

MAN2603

PRIMOS COMPUTER ROOM  
User Guide

Revision A  
January 1977

**PRIME**  
Computer, Inc.

145 Pennsylvania Ave.  
Framingham, Mass. 01701

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

Performance characteristics are  
subject to change without notice.

# CONTENTS

	Page
	-----
FOREWORD	
SECTION 1      PRIMOS II OPERATION	
OVERVIEW	1-1
DISKS	1-3
STORAGE MODULE	1-10
STARTUP OF PRIMOS III (FROM PRIMOS II)	1-15
STARTUP OF PRIMOS IV	1-15
INITIAL OPERATING SESSION	1-16
DISK BUILDING	1-16
PROGRAM DEVELOPMENT USING PRIMOS	1-20
BACKUP	1-23
SHUTDOWN	1-23
RESTARTING PRIMOS	1-24
SECTION 2      PRIMOS III OPERATION	
PRIMOS III SYSTEM OVERVIEW	2-1
PRIMOS III VERSIONS	2-1
PRIMOS III SYSTEM INITIALIZATION	2-3
USER LOGIN/LOGOUT	2-6
SYSTEM CONFIGURATION	2-7
WARM RESTART FOR PRIMOS III	2-15
SECTION 3      PRIMOS IV OPERATION	
OVERVIEW OF PRIMOS IV OPERATION	3-1
PRIMOS IV VERSIONS	3-2
SECTION 4      FIXRAT	
FIXRAT DESCRIPTION	4-1
RUNNING FIXRAT	4-2
FIXRAT OUTPUT EXAMPLE	4-6
PITFALLS	4-10
SECTION 5      MAKE	
OVERVIEW	5-1
RUNNING MAKE	5-1
PITFALLS	5-4

## CONTENTS (Cont)

### SECTION 6 COPY

RUNNING COPY	6-1
DISK CONFIGURATION	6-1
COPY METHOD	6-4
COPYING STORAGE MODULE	6-7

### SECTION 7 OPERATOR COMMANDS

PRIMOS III, IV INTERNAL COMMANDS	7 1
ADDISK	7-2
AMLC	7-3
CONFIG	7-5
DISKS	7-8
MESSAGE	7-9
SETIME	7-11
STARTUP	7-12
SHUTDN	7-13
STATUS	7-14
USRASR	7-16

### SECTION 8 COMPUTER ROOM OPERATIONS

DEFINITION OF TERMS	8-1
STARTUP OF PRIMOS	8-3
BOOTING IN (PRIMOS II)	8-4
CRASH PROCEDURE	8-5
WARM STARTING PRIMOS	8-7
POWERING UP OR DOWN SYSTEM	8-8
STARTING UP DISKS	8-11
USING MAGSAV MAGRST	8-12
MOUNTING A TAPE ON THE 9-TRACK TRANSPORT	8-13
LINE PRINTER OPERATION TIPS	8-13

### APPENDIX A BOOTSTRAPS

CONTROL PANEL BOOTS	A-1
SECOND LEVEL DISK BOOTS (BOOT)	A-5

### APPENDIX B DISK RECORD HEADER FORMATS

### INDEX

# ILLUSTRATIONS

<u>Figure No.</u>	<u>Title</u>	<u>Page</u>
4-1	Sample File Structure	4-3
4-2	Typical FIXRAT Traverse of File Structure	4-5
8-1	Disk Device Numbers	8-11

## TABLES

<u>TABLE NO.</u>	<u>TITLE</u>	<u>PAGE</u>
1 1	Physical Disk Numbers	1-1
1-2	Head Offset Definitions	1-6
1-3	Number of Heads Definition	1-7
2-1	Disk Space Required for 32K per User	2-10
2-2	Disk Space Required for 64K per User	2-11
5-1	Records Parameters for 30.M Word Disks	5-2
6-1	Partitioned Disk SIZE Specifications	6-4

## FOREWORD

This user guide provides machine room operating information for the Prime PRIMOS operating systems. The operating procedures are systems oriented rather than hardware oriented. Information is organized as follows:

- Section 1 Discusses operation of the PRIMOS II operating system, which is fundamentally a single-user disk-based operating system.
- Section 2 Discusses operation of PRIMOS III, which is a virtual memory, multi-user operating system for the Prime 300. Section 2 also gives many tips and details on PRIMOS III system generation and configuration.
- Section 3 Discusses operation of PRIMOS IV, the operating system for the Prime 400. PRIMOS IV is a virtual memory operating system which takes advantage of the Prime 400 architecture and segmentation to provide the user with both a vast amount of virtual memory space and faster execution of user programs.
- Section 4 Is a detailed description to guide the user in running FIXRAT, a file and disk maintenance command. This section is important to the user who wishes to maintain the integrity of his computer system files.
- Section 5 Describes MAKE, a command that allows the user to build his own unique disks from the Prime-supplied master disk volumes.
- Section 6 Describes PRIMOS III IV internal commands. These commands are normally used by the operator at the supervisor terminal in the computer room.
- Section 7 Describes COPY, a command that allows the user to copy one disk surface to another. COPY is important to PRIMOS II system initialization.
- Section 8 An overview of PRIMOS systems operation presented in the form of a sample of operating procedures at an existing computer site.
- Appendix A Discusses the different bootstrap facilities that are available to the user for bootstrapping the operating system into the computer system memory.
- Appendix B Describes disk record header formats for

## FOREWORD

both 448-word and 1040-word records.

### REFERENCE DOCUMENTS

This user guide is written with the assumption that the computer room operator is familiar with the mechanical operating procedures for equipment such as the CPU control panel, the various disks and their associated control panels, and peripheral devices. The following table references documents that detail operating procedures for various equipment items that are commonly present within Prime computer configurations.

<u>DOCUMENT NUMBER</u>	<u>TITLE</u>
MAN1671	SYSTEMS REFERENCE USER GUIDE (for Prime 100, 200, 300)
MAN1672	OPERATORS GUIDE
MAN1940	MAGNETIC TAPE USER GUIDE
MAN1942	LINE PRINTER USER GUIDE
MAN1946	TERMINAL USER GUIDE
MAN1948	FIXED AND MOVING HEAD DISKS USER GUIDE
MAN1949	DISKETTE USER GUIDE
MAN2798	PRIME 400 SYSTEMS REFERENCE USER GUIDE



## SECTION 1

## PRIMOS II OPERATION

This section describes how to load and start PRIMOS II, summarizes the essential operator tasks during an operating session, and describes PRIMOS II equipment configuration details.

## OVERVIEW OF PRIMOS II OPERATION

Sectored Mode and Relative Mode

Two versions of PRIMOS II are supplied on the master disk, sectored-mode and relative-mode PRIMOS II. Sectored-mode PRIMOS II is 11000 (octal) locations in size and is loaded in 32S addressing-mode. It cannot be loaded or even called from above 32K. Relative-mode PRIMOS II is currently 14000 (octal) locations in size and is loaded in 64R addressing-mode. Relative mode PRIMOS II can be loaded or called from above 32K, and it supports the storage module disk drive as file-system device.

Sectored-mode PRIMOS II and relative-mode 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 processor. The virtual-memory configurations run in user-space under the PRIMOS III or PRIMOS IV operating system on a Prime 300 or 400 processor.

The locations and names of the PRIMOS II configurations on the master disk as supplied as follows:

<u>UFD</u>	<u>File</u>	<u>Contents</u>	<u>Low</u>	<u>Restart</u>	<u>High</u>
DOS	*DOS16	16K real-memory S-DOS	027000	030000	037777
DOS	*DOS24	24K real-memory S-DOS	047000	050000	057777
DOS	SDOS32	32K real-memory S-DOS	067000	070000	077777
DOS	*DOS32	32K real-memory R-DOS	064000	070000	077777
DOS	*DOS64	64K real-memory R-DOS	164000	170000	177777
DOS	VDOS16	16K virtual-memory S-DOS	027000	030000	037777
DOS	VDOS24	24K virtual-memory S-DOS	047000	050000	057777
DOS	VDOS32	32K virtual-memory S-DOS	067000	070000	077777
DOS	UDOS32	32K virtual-memory R-DOS	064000	070000	077777
CMDNC0	DOSEXT	saved-file containing hybrid commands for S-DOS			
CMDNC0	DOSXT	saved-file containing hybrid commands for R-DOS			

The banner printed when relative-mode PRIMOS II is loaded is:

```
PRIMOS II REV 11.0 07/15/76 (AT nnnnnn)
```

where nnnnnn is the restart location. The banner printed when sectored-mode PRIMOS II is loaded is:

```
DOS REV. 10.0 2/19/76 (AT nnnnnn)
```

### Bootstrap Loading

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 \*DOSnn from UFD DOS, where nn is 16, 24, 32 or 64, depending upon main-memory size. The bootstrap no longer limits nn to 32 on 64K machines. Sense switches 1-3 are used to limit the memory size determination to less than the available physical memory. The sense switch settings are:

```
00XXXX highest DOS that will fit;
04XXXX *DOS16;
06XXXX *DOS24;
16XXXX *DOS32;
else error.
```

With the UFD named DOS, the bootstrap loads sectored-mode PRIMOS II in 16K and 24K machines and relative-mode PRIMOS II in 32K and 64K machines. An installation may desire to rename \*DOS32 to RDOS32 and SDOS32 to \*DOS32. Thereafter, the bootstrap loads sectored-mode PRIMOS II in a 32K machine at that installation. The selection can be reversed by renaming \*DOS32 to SODS32 and RDOS32 to \*DOS32 again.

It is possible but undesirable to load and use relative-mode PRIMOS II in 16K and 24K configurations, because of the reduced amount of main memory left. It is impossible, to load sectored mode PRIMOS II above 32K.

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. Relative-mode PRIMOS II supports the storage module fully as a file-system device.

#### Virtual-Memory PRIMOS in CMDNC0

With the availability of 64K PRIMOS II, the virtual-memory PRIMOS III supplied in UFD CMDNC0 of the master disk has been changed from VDOS32 (32K sectored-mode PRIMOS III) to UDOS64 (64K relative-mode PRIMOS III). VDOS32 continues to be supplied, but it is now in the UFD named DOS.

#### Supervisor Calls to Relative-Mode PRIMOS II

The restriction that all supervisor calls be confined to the first 32K of main memory is removed with relative-mode PRIMOS II. Programs operating in 64R addressing mode can run and make supervisor calls from anywhere in 64K of memory, above or below the operating system. Of course, programs operating in 32S or 32R addressing modes are confined to the first 32K of memory, and programs operating in 16S addressing mode are confined to the first 16K of memory. However, all of these programs may run with and make supervisor calls to a relative-mode PRIMOS II located anywhere in 64K of memory. Table 1-1 gives a list of the disk assignment numberse for all PRIMOS systems. For the storage module physical disk, refer to the paragraph on STORAGE MODULE in this section.

#### DISKS

The following table (Table 1-1) lists the physical numbers and the types of disks that may be used when running PRIMOS II. Subsequent paragraphs discuss logical disk number, disk data transfer and disk partitioning.

Table 1-1. Physical Disk Numbers.

---

<u>Physical Disk Drive Number</u>	<u>Controller Option</u>	<u>Description</u>	<u>Sectors/Track</u>
0	4000	Removable surface of first MH (moving head) disk drive (upper surface)	8
1	4000	Fixed surface of first MH disk drive (lower surface)	8
2	4000	Removable surface of second MH disk drive (upper surface)	8
3	4000	Fixed surface of second MH disk drive (lower surface)	8
4	4000	Removable surface of third MH disk drive (upper surface)	8
5	4000	Fixed surface of third MH disk drive (lower surface)	8
6	4000	Removable surface of fourth MH disk drive (upper surface)	8
7	4000	Fixed surface of fourth MH disk drive (lower surface)	8
10	4000	First fixed head disk drive	8
20	4300	First floppy disk drive	4
21	4300	Second floppy disk drive	4
22	4300	Third floppy disk drive	4
23	4300	Fourth floppy disk drive	4
24	4300	Fifth floppy disk drive	4
25	4300	Sixth floppy disk drive	4
26	4300	Seventh floppy disk drive	4

27	4300	Eighth floppy disk drive	4
30	4002	Removable surface of first MH (moving head) disk drive (upper surface)	8
31	4002	Fixed surface of first MH disk drive (lower surface)	8
32	4002	Removable surface of second MH disk drive (upper surface)	8
33	4002	Fixed surface of second MH disk drive (lower surface)	8
34	4002	Removable surface of third MH disk drive (upper surface)	8
36	4002	Removable surface of fourth MH disk drive (upper surface)	8
37	4002	Fixed surface of fourth MH disk drive (lower surface)	8
40	4002	Fixed head disk drive	64
50	4002	Removable surface of first MH (moving-head) disk drive (upper surface)	32
51	4002	Fixed surface of first MH disk drive (lower surface)	32
52	4002	Removable surface of second MH disk drive (upper surface)	32
53	4002	Fixed surface of second MH disk drive (lower surface)	32
54	4002	Removable surface of third MH disk drive (upper surface)	32
55	4002	Fixed surface of third MH disk drive (lower surface)	32
56	4002	Removable surface of fourth MH disk drive (upper surface)	32
57	4002	Fixed surface of fourth MH disk drive (lower surface)	32

5050	4003/4002	First 30-million word MH disk drive (controller address = 21)	32
5052	4003/4002	Second 30-million word MH disk drive (controller address = 21)	32
5054	4003/4002	Third 30-million word MH disk drive (controller address = 21)	32
5056	4003/4002	Fourth 30-million word MH disk drive (controller address = 21)	32
5250	4003/4002	First 30-million word MH disk drive (controller address = 23)	32
5252	4003/4002	Second 30-million word MH disk drive (controller address = 23)	32
5254	4003/4002	Third 30-million word MH disk drive (controller address = 23)	32
5256	4003/4002	Fourth 30-million word MH disk drive (controller address = 23)	32

---

## NOTES TO TABLE 1-1:

The logical-to-physical assignment depends on the order in which the physical device numbers are listed as parameters in the STARTUP command. The physical device number specified in the Ldisk0 position is assigned as logical disk unit 0, the physical device number specified in the Ldisk1 position is assigned as logical disk unit 1, and so on.

## Example:

```
STARTUP 5054 5056 5052 5050
```

The physical disks are 5054, 5056, 5052, and 5050; where:

physical 5054 is logical 0, physical 5056 is logical 1, physical 5052 is logical 3, and physical 5050 is logical 4. The number of parameters in STARTUP indicate to PRIMOS II the number of logical drives assigned to the system.

## CAUTION:

When changing disks, a SHUTDN is required. Otherwise, PRIMOS will use parameters (such as record availability) applicable to the previous replaced disk with possible loss of file directories. With SHUTDN, PRIMOS II shuts down all disks, and PRIMOS III allows specific disks to be shut down.

The codes shown in Table 1-1 are used in the STARTUP command and the ASSIGN command (refer to the Interactive User's Guide (MAN 2602)). The codes are also used by the utility commands FIXRAT, MAKE, and COPY (refer to Sections 4, 5, and 6).

The physical device codes are the same for three- or six-million word disk drivers connected to the controller.

The storage module disk is fully supported using a record of 102 words. For further information, refer to the paragraphs describing storage module support, later in this section.

#### Data Transfer Between Disks

An eight-sector/track disk pack written on a drive connected to the 4000 controller cannot be read on a drive connected to the 4002 controller and vice-versa, because the method of computing hardware checksum written on the pack is different on the two controllers. A special conversion program to convert packs written on one controller to read on the other controller must be written. A 32-sector/track pack cannot be read or written on a drive connected to the 4000 controller. An attempt to read a 32-sector/track pack using physical device numbers for an eight-sector/track pack will fail. Similarly, an attempt to read an eight-sector/track pack using physical device numbers for a 32 sector/track pack will fail. It is important to keep straight the identification of the disk pack. Each pack should be labeled with the range of physical device numbers appropriate to the disk pack.

Unlike the other disks, only even numbered physical disk drive numbers are allowed for the 30-million word disk drives. There may be up to four drives connected to a type 4001/4002 controller that is configured to the system. The default device address for a type 4003/4002 controller is 21, and the disk drive numbers associated with this drive are 5050, 5052, 5054, and 5056. Similarly, a 4003 or 4002 controller may have a device address of 23, and disk drive numbers associated with this controller are 5250, 5252, 5254 and 5256. A system configuration could have two type 4003/4002 controllers and up to eight 30-million word disks connected.

#### Partitioning Disks

A user may partition a 30-million word disk into two or more sub-disks, via use of the MAKE command (refer to Section 4). Each partition of a disk (sub-disk) is treated by the system commands, PRIMOS, FIXRAT and COPY, as if it were a physically separate disk. Each sub-disk contains its own MFD, DSKRAT, BOOT, CMDNC0, etc. Partitions are defined by a starting head address (relative to head 0 of the disk), a number of contiguous heads and the physical disk number. The minimum partition contains two heads (i.e., three million words). When a partition of a 30-million word disk is present, the physical disk number varies from those shown in Table 1-1. The number of heads is reflected in the second two digits. Tables 1-2 and 1-3 are useful in constructing partitions (the X's represent "don't care" octal digits).

The physical disk number defining a partition on a disk is generated by merging the head offset with the number of heads and with the disk device number. For the purposes of forming a physical disk number for a partitioned disk, the physical disk device numbers are considered to be: 50, 52, 54, and 56 for disks on disk drives 1 to 4 on the type 4003/4002 controller.

Example:

The physical disk number for a disk partition having a head offset of 00 and the number of heads of 20 would be calculated as follows: (assume that the device address of the type 4003/4002 controller is 21): first, look up at the appropriate numbers in Tables 1-2 and 1-3, then

head offset + number of heads + physical disk device number

00XXXX + XX50XX + 56 = 005956

As another example, consider a disk split into two partitions with the disk being the first disk on the type 4003/4002 controller with device address 23; the first portion has head offset = 0 heads and number of heads = 10. The physical disk for the first partition is:

00XXXX + XX26XX + 50 = 002650

The second partition has head-offset = 10 heads and number-of-heads = 10. The physical disk for the second partition is:

05XXXX + XX24XX + 50 = 05250

#### Physical Device Numbers Usage

The physical device codes described previously in this section are used in the ASSIGN, CONFIG, DISKS, SHUTDOWN, STARTUP, STATUS, and UNASSIGN commands. These device codes are also used by the utilities FIXRAT, FUTIL, MAKE, and COPY.



Table 1-2. Head Offset Definitions.

---

Offset	Physical Disk Numbers
0 heads	00XXXX
2 heads	01XXXX
4 heads	02XXXX
6 heads	03XXXX
8 heads	04XXXX
10 heads	05XXXX
12 heads	06XXXX
14 heads	07XXXX
16 heads	10XXXX
18 heads	11XXXX

---

Table 1-3. Number of Heads Definition.

Number of Heads	Type 4003/4002 Controller Address = 21 (default) Physical Disk Number	Type 4003/4002 Controller Address=23 (explicit) Physical Disk Number
2 heads (default)	XX00XX	XX02XX
2 heads (explicit)	XX0400	XX06XX
4 heads	XX10XX	XX12XX
6 heads	XX14XX	XX16XX
8 heads	XX20XX	XX22XX
10 heads	XX24XX	XX26XX
12 heads	XX30XX	XX32XX
14 heads	XX34XX	XX36XX
16 heads	XX40XX	XX42XX
18 heads	XX44XX	XX46XX
20 heads	XX50XX	XX52XX

## STORAGE MODULE DISK SUPPORT

Record Size and Format

Relative-mode PRIMOS II supports the storage module disk drive, both 80- and 300-million byte sizes.

The storage module uses a physical record size of 1040 words instead of 448 words. The storage module is the only disk device to use a physical record size other than 448 words. The physical record size is of concern to users in subroutine PRWFIL (refer to MAN 2604) in absolute position requests and in requests to return to the current position. It is also of concern in user calls to subroutine GINFO.

The concept of a virtual record size is introduced for programming convenience and consistency. A virtual record is constant on any disk and is 448 words per record (eight words of header followed by 440 data words). Thus, a file can be specified in terms of record number and word number. The record number/word number pair used to specify absolute position then becomes a convenient way of specifying a position in a file, if that absolute position requires more than 16 bits for complete specification.

#### Storage Module Disk Number

The encoding of the storage module disk numbers is as follows:

The type is 6.

The number-of-heads field and the head-offset field are identical to the corresponding fields on the 30-million word disk. For details, refer to the discussion on head offset and number-of-heads calculation in this section, and Tables 1-2 and 1-3. Unlike the 30-million word disk specification, the low-order bit of the disk number is combined with the number-of-heads field to specify the addition of one head to the partition. Only the last partition of a storage module disk may contain an odd number of heads, since there is no way to specify an odd head offset. The following examples demonstrate these principles.

<u>Value</u>	<u>Meaning</u>
60	Not allowed; does not default to two heads.
1061	Five-head partition, no offset, first unit of first controller.
1063	Five-head partition, no offset, second unit.
1261	Five-head partition, no offset, first unit of second controller.
1060	Four-head partition, no offset, first unit.
20061	One-head partition, four-head offset, first unit.

A more detailed explanation of the storage module physical device number involves its coding as seen below:

FFFFHHHHA0110UUH

Bits 1-4 (FFFF) comprise the head offset divided by two. Bits 5-8 and 16 (HHHH) are the number of heads in the partition (minimum 2, maximum 5; bit 9 (A) is the alternate controller bit (controller '26 if reset, '27 if set). Bits 10-13 comprise the device type, which is 6 for storage module, and bits 14-15 (UU) the unit number.

### Buffer Size

PRIMOS II selects its buffer size at the time of the STARTUP command. If no storage module devices are started up, PRIMOS runs with 448 (700 octal) word buffers. If any storage module devices are started up, buffers are 1042 (2022 octal) words long. The STATUS command types out the selected buffer size. When holding a storage module record, the larger buffer is organized as 16 (20 octal) words for header, 1024 (2000 octal) words of data, and two words for an error detecting and correcting code checksum. When holding a smaller record, the larger buffer is organized as eight (10 octal) words unused, eight (10 octal) words of header, 440 (670 octal) words of data, and the remainder unused. The format of a storage module record header is:

+0,+1	current record address;
+2,+3	father record address;
+4	data word count;
+5	file type;
+6,+7	next record address;
+8,+9	previous record address;
+10...+15	spares, must be zero.

### Header Size

To identify the header size as 16 words, the high-order bit of the type field is set in the headers of records 0 and 2 only of any storage module partition (the headers of the BOOT and the first record of the DSKRAT). Partition sizes on a storage module are limited to five heads in size. This restricts record addresses to single-word integers and permits storage module headers to be transformed, when in memory buffers, into the same eight-word format used by all other devices.

### File System Calls with Storage Module

All file system calls operate as described in the File System User Guide MAN 2604. The user program need not be aware that it is accessing a storage module device. Position information transmitted in PRWFIL calls is treated as if all records had a data size of 440 (670 octal) words, regardless of whether or not that is the actual record size.

PRWFIL calls, that request the read-convenient state, terminate transfers of 1024 (instead of 448) words on the storage module. If a smaller buffer size is specified (e.g., 448-words), the smaller buffer is not overrun and all existing user programs continue to run correctly with storage module devices.

#### UFD's on Storage Module

The size of a UFD is still limited to one physical record. However, the larger record size permits UFD's to contain  $(1024-8)/6 = 169$  entries on storage module devices. UFD's are limited to 72 entries on other devices.

#### DAM Files on Storage Module

Due to the larger size of index and data records, direct-access files on storage module devices have a longer reach before access becomes sequential. The first 1048576 (4000000 octal) words of a storage module file are within direct-access reach. Only the first 193600 (572100 octal) words of files on other devices are within direct-access reach.

Segment directory operation is unchanged. Any user application program employing segmented files will continue to run correctly on storage module devices at this revision. The limitation of storage module partition sizes to five heads keeps file-system record addresses confined to single-word integers for Revision 11.

#### Storage Module Error Detection

The storage module controller writes a two-word correcting code checksum on each record. It is a five-code 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 storage module reads by the operating systems.

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).

The normal disk error message is of the form:

DISK PD ER <physical device><record address><CRA><disk status word>

CRA is valid only on read requests. CRA is the identifier on the record read, which should match the requested record <record address>. The disk status word for the storage module disk is as follows:

<u>Status Word</u>	<u>Meaning</u>
177777	Bad record identifier
177776	Device not ready
100000	Always set
040000	DMX overrun
010000	Check error
004000	Checksum error
002000	Header check failure
000010	Disk drive seeking
000004	Illegal seek
000002	Select error
000001	Not available or not ready

The storage module controller supports ECC. A few existing customers may require a field update to the controller to run current software.

Control Panel BOOT from Storage Module

The control panel boot program (refer to Appendix A) has been modified to support the storage module. Customers must obtain Rev. G PROM for the purpose of BOOTING from the storage module. The sense switch setting to boot from the storage module is 114 octal. The boot types the message:

PHYSICAL DEVICE =

The user then types the appropriate device number for the storage module.

#### CAUTION

The BOOT does not boot from the storage module if the boot is read from a disk other than a storage module.

#### Procedure for Storage Module Users to Convert to a 1040-Word Format

1. Bring up REV. 10, PRIMOS III or IV.
2. Move all files from the storage module disks to either magnetic tape or other disks. Use MAGSAV to move files to magnetic tape. Use FUTIL (MAN 2604) to move files to non-storage module disks.
3. Bring up PRIMOS II (formerly DOS).
4. MAKE the storage module disks with saved information (refer to Section 5). Use the VIRGIN DISK and VERIFY DISK option of MAKE to reformat the entire disk for large records. Be sure to run the current version of MAKE. MAKE all partitions on the disk, including the paging partition.
5. Use MAGRST (MAN 2602) or FUTIL to move files back to the storage module packs.
6. If a pack or partition of a pack is to be used for BOOTING, move \*DOS32 and \*DOS64 to the named UFD DOS from the master disk volume 1.
7. Flush all commands from CMDNC0 and replace with commands from the current master disk.
8. Bring up PRIMOS III or IV according to the directions in Sections 2 and 3 of this user guide.

User File Directories (UFD's) are constrained to be one physical record. Because the storage module has larger records, it is possible that more files than may be described by the entries in a single UFD may possibly be stored in a UFD (up to 169 files).

As the storage capacity of the storage module disk is so large, the disk must be partitioned to keep the number of records per partition below 65,536 (64K). The largest allowable partition is eight heads containing 59,256 records. The 80MB five-head storage module can be one partition; the 19-head storage module must be split into three partitions.

If a storage module is available, PRIMOS II can be loaded from it. The sense switch settings are 000114 (highest PRIMOS II that will fit into available memory) or 100114 (32K PRIMOS II). Sectored-mode PRIMOS II does not support the storage module as a device; therefore, 16K or 24K PRIMOS II may not be bootloaded when it is necessary to access a storage module.

#### STARTUP OF PRIMOS III

Once PRIMOS II is started and running, PRIMOS III can be started (for details, refer to Section 2). To invoke PRIMOS III, give the command: PRIMOS3. Note that the response of PRIMOS III to a valid command is: OK, ("OK" followed by comma, not a colon), which indicates which of the operating systems has control.

#### STARTUP OF PRIMOS IV

PRIMOS IV is also started from PRIMOS II. For details, refer to Section 3. To invoke PRIMOS IV, give the command:

PRIMOS

#### 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. To determine which spare UFD's are available, ATTACH to the MFD and do a LISTF:

OK: A MFD XXXXXX

OK: LISTF

UFD=MFD 0

DSKRAT	MFD	BOOT	CMDNC0	LIB	SRCLIB	DIAG	PMA
FORTRN	LDR	BASIC	FLIB1	FLIB2	FLIB3	FLIB4	FLIB5
FLIB6	LIB7	LIB8	IOCS	AIDS	ED	BINED	TM
PRIMII	RTOS	PRIMOS	IIVM	RTOSVM	INDEX		

Note that the MFD has a password, XXXXXX, which acts as the default at the time the disk is MADE'd, to discourage casual or inadvertent use of this important directory.



## CAUTION

Do not attach to MFD for a program development or normal file handling tasks. Be very careful in entering commands while attached to this UFD. FIXRAT can be used to fix the disk if the UFD's DOS, BOOT, or CMDNC0 are intact; but if any of the files in this MFD are damaged, the master disk is compromised.

## DISK BUILDING (COPYING MASTER DISK PACK)

Disk building consists of three phases: format the disk; move run files of PRIMOS II (\*DOS16, \*DOS24, \*DOS32, \*DOS64, as appropriate) onto the UFD named DOS; move any desired external commands onto the UFD CMDNC0 and/or move libraries onto the UFD LIB.

Format of PRIMOS II Disk

If a PRIMOS II master disk (or any other PRIMOS II disk) is available, it can be COPY'd onto the virgin disk. Unneeded files on a new pack can be deleted and new files can be created.

If no PRIMOS II disk is available and an empty disk is desired, the MAKE program can be run (the COPY and MAKE programs are described in Sections 5 and 6. When a disk is formatted using MAKE, any needed files are then copied from the master disk onto the new disk; other files and programs can be created later. If you plan to bootload from this new disk, then the UFD named DOS must contain the files \*DOS16, \*DOS24, \*DOS32, and \*DOS64. MAKE ensures that an executable and correct BOOT is written onto record 0 of the disk.

PRIMOS II supports the storage module disk and the Terminet system terminal. Those users with 64K or more contiguous memory invoke a PRIMOS II loaded at the top of 64K, following the BOOT procedure (refer to Appendix A). If this version of PRIMOS II is inconvenient for users with programs accessing these locations, the user may request that a 32K PRIMOS II be loaded by setting sense switches 1-4 before BOOTing.

The newer features of PRIMOS II provide 3000 additional octal locations; however, 16K and 24K PRIMOS are unchanged. The commands LOAD and PSD, as supplied on the master disk, do not run with 32K PRIMOS II. In the event that users have programs too large to run with the larger versions of PRIMOS II, 32K PRIMOS II is supplied as file SDOS32 in the UFD named DOS. To cause this version to be BOOTed, ATTACH to the UFD named PRIMOS and perform the following CNAME comments:

```
CNAME *DOS32 RDOS32
CNAME SDOS32 *DOS32
```

Then, BOOT DOS, as described in this section and Appendix A.

The commands FTN, ED, and PMA cannot be used with \*DOS16 or \*DOS24. \*DOS16 or \*DOS24 users must use the commands SFTN, SED, and SPMA. SFTN does not support 32-bit integers. It is suggested that DOS16 and DOS24 users delete FTN, ED, and PMA from their command directory (CMDNC0) and rename (CNAME): SFTN to FTN, SED to ED, and SPMA to PMA for ease of operation.

### Building a PRIMOS II Disk from Paper Tape

Disk packs can be ordered with full CMDNC0 etc.; however, if no PRIMOS II formatted disks are available, one must be created from Prime-supplied paper tapes. All tapes provided are MDL self-loading tapes and are loaded into memory using the control panel boot. The first step is to load MAKE and run it (refer to Section 5). The following procedure is required:

1. Load BDOSV2 (PRIMOS II bootstrap tape). This loads the loader \*PRIMOS I116 as well as other necessary modules.

Start at '6765.

After the header is typed and PRIMOS II prompts OK:, type the following commands:

STARTUP (Pdev) where Pdev is a physical device number.

ATTACH PRIMOS II

SAVE \*PRIMOS I116 7000 17777 31000 20000 0 0 2000

2. Boot the new \*PRIMOS I116 from the disk using either:

Control panel boot (Sense Switch 2 set)

Paper tape PRIMOS IIBOOT (SLT-start at '1000). Set Sense Switch 2 after the tape has been loaded.

3. Any other command can be added by loading it into memory (control panel boot), starting at '30000 (Starting Address, SA, of PRIMOS II, ATTACHing to CMDNC0 and SAVEing the command). Refer to the Interactive Users' Guide (MAN 2602) for a description of these commands.

4. Use EDB to read any binary files (e.g., FTNLIB) from paper tape.

5. Use ED to read any source files (e.g., DRATIT) from paper tape.

Once the disk is READY, the user can resume PRIMOS II operation. A new STARTUP is required.

Example:

OK: STARTUP 5051

This establishes the fixed surface as logical unit 0; all PRIMOS II disk activity supporting the assembler, compiler, editor (etc.) uses logical unit 0.

#### Creating Additional PRIMOS II Disks

Every PRIMOS II disk must contain a BOOT, a DSKRAT file, an MFD, the command UFD (e.g., CMDNC0). The user must add the command programs FIXRAT, COPY, and the required UFD's. The easiest way to convert a blank disk pack to a PRIMOS II disk is to run the MAKE program (refer to the MAKE command description in Section 5). Another method is to copy the active PRIMOS II disk from its present location (for example, from the fixed surface to the removable surface of a moving head disk drive which is accomplished as follows:

OK: COPY (Copy operation begins)

Any number of PRIMOS II compatible disk packs can be made in this way. Of course, much of the available file space on an original PRIMOS II master disk is occupied. To make room, the user can delete duplicate UFD's or files on a new pack (original) as required. Only the PRIMOS II disk assigned as logical unit 0 needs to have the full set of PRIMOS II command files, UFD's, library, etc. On other disks to be used mainly for user's data or program files, the surplus UFD's and the files within them, can be deleted. To determine that the files provide the most space, run a FIXRAT to observe the number of disk records occupied by each UFD. To delete UFD's, attach to the MFD and enter DELETE commands.

Example:

OK: A MFD XXXXXX

FUTIL  
 >TREDEL LIB  
 >TREDEL PMA  
 >TREDEL FORTRAN  
 >QUIT

OK:

New UFD's can be created with the CREATE command. Alternatively, if the user needs many UFD's, he could use CNAME to change the UFD names, then attach to the UFD's and delete the files within them, using FUTIL (refer to the File System User's Guide, MAN 2604).

Entering New UFD's

Another method for coining new UFD names is to attach to the MFD and use the CREATE command.

Example:

OK: A MFD XXXXXX

OK: CREATE BDAWES

The above creates a new UFD named BDAWES in the MFD. The user must attach to a UFD other than the MFD as soon as possible, to reduce the likelihood of spoiling any of the MFD files.

Creating a Sub-UFD

The command sequence:

A BDAWES  
CREATE BOB

creates a UFD named BOB that is subordinate to the UFD named BDAWES. To attach to this sub-UFD, the user issues the command:

A BOB 1/2

For further details about attachment, refer to the Interactive Users' Guide (MAN 2602).

PROGRAM DEVELOPMENT USING PRIMOS II

The user is free to use PRIMOS II and its supporting software to create, assemble or compile, load, save, and execute user programs. The internal and external PRIMOS II commands are described in the Interactive Users' Guide (MAN 2602). The appropriate manuals provide detailed information on the Editors, FORTRAN, BASIC. The Macro Assembler, Loader, and other programs are described in the Programs Software Development User Guide (MAN 1879).

Refer to the Interactive Users' Guide (MAN 2602) for an example of the terminal printout resulting from the development of a simple FORTRAN program. The user may study the example and use its procedures as a guide during initial program development efforts.

Recovering from Errors

If an equipment failure or program error causes the CPU to leave PRIMOS II control, it is usually possible to return to PRIMOS II by starting the CPU at location '30000, '50000, or '70000, depending on the hardware configuration and available memory. See the Operators Guide for instructions to restart at these locations.

Installing New External Commands

The user can install his own custom utility or device control programs to be invoked by external command to PRIMOS II. One way this is done is by restoring a program from the user's UFD, and then saving it under the command UFD CMDNCØ. Assume, for example, that the user wants to install a cassette recording and playback monitor program to be invoked by the name CASS:

Example:

When PRIMOS II prints the message, OK:, the first command to be entered must be a STARTUP command that assigns the logical disk Ø. PRIMOS II expects to find all its command and utility files on disk Ø. For detailed information on the STARTUP command, refer to the Interactive User's Guide. The usual initial STARTUP is:

OK: STARTUP 5Ø 51

This assigns the removable surface of the moving head disk (the master disk pack) as logical unit Ø, and the fixed surface as logical unit 1.

OK: A USER1

OK: REST CSETV1

OK: PM

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

ØØØ1ØØ Ø111ØØ ØØ1ØØØ ØØØØØØ ØØØØØØ ØØØØØØ ØØØØØØ

OK: A CMDNCØ

OK: SAVE CASS 1ØØ 111ØØ 1ØØØ Ø Ø Ø

OK: A USER1

OK: CASS

GO

(CASS program begins running)

In this example, the user restores file CSETV1 from his own UFD (USER1), and does a PM to determine the RVEC parameters (discussed in the Interactive User's Guide). He then attaches to CMDNCØ and saves the program under the name CASS, with the same parameters as the original. Thereafter, when he uses the name CASS as an external command, PRIMOS II resumes the saved CASS program.

One way to install a new external command is:

OK: A NEWUFD

OK: FUTIL

>FROM USER1

>TO CMDNCØ

>COPY CSETV1 CASS

## NOTE:

This copies file CSETV1 to the UFD named CMDNC0 as a file named CASS.

>QUIT  
OK:

Commands for starting up disks under PRIMOS III:

Another way to install a new external command is:

ADDISK [PHYDEV] [PHYDEV] ...

This command searches for available logical number (up to octal 21) and does the STARTUP.

## BACKUP

General

Each installation can develop its own procedures to save copies of files and disks for backup purposes. The techniques are simple. PRIMOS II disks can be copied to removable disk packs by careful use of the COPY command described in Section 6, or individual files can be saved on paper tape through the Text or Binary Editors.

To copy the fixed disk surface to a removable backup pack (not the master disk): first, do a FIXRAT of the fixed surface and do not proceed until an error-free FIXRAT pass is obtained. Then, do the appropriate STARTUP, attach to some UFD, and use the COPY command.

## CAUTION

Before entering the COPY command, make sure the FROM surface being copied is in WRITE PROTECT mode.

Use of MAGSAV

If magnetic tape devices are present, files are copied to them by the MAGSAV command. This is the most convenient and simplest method of implementing system backup. The tapes produced by MAGSAV can also be read back into the system configuration by use of the MAGRST command (refer to MAN 2602).

## SHUTDOWN

Before terminating an operating session with PRIMOS II by loading another operating system or turning off power, enter the following commands:

OK: FIXRAT (This step is optional)

OK: SHUTDN

The SHUTDN command writes to disk PRIMOS II data that is buffered in memory.

### Changing Disk Packs

To change removable disk packs in the moving-head disk drive, 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.

### Using FIXRAT

The external command FIXRAT loads and starts the PRIMOS maintenance program that checks the file integrity on any disk pack. FIXRAT fully supports nested UFD's and nested segment directories. Section 4 gives further information on the FIXRAT command.

FIXRAT must be run whenever there is reason to expect that the file structure is damaged - for example, if a program being debugged runs wild and writes over part of PRIMOS II. Until the user gains experience with the system, FIXRAT should be run at the close of every operating session.

The suggested procedure is to maintain a PRIMOS II disk pack and to run FIXRAT every morning; if no error occurs, copy the disk pack onto a daily backup disk pack. If any files are truncated or deleted, these may be copied from the daily backup disk pack, provided they were stored previously on the daily backup disk pack.

### 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

## CAUTION

If you are unfamiliar with the system, do not attempt to restart PRIMOS II. Check with someone who knows the system's hardware status, the contents of all disk surfaces, and the correct STARTUP procedure for the particular installation.

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. For example, in a system with a fixed/removable moving-head disk drive, the usual startup is STARTUP 50 51. This establishes the fixed surface as the PRIMOS II command disk.
3. ATTACH to an authorized UFD and resume operation.

Terminal Patch Locations

PRIMOS can be patched to drive terminals of various brands and speeds, as explained in the Terminal User Guide (MAN 1946). The patch locations are shown in the table below.

Some terminals require a few character-times of delay at the end of each line to avoid garbling the next line. Relative-mode PRIMOS II is provided with a patch location to determine the delay it transmits. The location must be set to the number of character-times of delay desired. Rubout characters (octal 377) are transmitted during the delay. Sectoried-mode PRIMOS II does not have such a patch location.

<u>Function</u>	<u>Factory Setting</u>	<u>Location in S&lt;-DOS</u>	<u>Location in R&lt;-DOS</u>
Number delays at line end	0	(none)	restart + 310
Option-A control word	000110	restart + 246	restart + 311
SOC control word one	000027	restart + 247	restart + 312
SOC control word two	074000	restart + 250	restart + 313

Building Relative-Mode PRIMOS II

As PRIMOS II is supplied already built on the master disk, it is unlikely that a customer will need to build it. However, the procedure is as follows:

All source modules and command files for building relative-mode PRIMOS II are in UFD PR2SRC of Volume 2 of the master disk. UFD PR2BIN, also on Volume 2, is supplied initially empty, but receives the binary and run files as they are constructed.



The command files (in order) which build relative-mode PRAIMOS II are:

<u>File</u>	<u>Function</u>
C<-LDUT	Builds a utility for relocating run-file address parameters as required by the bootstrap loader (refer to Appendix A);
C<-FTN	Performs all FORTRAN compilations;
C<-PMA	Performs all PMA assemblies;
C<-LDX	Loads DOSXT, the hybrid-command run-file;
C<-LD32	Loads RDOS32 and UDOS32 (CAUTION: do not run this command file under 32K DOS - it will overwrite the operating system);
C<-LD64	Loads RDOS64 and UDOS64 (CAUTION: do not run this command file under 64K DOS - it will overwrite the operating system);
C<-COPY	Copies the created run-files into the UFDs named DOS and CMDNC0.

### Pitfalls to Avoid

Relative-mode PRIMOS II is larger than sectored-mode PRIMOS II. Some programs that ran under 32K sectored-mode PRIMOS II will not run under 32K relative-mode PRIMOS II. In particular, the commands LOAD and PSD supplied in UFD CMDNC0 of the master disk are loaded too high to operate under 32K relative-mode PRIMOS II, use LOAD20 and PSD20 instead. Of course, there is no problem using LOAD and PSD under 64K PRIMOS II.

Furthermore, upcoming revisions of the master disk may add considerable functionality to relative-mode PRIMOS II, and, consequently, its size is expected to grow larger in the future. Users should avoid configuring their applications programs to be dependent upon a specific size operating system. But in consideration of existing programs already demanding as much memory as possible, a compact sectored-mode PRIMOS II will continue to be supplied.

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, 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.

Floppy Disk Troubleshooting

If floppy disks do not run correctly under relative-mode PRIMOS II, try the following procedure:

1. Bootload PRIMOS II from the floppy disk or any other device.  
Do not issue STARTUP command.
2. From the processor control panel, enter the patch shown below and restart PRIMOS II at X70000 (X=0 for 32K, X=1 for 64K).  
The PRIMOS II in memory now runs floppy disks properly.
3. STARTUP devices.
4. Make permanent patches to files \*DOS32 and \*DOS64 in UFD DOS.
  - 4.1 RESTORE \*DOS32 (or \*DOS64) and PM.
  - 4.2 Use TAP to enter patch (not PSD).
  - 4.3 SAVE with original PM parameters.
5. If your installation uses UDOS32 or UDOS64, the same patch may be necessary in them. The patch is best done under PRIMOS III or PRIMOS IV).

<u>SAVE/REST Loc</u>	<u>Loc</u>	<u>Contents Were</u>	<u>Contents Should Be</u>
007763:	X67763:	junk	005005
4:	4:	junk	022020
5:	5:	junk	010020
6:	6:	junk	000001
7:	7:	junk	005002
007770:	X67770:	junk	103771
1:	1:	junk	162000
2:	2:	junk	000020
013340:	X73340:	junk	X67762
013546:	X73546:	005362	121571

## SECTION 2

## PRIMOS III OPERATION

## PRIMOS III SYSTEM OVERVIEW

PRIMOS III shares the computer resources among a community of up to 31 simultaneous users in a virtual memory environment.

The resources shared are the central processor, high-speed memory, the file system, and the peripheral devices which are also 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. Other 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.

## PRIMOS III VERSIONS

There are three general classes of PRIMOS III: four-user versions, 15-user versions, and 31-user versions. The following paragraphs describe in detail the content of the existing versions of PRIMOS III. Command files that may be invoked to generate the various versions of PRIMOS II are described after the description of their contents.

Standard PRIMOS III

PRIMOS III now consists of three run files in the UFD named PRIMOS, PRIMOSL, and PRIMOSU. PRIMOS is a utility program that preloads the paging device with the file PRIMOSU (the upper part of PRIMOS III) and RESUMES PRIMOSL.

To invoke PRIMOS III, the user must give the command:

PRIMOS (standard default)

This version of PRIMOS supports 15 users on the Prime 300 with either an 8- or 16-line AMLC. This version also supports the SMLC and the Inter Processor Controller (IPC). File sharing between two Prime 300 or 400 computers is supported with the help of the File Access Manager (FAM) (available at extra cost).

As described in Section 1, the storage module with a record size of 1040 words is supported by PRIMOS III.

CAUTION

TAP is no longer loaded at 53000 for patching the PRMOSL or PRMOSU run file; code is not loaded there. To patch PRIMOS, use the command PSD160. PSD160 cannot be given while running under control of 64K PRIMOS II.

Other PRIMOS III Versions

In the UFD named PRM32, three files: PRIMOS, PRMOSL, and PRMOSU comprise the 31-user version of PRIMOS III. The IPC and file sharing are not supported in this version; however, the SMLC is supported. To run this 31-user version of PRIMOS III, ATTACH to PRM32 and issue the command RESUME PRIMOS.

In the UFD named PRM330, three files: PRIMOS, PRMOSL, and PRMOSU comprise the four-user version of PRIMOS III that runs the bit banger with the 330 cycle clock. The OPC and file sharing are not supported by this version. If desired, the user may move this version to CMDNC0.

CAUTION

Be sure to copy all three files to avoid unspecified results.

The source and command files of PRIMOS III are contained in the UFD named PRI300 on Volume 5 of the master disk. PRI300 contains several subdirectories with the following contents:

<u>Subdirectory Name</u>	<u>Content</u>
PRISRC	Sources for PRIMOS
NETSRC	Sources for IPC support in PRIMOS
FUFD	UFD used for FORTRAN object files during system building
PUFD	UFD used for PMA object files during system building

Command Files for System Building

All command and run files are found in PRI300. The command files to build various versions are as follows:

<u>Command File</u>	<u>System</u>
C<-16	Builds 15-user version generating run files PRIMOS, PRIMOSL, PRIMOSU, in UFD PRI300.
C<-32	Builds 31-user version generating run files with the same name as above.
C<-330	Builds four-user version of PRIMOS III with 330 cycle clock generating run files with same names.
C<-300	Builds four-user version of PRIMOS III with 300 cycle clock generating run files with same names.

To build a version of PRIMOS III to support the Gould Printer/Plotter, delete the file VERDIM and rename the file GLDDIM to VERDIM and run the appropriate command file.

#### PRIMOS III SYSTEM INITIALIZATION

Once the system configurator has installed and appropriately modified PRIMOS III in CMDNC0, as explained in the preceding paragraphs, PRIMOS III is started from PRIMOS II by the system operator.

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 1.
2. To ensure that PRIMOS III is available, enter the command:  
STARTUP Command-disk-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. The erase character, double quote (") may not be used to modify input to 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 l0/1 must be given.

Example:

```
CONFIG l0 51 50 l0/1
```

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 l0 51 50
```

Complete information about CONFIG, SETIME, and other operator commands is given in the Interactive Users Guide (MAN 2602).

### Sharing Files

Sharing of files is possible under PRIMOS III and IV. Two or more users may be attached to the same UFD at the same time. Furthermore, two or more users may have the same file open for reading, and thus may be reading the same file at the same time. File interlocks are provided, as under PRIMOS II, to prevent one user from reading the file while another is writing. This interlock may be modified by the PRIMOS III or IV system user.

### File Access Protection

Under PRIMOS III and IV, a user attached with owner rights (hereafter called the owner) has the ability to open his file directories to other users, giving restricted access rights. The owner of a file directory can declare the access rights that nonowner users have over each file. File access protection is not available under PRIMOS II.

The declaration of access can be made on a per-file basis, thus the owner has a degree of flexibility in the manner that file access is specified. Access rights are separated into three categories:

Read Access (includes Execute Access)

Write Access (includes Overwrite Access and Append Access)

Delete/Truncate Access

The access rights to a file are declared and specified through the PASSWD and PROTECT commands. For details and example, refer to the Interactive Users' Guide (MAN2602).

The owner of a UFD can establish two passwords for access to any file in the UFD. An owner password is required by the owner to obtain owner privileges, and a nonowner password (if any) is required to obtain nonowner privileges.

The PROTECT command replaces the existing protection keys on a file. It is used by an owner to specify the access rights to be given other users of a specific file.

### Bypassing Bad Memory

PRIMOS III includes features to detect and bypass bad memory chips. These features are categorized as follows:

- On a START, 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 call that contains incorrect parity.

### Memory Parity

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 PAGMAPS, WARM START
```

The operator must stop the machine, fix MMAP and PAGMAPS if possible and do another warm start.

### USER LOGIN/LOGOUT

When a user of the system LOGS IN or LOGS OUT, any existing program LOGIN in CMDNC0 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 performs a CALL BREAK\$ (.FALSE.) before exiting.

The command line that called LOGIN is available via the library subroutine 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.

### 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 TIMOUT.



The parameter N is set by changing LOUQOM, which is the first variable in the common area FIGCOM in subroutine TMAIN. For example, to change the inactivity Logout time to 10 minutes, proceed as follows:

```

OK, ATTACH CMDNC0
OK, RESTOR PRIMOS
OK, START 53000 (This starts TAP)
GO
$A 13640
13640 1750 12
      --
13641      1 /

CNTRL P
QUIT, SAVE PRIMOS  LOW ADDR, HI ADDR, STRT ADDR
OK,

```

#### NOTE

The above locations may be subject to change without notice.

#### SYSTEM CONFIGURATION

Upon obtaining a master disk from Prime, the system configurator (user) must install a PRIMOS III for his installation into CMDNC0. PRIMOS III supports 1 to 31 users 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 31-user version of PRIMOS III previously described. It is suggested that 8K words of memory be available per user. Therefore, a 31-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 users. For example, an installation to run 20 users would require proportionally less memory.

#### AMLC Requirements

Two 16-line AMLC's are required to handle user terminals for 31 users. For 23 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. Alternatively, two 3.0-million word partitions of a 30-million word disk may be used. Only one head is needed for the storage module. The CONFIG command lines format to accomplish this must be given as follows:

CONFIG Nuser Pagedev1 Comdev [Availm] [Pagdev2] [Namlc] [Nphan]

For details, refer to the description of the CONFIG command in Section 7. If Pagedev1 is not large enough, a failure to give the parameter Pagdev2 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

For recent versions of PRIMOS III, 32 additional pages are required on the paging disk. Some users may require a larger paging space to run the same number of users. For all disks except the storage module and fixed-head disk, this translates into 64 additional records required for paging. When used for paging, the storage module requires one record for each page instead of two records per page.

#### Modification of Page Maps

A system configurator 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 and others on a moving head disk.

#### AMLC Driver Configuration

Many users modify the line speed or group tables for AMLC lines for their installations. The position of these tables may change with different software revisions and PRIMOS III classes; however, the AMLC driver is loaded at 55000 octal for both the 15-user and 31-user versions of PRIMOS III. The routines CRDIN and CRDOUT are included in the PRIMOS III load package so that serial card reader users may modify the AMLC tables to run a serial card reader.

The AMLC driver is self-configuring. With the 15-user version of PRIMOS III, the driver modifies itself to run either an 8-line or 16-line AMLC with a device address of either 53 octal or 54 octal. Users with an 8-line AMLC do not have to modify the tables to run their system configuration. On a 31-user PRIMOS III configuration, the AMLC driver modifies itself to run any combination of 8-and 16-line AMLC boards, so long as one AMLC has a device address of 54 and the other AMLC has a device address of 53. The 31-user PRIMOS III version of the AMLC driver also runs with a single AMLC board.

A system that consists of two 8-line AMLC boards has user terminals 2 through 9 connected to the AMLC with a device address of 54 and user terminals 18 through 25 connected to the other AMLC.

Users that modify the AMLC driver group tables must set the tables up for their particular AMLC configuration and must not depend on the auto-configuration feature of the PRIMOS III systems AMLC driver.

Through the use of the AMLC command, the configuration word for any user terminal AMLC line may be changed. This command is issued at the PRIMOS III systems terminal. For further information, refer to the description of the AMLC command in this section.

TAP may be used to modify the AMLC table or make other changes to PRIMOS III. The user must exercise caution when patching PRIMOS III and remember that PRIMOS III is loaded in 64R mode.

All AMLC lines are set to run at 1200 baud. System configurators who wish to set lines for other speeds must refer to the discussion of "Changing Configuration Table".

#### A Forced Logout

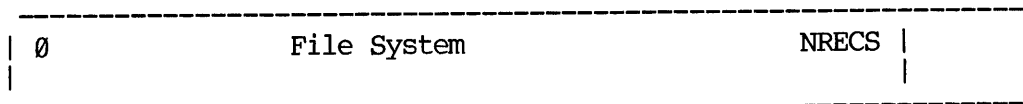
An option exists to cause a forced logout if a user hangs up the phone on an AMLC line connected to a dataset. This option increases security; as another person cannot dial the same number and upon connection be logged into PRIMOS III, as happens automatically without a forced logout. This option is enabled by setting FIGCOM+3 to a 1. FIGCOM stands for configuration common and is part of both TMAIN and the PRIMOS III load map. The variable in TMAIN corresponding to FIGCOM+3 is DLOGOT.

#### Running Split Disk

The system configurator 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:

Ø	NRECS	PAGES
File System	Paging	

The program MAKE must be run to configure a split disk. For further information, refer to Section 6.

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.

Paging space on the fixed head disk is 128 records per user (64K address space), plus 176 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:

$$\text{PAGES} = 256 \times n + 352 \text{ records on MHD}$$

$$\text{PAGES} = 128 \times n + 176 \text{ records on FHD}$$

Example:

For a six-user system on a moving head disk:  $\text{PAGES} = 256 \times 6 + 352 = 1888$  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:

```
User:      MAKE
Response:  PHYSICAL DEVICE =
User:      52
Response:  1.5M WORD PACK?:
User:      NO
Response:  SPLIT DISK:
User:      YES
Response:  PAGING RECORDS (DECIMAL):
```

User: 1888

Response: DISK        FILE-RECORDS    PAGE RECORDS (DECIMAL)  
          000031     1360                1888

PARAMETERS OK?

User: YES

Response: BAD SPOTS ON DISK:

User: NO

Response: VIRGIN DISK?

User: YES

Response: VERIFY DISK

User: YES

Rresponse: BEGINNING WRITE  
Response: BEGINNING VERIFY

Response: DISK CREATED (after a while)

The system configurator must then use the UFDCPY feature of FUTIL to copy the UFD's: 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.

Many systems are shipped with only one disk - a 30M word disk. It is suggested that the user not run the 30M word disk as a split disk. When a 30NW disk is shipped, it is partitioned into four subdisks: a 50, 10050, 20050, and 33450. The system configurator should make a 3.0M word partition for paging if PRIMOS III is for 23 or fewer users, or two 3.0M word partitions if PRIMOS III is for 24 or more users. The system configurator does this using the command MAKE, to make physical device 31050, for example. The rest of the disk must be then be made as 1 or more partitions. If it is made as one partition, the disk number would be 52450.

Tables 2-1 and 2-2 provide a guide to disk addresses for system configurators. They show disk space required for the supervisor and up to 31 users on both 32K and 64K configurations.

Changing Configuration Table

The baud rate for the AMLC configuration can be changed easily by modifying the line configuration table for lines 0-6. The line configuration table is identified by a comment in the PRIMOS III source program. The values that may be specified are:

110 baud:	XX0033
300 baud:	XX0213
1200 baud:	XX0313

Table 2-1. Disk Space Required for 32K Per User

<u>User</u>	<u>MHD Disk Addresses</u>	<u>FHD Disk Addresses</u>
Supervisor	0-351	0-175
1	352-479	176-239
2	480-607	240-303
3	608-735	304-367
4	736-863	368-431
5	814-991 limit 256K disk=512	432-495
6	992-1119	496-559
7	1120-1247	560-623
8	1248-1375	624-687
9	1376-1503	688-751
10	1504-1631	752-815
11	1632-1759	816-879
12	1760-1887	880-943
13	1888-2015 limit 512K disk=1024	944-1007
14	2016-2143	1008-1072
15	2144-2271	
16	2272-2399	
17	2400-2527	
18	2528-2655	
19	2656-2783	
20	2784-2911	
21	2912-3039	
22 limit 1.5M platter = 3248	3040-3167	
23	3168-3295	
24	3296-3423	
25	3424-3551	
26	3552-3779	
27	3780-3907	
28	3908-4035	
29	4036-4163	
30	4164-4291	
31	4292-4419	
32	4420-4547	

Table 2-2. Disk Space Required for 64K Per User

User	MHD Disk Addresses	FHD Disk Addresses
Supervisor	0-351	0-175
1	352-607	176-303
2	608-863	limit 256K 304-431
3	864-1119	432-559
4	1120-1375	560-687
5	1376-1631	limit 512K 688-815
6	1632-1887	816-1044
7	1888-2143	
8	2144-2399	
9	2340-2655	
10	2656-2911	
11 limit 1.5M platter = 3245	2912-3167	
12	3168-3423	
13	3424-3779	
14	3780-4035	
15	4036-4291	
16	4292-4547	
17	4548-4803	
18	4804-5059	
19	5060-5315	
20	5316-5571	
21	5572-5827	
22	5828-6083	
23 limit 3.0M platter = 4956	6084-6339	
24	0-255 pagedev 2	
25	256-511 "	
26	512-767 "	
27	768-1023 "	
28	1024-1279 "	
29	1280-1535 "	
30	1536-1791 "	
31	1792-2047 "	



Detection of Bad Memory

If PRIMOS III fails to type its introductory message and halts, it usually means that it has detected bad memory. Refer to Section 1 and 8 for a description of how to recover from the situation. The following example shows a typical operator procedure to bring up PRIMOS III for use. After the example, the operator terminal commands issued at the start and during the running of PRIMOS III are explained in detail. An example startup of PRIMOS III follows:

```
OK: STARTUP 50
OK: PRIMOS
GO
```

```
PRIMOS REV 11.15
```

```
64.0K MEMORY IN USE
```

```
PLEASE ENTER CONFIG
```

```
OK, CONFIG 12 51 50
PLEASE ENTER DATE
OK, SE -0426 -0905
```

```
...
```

WARM RESTART FOR PRIMOS III

## CAUTION

Do not warm restart blindly.

If PRIMOS III halts because of an error or because of a machine-check, it is usually possible to institute a warm restart as follows:

1. At the control panel, set the rotary switch to STOP/STEP. Press MASTER CLEAR.
2. START at 1001 for machine check or 1002 for no machine check. Store the starting address (e.g., 1001) in Location 7 and set rotary switch to RUN.
3. Set START.
4. At each terminal connected to PRIMOS III at the time of the halt, type:

S

followed by a CARRIAGE RETURN.

## CAUTION

Warm start now runs the memory parity finder before restarting PRIMOS and no longer types QUIT, on the system terminal as a response. There is no response to a warm start at the system terminal. Furthermore, the warm start may run for 20 seconds or more before the QUIT, message appears on user terminals. Do not assume the warm start has failed without waiting 20 seconds and checking to see if QUIT, has failed to appear on a user terminal.

## SECTION 3

## PRIMOS IV OPERATION

## OVERVIEW OF PRIMOS IV

Introduction

PRIMOS IV consists of three run files in the UFD named CMDNC0 on Volume 6 of the master disk: PRIMOS, PRMOSL, and PRMOSU.

## WARNING

The three files with the same names in the UFD named CMDNC0 on Volume 1 are PRIMOS III for the Prime 300.

To use PRIMOS, PRMOSL, and PRMOSU, the files must be copied to the UFD named CMDNC0 of the user's command disk.

PRIMOS is a utility program that preloads the paging device, RESTORES the file PRMOSU (the upper part of PRIMOS IV), and RESUMES PRMOSL.

Configuration Features

This version of PRIMOS IV supports up to 31 users on the Prime 400 computer system with one or two 8- or 16-line AMLC controllers. This version also supports the SMLC and the Inter Processor Controller (IPC). File sharing between two Prime 300 or 400 computers is supported with the help of the File Access Manager (available at extra cost). The storage module is supported, with the large record size (1024 words).

STARTUP OF PRIMOS IV PRIMOS IV is started up from PRIMOS II. To invoke PRIMOS IV, give the command:

PRMOS4

Typical response is:

```
PRIMOS 4 REV 11.32
XX K MEMORY IN USE
PLEASE ENTER CONFIG AND DATE
```

The operator may enter the CONFIG command DATE exactly as with PRIMOS III. The remaining details for entering CONFIG, etc., are the same as those described under Bringing PRIMOS III Up in Section 2.

PRIMOS IV Versions

There is one version of PRIMOS IV. The sources and command files of PRIMOS IV are in the UFD named PRI400 on Volume 6 of the master disk. PRI400 contains subdirectories with the same names as those described in Section 2. All command and run files are available in PRI400. The command file to build PRIMOS IV is C<-ALL.

Description of 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 Prime 400 Preliminary Systems Reference Manual. 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:

State    Meaning of

-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:

$$\text{page map number} = (\text{MMAP entry} - 1400) / 200$$

where all numbers are octal and 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.

#### Description of 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. These tests are performed in the COLD START initialization routine (INIT).

Upon detecting a bad memory cell, INIT halts. The following information is then available:

register 3 = page-number

register 0 =  $172000 + \text{page displacement}$

register 1 = bad bits if 0, one of the two parity bits is bad

If INIT is continued (by placing function switch in RUN and pushing START), the appropriate physical page is marked unavailable (see MMAP description) and the scan continues. If the operator wishes to make this page permanently unavailable, he must follow the procedure described below. If no bad memory is found, PRIMOS IV prints the actual total memory in use as part of its COLD START message. If a bad page is found and its number is below 60 octal, the bad page is within the supervisor and PRIMOS IV cannot be run until the memory is fixed.

#### 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:

MMAP + 177 = 22000 + 177 = 22177

From PRIMOS II: (operator input underlined>)

OK: ACMDNC0

OK: REST PRMOS4

OK: PM

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

60 64777 1001 0 0 0 2000

OK: S 56000

\$ U 22177 777777

22177 000000 177777

\$ R 7000

OK: SAVE PRMOS4 60 65777 1001 0 0 0 4000

OK:

System Crash (Memory Parity Halt at 31565)

1. Do not MASTER CLEAR.
2. Turn on run knob to fetch Y.
3. Read register 34, which is DSWRMA, the 32-bit address of the location that caused the parity halt. This address is virtual if caused by the CPU or absolute if caused by a DMX I/O transfer. To read absolute register 34, enter 34 in the address, set sense switch 1 (SS1) up and SS2 up. Place SS4 up and the address/data switch to data. Press the START switch, then record the data in the lights as DSWRMA-H. Reset SS4, press the START and record the lights as DSWRMA-L. Next, read the high side of register 35 by a similar procedure. Record this as DSWSTAT. Verify that bit 15 of DSWSTAT is on, indicating a parity error occurred (bit 15 for 4-row PROM, bit 3 for 3-row PROM). If bit 10 is on, the parity error occurred during Direct Memory (DMx) I/O and location DSWRMA is an absolute address. Extract the page number from the address. It consists of bits 15 and 16 of DSWRMA-H concatenated with bits 1-6 of DSWRMA-L. Then refer to the section entitled 'Removing a Bad Page' before a WARM START. If DMx was not in progress, the address is virtual. Determine the physical address corresponding to that virtual address as described in step 4.
4. Bits 5 and 6 of DSWRMA-H should be 1 and 0 respectively, indicating a reference to the user's Segment Table 2. If these bits are not 1, 0, go through the rest of the procedure to find the bad physical page, but PRIMOS cannot be warm-started (because there was bad parity in the supervisor).
5. Depending on bits 5 and 6 of DSWRMA-H, determine to access one of the descriptor table address registers (DTAR's) as follows:

<u>Value of</u> <u>Bits 5-6</u>	<u>Absolute Register</u>	
00	DTAR0	23
01	DTAR1	22
10	DTAR2	21
11	DTAR3	20

Access the appropriate DTAR by placing the corresponding register number in the address, setting SS1 up, SS2 and SS4 down. Fetch the contents and read out the data. Double the number read to get a pointer to the segment table. If DTAR2 is examined, this pointer must point in the range of the STARL table in TMAIN.

6. Double the segment number in bits 7-16 of DSWRMA-H and add it to the address of the segment table determined in step 5. Fetch the two-word segment table entry at that address and write down the entry.
7. The first word of the entry contains a pointer to the page table for the segment. The second word contains the high-order bits of the page table address. Add the page number contained in bits 1-6 of DSWRMA-L to the page table address to get a 22-bit physical pointer to the page table entry. Access the entry and write it down. Make sure the access is done with SS4 up, indicating an absolute access.
8. Bits 5-16 of the page map entry indicates the physical page number where the memory parity occurred.

#### Removing a Bad Page Before a Warm Start

1. Add the bad page number to 22000 to get a computed entry. Access the computed entry and write it down. Replace the entry with a -1, which tells PRIMOS the page is not available.
2. The computed entry points to a page map entry in a page map. Determine which page map the entry is in, as explained in the previous paragraphs about PRIMOS Memory Map.
3. If the computed entry address is below 2000 octal, the bad page belongs to the supervisor and PRIMOS cannot be warm-started.
4. Otherwise, the computed entry must contain the virtual address in segment 0 of the page map entry. Replace that entry with a 0.
5. Warm Start by MASTER CLEARing and pushing the START switch twice. Then, the system should come up and start running.





## SECTION 4

## FIXRAT

## INTRODUCTION

The external command FIXRAT loads and restarts a maintenance program that checks the PRIMOS file integrity on any disk pack. FIXRAT fully supports nested UFD's and nested segment directories, and may be run from a command file.

## FIXRAT DESCRIPTION

The external command FIXRAT runs under PRIMOS; it loads and starts a maintenance program that checks the file integrity. Before reading this document, the user should read a description of the file structure found in the File System Users Guide (MAN 2604). Existing PRIMOS users should also read this guide for a description of segment directories, nested directories, and FIXRAT printout options.

FIXRAT reads every record in every file, UFD, and segment directory, and checks that information in each record header is consistent with both 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 available 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.

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.

FIXRAT must be run whenever there is reason to expect that the file structure is damaged; for example, if a program being debugged runs wild and writes over part of PRIMOS II. Until the user gains experience with the system, FIXRAT should be at the close of every operating session. Never attempt to run FIXRAT after a COPY has aborted.

The suggested procedure to maintain a disk pack is to run FIXRAT every morning and, if no errors occur, to 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. Then this new disk pack is copied to an updated daily loading disk. The owners of the bad files must be notified that those files have been

copied from the backup and any modifications to those files may have been lost.

### FIXRAT and Disks

FIXRAT will maintain files on a Y disk, including the storage module, and handles both 448-word and 1040-word records. Under \*DOS16, only FIXRAT is limited to 3-million word disks; a command file cannot be used in this case to run FIXRAT.

### Running FIXRAT

The command is:

```
FIXRAT [OPTIONS]
```

If the word OPTIONS is included, FIXRAT requests printout options. Otherwise, FIXRAT prints the name and number of records used (in decimal) in the MFD and in each directory in the MFD. When entered, FIXRAT asks the question:

```
FIX DISK ?
```

If the answer is YES .CR., FIXRAT truncates or deletes defective files and generates a corrected DSKRAT file, in addition to checking the file structure and repairing all file structure errors. FIXRAT then asks the question:

```
PHYSICAL DISK DRIVE =
```

The user types in octal the physical disk drive on which FIXRAT is to be run, followed by .CR.; FIXRAT then prints the disk pack identification (which is the name of the DSKRAT) and begins processing the file structures. The DSKRAT is always the first file in the disk pack ID.

Figure 4-1 is 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 subdirectories that have their root in that directory. In Figure 4-1, 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 4-1, 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 by the snaked line in Figure 4-2, generating typeout at the various points below.

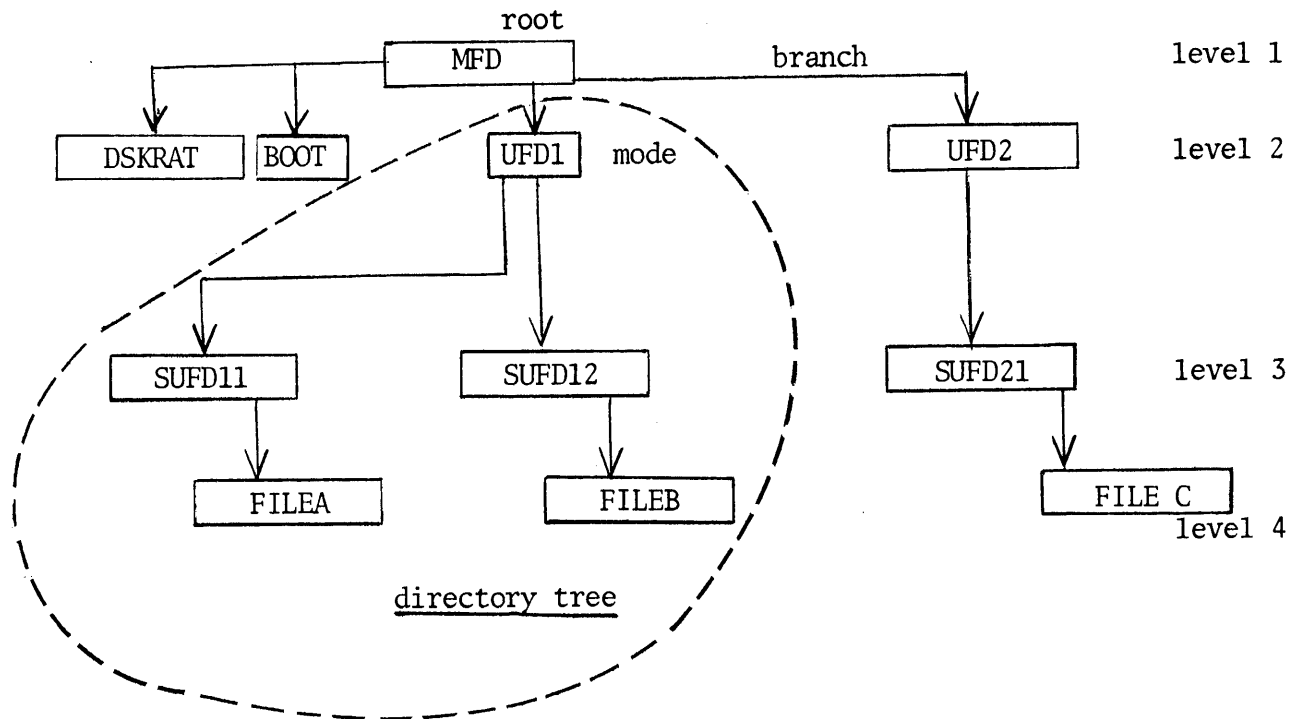


Figure 4-1. Sample File Structure

The terminal output appears as follows for the above file structure:

```

BEGIN MFD
  BEGIN UFD
    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 = 35
RECORDS LEFT = 6223
DSKRAT OK
OK,
  
```

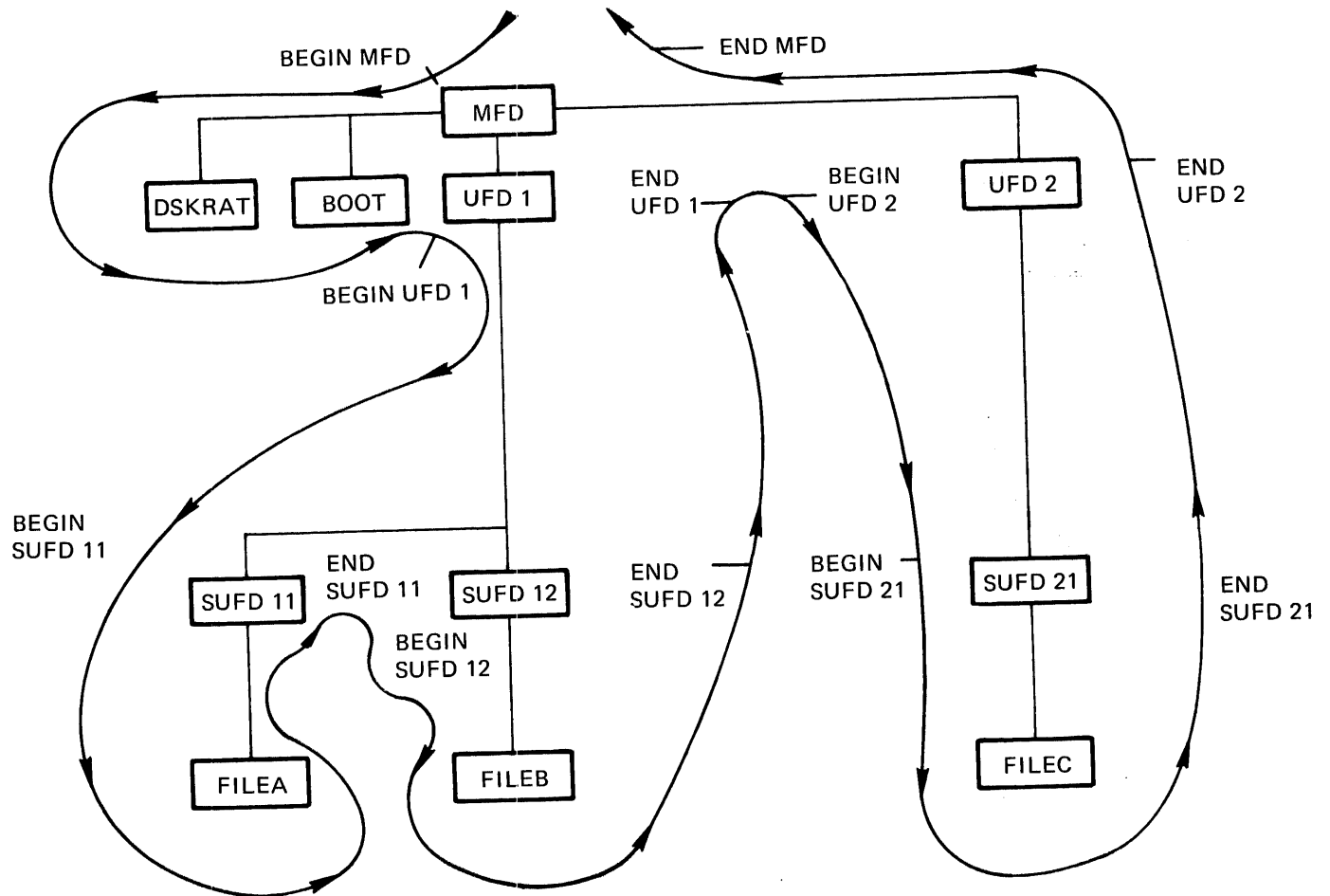


Figure 4-2. Typical FIXRAT Traverse of File Structure

FIXRAT prints the word BEGIN followed by the directory name when beginning processing of a directory tree. On leaving a directory tree, FIXRAT prints END Directoryname followed by the number of physical records (in decimal) used by all files and directories in the directory tree. In the above example, this is the number following MFD (35).

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 4-1 generates the following default output:

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

If the command FIXRAT OPTIONS is given, FIXRAT asks the question FIX DISK ? and PHYSICAL DISK DRIVE =, as before, and also asks:

TYPE DIRECTORIES TO LEVEL =

The user must type 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)
etc.	etc.

FIXRAT will then ask:

TYPE FILE NAMES ?

If the answer is YES followed by .CR., FIXRAT prints all filenames in all directories, indented appropriately. This option is useful for listing the contents of a disk. Note that, unless the user requests suppression of directory name output by answering the TYPE DIRECTORIES to LEVEL = question with the parameter one, directories are printed three times; twice as directories and once as files.

FIXRAT will then ask:

TYPE FILE CHAINS ?

If the answer is YES, followed by .CR., 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.

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    CMDNCO    000021
BEGIN  DOS
END    DOS       000011
END    MFD       000035
RECORDS  USED  =  000035
RECORDS  LEFT  =  006223
DSKRAT  OK
OK:

```

The first line prints the disk pack identification as the name of the disk. Generally, this name is DSKRAT.

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, CMDNCØ 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 a number that is 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.

Following the printing of 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, .CR. 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:

```
reason for error
```

```
FILE - filename BAD RECORD = octal record address.
```

```
DIRECTORY PATH = list of directories
```

```
FILE DELETED, FILE TRUNCATED or blank
```

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 4-2 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 how it disposed 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.

#### Segment Directories

A segment directory may contain references to files, other segment directories and User File Directories (UFD). 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. Recall that each entry in a UFD consists of a one-word disk address that is the beginning record address of the file, followed by a six-letter name and two spare words. Refer to the File System User Guide (MAN 2604). In a segment directory, FIXRAT prints the absolute position of the file in the segment directory as a decimal number-pair (record number, word number). For example, the first entry in a segment directory is printed as (0, 0), the second entry as (0, 1), the 440th as (0, 440), and the 441st entry as (1, 0). Note that, as with user file directories, identically named files in different segment directories represent unique files.

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 zero. On the next pass, FIXRAT skips the zero entry. The convention, then, is that a zero in a segment directory represents a null file.

#### Disks With Badspots

FIXRAT handles disks with badspots (Refer to MAKE, Section 5). 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 need perform no badspot handling.

The format of the file BADSPT in the MFD is described in Section 6. If the format of the file BADSPT is incorrect, FIXRAT prints the message:



BAD BADSPT FILE, IGNORED

Otherwise, FIXRAT types:

BADSPT FILE HAS ENTRIES = No.

where No. is the 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 = No.

## PITFALLS

### 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, RESTOR from a good MFD and SAVE the BOOT into the MFD before doing anything else.

### Directory Nesting Limit

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

### Writing Into Directories

Because directories may be nested, the possibility of accidentally writing bad data into directories is increased. A bad file structure will be detected by FIXRAT. To minimize this possibility, it is suggested that users preface all except those in the MFD by U\_ and all segment directories by S\_.

### Deleting Directories

Delete all files contained in the directory before deleting a UFD or segment directory; otherwise, the records used by the files in the directory will not be returned to the DSKRAT. When the next FIXRAT is run, the message DSKRAT, FILE DIRECTORIES MISMATCH is given, and the records of files not deleted explicitly will not be recovered for use until the next FIXRAT.

## 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 BAD RECORD = octal number

DIRECTORY PATH = list of nested directories

Description of Messages

<u>Message</u>	<u>Meaning</u>
DSKRAT BAD	<p>This message is obtained if the DSKRAT file contains any bad record pointers, the DSKRAT data header word is not 5, or the words in the DSKRAT file does not match <math>(NRECS+15)/16-+5</math>. 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:</p> <pre> INPUT DECIMAL RECORD SIZE INPUT DECIMAL FILE SPACE RECORD COUNT = INPUT DECIMAL CYLINDERS = INPUT DECIMAL HEADS = </pre> <p>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 Section 2 for tables of disks and values.</p>
BAD DISK ADDRESS	<p>A pointer to a disk record is out range. Acceptable range is between 1 and <math>NRECS - 1</math>, 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</p>

the MFD.

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.

BRA POINTER MISMATCH

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

FATHER POINTER MISMATCH

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

BACK POINTER MISMATCH

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 point is not 0. (For the storage module, the data word count is word 4 of the record.)

BAD WORD COUNT

The data word count of a record is not between 0 and 440. A word count of 0 indicates an empty record.

NOTE:

For the storage module, the data word count is word 5 of the record.

BAD FILE TYPE

The data word count for the storage module, word 6, of the first record of a file is not between 0 and 4.

TWO 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.

## 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.

UFD LONGER THAN  
RECORD

A UFD is longer than one record. PRIMOS expects all UFD's to be only one record long.

## BAD UFD HEADER

When the storage module is present, data word 1 of a UFD file does not contain 8 (decimal), the first word of a UFD header.

DIRECTORIES NESTED  
TOO DEEP

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.

## BAD STRUCTURE MESSAGES

FILE = MFD           BAD RECORD = 7  
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 same as suggested in the previous paragraph.

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.

#### FIXRAT and 30-Million Word Disk

FIXRAT supports the 30-million word disk (storage module). If the 30-million word disk is treated as a single disk device (no partitioning), the disk numbers 0, 1, 2, 3, attached to the controller are 5252, 5253, 5254, and 5256, respectively. 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

## 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 the MAKE command. 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 by MAKE. When FIXRAT is rerun as a result of the unrecovered read error recovery procedure described here, multiple errors of the form:

## TWO 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

The following is an example of FIXRAT run under PRIMOS II with the optional argument, OPTIONS:

```
PRIMOS II REV 11.0 07/15/76 (AT 070000)
```

```
OK:  STARTUP 30
OK:  FIXRAT OPTIONS
GO
REV.11.00
FIX DISK?  YES
PHYSICAL DISK = 30
TYPE DIRECTORIES TO LEVEL = 3
TYPE FILES NAMES?  YES
TYPE FILE CHAINS?  YES
```

```
DISK PACK ID IS MD11V1
BEGIN MFD
000001
BEGIN MFD
  BEGOM ,D11?BEGIN MD11V1
000002
  BEGIN BOOT
000000
  BEGIN CMDNC0
000003
  BEGIN CMDNC0
  BEGIN MCG
000006
000011
000014
```

BEGIN NUMBER  
,BR?,BER?.BR 000017  
000005  
000010  
000013  
000016  
000007  
000012  
BEGIN BASINP  
000015  
000020  
BEGIN SIZE  
000023  
000026  
000031  
BEGIN FILVER  
000034  
PHYSICAL DEV=30

PRIMOS II REV 11.0 07/15/76 (AT 070000)

OK: STARTUP 30  
OK: FIXRAT  
GO  
REV. 11.00  
FIX DISK? 30  
PHYSICAL DEV=30

PRIMOS II REV 11.0 07/15/76 (AT 070000)

OK: STARTUP 30  
OK: FIXRAT OPTIONS  
GO  
REV. 11.00  
FIX DISK? YES  
PHYSICAL DISK = 30





## SECTION 5

## MAKE

## OVERVIEW

MAKE creates a disk for any disk or storage module supported by PRIMOS. MAKE may run from a command file under any version of PRIMOS. Under PRIMOS III or IV, the disk to be created by the MAKE command must be ASSIGNED before MAKE is invoked. MAKE must be used to create partitions and disks, including paging and user disks. MAKE creates a PRIMOS disk that has the following:

DSKRAT  
MFD  
BOOT  
DOS  
CMDNC0

The MAKE program writes the bootstrap (BOOT) into Record 0 of the disk. MAKE places the proper length Record 0 BOOT on the disk being formatted by MAKE.

## RUNNING MAKE

To run MAKE, type the command:

MAKE

The response is:

PHYSICAL DISK

The user must then type the number of the physical disk to be created. All user-typed responses must be followed by a carriage return. This disk must not be connected to PRIMOS by the STARTUP command. Possible disk numbers are listed in Section 1; a summary of possible disk numbers is given in the description of the COPY command.

It is recommended that when running MAKE under PRIMOS II, all running disks be write-protected except the disk to be created by MAKE. Under PRIMOS III and IV, it is recommended that only the disk to be created by MAKE be ASSIGNED to the terminal.

MAKE computes the number of records on the disk pack from the disk number. In case of ambiguity, MAKE asks resolving questions such as:

## 1.5M WORD PACK?

MAKE then asks:

## SPLIT DISK?

The user responds by typing YES followed by carriage return at the terminal, if part of the disk is to be used for paging (for an example, refer to Section 2). If 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 or IV. Refer to Section 2 for details of this calculation. MAKE then responds by printing the disk number, file records, and paging records at the terminal. Then, MAKE asks the user if the parameters are OK. The following table of file records is a guide to help the user verify that the parameter input was correct.

<u>Disk</u>	<u>Decimal Records</u>
Diskette	460
1.5 million word disk	3248
3.0 million word pack	6496
30 million word disk	64960
128 thousand word fixed head disk (32 track)	256
256 thousand word fixed head disk (64 track)	524
512 thousand word fixed head disk (128 track)	1024
1025 thousand word fixed head disk (256 track)	4096

If the 30-million word disk is partitioned, other RECORDS parameters may be specified as shown in Table 5-1.

Table 5-1. RECORDS Parameters for 30-Million Word Disk

Partition	Device Address 23 Disk Number	Device Address 21 Disk Number.	RECORDS
2 head (default)		XX025X	6496
2 head (explicit)		XX065X	6496
4 head	XX125X	XX105X	12992
6 head	XX165X	XX145X	19488
8 head	XX225X	XX205X	25984
10 head	XX265X	XX245X	32480
12 head	XX325X	XX305X	38976
14 head	XX365X	XX345X	45472
16 head	XX425X	XX405X	51968
18 head	XX465X	XX445X	58464
20 head	XX525X	XX505X	64960

In Table 5-1, the X's represent octal digits that must be set appropriately. The leftmost X's specify head offset, and rightmost X specifies one of four possible drives connected to the controller.

MAKE also handles the storage module disk, using disk numbers as described in Section 1.

For a 128 thousand word fixed-head disk, a diskette, or a 1.5 million word disk pack, the user can type CARRIAGE RETURN and MAKE defaults to the correct number of records for that disk. MAKE echoes the user input as follows:

DEVICE NUM	RECORD COUNT	where <number> is
<number>	<number>	one of the above
		octal numbers

OK?

If the number is correct, type YES in response to the OK? query followed by CARRIAGE RETURN. If not, type NO followed by CARRIAGE RETURN, and MAKE requests the input again. After the number of records are specified, MAKE then asks the question:

BADSPOTS ON DISK?

Badspots are part of a disk pack that cannot hold data. On 30-million word 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 of the first badspot. MAKE then prints:

HEAD=

The user responds by typing the head of the first badspot. This dialogue between MAKE and the user continues. Thus, the user may record the track and head of all badspots on a given disk. The TRACK, HEAD pair 0, 9 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 BADSPOT 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.

If there are no badspots on the disk, the user responds by typing NO when the query BADSPOTS ON DISK? is printed by MAKE, and no BADSPT file is created. MAKE then asks the question:

#### VIRGIN DISK?

If the user answers YES, followed by CARRIAGE RETURN, MAKE writes the first records (n=RECORD-COUNT) of the disk. The first word of each record is set to the record address; record size is 448 words. This action also writes a valid hardware checksum for each record. If the user answers NO followed by CARRIAGE RETURN, MAKE does not initialize the records. 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.

If the disk is a storage module, upon a request to MAKE a disk with the VIRGIN DISK question answered YES, MAKE formats the disk or partition; and the MAKE writes all records on the disk or partition to generate correct checksums.

If the user desires to make a split (partitioned) 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 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. After the VIRGIN DISK query, MAKE then asks the question:

#### VERIFY DISK?

If the user types YES, MAKE reads every record in the file system part of the disk to verify that each record can be read.

#### PITFALLS

During the reading and writing of all records, MAKE displays the record 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
```

If the write is not successful after ten 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 17776. 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 #

If the BADSPT file exists, MAKE then asks:

ADD TO BADSPT 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 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.

After MAKE is run, the user must use FUTIL to copy \*DOS16, \*DOS24, and \*DOS32 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. The user must also use FUTIL to copy DOSEXT from UFD CMDNC0 on a master disk to UFD CMDNC0 on the newly created disk.

It may be desirable to use part of a disk pack for the file structure and part for the paging device under PRIMOS III and IV. The user must follow the directions given in Section 1.

#### Description of BADSPT

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 TAP. BADSPT is restored into consecutive memory locations starting at location 1000 octal. 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 off 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 EA+1 and EA+2 (EA is the second parameter printed by the PM command); then, give the command:

SAVE BADSPT 1000 EA+2

The BADSPT file may be used in conjunction with disks other than the 30-million word disk, with the exception of fixed head disks and diskettes. MAKE handles badspots on the storage module disk in the same manner as it does on the 30-million word disk.

## CAUTION

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

## BUILDING MAKE

If the system is being bootstrapped, or if the storage module is to be formatted, it may be necessary to build MAKE by running the command file C MAKE before formatting any disks. During the process of building MAKE using the command file C MAKE, the file \*BOOT is created. \*BOOT is an image of the record 0 bootstrap, as it would appear on the storage module. This run file may be used to place the storage module bootstrap (See Appendix B) on old disk volumes other than storage modules without rerunning MAKE. \*BOOT may also be used to change the start-up parameters for the supervisor terminal.

To save \*BOOT as BOOT, perform the following:

RESTORE \*BOOT  
SAVE BOOT (with the following parameters)

<u>Parameter</u>	<u>448-record</u>	<u>1040-word record</u>
SA	3011	3011
EA	3667	3771
P	0	1
A	0	0
Bx	Controller Control Wd	
KEYS	SOC Control WD 1	
	SOC Control WD 2	

The proper SAVE command for all disks except the storage module is:

SAVE BOOT 3011 3667 0 0 110 27 74000

The SAVE command for the storage module is:

SAVE BOOT 3011 3771 1 0 110 27 74000

## \*BOOT

An image of \*BOOT is stored as part of the program MAKE in the COMMON BOOT1. The first nine decimal words of the COMMON correspond to the SAVE parameters given in the previous paragraphs. The serial interface controller control word is then at location: (LOC((BOOT1)+4)). The

procedure for changing MAKE to create bootstraps (See Appendix B) for systems which do not use TTY as the supervisor terminal is:

Build MAKE user command file C MAKE to get the value of LOC (BOOT1) from the load map

RESTORE MAKE

PM

Use PSD or TAP to insert new initial control words

SAVE MAKE (use original parameters)

The serial interface controller control words (Port 1) are:

<u>Speed (Baud)</u>	<u>Serial Interface Control Words</u>	<u>SOC Port Sel and Speed</u>	<u>SOC Character Definition</u>
110	110	27	74000
300	1010	76	34000
1200	2010	373	34000
9600	3410	3735	34000





## SECTION 6

## COPY

## OVERVIEW

COPY is an external command that copies and verifies a disk. COPY copies any disk to any other disk under PRIMOS. Under PRIMOS III or IV, both disks must be ASSIGNED before invoking COPY.

## 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.

## DISK CONFIGURATION

When asked, the user must specify the disk to be copied from (FROM), and the disk to be copied to (TO). The user is also asked whether or not the FROM and TO disks are 1.5M word disks. The size of the FROM disk and the TO disk must be equal; the user must be sure to specify disks of the same size when replying to the FROM and TO queries. The parameters specified are terminated by the CARRIAGE RETURN character. The FROM and TO parameters are physical device numbers; disk device numbers are listed in Section 1. Possible disk numbers are summarized as follows:

Physical Disk No.	Associated Devices
0-7	Type 4000 controller moving head disks; 8 sectors/track
10	Type 4000 controller fixed head disks; 8 sectors/track
20-27	Type 4300 controller diskettes; 4 sectors/track
30-37	Type 4002 controller moving head disks; 8 sectors/track
40	Type 4002 controller fixed head disks; 64 sectors/track
50-57	Type 4002 controller moving head disks; 32 sectors/track
5050-5056	Type 4001 controller moving head disks; 32 sectors/track
	(even numbers only) controller device address=21
5250-5256	Type 4001 controller moving head disks; 32 sectors/track
	(even numbers only) controller device address=23

## NOTE:

The above information is also required to MAKE and  
FIXRAT.

Controllers are normally delivered with address=23.

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.

If the 30-million word disk is partitioned, the user may COPY an individual partition of the disk.

Table 6-1. Partitioned Disk Address Specification

Partition	Device Address 23 Disk No.	Device Address 21 Disk No.
2 head (default)	XX025X	XX005X
2 head (explicit)	XX065X	XX045X
4 head	XX125X	XX105X
6 head	XX165X	XX145X
8 head	XX225X	XX205X
10 head	XX265X	XX245X
12 head	XX325X	XX305X
14 head	XX365X	XX345X
16 head	XX425X	XX405X
18 head	XX465X	XX445X
20 head	XX525X	XX505X

In Table 6-1, the X's represent octal digits that must be set appropriately. The leftmost X's indicate head offset; and the rightmost X indicates one of the four possible drives connected to the controller. Refer to Section 1 for a complete explanation of the partitioned disks.

Example:

100252

means a 3M word size partition with a head offset of 10 on the disk that is connected to the second drive (drive 2) connected to the controller.

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 or IV, 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.

## COPY 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. Bit 1 may be ON as the high-order address bit of the storage module record number, during the copy phase. During this process, COPY displays the disk record 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:

```
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:

```
ERROR READING 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.

COPY Success or Abort

If the COPY is successful, the message

DONE

is printed at the terminal (only if both the copy and verify were successful).

Disk with Badspots

COPY can copy FROM a disk that has badspots. COPY looks for file BADSPT in the MFD (refer to the MAKE command description for details). 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.

COPY CHECKSUM

When copying a disk pack written using a 4002 controller or a 4000 controller (or vice versa), the two controllers 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.

## NOTE:

A FIXRAT of a disk copied from a disk with bad spots may cause the error message:

DSKRAT, FILE DIRECTORIES MISMATCH

This message is a result of the truncation of the BADSPT file, and does not indicate that the COPY was inaccurate.

If the BADSPT file exists on the FROM disk, but has a bad format, the message:

## BAD BADSPT FILE, IGNORED

is printed; and COPY attempts to copy the disk as if the file BADSPT did not exist. Badspots are likely to occur on 30M-word disk packs. On these packs, a list of pairs of numbers is maintained inside the bottom cover. These pairs (track number, head number) represent probable badspots. 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.

Copying Storage Module

The COPY command handles the storage module disk (40-80M word disk) using disk numbers as described in Section 1. In this case, the disk number is related to the number of heads in the disk partition as follows:

<u>Partition</u>	<u>Disk Number</u>
1 head	XX006X+1
2 heads	XX046X
3 heads	XX046X+1
4 heads	XX106X

The storage module must be formatted by MAKE before it can be used. Therefore, a disk to be copied must be created by the MAKE command with the VIRGIN DISK question answered YES before it can be copied to. For further information, refer to MAKE in Section 5. The storage module disk need only to be created once (using MAKE) after it is recovered.

Badspots on a storage module are handled the same way as on a 30-million word disk.

## EXAMPLE OF USE OF COPY COMMAND

The following is an example of COPY use under PRIMOS II. Operation under PRIMOS III and IV is similar.

OK: COPY  
GO

FROM PHYS DISK= 50  
1.5M WORD PACK? YES  
TO PHYS DISK= 51  
1.5M WORD PACK? YES  
FROM, TO, RECORDS = 000050, 000031, 3248  
PARAMETERS OK? YES

## SECTION 7

## OPERATOR COMMANDS

This section describes commands that are normally used by the operator at the PRIMOS III or IV 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 INTERNAL COMMANDS

When started, PRIMOS III and IV wait for a command at the supervisor terminal. A typical sequence of commands is:

CONFIG

SETIME

STARTUP

These commands and their arguments are described in the following paragraphs. All system commands are issued at the supervisor terminal.

After PRIMOS III or IV is started, the PRIMOS II supervisor terminal becomes the PRIMOS III or IV supervisor terminal. Unless the USRASR or OPRPRI command is given, the supervisor terminal can be used only for a restricted set of operator commands, given in this section. Since the supervisor terminal operates in the supervisor address space, any external commands and/or the RESTOR, RESUME, and START commands will crash PRIMOS III and IV if they are given at the supervisor terminal. 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.

\*\*\*\*\*  
\* ADDISK \*  
\*\*\*\*\*

The command format is:

ADDISK Pdisk1 [Pdisk] ... [Pdisk]

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

NO ROOM

is printed at the supervisor terminal.

ADDISK is available only as a PRIMOS III or IV command to be issued at the supervisor terminal. Since PRIMOS III and IV support 18 logical disks, ADDISK must be used to start up disks beyond logical 8.



```
*****
* AMLC *
*****
```

The command format is:

```
AMLIC [Protocol] Line [Config] [Lword]
```

The arguments to the AMLC command have the following significance.

#### Protocol

After the system is running, users may assign the AMLC lines through the following commands:

```
ASSIGN AMLC [Protocol] Line [Config]
```

```
UNASSIGN AMLC Line
```

Line number (0 to 17 octal for PRIMOS II, 0 to 37 octal for the 31-user PRIMOS III and IV) is assigned to the user, and I/O protocol and the line configuration word is established for Line. The following protocols are available:

```
TTY      - terminal protocol
TTYHS    - high-speed terminal protocol
TRAN     - transparent protocol
TRANHS   - high-speed transparent protocol
TTYNOP   - disconnects terminal
```

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, in addition, a carriage return is echoed following carriage return. Bit 8 of each character is forced on. Note that .CTRL. P or BREAK does not cause the AMLC input/output program to QUIT. These characters are significant only at a PRIMOS III or IV user terminal. Both characters are ignored. 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 that protocol used to control PRIMOS III and IV user terminals, except for the action of .CTRL. P and BREAK.

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 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 less 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 specified in octal and must be equal to or less than the parameter Nuser. See Figure 7-1 for details about the line configuration word.

### Line

Line, the AMLC line number, is an octal number from 1 to 17.

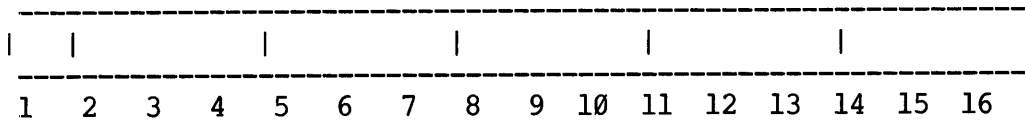
### Config

The Config argument which sets the line configuration, is an octal number that corresponds to the bit pattern illustrated in Figure 7-1. For example, the number 2213 would specify a data set at 300 baud with parity disabled and eight-bit character length (see Figure 7-1).

### Lword

The optional parameter Lword is an octal parameter divided into a number of significant parts. If Bit 1 of Lword is set, the line is half duplex; if Bit 1 of Lword is reset, the line is full duplex. Bit 2 set indicates that LINE FEED is not to be echoed for CARRIAGE RETURN. Bit 2 reset indicates that LINE FEED is to be echoed for CARRIAGE RETURN. Bits 8 through 15 of Lword contain the number of the user to which the AMLC is connected. This user number is the number printed at the terminal upon LOGIN or LOGOUT, or printed by the STATUS command indicating user number. Although STATUS prints the User Number in decimal, the user must specify the User Number in octal when specifying this value in the AMLC command. If the rightmost eight bits of Lword are zero, the AMLC line is not associated with any user space and is available to be assigned.

The AMLC command may be used to configure ASSIGNED AMLC lines as well as terminal AMLC lines.



<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 7-1. Setup of Line Configuration

<u>CONFIG</u>	Nuser	Pagedev1	Comdev	[Availm]	[Pagedev2]	[Namlc]	[NPhan]
	Nuser	An integer less than or equal to octal 40 that defines the number of users, including the supervisor (e.g., for a four-user system, enter 5; for a seven-user system, enter 10).					
	Pagedev1	A physical disk number that specifies the device to be used for paging. See Table 3-1 for possible values.					
	Comdev	A parameter that specifies the physical device number initially assigned as logical 0. When a user invokes an external command, the command directory CMDNC0 is searched on this device. If Comdev and Pagedev 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. (See the paragraph on configuration at the beginning of this section.)					
	Availm	An optional argument that defines available physical storage. It corresponds to the last sector number (octal) to be used. The values for Availm and associated storage used are as follows:					
	Blank or 0	All of Memory		(must be at least 32K)			
	'117	40K					
	'137	48K					
	'157	56K					
	'177	64K					
	...	...					
	'777	256K					
	Pagedev2	CONFIG may specify either one or two disk devices on which paging is to take place. CONFIG does not allow the alternate paging device to be a split disk.					
	Namlc	An optional argument that defines the number of assignable AMLC lines.					
	NPhan	An optional argument that defines the number of phantoms.					

The CONFIG command uses the range of acceptable Pagedev and Comdev codes (Physical Disk Numbers), as shown in Table 2-1.

**Example:**

A supervisor terminal operator may wish to specify two paging devices to run a 30-user system using 3.0-million word disk packs. If each user's virtual space is set to be 64K, only 23 users will fit on a 3.0-million word disk pack. Thus the use of two paging devices is required. The command line:

```
CONFIG 30 31 30
```

allows paging device to be physical devices 30 and 31 for a 30-user system.

Command File with CONFIG

It is possible to start up PRIMOS III and IV with a command file that contains the CONFIG command. To do so, the user generates a command file named C\_DOSV and places this file in the UFD CMDNC0. The first line in the command file C\_DOSV must be a CONFIG command. The command file C\_DOSV may contain other PRIMOS III or IV internal commands such as STARTUP and MESSAGE, but the command file must terminate with the command line:

```
CO TTY
```

If CMDNC0 has a password, the nonowner password must be null and C\_DOSV must have read protection rights for the nonowner.

Starting PRIMOS III or IV from a Command File

To bring up PRIMOS III or IV from the command file C\_DOSV, 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 or IV. The operator then enters the command:

```
PRIMOS
```

The PRIMOS III or IV operating system then starts running and checks if a command file C\_DOSV exists in the UFD named CMDNC0 or logical disk 0. If C\_DOSV does not exist, PRIMOS III or IV will take commands from the system terminal. If C\_DOSV 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 or IV 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.

```
*****  
* DISKS *  
*****
```

The command format is:

```
DISKS [NOT] Pdisk0 [Pdisk1] ... [Pdisk8]
```

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. Pdisk0 ... Pdisk8 are physical disk 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:

```
OK, DISK 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:

```
OK, DISK NOT 20250 OK, UNASSIGN 20250
```

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

```
*****
* MESSAGE *
*****
```

The PRIMOS III and IV internal command MESSAGE 1) provides a message facility that at the supervisor terminal; 2) allows an operator, at the supervisor terminal, to communicate with all users at all terminals connected to the system; or 3) allows an operator to communicate with a specific user at a specific terminal. The format for user to operator message is:

```
MESSAGE
text of message
```

where (text of message) is a one-time message. Two lines are printed at the supervisor terminal. Their format is:

```
*** 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
MESSAGE ALL NOW
MESSAGE -uu
MESSAGE -uu NOW
```

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 minus 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 or IV 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 an unfriendly thing to do if the user is in the midst 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.

To cancel a stored message, a null line must be entered as the text of message.



```
*****  
* SETIME *  
*****
```

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 digit only), 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 minus sign as the first character.

Example:

```
SETIME -09296 -1630
```

sets the date and time: September 19, 1974, 4:30 PM.

```
*****
* STARTUP *
*****
```

The STARTUP command defines a list of physical disk devices to be used by PRIMOS II, III or IV. A disk is considered started if it has been mentioned in a previous STARTUP command. Additional disks 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 [Pdev1 Pdev2 ... Pdevn]
```

where Comdev and Pdev1 ... Pdevn 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.) Physical device codes are listed in Section 1.

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 and IV and associates the following logidevice numbers with the physical device numbers specified: 52 is logical 0; 53 is logical 1; and 54 is logical 2. In PRIMOS III and IV 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
```

```
*****
* SHUTDOWN *
*****
```

The SHUTDOWN command performs tasks necessary to shutting down the PRIMOS III or IV system in an orderly manner. The format is:

```
SHUTDOWN
SHUTDOWN Pdev [Pdev2 ... Pdevn]
SHUTDOWN ALL
```

The command form: SHUTDOWN ALL performs a complete PRIMOS II or IV system shutdown. All user files are closed, physical disks are closed, and the PRIMOS III and IV 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.

If the SHUTDOWN command is issued with a list of physical devices (Pdev0 ... Pdevn), the listed devices are closed by closing all files opened in the listed devices and by detaching all users attached to the listed devices. Refer to Section 1 for a complete discussion of physical device numbers). Then, the specified disks are not available for PRIMOS file I/O operations until the devices are specified on 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 and IV will lose the command directory (from memory, not disk). To recover, STARTUP the disk and ATTACH CMDNC0.

#### Example of Selective Shutdown

Assume the initial STARTUP command was:

```
STARTUP 51 50 56 54
```

The operator wishes to replace the pack on physical device 56, which is logical device 2. The operator gives the command SHUTDOWN 56, stops the drive, replaces the pack, and restarts the drive. The operator then gives the command:

```
STARTUP 2/56
```

to startup physical drive 56 as logical device 2.

```
*****
* STATUS *
*****
```

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

1. Prior to mounting a new disk pack to determine what physical disk assignments are available.
2. After a request that all users release a given disk or disks, to determine that they have done so before shutting down that disk or disks.
3. As a check that all users have logged out before shutting down PRIMOS III and IV. (No harm to the system results if the users of a particular disk are still logged-in when the disk or the system is shut down. However, the user's files are closed and a message is printed at the terminal to that effect.)

Examples:

STATUS Command at the Supervisor Terminal

OK, STATUS

USR = SYSTEM

FUNITS

DISK	LDEV	PDEV
TS	0	250
SPOOLD	1	250
DUD	2	20250
DSKRAT	3	50
ADMIN	4	110250
ETCH	5	40250
PMFII	6	100250
MD6V2	7	4
TRANS	10	50250

PAGEDEV = 10252 COMDEV = 250

USER	LINE	PDEVS
JOEL	2	110250
SPOOL	3	252 PRI
COHEN	4	50 PTR PUNCH
MERRIC	5	250
GOUDY	8	20250
PODUSK	13	250
JDOAKS	16	110250

OK,

STATUS Command Issued at a User Terminal

OK, STAT

FUNITS

DISK	LDEV	PDEV
TS	0	252
SPOOLD	1	250
DUD	2	20250
ETCH	3	40252
LSTFIL	4	50250
DOSDVM	5	60252
WORKII	7	100252
ADMIN	10	110252

\*\*\*\*\*  
\* USRASR \*  
\*\*\*\*\*

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 by means of the proper form of the USRASR command. The format is:

USRASR Usrno

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. If connected to a current loop bit-banger line, the input leads must be shorted (or a terminal must be connected to the line).

Return

To return to operations as a normal supervisor terminal, type:

USRASR 1

## SECTION 8

## COMPUTER ROOM OPERATIONS

## OPERATIONS EXAMPLE

This section describes computer room operation for a typical installation. It is intended as a guideline for programming personnel who may have to startup or troubleshoot the system unaided late at night or on weekends. While some of the details may differ from operation of the user's computer configuration, the principles are similar and the techniques are useful as a general guideline for computer room operation.

## DEFINITION OF TERMS

<u>Term</u>	<u>Definition</u>
Set Sense Switch	On the computer's control panel, a sense switch is set if it is in the UP position.
SSW=XXXXXX	Where XXXXXX is an octal number, set sense switches to equal the number XXXXXX. Example: to set SSW equal to the number, 14, set bits of sense switches 13 and 14.
Bits	A bit corresponds to a sense switch 8-16 on the control panel.
Press Sense Switch	On the computer's control panel, press the sense switch downward.
Press Sense SSW=XXXXXX	Where XXXXXX is an octal number, press sense switches to equal the octal number XXXXXX.
DATA CLEAR	Press DATA CLEAR switch on control panel.
MASTER CLEAR	Press MASTER CLEAR switch on control panel.

## STARTING UP PRIMOS

To start up PRIMOS, proceed as follows:

BOOT in PRIMOS II (see Section 1)

Startup disk with the UFD that contains PRIMOS.

The following is an example of how PRIMOS is currently being started up.

Example:

PRIMOS NOT IN OPERATION

PHYSICAL DEV = 50  
PRIMOS REV. XX.XX MM/DD/YY (AT 170000)

OK: STARTUP 50  
OK: A SYSTEM  
OK: R PRIMOS  
GO

CONFIG SYSA 30 20061 50 5/2 5  
PRIMOS 4 <REV 11.31>  
506.0 K MEMORY IN USE

PLEASE ENTER DATE

OK, ADDISK  
OK, A SPOOL  
OK, PH C<-SPOOL  
OK, A FAM  
OK, CO C<-FAM  
OK, A SYSTEM  
OK, PH C<-CX

.  
.  
.

PRIMOS has been started up. After typing 'R PRIMOS', only the date and time need be typed. The command file enters the CONFIG and starts up all normally started disks.



## BOOTING IN (PRIMOS II)

The technique of Booting in PRIMOS II is as follows:

Set sense switches according to Table 8-1 in order to BOOT in PRIMOS II from the device listed.

Set rotary switch to STOP STEP.

MASTER CLEAR.

Set rotary switch to LOAD.

Press START switch.

Table 8-1. Sense Switch Settings to Specify Bootload Device

<u>DEVICE</u>	<u>SSW=</u>
TS TIMESHARE DISK	14
FLOPPY DRIVE	6
PAPER TAPEHIGH SPEED	2
PAPER TAPEASR	1
STORAGE MODULE	10014

The Systems Terminal prints: PHYSICAL DEV =. The operator then types the device number of the device containing a version of PRIMOS II.

In most cases, PRIMOS II is booted in from the Time-Share (TS/A) disk (SSW=14 and PHYSICAL DEV = 50)

## CRASH PROCEDURE

## NOTE

It is important that all actions and analysis be recorded, preferably in the log book entitled 'log book - System A')

Illegal Commands Typed at the Systems Terminal

If a crash is caused by typing an illegal command at the systems terminal, all disks have to be checked using FIXRAT (refer to Section 4 and the File System User Guide (MAN 2604) for details) before PRIMOS can be restarted. Certain command files may be used to run FIXRAT for the purpose of fixing disks.

## NOTE

If the systems terminal is left at the default CPPI 0, the system does not allow this type of crash to occur.

Hung System

If a disk is added to the system with the ADDISK command and is not in the ready state (ready light on disk drive indicates ready state), the system hangs for about 30 seconds. To check other possible causes of a hung system: (1) make sure line printer is on, (2) check select lock on 30-million word disk drives. If locked, clear and report lock to maintenance department.

Crash

Upon a systems crash, the rotary switch on the control panel is set to STOP STEP, data/address (DA/AD) switch is set to ADDRESS. Do not MASTER CLEAR unless specifically instructed to do so. If the number displayed in the data light '26303, the crash is a memory parity halt. If any other number is displayed, see a programmer if possible. If not, follow the procedure below to take down registers, then BOOT in PRIMOS II FIXRAT all disks, and restart PRIMOS III. If data lights show a '26303 (memory parity halt), follow the procedure below; if available run the memory parity finder, if possible. The memory parity finder locates bad pages in memory and allows the user to make them unavailable for use. This procedure often allows the system to be warm-started. Under PRIMOS III or IV, the user with the bad page of memory at the time of the crash must log out immediately after PRIMOS is restarted. This logout can be done from the user terminal or from the systems terminal by the command 'LO -(-UU)', where UU is the number of user with the bad page of memory at the time of the crash.

The following paragraphs give a checklist of procedures to follow in the event of a system crash.

Registers 0-37 are taken down (recorded in log).

The procedure is:

Set rotary switch to fetch Y.

Set DA/AD switch to ADDRESS.

DATA CLEAR.

Press START switch; address 0 is displayed. By setting DA/AD switch to DATA data at ADDRESS or register 0 is displayed on panel lights. Take down data at address 0 .

Set rotary switch to fetch Y+1. Press START switch. DA/AD switch set to DATA shows data at address 1 . DA/AD switch set to address shows a 1 for address 1 .

Every time START switch is pressed, the next data/address correspondence can be accessed and taken down. Continue this process for registers 8-37.

Data at addresses '25463 (SUPSW) and '25477 (CUSR), are also recorded into the log.

The procedure is:

Set rotary switch to fetch Y.

Set DA/AD switch to ADDRESS.

DATA CLEAR.

Press SSW='25463 . (Press bits 3,5,7,8,11,12,13 )

Press START switch.

Set DA/AD switch to DATA.

Record data in log, and repeat above procedure for CUSR. (Press SSW= '25477 instead of '25463)

Run memory parity finder (if available) by warm starting PRIMOS. If any bad pages are found, the bad page numbers are typed with the following message:

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

#### Locking Out Bad Pages

(For example, assume that page 247 is found to be bad:  
The procedure is:

Add 22000 to the bad page number (e.g., 22000+247=22247).

Set rotary switch to fetch Y DATA CLEAR.

Press SSW=bad (the calculated number of the bad page;  
in this case '22247) page number+22000.

Set DA/AD switch to DATA.

NOTE

1. The number displayed by setting the DA/AD switch corresponds to a memory location.
2. If the number displayed is less than 2400, the system cannot be warm-started.
3. The user number found by the MMAP is a user who must logout immediately after the system is warm-started.

Set rotary switch to store Y.

Press SSW=17777.

Press START switch.

Continue this process for any other bad pages found.

WARM-STARTING PRIMOS

The procedure is:

Set rotary switch to RUN.

Press START switch twice.

If applicable, LOGOUT users with bad memory pages at the time of crash.

Example log containing data taken after a systems memory parity halt:

December 10 1975

Registers-Contents	Bad Page No.	Displacement	Contents
0 15	247	622	120261
1 0	247	727	40763
2 63670	251	127	12401
3 177777			
4 37254			
5 177777			
6 142400			
7 26272			
10 4			
11 4			
12 1400			
13 217			
14 14			
15 14			
16 4015			
17 5511	CUSR=5		
20 176402	SUPSW=1		
21 22317			
22 0			
23 0			
24 0			
25 0			
26 0			
27 0			
30 0			
31 0			
32 0			
33 0			
34 1			
35 16000			
36 0			
37 133534			

## POWERING UP OR DOWN SYSTEM

### Powering Down

To power down, key top left rotary switch to OFF, after shutting down all disks with the command 'SH ALL'. Shut down all other devices if the main power is to be shut down. Shut off the 30-million-word disk drives with their STOP-START switches. Then shut off the main circuit breakers on the drives. These circuit breakers may be accessed by opening the front panels on the drives.

### Powering Up

To power up, turn on all devices that were turned off. Power up the 30-million word drives by turning on the main circuit breakers. Then turn on START-STOP switches.

## COMMAND FILES FOR FIXRAT AND COPY

The following command files are site-specific to one computer site, but they are described to show that FIXRAT and COPY may be run from command files.

### FIXYES

FIXYES is a command file in UFD=System Name. FIXYES runs the FIXRAT option on all disks normally started up. It does not FIXRAT the 1.5-million word drive(s). If the 1.5-million word disk is to be FIXRATed, it must be done by hand to run FIXYES, BOOT in, 'STARTUP 50', 'A System', and 'CO FIXYES'.

### FIXNO

FIXNO runs the same as FIXYES but does not fix the disk. FIXNO does not FIXRAT the 1.5-million word disk drives.

### C<-COPY

The command file C\_COPY writes out a map to the terminal (C\_COPY is run from the supervisor terminal) showing where the backup disk is to be placed for each copy. (At present, copying 30-million word drives requires three COPYs). C\_COPY checks packnames in between copies, the command file pauses so that disk packs may be changed. The backup disk is indicated by BK following the backup drive number in the map that C\_COPY prints.

### COPY by Hand

COPY may be invoked by hand. To COPY all ten partitions of a 30-million word disk drive, 'the FROM:, TO:', queues responses are 50XX 50YY 30, where XX is the FROM device number, YY is the TO device number (in both cases, XX and YY are 50, 52, 54, or 56). For further information about COPY, refer to Section 6.

## STARTING UP DISKS

PRIMOS III and PRIMOS IV

Under PRIMOS III and IV, all disks are started up with the ADDISK command (described in the Interactive User Guide MAN 2602). The 1.5-million word disk drive is started up as device 36 for an 8 sectored disk or 56 for a 32 sectored disk. Physical disk numbers are discussed in Section 1. Figure 8-1 shows the location of the disk pack number and disk pack label on the disk pack.

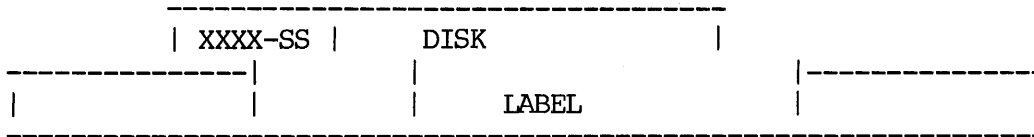


Figure 8-1 Disk Device Numbers.

The number on the face of the disk pack, SS, indicates the number of sectors. Sectoring may also be determined by counting notches on the bottom of the disk pack. SS will be either an 8 or a 32.

PRIMOS II

Only disks started up at the last 'STARTUP' are currently started up.

VDOS64

Any disk that is not started up under PRIMOS II may be started up under PRIMOS III or IV. The user who starts up a disk under PRIMOS III or IV is the only user that may access the disk. Only commands under CMDNC0 of that disk may be invoked, unless a program is restored before invoking PRIMOS III or IV. The following is an example procedure for executing a program under PRIMOS III or IV. In this example, the program is not under CMDNC0 of the disk started up in VDOS64, but under CMDNC0 of an ADDED disk.

<u>Step</u>	<u>Command Given at Systems Terminal</u>	<u>Command Given at User Terminal</u>
1	OK, DISK 56	
2		OK, AS DISK 56
3		OK, REST PROGMM
4		OK, SVC 1
5		OK, VDOS64

PRIMOS REV 11.33 2/2 '72 (AT 170000)

```

6          OK: STARTUP 56
7          OK: A UFDNAM
8          OK: S 1000
          (PROGRAM EXECUTES HERE)
9          OK: SH
10         (TYPE CONTROL P)
          QUIT,
11         C ALL
12         OK, SVC 0
13         OK, UN DISK 56
14 OK, DISK NOT 36

```

#### WRITING TO A WRITE-PROTECTED DISK

If a request is made to write on a WRITE-PROTECTED disk, an error message indication is printed at the user terminal as well as the supervisor terminal. The first six-digit number typed as part of the error message is the WRITE-PROTECTED disk number. To shutdown the disk from the supervisor terminal, use the SHUTDN command as many times as necessary.

#### USING MAGRST AND MAGSAV

##### MAGSAV

Responses to the following MAGSAV, MAGRST request are:

**TAPE UNIT:** The physical number as dialed on device (9-track 1) only; at Rev. 11, MAGSAV does not accept 7-track tape.

**ENTER LOGICAL TAPE NUMBER:** The response is 1 for the first logical tape, 2 for the second, etc. MAGSAV will rewind and reposition. A response of 0 implies that the tape is already positioned correctly and MAGSAV takes no action.

**TAPE NAME:** Any six-character name.

**DATE:** MM DD YY

**REV NO:** An arbitrary number.

**NAME:** The response is either a file name or a MFD that saves the MFD to which the disk is currently attached. Alternate responses are: \$A, \$R, \$Q. \$A changes the home UFD via an ATTACH.



Example:

```
$A USERX PASSWD LDEV
```

\$Q and \$R each terminate a logical tape and return to the operating system. \$R also rewinds the tape. NAME: is printed when writing has been completed, so that further writing may be requested or the current logical tape may be terminated.

### MAGRST

All restoring takes place in the home UFD. MAGRST asks for the tape unit and the logical tape number exactly as in MAGSAV. MAGRST then prints out the name, date, and Rev on the user terminal and asks:

READY TO RESTORE: The responses are YES, NO, or PA (Partial). YES restores the whole logical tape. NO requests a new tape unit and logical tape combination. PARTIAL permits a RESTOR of part of a tape.

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

```
NAME1 > NAME2 > ... > NAMEx
```

Any file on the logical tape whose tree name begins with the sequence entered is restored.

Example:

All tree names in a level 1 SAVE (MFD) begin with MFD, so the tree name to restore an entire UFD would be MFD>UFDX.

For further information about MAGSAV and MAGRST, refer to the Interactive User Guide (MAN 2602).

### MOUNTING A TAPE ON THE 9-TRACK TAPE TRANSPORT

Mount the tape as shown on the diagram on the inside of the tape transport's front cover. Turn bottom reel by hand three revolutions to secure tape. Press LOAD button twice. Press ON LINE button. To rewind, press RESET button and then rewind button. If the WRT EN (WRITE ENABLE) light is lit, writing is allowed. If not lit, then the WRITE ENABLE ring is not mounted on the reel of tape, and writing to the tape is not allowed.

## LINE PRINTER OPERATION TIPS

For efficient line printer operation, observe the following:

**PAPER JAM:** Press STOP button and re-insert paper. If part of a file is lost, the file may be reprinted from its start by logging out SPOOL (by typing LO -30 at the systems terminal) and restarting it after performing the following steps:

1. Dump the output buffer of the printer.
2. Press the RUN button to dump anything left in the buffer.
3. Then restart the SPOOL program. For example, at a given computer site, the SPOOL program may be started from a command file by typing:

```
CO START SPOOL
```

**ALIGNING PAPER:** After paper has been clamped on the line printer, adjust the paper so that its fold mark is about one quarter of an inch above the hammers. Refer to the Line Printer User Guide (MAN 1942).

## PRIMOS IN NETWORKS

The 15-user version of PRIMOS III or IV, together with a symbiont called FAM, provide a file-sharing capability to all users in a two-node network comprised of two Prime 300, two Prime 400, or one Prime 300 and one Prime 400 computer systems. The operation of the network is described in the Interactive User Guide MAN 2602; the commands used to startup and run the network software are also described in that user guide. The network supports file sharing, which is the ability to use standard file system calls and commands to manipulate files on a set of disks that may be physically connected to the remote as well as the local computer. All computers are connected by an Interprocessor Controller (IPC) that allows high-speed communication between the memories of two Prime computers.

## APPENDIX A

## BOOTSTRAPS

## INTRODUCTION

When power is first turned on, the semiconductor read/write memory of a Prime CPU is empty. To start operation, the operating system must be loaded from disk. 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 disk into main memory. Control then passes to this longer bootstrap, which completes the job of loading the memory image of the operating system from disk.

This appendix describes Prime bootstraps: control panel bootstraps (and key-in substitutes); and second level bootstraps stored on paper tape, magnetic tape, disk, or the storage module.

## CONTROL PANEL BOOTS

Control Panel u--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 u-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 with the following instruction sequence:

```
LDA (PROM address)
OTA '1720 (address setup - same as display data lights)
INA '1420 (input PROM location)
STA (memory location)
```

For some applications, the initial '51 words may be sufficient to code a complete bootstrap. Caution must be exercised when coding a program to execute in the register file (locations '0 to '37), because some instructions alter registers.

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:

<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 2): Modifies itself for either ASR or high speed paper tape (by sense switches) and reads a second-level MDL boot into memory. This boot requires that the first nonzero frame on the tape be '20 and the next two frames be '004/010 = '2010 =JMP '10. If the initial A-register setting (saved by the pre-boot in location '57) is to be used, it must be saved before location '57 is loaded by the bootstrap. The first zero frame on the tape causes the JMP '10 instruction in location '20 to be skipped. When execution starts at location '21, the following locations have been set up:

<u>Location</u>	<u>Contents</u>	<u>Instruction (X=1 for PTR, =4 for ASR)</u>
'2	'3000X	OCP X
'3	'13100X	INA '100X
'10	'13100X	INA '100X
'11	'002010	JMP *-1
'12	'141240	ICR
'13	'13000X	INA X
'14	'002013	JMP *-1
'15	'050000	STA 0,1
'16	'140114	IRX
'17	'100040	SZE
'20	'002010	JMP '10 (from tape)

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.

SS =	$\frac{12}{X}$	$\frac{13}{0}$
	0	1
	1	1

MHD (4): Moving head disks come in three varieties: two platter drives (3M or 6M words) on either 4000 or 4002 controllers, 20-surface drives on the 4001/4002 controller, and 80MB or 300MB storage modules. Sense Switches 11, 12, and 13 are used as follows (X - don't care):

SS =	<u>11</u>	<u>12</u>	<u>13</u>	<u>14</u>	
	X	0	X	0	4000, upper surface
	X	1	X	0	4000, lower surface
	0	0	0	1	4002, lower surface ('21)
	0	1	0	1	4002, upper surface ('21)
	0	0	1	1	4002, upper surface ('23)
	0	1	1	1	4002, lower surface ('23)
	1	X	0	1	Storage module ('26)
	1	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. Selection of the lower surface on a 20-surface MHD translates to:

Record 0 on Head 2

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 a loadpoint (space forward, abort, and rewind), and reads one tape record into memory starting at location '220 and through '777 (4K). Execution begins at '1000 (via a JST '777).

FLOPPY (6): Reads record 0 (track 1, sector 1, 448-word PRIMOS II record format) 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.

PROM Generation

Generation of control panel PROM is a three-step operation: write, assemble (PMA), and load (LOAD) the control panel boot program (CPBOOT); generate a PROM simulator paper tape with the CPBGEN program; and physically make the PROM.

CPBOOT: CPBOOT is the standard Prime control panel bootstrap program. It resides on UFD=AIDS on the master disk. There are three general rules for generating a control panel boot:

1. It must be loaded at '1000 in 16S mode, but executable in sector zero (all sector bits reset).
2. All unused locations in sector '1000 must be set to 0.
3. A maximum of 256 locations can be used (512 for larger control panels).

Rules 1 and 2 are satisfied by the use of absolute offsets to the proper values and added to all addresses in memory reference instructions and an initial instruction sequence of:

```

D16S
ABS
ORG '1000
BSZ 512
ORG '1000

```

Offsets are computed and used as follows:

	PBD	EQU	6-*	(*='1000)
'1000	PB2	DATA	7	at '1000, to be loaded at 6
'1001		LDA	*-1+PBD	(*='1001, '1001-1+6-'1000=6)
'1002		LDA	PB2+PBD	(PB2='1000, '1000+6-'1000=6)
	.			
	.			
	OVER	EQU	'1042	for subsequent device boot offsets actual location = '50
	.			
	.			
	STD	EQU	OVER+PBD-*	(*='1051)
'1051	START	ANA	S1+STD	(S1='1056, '1056+'1042+6-'1000-'1051 ='55)
'1052		SNZ		
'1053		JMP*	STE+STD	(STE='1060, '1060+'1042+6-'1000- '1051='57)
'1054		E64R		
'1055		JMP*	1	
'1056	S1	DATA	'177700	
	STE	EQU	*+1	(next location contains '1000)
'1057		DATA	'1000	

The number of locations in the auto-start boot is computed by STE-START= '1060-'1051=7. Each device boot, as well as the pre-boot, defines its own base offset to make the code easier to read. The use of these bases as memory reference modifiers requires that no literals be used. A FIN pseudo-op is placed at the end of each boot to allow easy identification of any literals so that they may be replaced with appropriately named locations. Since the boot program must be wholly contained within sector 1 (no LOAD generated cross sector links), instructions of the form LDA -1, 1 cannot be used. The command file C CPB, also on UFD=AIDS, produces a SAVE file named \*CPB. CPBOOT occupies 240 ('360) locations of PROM.

CPBGEN: CPBGEN punches a PROM simulator tape of locations '1000-'1777. Since the control panel u-code expects the PROM to contain the one's complement of the desired locations, CPBGEN performs a preliminary backscan and inverts all locations except the unused trailing zeroes (a 0 is inherently more reliable than a 1 in PROM). Since PROM comes in eight X 512-bit chips and the PROM simulator loads two parallel 8-bit banks, CPBGEN first punches all left bytes and then all right bytes. A given byte is punched as two ASCII hexa-decimal digits followed by an ASCII apostrophe. For example, the bit pattern 10100110 is viewed as 1010/0110 = C6 and is punched as '303/'266/'247. A TAPE-ON ('222) turns the reader on and a TAPE-OFF ('224) turns the reader off. The final tape format is:

leader (48 inches): TAPEON: left bytes (0-'777): TAPEOFF:  
 blank tape (48 inches): TAPEON: right bytes (0-'777): TAPEOFF:  
 trailer (48 inches)

CPBGEN resides on UFD=AIDS of the master disk and the command file C CPBG produces a run file named \*CPBG. To punch a paper tape of CPBOOT, the following sequence of instructions must be used:

FILMEM		ensure sector '1000 filed with zeroes
RESTORE	*CBP	into '1000-'1777
ASSIGN	PUNCH	(PRIMOS III only)
RESUME	*CPBG	at '2000
UNASSIGN	PUNCH	(PRIMOS III only)

Physical PROM: To create the physical PROM, load the paper tape produced by CPBGEN into the PROM simulator and verify that the load is good. Use the PROM simulator to blow the actual PROM chips and insert them into the control panel.

#### Key-In Substitutes for Control Panel Boots

Since the auto-load control panel PROM function is optional on some Prime computers, 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 control panel disk bootstraps (FHD/MHD/FLOPPY) read one 448-word PRIMOS record from record 0 of the selected device into memory, starting at location 770, and begin execution at 1000 in 16S mode. Regardless of the device booted from, the second level boot is the same and can, in turn, read PRIMOS II into memory from any PRIMOS II disk in the system. The source is named BOOT and resides in the UFD=FILEID on the master disk. The run file on record 0 is also named BOOT and resides in the MFD. The storage module bootstrap is a 1040-word record, and it can bootstrap PRIMOS II from any device.

## 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 unit number as defined for the PRIMOS II STARTUP command. The format of the unit number as of revision 11 is:

- Bits 1-4: head offset divided by 2 (4002 controller and storage module only)
- 5-8: # heads divided by 2 (4002 controller and storage module only)
- 9: alternate device (4002 controller and storage module only) address (Bit 9 set)
- Bits 10-13: Type = 0 type 4000 controller and MHD
  - 1 Type 4000 controller and FHD
  - 2 floppy
  - 3 Type 4000 controller 8-sector MHD
  - 4 Type 4000 controller 64-sector FHD
  - 5 Type 4000 controller 32 sector MHD
  - 6 storage module
  - 7-17 unused

Bits 14-16: Unit = Physical drive number. For 4000



MHD's, even units are upper platters and odd units are lower platters of the drive number in bits 14 and 15. For the storage module, bit 16 is the low order bit of the number of heads/partition (bits 5-8).

Which DOS (\*DOS16, \*DOS24, \*DOS64) is read in is determined by either memory size of Sense Switches 1, 2, and 3 in the following manner:

If all sense switches are reset, the highest memory DOS that will fit in available memory is read.

If any of the sense switches are set, 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>DOS</u>
0	highest that will fit
'20000	error
'30000	*DOS16
'40000	*DOS16
'50000	*DOS24
'60000	*DOS24
'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 reexecuted 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 finally the BOOT program itself. 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) (must = '3011)  
 1 = end address (EA) (must be correct)  
 2 = program counter (PC)  
 3 = A-register  
 4 = B-register  
 5 = X-register  
 6 = Keys  
 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 (thus requiring the end address to be correct). Because BOOT can never be executed as a console 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	Short boot (448-word record)
	= 1	Long boot (1040-word record)
A	= 0	Not used
B	= '110	Option A master clear default control register
X	= '27	SOC master clear default control register 1
Keys	= '74000	SOC master clear default control register 2

Once a BOOT has been placed on a disk, it can be copied to another disk with the following command sequence:

```

A   MFD   XXXXXX   a
RESTORE  BOOT
A   MFD   XXXXXX   b
SAVE    BOOT

```

Since the save parameters can be specified, the PC, A, B, X, and Keys registers can be set to other than master clear defaults for option A or SOC to allow other types of system consoles to be configured to the system. When changing any of these values, care must be taken not to change any of the others. Also, special care must be taken if transferring a boot to or from the storage module. If the boot is restored from other than the storage module, the high address will be '3667 and the PC=0. This must not be changed, even if saving onto a storage module. When restoring from the storage module, however, the high will be greater than '3667 and PC=1. To save it onto other than a storage module, the high must be specified as '3667 and PC=0.

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

## DISK RECORD HEADER FORMATS

Disk Record Header Formats

<u>Word No.</u>	<u>448-word record</u>	<u>1040-word record</u>
0	Record Address	Record Address-high
1	Beginning Record Addr	Record Address-low
2	Forward Pointer	Beginning Record Addr-high
3	Backward Pointer	Beginning Record Addr-low
4	No. of Data Words	No. of Data Words
5	File Type	File Type
6	Spare	Forward Record Addr-high
7	Spare	Forward Record Addr-low
8	Data Words	Backward Record Addr-high
9	Data Words	Backward Record Addr-low
10	Data Words	Spare
11	Data Words	Spare
12	Data Words	Spare
13	Data Words	Spare
14	Data Words	Spare
15	Data Words	Spare
16	Data Words	Data Words

## NOTES:

All disks except the storage module have 448-word records. Storage modules have 1040-word records.

The Beginning Record Address of the first record in a file points to the directory (UFD or segment) entry of the file. In all other records, the BRA points to the first record of the file.

The forward pointer contains the address of the next sequential record in the file or 0 if it is the last record in the file.

The backward pointer contains the address of the previous record in sequence or 0 if it is the first record in the file.

The file type is only valid in the first record in the file. Legal values are:

0	SAM file
1	DAM file
2	SAM segment directory
3	DAM segment directory
4	UFD

If the file is the record zero bootstrap (BOOT) or the disk record availability table (DSKRAT or volume name) and the disk has a 1040 record size (storage module), Bit 1 (100000 octal) of the file type will be set.

## INDEX

- 9-TRACK TAPE TRANSPORT 8-11
- ADDISK 7-2
- AMLC 7-3
- AMLC REQUIREMENTS
- AMLC DRIVER CONFIGURATION 2-8
- ATTACHING TO UFD 1-16
- BACKUP 1-22
- BAD BOOT 4-9
- BEGIN 4-4
- BOOT 5-6
- BOOTIN IN (PRIMOS II) 8-3
- BOOTSTRAP LOADING 1-2
- BROKEN FILE STRUCTURE MESSAGES 4-7
- BUFFER SIZE 1-12
- BUILDING A PRIMOS II DISK FROM PAPER TAPE 1-18
- BUILDING MAKE 5-6
- BUILDING RELATIVE-MODE PRIMOS II 1-24
- BULLETIN 7-9
- BYPASSING BAD MEMORY 2-5
- C<-COPY 8-8
- CHANGING CONFIGURATION TABLE 2-12
- CHANGING DISK PACKS 1-23
- COMMAND FILE WITH CONFIG 7-7
- COMMAND FILES FOR FIXRAT 8-8
- COMMAND FILES FOR SYSTEM BUILDING 2-2
- COMMAND FILESS FOR COPY 8-8
- COMPUTER ROOM OPERATIONS 8-1
- CONFIG 7-6
- CONFIGURATION FEATURES 3-1
- CONTROL PANEL BOOT 1-14
- CONTROL PANEL BOOT FROM STORAGE MODULE 1-14
- COPY 6-1
- COPY BY HAND 8-8
- COPY CHECKSUM 6-5
- COPY METHOD 6-4
- COPY SUCCESS OR ABORT 6-5
- COPYING MASTER DISK PACK 1-17
- COPYING STORAGE MODULE 6-6
- CRASH 8-4
- CREATING A SUB-UFD 1-20
- CREATING ADDITIONAL PRIMOS II DISKS 1-19
- DAM FILES ON STORAGE MODULE 1-13
- DATA TRANSFER BETWEEN DISKS 1-7
- DEFINITION OF TERMS 8-1
- DELETING DIRECTORIES 4-9
- DESCRIPTION OF BADSPT 5-5
- DESCRIPTION OF MESSAGES 4-10
- DETECTION OF BAD MEMORY 2-15
- DIRECTORY NESTING LIMIT 4-9
- DISK BUILDING 1-17
- DISK CONFIGURATION 6-1

## INDEX

- DISK SPACE 2-7
- DISK WITH BADSPOTS 6-5
- DISKS 1-3
- DISKS 7-8
- DISKS WITH BADSPOTS 4-8
- ENTERING NEW UFD'S 1-20
- ERRORS 7-7
- EXAMPLE OF FIXRAT 4-14
- EXAMPLE OF SELECTIVE SHUTDOWN 7-13
- EXAMPLE OF USE OF COPY COMMAND 6-6
- FHD DISK ADDRESSES 2-13
- FHD DISK ADDRESSES 2-14
- FILE ACCESS PROTECTION 2-5
- FILE SYSTEM CALLS WITH STORAGE MODULE 1-12
- FIXNO 8-8
- FIXRAT 4-1
- FIXRAT AND 30-MILLION WORD DISK 4-13
- FIXRAT AND DISKS 4-2
- FIXRAT DESCRIPTION 4-1
- FIXRAT ERROR MESSAGES 4-10
- FIXRAT OUTPUT EXAMPLE 4-6
- FIXRAT TRAVERSE OF FILE STRUCTURE 4-4
- FIXYES 8-8
- FLOPPY DISK TROUBLESHOOTING 1-26
- FORMAT OF PRIMOS II DISK 1-17
- HEAD OFFSET DEFINITIONS 1-9
- HEADER SIZE 1-12
- HUNG SYSTEM 8-4
- ILLEGAL COMMANDS TYPED AT SYSTEMS TERMINAL 8-3
- INACTIVITY TIMEOUT 2-6
- INITIAL PRIMOS II OPERATING SESSION 1-16
- INSTALLING NEW EXTERNAL COMMANDS 1-21
- LINE 7-4
- LINE PRINTER OPERATION TIPS 8-12
- LOADING, BOOTSTRAP 1-2
- LOCKING OUT BAD PAGES 8-5
- LOGOUT, FORCED 2-9
- LWORD 7-4
- MAGRST 8-11
- MAGSAV 8-10
- MAKE 5-1
- MEMORY PARITY 2-6
- MEMORY PARITY HALT AT 31565 3-4
- MEMORY REQUIREMENTS 2-7
- MESSAGE 7-9
- MHD DISK ADDRESSES 2-13
- MHD DISK ADDRESSES 2-14
- MODIFICATION OF PAGE MAPS 2-8
- MOUNTING A TAPE 8-11
- MOUNTING A TAPE ON THE 9-TRACK TAPE TRANSPORT 8-11

## INDEX

- NUMBER OF HEADS DEFINITION 1-10
- OPERATIONS EXAMPLE 8-1
- OPERATOR COMMANDS 7-1
- OTHER PRIMOS III VERSIONS 2-2
- OVERVIEW OF COPY 6-1
- OVERVIEW OF MAKE 5-1
- OVERVIEW OF PRIMOS IV 3-1
- OVERVIEW OF PRIMOS OPERATION 1-1
- PAGING SPACE REQUIREMENTS 2-8
- PARTITIONED DISK ADDRESS SPECIFICATION 6-3
- PARTITIONING DISKS 1-7
- PERMANENT BAD PAGE 3-3
- PHYSICAL DEVICE NUMBERS USAGE 1-8
- PHYSICAL DISK NUMBERS 1-4
- PITFALLS 4-9
- PITFALLS 5-4
- PITFALLS TO AVOID 1-25
- POWERING DOWN 1-23, 8-8
- POWERING UP 8-8
- POWERING UP OR DOWN SYSTEM 8-8
- PRIMOS II 2-1, 8-9
- PRIMOS III 8-9
- PRIMOS III SYSTEM INITIALIZATION 2-3
- PRIMOS III SYSTEM OVERVIEW 2-1
- PRIMOS III VERSIONS 2-1
- PRIMOS III, IV INTERNAL COMMANDS 7-1
- PRIMOS IN CMDNC0 1-3
- PRIMOS IN NETWORKS 8-12
- PRIMOS IV 8-9
- PRIMOS IV MEMORY MAPS 3-2
- PRIMOS IV MEMORY PARITY TESTS 3-3
- PRIMOS IV PAGE MAPS 3-2
- PRIMOS IV VERSIONS 3-2
- PROCEDURE FOR STORAGE MODULE USERS 1-15
- PROCEDURE TO CONVERT TO A 1040-WORD FORMAT 1-15
- PROGRAM DEVELOPMENT USING PRIMOS II 1-20
- PROTOCOL 7-3
- RECORDS PARAMETERS FOR 30-MILLION WORD DISK 5-2
- RECOVERING FROM ERRORS 1-20
- RELATIVE MODE 1-1
- RELATIVE-MODE PRIMOS II 1-3
- REMOVING BAD PAGE BEFORE A WARM START 3-5
- RESTARTING PRIMOS II 1-24
- RETURN 7-16
- RUNNING COPY 6-1
- RUNNING FIXRAT 4-2
- RUNNING MAKE 5-1
- RUNNING SPLIT DISK 2-9
- SAMPLE FILE STRUCTURE 4-3
- SECTORED MODE 1-1
- SEGMENT DIRECTORIES 4-8

## INDEX

SETIME 7-11

SHARING FILES 2-5

SHUTDN 7-13

SHUTDOWN 1-22

STARTING PRIMOS III OR IV FROM A  
COMMAND FILE 7-7

STARTING UP DISKS 8-9

STARTING UP PRIMOS 8-2

STARTUP 7-12

STARTUP 2/56 7-13

STARTUP OF PRIMOS III 1-16

STARTUP OF PRIMOS IV 1-16

STATUS 7-14

STATUS COMMAND AT SUPERVISOR  
TERMINAL 7-14

STATUS COMMAND ISSUED AT A USER  
TERMINAL 7-15

STORAGE MODULE DISK NUMBER 1-11

STORAGE MODULE DISK SUPPORT 1-10

STORAGE MODULE ERROR DETECTION  
1-13

SUPERVISOR CALLS 1-3

SYSTEM CONFIGURATION 2-7

SYSTEM CRASH 3-4

TERMINAL PATCH LOCATIONS 1-24

TURNING POWER OFF 1-23

UFD'S ON STORAGE MODULE 1-13

UNRECOVERED DISK READ ERROR 4-14

USE OF MAGSAV 1-22

USER LOGIN/LOGOUT 2-6

USING FIXRAT 1-23

USING MAGRST 8-10

USING MAGSAV 8-10

USRASR 7-16

VDOS64 8-9

VIRTUAL-MEMORY 1-3

WARM RESTART FOR PRIMOS III 2-15

WARM START 3-5

WARM-STARTING PRIMOS 8-6

WRITE PROTECTED DISK 8-10

WRITING INTO DIRECTORIES 4-9

WRITING TO A WRITE-PROTECTED DISK  
8-10