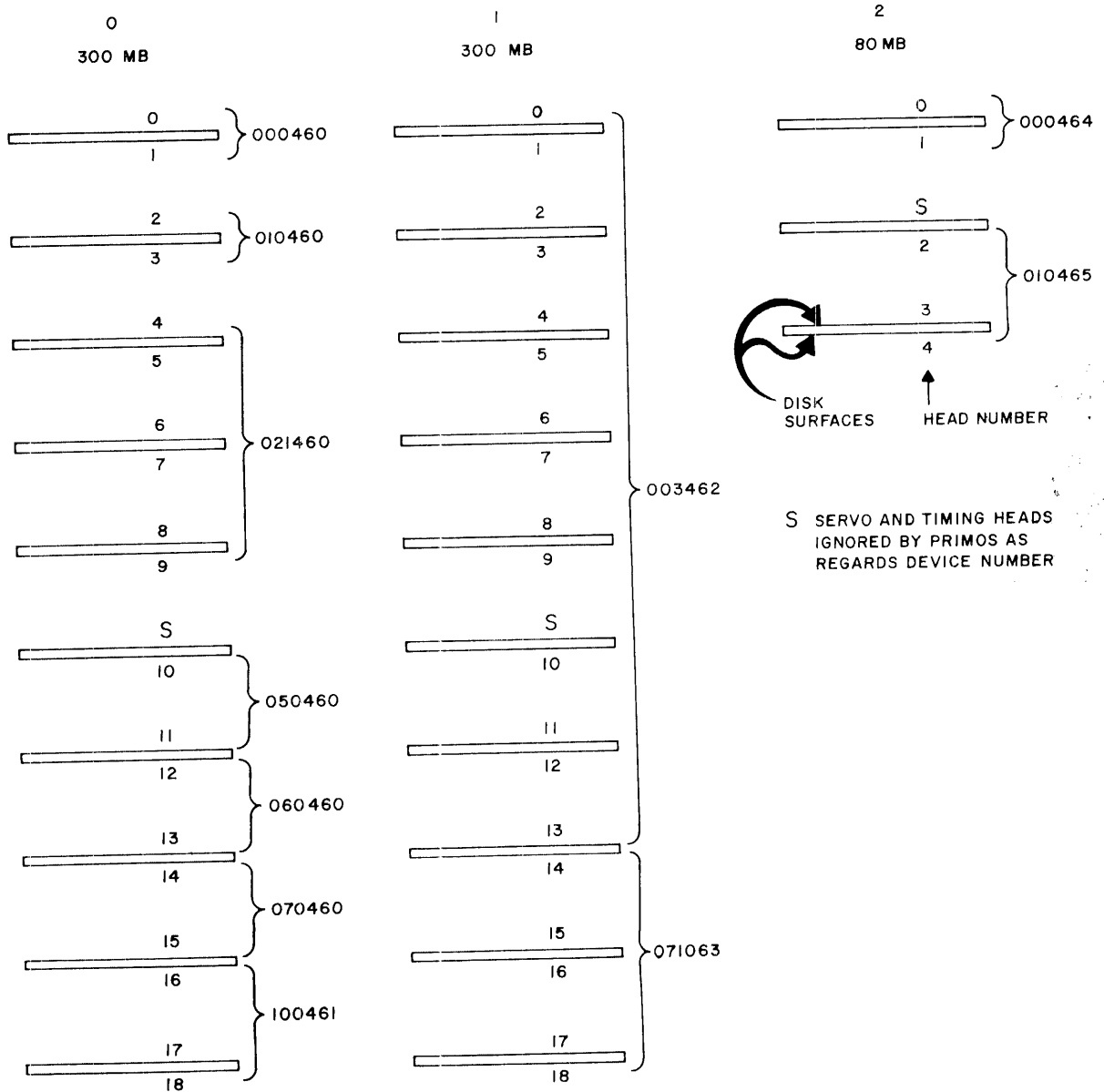


Figure G-2. Example of Storage Module Partitions

INDEX

- *** BULLETIN *** 15-13
- *COLDS 16-6
- *LOGPRT program 5-9
- *SPPR B-1
- ABORT (SPOOL Command) 9-4
- ALL (CX Command) 5-3
- BACK (SPOOL Command) 9-5
- BREAK (TERM Command) 3-6
- CANCEL (SPOOL Command) 9-9
- DEFER (SPOOL Command) 9-9
- DELETE (LOGPRT Command) 5-9
- DROP (SPOOL Command) 9-4
- ERASE (TERM Command) 3-6
- EXPAND (SPOOL Command) 9-9
- FINISH (SPOOL Command) 9-3
- FORM (SPOOL Command) 9-9
- FROM (LOGPRT Command) 5-10
- FTN (SPOOL Command) 9-9
- FULL (TERM Command) 3-6
- FUNIT (SPOOL Command) 9-10
- HALF (TERM Command) 3-6
- HANG (SPOOL Command) 9-3
- HELP (LOGPRT Command) 5-9
- HOME (SPOOL Command) 9-10
- INPUT (LOGPRT Command) 5-10
- KILL (TERM Command) 3-6
- LENGTH (SPOOL Command) 9-5
- LF (TERM Command) 3-6
- LIST (SPOOL Command) 9-10
- LNUM (SPOOL Command) 9-11
- LOGOUT (SPOOL Command) 9-6
- NOFMT (SPOOL Command) 9-11
- NOLF (TERM Command) 3-6
- NOXOFF (TERM Command) 3-6
- ON logical-device (CX) 5-4
- ON node-name (remote login)
3-2
- OPEN (SPOOL Command) 9-11
- PAPER (SPOOL Command) 9-4
- PLOT (SPOOL Command) 9-11
- PRINT (CX Command) 5-4
- PURGE (LOGPRT Command) 5-9
- QUEUE (CX Command) 5-4
- RESTART (SPOOL Command) 9-5
- Sjob# (CX Command) 5-4
- SPOOL (LOGPRT Command) 5-9
- TIME (SPOOL Command) 9-3
- TUNIT (SPOOL Command) 9-11
- TYPE (LOGPRT Command) 5-10
- USER (SPOOL Command) 9-2
- XOFF (TERM Command) 3-6
- Abort, FIXRAT 6-6

INDEX

- Acknowledge time, spooler 9-3
- Activity file 5-4
- Adding event types D-2
- Adding PRIMOS II to new disks 7-6
- Adding software 4-11
- Adding UFDs 4-11
- ADDISK (PRIMOS Command) 15-2
- Address display F-9
- Administrator, Roles of 1-2
- ALTDEV (CONFIG subcommand) 16-5
- Alternate paging device 16-5
- Alternate paging device 19-6
- Ambiguity, device 7-2
- AMLBUF (CONFIG subcommand) 16-5
- AMLC (PRIMOS Command) 15-3
- AMLC line configuration 2-6
- AMLC lines, configure 15-3
- AMLC requirements, PRIMOS III 18-4
- ASR terminal I/O buffer size 16-6
- ASRATE (CONFIG subcommand) 16-6
- ASRBUF (CONFIG subcommand) 16-6
- Assignable disk table, changing 4-1
- Assignable disks table, change 15-8
- ATTACH (PRIMOS Command) 3-3
- Attaching 3-3
- Auto-start boot A-2
- Automatic program load F-16
- Automatic restart F-17
- AVAIL (PRIMOS Command) 5-1
- Available segments 16-14
- Backing up 4-7
- Backing up 8-1
- Backing up, disk 4-8
- Backing up, file recovery 4-8
- Backing up, magnetic tape 4-8
- Backup, disk-to-disk 8-3
- Backup, disk-to-tape 8-8
- Backup, scheduling 8-1
- Bad BOOT 6-11
- Bad memory, PRIMOS III 18-10
- Bad page, permanent 19-12
- Bad pages, map out 10-4
- Badspot file, details 7-6
- Badspot file, modifying -76
- Badspots 6-10
- Badspots 7-3
- Badspots 8-5
- BADSPT file 8-5
- Basic concepts 1-6

INDEX

- BASIC/VM, shared, startup 2-7
- Binary file, definition 1-6
- BOOT file 7-4
- Boot operations A-4
- Boot, auto-start A-2
- BOOT, bad 6-11
- BOOT, building A-5
- Boot, diskette A-3
- Boot, magnetic tape A-3
- Boot, mass storage A-2
- Boot, paper tape A-2
- Booting PRIMOS II 17-3
- Booting PRIMOS II 2-2
- Boots, control panel A-1
- Bootstraps A-1
- Broken file structure messages 6-8
- Building BOOT A-5
- Building PRIMOS III 18-2
- BULLETIN 15-13
- Byte, definition 1-6
- Cancel message 15-13
- Cartridge device number G-2
- Cartridge errors E-3
- Change assignable disks table 15-8
- Change eligibility timeslice 15-9
- Change maximum users 15-11
- Change priority 15-6
- Change timeslice 15-6
- Changing disk packs under PRIMOS II 17-5
- Changing LOGPRT filenames D-5
- Changing LOGREC size D-5
- Changing paper 9-4
- Changing priority 2-6
- Changing priority 4-1
- Changing timeslice 4-1
- CHAP (PRIMOS Command) 15-6
- Checksums 8-6
- CMDNCØ (UFD) 4-2
- CNAME (PRIMOS Command) 7-3
- COBOL library 11-1
- Cold start 1Ø-4
- COMDEV (CONFIG subcommand) 16-7
- Command device 16-7
- Command device 2-3
- Command device 2-4
- Command Device, PRIMOS II 2-2
- Command file with CONFIG 16-2
- Command line conventions 1-5
- Command, external, definition 1-7
- Command, internal, definition 1-7

INDEX

| | | | |
|-----------------------------|-------|---------------------------|-------|
| Commands, Spooler, Operator | 9-2 | CONFIG subcommand MYNAME | 16-11 |
| Commands, user, spooler | 9-8 | CONFIG subcommand NAMLC | 16-11 |
| Compression of UFDs | 6-5 | CONFIG subcommand NET | 16-12 |
| Concepts, basic | 1-6 | CONFIG subcommand NPUSR | 16-14 |
| CONFIG | 16-1 | CONFIG subcommand NRUSR | 16-14 |
| CONFIG (CONFIG subcommand) | 16-7 | CONFIG subcommand NSEG | 16-14 |
| CONFIG command for PRIMOS | III | CONFIG subcommand NTUSR | 16-15 |
| 18-5 | | CONFIG subcommand PAGDEV | 16-15 |
| CONFIG file | 2-4 | CONFIG subcommand PREPAG | 16-16 |
| CONFIG file (contents) | 2-12 | CONFIG subcommand RLOGIN | 16-16 |
| CONFIG subcommand ALTDEV | 16-5 | CONFIG subcommand RWLOCK | 16-16 |
| CONFIG subcommand AMLBUF | 16-5 | CONFIG subcommand SMLC | 16-17 |
| CONFIG subcommand ASRATE | 16-6 | CONFIG subcommand TYP OUT | 16-17 |
| CONFIG subcommand ASRBUF | 16-6 | CONFIG, single-line | 16-1 |
| CONFIG subcommand COMDEV | 16-7 | Configuration data file | 16-4 |
| CONFIG subcommand CONFIG | 16-7 | Configuration data file | |
| CONFIG subcommand DISLOG | 16-7 | (contents) | 2-12 |
| CONFIG subcommand ERASE | 16-8 | Configurations | 2-4 |
| CONFIG subcommand FAM | 16-8 | Configure AMLC lines | 15-3 |
| CONFIG subcommand GO | 16-8 | Configure FAM on network | 16-8 |
| CONFIG subcommand KILL | 16-9 | Configure network | 16-12 |
| CONFIG subcommand LOGLOG | 16-9 | Configure phantom users | 16-14 |
| CONFIG subcommand LOGMSG | 16-9 | Configure remote login | 16-16 |
| CONFIG subcommand LOGREC | 16-10 | Configure remote users | 16-14 |
| CONFIG subcommand LOUTQM | 16-10 | Configure SMLC lines | 16-17 |
| CONFIG subcommand MAXPAG | 16-10 | Configure terminal users | 16-15 |

INDEX

- Contents, Logbook 5-16
- Control Panel F-1
- Control panel boots A-1
- CONTROL-Q 3-6
- CONTROL-S 3-6
- Controllers G-1
- Conventions 1-5
- Conventions, command line 1-5
- COPY (PRIMOS Command) 8-3
- Copying method (COPY) 8-4
- CPU power 2-1
- Crash SEE ALSO Halt
- Crashes, system 4-9
- CREATE (PRIMOS Command) 3-3
- Creating directories 3-3
- Current register set, accessing F-13
- Current UFD 3-3
- Current version of document 1-1
- CX (PRIMOS Command) 5-3
- CX activity file 5-4
- CX Command -ALL 5-3
- CX Command -PRINT 5-4
- CX Command -QUEUE 5-4
- CX Command -Sjob# 5-4
- CX job file 5-3
- CX monitor 4-5
- CX operation 4-5
- CX phantom user number 2-6
- CX Processor messages 5-7
- CX queue 5-3
- CX queue on remote devices 5-4
- CX SEE ALSO Sequential job processor
- CX slave 4-5
- CX startup 2-6
- CX*** (UFD) 4-3
- C_PRMO 2-4
- C_PRMO command file 16-2
- C_PRMO file (contents) 2-13
- C_SHLB file 11-2
- Data display F-9
- Dataphone power 2-2
- Date, set system 15-14
- Date, system 2-9
- DBMS, shared, startup 2-8
- Default paper size (spool) B-2
- Definitions 1-5
- DELAY (PRIMOS Command) 3-7
- DELETE (PRIMOS Command) 3-5
- Deleting files 3-5
- Deletion of file 6-9

INDEX

- Device ambiguity 7-2
- Device connection 2-6
- Device number, cartridge G-2
- Device number, definition 1-6
- Device number, diskette G-2
- Device number, fixed head disk
G-2
- Device number, moving head disk
G-3
- Device number, storage module
G-4
- Device shutdown 15-17
- Device startup 15-18
- Directories, segment 6-10
- Directory protection 3-4
- Directory tree 6-2
- Directory, definition 1-6
- Directory-chain, definition 1-7
- Disable BREAK 3-6
- Disable Newline 3-6
- Disable X-OFF/X-ON 3-6
- DISCON (PRIMOS Command) 15-7
- DISCON (PRIMOS Command) 4-6
- Disconnect remote users 15-7
- Disconnecting users 4-6
- DISCS file 5-3
- Disk connection 2-6
- Disk drive power 2-1
- Disk Error Messages 5-8
- Disk errors E-1
- Disk formatting 4-4
- Disk formatting 7-1
- Disk initialization 7-4
- Disk packs, changing, under
PRIMOS II 17-5
- Disk partition 7-2
- Disk space utilization 5-1
- Disk space, monitoring 4-4
- Disk space, PRIMOS III 18-5
- Disk table, assignable, changing
4-1
- Disk verification 7-5
- Disk, logical, definition 1-7
- Disk, split 7-2
- Disk, virgin 7-4
- Disk-to-disk backup 8-3
- Disk-to-tape backup 8-8
- Diskette boot A-3
- Diskette device number G-2
- Diskette errors E-4
- DISKS (PRIMOS Command) 15-8
- DISLOG (CONFIG subcommand) 16-7
- Document organization 1-3
- Documents, related 1-4

INDEX

- DOSEXT 17-2
- DOSXT 17-2
- Drive unit numbers G-1
- DSKRAT file 6-1
- DSKRAT file 7-3
- Editor, shared, startup 2-7
- Eligibility timeslice, change 15-9
- ELIGTS (PRIMOS Command) 15-9
- Enable BREAK 3-6
- Enable Newline 3-6
- Enable X-OFF/X-ON 3-6
- Entering badspots 7-3
- Equipment shutdown 2-11
- Equipment startup 2-1
- ERASE (CONFIG subcommand) 16-8
- Erase character set 3-6
- Erase character, set 16-8
- Error correction E-2
- Error messages, FIXRAT 6-12
- Error messages, PRIMOS initialization 16-18
- Error recovery, PRIMOS II 17-5
- Error, write 7-5
- Errors, cartridge E-3
- Errors, disk E-1
- Errors, diskette E-4
- Errors, storage module E-1
- Event logger 2-5
- Event logger D-1
- Event logger, modifying D-2
- Event logger, use of 4-7
- Event Logging 5-9
- Event message format D-2
- Event Messages, LOGPRT 5-11
- Event types, current D-3
- Event types, LOGPRT 5-10
- Example of FIXRAT 6-18
- Examples, spooler 9-7
- Extended instruction set 19-7
- External command, definition 1-7
- External login program 14-1
- FAM (CONFIG subcommand) 16-8
- FAM (sub-UFD) 4-2
- FAM = File Access Manager
- FAM configuration 2-5
- FAM operation 4-4
- FETCH F-3
- File access, remote C-4
- File chains 6-5
- File deletion 3-5
- File deletion 6-9

INDEX

- File integrity 6-1
- File protection 3-4
- File structure 3-1
- File structure 6-2
- File system integrity 4-8
- File truncation 6-9
- File, binary, definition 1-6
- File, definition 1-7
- File, event logging 5-9
- File, object, definition 1-8
- File, source, definition 1-9
- File-unit, definition 1-7
- Filename conventions 1-9
- Filename, definition 1-7
- Fixed head disk device number
G-2
- FIXRAT (PRIMOS Command) 6-1
- FIXRAT abort 6-6
- FIXRAT Error messages 6-12
- FIXRAT example 6-18
- FIXRAT output 6-7
- FIXRAT printout options 6-5
- FIXRAT usage 6-5
- Forcing logouts 4-6
- Format, Logbook 5-15
- Formatting disks 7-1
- Formatting, disk 4-4
- FORMS library 11-1
- FORMS* (UFD) 4-3
- FORTRAN library 11-1
- FORTRAN printer form control
9-9
- FROM disk 8-3
- Full duplex set 3-6
- Functions, Terminal 1-5
- FUTIL (PRIMOS Command) 8-6
- GO (CONFIG subcomand) 16-8
- Half duplex set 3-6
- Halt actions 10-4
- Halt address 10-3
- Halt segment number 10-3
- Halt, PRIMOS II 10-1
- Halt, PRIMOS III 10-2
- Halt, PRIMOS IV 10-3
- Halt, PRIMOS V 10-3
- Halt, system 10-1
- Halts, PRIMOS III 18-11
- Halts, PRIMOS IV 19-13
- Hardware, network C-1
- Home UFD 3-3
- Hybrid command file 17-2
- Identity, definition I-7

INDEX

- ILLEGAL DEVICE NUMBER 7-2
- Implicit logout 16-9
- Inactivity logout 2-5
- Inactivity timeout 13-1
- Inactivity timeout 16-10
- Incorporating shared segments 4-2
- Increasing LOGBUF size D-2
- Incremental dump to tape 8-8
- Indexing files 8-9
- Indexing files on tape 8-12
- Initialization error messages, PRIMOS 16-18
- Initialization, disk 7-4
- Initialization, PRIMOS III 18-9
- Installation of shared libraries 11-2
- Installation options, spool B-1
- Installing shared utilities 15-15
- Integrity, file 6-1
- Integrity, file system 4-8
- Inter Processor Controller 19-2
- Internal command, definition 1-7
- Interprocessor Controller C-2
- IPC 19-2
- IPC, PRIMENET C-2
- Jam, line printer 4-10
- Job file 5-3
- KILL (CONFIG subcommand) 16-9
- Kill character set 3-6
- Kill character, set 16-9
- Large/short files (spool) B-2
- Ldev, definition 1-7
- Ldisk, definition 1-7
- LIB (UFD) 4-2
- Libraries, shared 11-1
- Limiting logins 4-6
- Line printer 4-9
- Line printer jam 4-10
- Line, AMLC, configuration 2-6
- Link protocol C-2
- Link, synchronous C-3
- LOAD F-3
- Local nodename 16-11
- Local nodename 2-5
- Log out spooler 9-6
- Logbook contents 5-16
- Logbook Format 5-15
- Logbook, keeping a 4-9
- Logbook, System 5-15
- LOGBUF (event buffer) D-1
- LOGBUF size, increasing D-2

INDEX

- LOGEV1 D-3
- Logging in 3-2
- Logging of Events 5-9
- Logging out 3-7
- Logical disk, definition 1-7
- LOGIN (PRIMOS Command) 3-2
- Login program, external 14-1
- Login, remote C-5
- Login, remote, specification 2-5
- Logins, limiting 4-6
- LOGLOG (CONFIG subcommand) 16-9
- LOGLST file 5-9
- LOGMSG (CONFIG subcommand) 16-9
- LOGOUT (PRIMOS Command) 3-7
- Logout on disconnect 16-7
- Logout, automatic 16-10
- Logout, implicit 16-9
- Logout, inactivity 2-5
- Logouts, forcing 4-6
- LOGPRT Command -DELETE 5-9
- LOGPRT Command -FROM 5-10
- LOGPRT Command -HELP 5-9
- LOGPRT Command -INPUT 5-10
- LOGPRT Command -PURGE 5-9
- LOGPRT Command -SPOOL 5-9
- LOGPRT Command -TYPE 5-10
- LOGPRT Event Messages 5-11
- LOGPRT event types 5-10
- LOGPRT filenames, changing D-5
- LOGPRT Messages 5-11
- LOGPRT utility 5-9
- LOGPRT, modifying D-4
- LOGREC (CONFIG subcommand) 16-10
- LOGREC file 5-9
- LOGREC size, changing D-5
- LOGREC, creating D-1
- LOOK (PRIMOS Command) 15-10
- LOUTQM (CONFIG subcommand) 16-10
- Magnetic tape boot A-3
- Magnetic tape drive power 2-2
- Magnetic tape dump, crash 10-4
- Magnetic tape utility 8-11
- Magnetic tape utility 8-8
- MAGRST (PRIMOS Command) 8-11
- MAGSAV (PRIMOS Command) 8-8
- MAKE (PRIMOS Command) 7-1
- Manager, network C-2
- Map out bad pages 10-4
- Mapped memory F-1
- Mass storage boot A-2

INDEX

- Master clear F-9
- Master File Directory 3-1
- Maximum users, change 15-11
- MAXPAG (CONFIG subcommand) 16-10
- MAXSCH (PRIMOS Command) 15-11
- MAXUSR (PRIMOS Command) 15-11
- Memory parity halt 19-10
- Memory parity tests, PRIMOS IV 19-10
- Memory Parity, PRIMOS III 18-10
- Memory parity, uncorrected 10-7
- Memory requirements, PRIMOS III 18-4
- Memory, loading F-12
- Memory, mapped, accessing F-12
- Memory, physical, accessing F-12
- MESSAGE (PRIMOS Command) 15-12
- MESSAGE NOT SENT 15-13
- Message, login/logout 16-9
- Messages, broken file structure 6-8
- Messages, error, FIXRAT 6-12
- Messages, LOGPRT 5-11
- Messages, PRIMOS 2-10
- Messages, sending 4-10
- Messages, shutdown 2-11
- Messages, Spooler 9-6
- Messages, Startup 2-10
- Messages, Supervisor Terminal 5-6
- MFD 3-1
- MFD, definition 1-7
- MIDAS library 11-1
- Mode, definition 1-8
- Modifying event logger D-2
- Modifying LOGPRT D-4
- Modifying PRIMOS IV 19-4
- Monitoring disk space 4-4
- Monitoring the system 5-1
- Moving head disk device number G-3
- Multiple spoolers 9-2
- MYNAME (CONFIG subcommand) 16-11
- M_PRMO 18-2
- NAMLC (CONFIG subcommand) 16-11
- NET (CONFIG subcommand) 16-12
- Network configuration 16-12
- Network configuration 2-5
- Network configuration C-6
- Network manager C-2
- Network Messages 5-8
- Network operation C-7

INDEX

| | | | |
|------------------------------|-------|-----------------------------|-------|
| Network primitives | C-7 | Order of Startup | 2-1 |
| Network subroutines | C-7 | Organization, document | 1-3 |
| Network, operator messages | 15-12 | Output stream, definition | 1-8 |
| Network, remote devices | 15-2 | Overview | 1-1 |
| Networks, PRIMOS III | 18-2 | Pack name | 7-3 |
| nodename, local | 16-11 | Packname | 6-1 |
| Nodename, local | 2-5 | Packname, definition | 1-8 |
| Non-standard terminal | 3-7 | PAGDEV (CONFIG subcommand) | 16-15 |
| NPUSR (CONFIG subcommand) | 16-14 | Page length, printer | 9-5 |
| NRUSR (CONFIG subcommand) | 16-14 | Page maps, PRIMOS III | 18-5 |
| NSEG (CONFIG subcommand) | 16-14 | Page maps, PRIMOS IV | 19-9 |
| NTUSR (CONFIG subcommand) | 16-15 | Page states | 19-10 |
| Number of users | 5-5 | Page, definition | 1-8 |
| Object file, definition | 1-8 | Paging device | 16-15 |
| Operating the Spooler | 9-1 | Paging device | 2-4 |
| Operator as user | 3-1 | Paging device, alternate | 16-5 |
| Operator commands | 15-1 | Paging device, alternate | 19-6 |
| Operator Commands, Spooler | 9-2 | Paging space, PRIMOS III | 18-5 |
| Operator tasks | 4-1 | Paging space, PRIMOS IV | 19-5 |
| Operator, Roles of | 1-2 | Paper size, default (spool) | B-2 |
| Operator-to-operator message | 15-12 | Paper tape boot | A-2 |
| Operator-to-user message | 15-12 | Paper, changing on printer | 9-4 |
| OPRPRI (PRIMOS Command) | 15-14 | Parameters, System (list) | 5-5 |
| OPTIONS (FIXRAT option) | 6-5 | Partition | 7-2 |
| Options, FIXRAT printout | 6-5 | Partition, definition | 1-8 |

INDEX

Partitions G-3

PASSWD (PRIMOS Command) 3-4

Password 3-2

Pathname, definition 1-8

Pdev, definition 1-8

Pdisk, definition 1-8

Peripheral device power 2-2

Permanent bad page 19-12

Phantom user messages 5-7

Phantom user number, CX 2-6

Phantom user number, spooler 2-7

Phantom user, definition 1-8

Phantom users 2-4

Phantom users, configure 16-14

Physical device number 6-6

Physical device number 7-1

Physical device numbers G-1

Physical memory pages 2-4

PH_CEN 9-6

PH_FAM (FAM startup) 4-5

PH_GO (CX startup) 4-5

PH_PLT 9-6

PH_PR0 9-6

PH_PRL 9-6

PLEASE ENTER CONFIG 2-4

Plot/Print option (spool) B-2

Power failure F-17

Powering down PRIMOS II 17-6

Preloader 16-3

PREPAG (CONFIG subcommand) 16-16

Prepaging 16-16

PRIMENET 19-2

PRIMENET C-1

PRIMENET local printer 9-10

PRIMENET size C-1

PRIMENET structure C-1

PRIMOS Command ADDISK 15-2

PRIMOS Command AMLC 15-3

PRIMOS Command ATTACH 3-3

PRIMOS Command AVAIL 5-1

PRIMOS Command CHAP 15-6

PRIMOS Command CNAME 7-3

PRIMOS Command COPY 8-3

PRIMOS Command CREATE 3-3

PRIMOS Command CX 5-3

PRIMOS Command DELAY 3-7

PRIMOS Command DELETE 3-5

PRIMOS Command DISCON 15-7

PRIMOS Command DISCON 4-6

PRIMOS Command DISKS 15-8

INDEX

- PRIMOS Command ELIGTS 15-9
- PRIMOS Command FIXRAT 6-1
- PRIMOS Command FUTIL 8-6
- PRIMOS Command LOGIN 3-2
- PRIMOS Command LOGOUT 3-7
- PRIMOS Command LOOK 15-10
- PRIMOS Command MAGRST 8-11
- PRIMOS Command MAGSAV 8-8
- PRIMOS Command MAKE 7-1
- PRIMOS Command MAXSCH 15-11
- PRIMOS Command MAXUSR 15-11
- PRIMOS Command MESSAGE 15-12
- PRIMOS Command OPRPRI 15-14
- PRIMOS Command PASSWD 3-4
- PRIMOS Command PROTEC 3-4
- PRIMOS Command SETIME 15-14
- PRIMOS Command SHARE 15-15
- PRIMOS Command SHUTDN 15-16
- PRIMOS Command SPOOL 9-2
- PRIMOS Command STARTUP 15-18
- PRIMOS Command STATUS 15-20
- PRIMOS Command TERM 3-6
- PRIMOS Command USERS 5-5
- PRIMOS Command USRASR 15-24
- PRIMOS Commands 15-1
- PRIMOS II 17-1
- PRIMOS II Command Device 2-2
- PRIMOS II halt 10-1
- PRIMOS II Operations 17-3
- PRIMOS II power down 17-6
- PRIMOS II restart 17-6
- PRIMOS II Shutdown 17-5
- PRIMOS II, adding to new disks 7-6
- PRIMOS II, Booting 2-2
- PRIMOS II, Bootstrapping 17-3
- PRIMOS II, Changing disk packs 17-5
- PRIMOS II, Error recovery 17-5
- PRIMOS II, Program restrictions 17-4
- PRIMOS II, Real-memory 17-2
- PRIMOS II, Versions of 17-2
- PRIMOS II, Virtual-memory 17-2
- PRIMOS III 18-1
- PRIMOS III halt 10-2
- PRIMOS III halts 18-11
- PRIMOS III in networks 18-2
- PRIMOS III loadmap 18-2
- PRIMOS III startup 18-9
- PRIMOS III versions 18-1
- PRIMOS III, building 18-2
- PRIMOS III, modifying 18-4

INDEX

PRIMOS III, Starting 2-3
 PRIMOS Initialization Error messages 16-18
 PRIMOS IV 19-1
 PRIMOS IV configuration 19-5
 PRIMOS IV halt 10-3
 PRIMOS IV halts 19-13
 PRIMOS IV memory parity tests 19-10
 PRIMOS IV page maps 19-9
 PRIMOS IV segments 19-3
 PRIMOS IV startup 19-7
 PRIMOS IV, building 19-2
 PRIMOS IV, modifying 19-4
 PRIMOS IV, network versions 19-2
 PRIMOS IV, Starting 2-3
 PRIMOS IV, versions 19-1
 PRIMOS messages 2-10
 PRIMOS V 19-1
 PRIMOS V halt 10-3
 PRIMOS V, Starting 2-3
 PRIMOS, definition 1-8
 PRIMOS, Single-user version 17-1
 PRIMOS, starting from a command file 16-3
 PRINET (UFD) 4-2
 Print/Plot option (spool) B-2
 Printer page length 9-5
 Printout options 6-5
 Priority, change 15-6
 Priority, changing 2-6
 Priority, changing 4-1
 Program development using PRIMOS II 17-4
 Program development, PRIMOS II 17-4
 Program load, automatic F-16
 PROTEC (PRIMOS Command) 3-4
 Protecting files 3-4
 Protection keys 3-4
 Protection, directory 3-4
 Protocol, link C-2
 Protocols 15-3
 Q-DOS 17-1
 Queue structure, spool B-2
 Queue, CX 5-3
 Queue, spool 5-3
 R-DOS 17-2
 RAT 6-1
 Read/write lock 16-16
 Real-memory PRIMOS II 17-2
 Reboot F-17
 Record availability table 6-1

INDEX

- Recovering backed up files 4-8
- Register contents 10-5
- Register sets F-1
- Registers, loading F-12
- Related documents 1-4
- Remote device shutdown 15-17
- Remote device startup 15-19
- Remote devices 15-2
- Remote devices, CX queue 5-4
- Remote disk startup 2-9
- Remote file access C-4
- Remote login 3-2
- Remote login C-5
- Remote login specification 2-5
- Remote login, configure 16-16
- Remote users 2-4
- Remote users, configure 16-14
- Remote users, disconnect 15-7
- Replacing software 4-11
- Resource allocation 4-1
- Resource monitoring 4-1
- Restart spool phantom 9-6
- Restart, automatic F-17
- Restarting PRIMOS II 17-6
- Restarting spooler 9-5
- Restoring data from disk 8-6
- Restoring data, tape-to-disk 8-11
- RLOGIN (CONFIG subcommand) 16-16
- Roles of Administrator 1-2
- Roles of Operator 1-2
- RUN mode F-1
- Runfile, definition 1-9
- Running MAKE 7-2
- RWLOCK (CONFIG subcommand) 16-16
- S-DOS 17-2
- Scheduling backup 8-1
- Segment Descriptor Table 19-9
- Segment directories 6-10
- Segment directory 3-1
- Segment number, halt 10-3
- Segment, definition 1-9
- Segments available 16-14
- Segments, PRIMOS IV 19-3
- Segments, shared utilities 4-2
- Segments, user 19-6
- Sequential job processor SEE CX
- Sequential job processor startup 2-6
- Set address display F-9
- Set data display F-9
- Set printer page length 9-5

INDEX

- Set system date 15-14
- Set system time 15-14
- SETIME (PRIMOS Command) 15-14
- SHARE (PRIMOS Command) 15-15
- Shared BASIC/VM startup 2-7
- Shared DBMS startup 2-8
- Shared editor startup 2-7
- Shared libraries 11-1
- Shared libraries startup 2-7
- Shared libraries, installation 11-2
- Shared libraries, re-installation 11-3
- Shared libraries, rebuilding 11-3
- Shared segment incorporation 4-2
- Shared segments 15-15
- Shared SPSS startup 2-7
- Shared subsystems 2-7
- Shared UII subroutines startup 2-7
- Shared utility segments 4-2
- Short/large files (spool) B-2
- SHUTDN (PRIMOS Command) 15-16
- Shutdown messages 2-11
- Shutdown, device 15-17
- Shutdown, equipment 2-11
- Shutdown, remote device 15-17
- Shutdown, system 15-16
- Shutdown, System 2-11
- Shutting down PRIMOS II 17-5
- Single user operations 17-3
- Single-line CONFIG 16-1
- Size, PRIMENET C-1
- SMLC 19-2
- SMLC C-3
- SMLC (CONFIG subcommand) 16-17
- SMLC enabling 2-5
- SMLC lines, configure 16-17
- Software, adding 4-11
- Software, replacing 4-11
- Source file, definition 1-9
- Space utilization 5-1
- Specifying spooler 9-2
- Split disk 7-2
- Split disk, PRIMOS III 18-6
- SPOOL (PRIMOS Command) 9-2
- SPOOL Command -ABORT 9-4
- SPOOL Command -BACK 9-5
- SPOOL Command -CANCEL 9-9
- SPOOL Command -DEFER 9-9
- SPOOL Command -DROP 9-4
- SPOOL Command -EXPAND 9-9

INDEX

- SPOOL Command -FINISH 9-3
- SPOOL Command -FORM 9-9
- SPOOL Command -FTN 9-9
- SPOOL Command -FUNIT 9-10
- SPOOL Command -HANG 9-3
- SPOOL Command -HOME 9-10
- SPOOL Command -LENGTH 9-5
- SPOOL Command -LIST 9-10
- SPOOL Command -LNUM 9-11
- SPOOL Command -LOGOUT 9-6
- SPOOL Command -NOFMT 9-11
- SPOOL Command -OPEN 9-11
- SPOOL Command -PAPER 9-4
- SPOOL Command -PLOT 9-11
- SPOOL Command -RESTART 9-5
- SPOOL Command -TIME 9-3
- SPOOL Command -TUNIT 9-11
- SPOOL Command -USER 9-2
- Spool installation options B-1
- Spool phantom restart 9-6
- Spool print program B-1
- Spool queue 5-3
- Spool queue structure B-2
- Spooler B-1
- Spooler acknowledge time 9-3
- Spooler examples 9-7
- Spooler features 9-2
- Spooler logout 9-6
- Spooler Messages 9-6
- Spooler Operation 9-1
- Spooler Operator Commands 9-2
- Spooler phantom user number 2-7
- Spooler restart 9-5
- Spooler startup 2-7
- Spooler user commands 9-8
- Spoolers, multiple 9-2
- SPOOLQ (UFD) 4-3
- SPOOLQ (UFD) 9-1
- Starting PRIMOS from a command file 16-3
- Starting PRIMOS III 2-3
- Starting PRIMOS IV 2-3
- Starting PRIMOS V 2-3
- Starting up devices 15-2
- STARTUP (PRIMOS Command) 15-18
- Startup command file (contents) 2-13
- Startup CX 2-6
- Startup messages 2-10
- Startup of Equipment 2-1
- Startup order 2-1
- Startup, device 15-18
- Startup, PRIMOS III 18-9

INDEX

| | | | |
|-----------------------------------|-------|--|-------|
| Startup, PRIMOS IV | 19-7 | Supervisor terminal as a user terminal | 15-24 |
| Startup, remote device | 15-19 | Supervisor Terminal Messages | 5-6 |
| Startup, remote disk | 2-9 | Synchronous link | C-3 |
| Startup, Sequential job processor | 2-6 | SYSCOM (UFD) | 4-3 |
| Startup, Shared BASIC/VM | 2-7 | SYSOVL (UFD) | 4-3 |
| Startup, shared DBMS | 2-8 | SYSTEM (UFD) | 4-3 |
| Startup, shared editor | 2-7 | System configuration | 16-1 |
| Startup, shared libraries | 2-7 | System configuration, PRIMOS III | 18-4 |
| Startup, shared SPSS | 2-7 | System crash | 10-1 |
| Startup, shared UII subroutines | 2-7 | System crash | 19-10 |
| Startup, spooler | 2-7 | System crashes | 4-9 |
| Startup, system | 15-18 | System date | 2-9 |
| Startup, System | 2-1 | System halt | 10-1 |
| STATUS (PRIMOS Command) | 15-20 | System halts | 4-7 |
| Status words | E-1 | System intialization, PRIMOS III | 18-9 |
| stop spool printing | 9-3 | System Logbook | 5-15 |
| Storage module device number | G-4 | System monitoring | 5-1 |
| Storage module errors | E-1 | System Parameters (list) | 5-5 |
| STORE | F-3 | System shutdown | 15-16 |
| Stream, output, definition | 1-8 | System Shutdown | 2-11 |
| Structure, PRIMENET | C-1 | System startup | 15-18 |
| Sub-UFD, definition | 1-9 | System Startup | 2-1 |
| Subroutines, network | C-7 | System status | 15-20 |
| Subsystem usage | 4-4 | System terminal Baud rate | 16-6 |

INDEX

- System time 2-9
- System UFDs 4-2
- SYSTEM, UFD for spooler 9-1
- SYSTEM>DISCS file 5-3
- tape drive, magnetic, power 2-2
- Tape-to-disk data restoring 8-11
- TERM (PRIMOS Command) 3-6
- TERM Command -BREAK 3-6
- TERM Command -DISPLAY 3-6
- TERM Command -ERASE 3-6
- TERM Command -FULL 3-6
- TERM Command -HALF 3-6
- TERM Command -KILL 3-6
- TERM Command -LF 3-6
- TERM Command -NOLF 3-6
- TERM Command -NOXOFF 3-6
- TERM Command -XOFF 3-6
- Terminal characteristics 3-6
- Terminal Functions 1-5
- Terminal I/O buffer size 16-5
- Terminal user messages 5-6
- Terminal users 2-4
- Terminal users, configure 16-15
- Terminal, non-standard 3-7
- Time, set system 15-14
- Time, system 2-9
- Timeout, inactivity 13-1
- Timeout, inactivity 16-10
- Timeslice, change 15-6
- Timeslice, changing 4-1
- TIMOUT 13-1
- TO disk 8-3
- Tree, directory 6-2
- Treename, definition 1-9
- Truncation of file 6-9
- TYPOUT (CONFIG subcommand) 16-17
- UFD 3-1
- UFD compression 6-5
- UFD, Current 3-3
- UFD, definition 1-9
- UFD, Home 3-3
- UFDs, Adding to the system 4-11
- UFDs, System 4-2
- UII package 19-7
- Uncorrected memory parity 10-7
- User commands, spooler 9-8
- User File Directory 3-1
- User number from control panel 10-6
- User segments 19-6
- User, phantom, definition 1-8

INDEX

User-to-operator message 15-12
User-to-user communication C-7
USERS (PRIMOS Command) 5-5
Users, number of 5-5
USRASR (PRIMOS Command) 15-24
Verification, disk 7-5
Virgin Disk 7-4
Virtual-memory PRIMOS II 17-2
Volume name 7-3
Volume, definition 1-9
Volume-name, definition 1-9
Warm start 10-7
Warm Start 19-12
Word, definition 1-9
write error 7-5
X-OFF 3-6
X-ON 3-6

