TeleVideo®

MmmOST™
Programmer's Manual

TELEVIDEO

MmmOST™ PROGRAMMER'S MANUAL
(for Version 2.1)

TeleVideo Document No. 2003200

October 1982

Disclaimer

TeleVideo Systems, Inc. makes no representations or warranties with respect to the contents hereof and specifically disclaims any implied warranties of merchantability or fitness for any particular purpose. Further, TeleVideo Systems, Inc. reserves the right to revise this publication and to make changes from time to time in the content hereof without obligation of TeleVideo Systems, Inc. to notify any person of such revision or changes.

TeleVideo Systems, Inc.
1170 Morse Avenue
Sunnyvale, CA 94086
(408) 745-7760

TABLE OF CONTENTS

List of Figures. v
List of Tables v

1. INTRODUCTION 1

1.1 An Overview of MmmOST 1

1.2 Using the Manual. 1

1.3 Before You Start. 2

2. THEORY OF OPERATION. 3

2.1 Introduction. 3

2.2 An Overview of the CP/M Operating System. 3

2.3 The Single-User Environment 3

2.4 TeleVideo MmmOST and the Multi-User Environment . . . 4
2.4.1 Communication Between User Stations and the Service Processor 5
2.4.2 What Happens at Power On/Reset? 5
2.4.3 Operating the Service Processor as a Stand-Alone. . 5

3. DRIVE AND FILE ACCESS UNDER MmmOST 8

3.1 Introduction. 8

3.2 Types of Drives 8

3.3 Types of Directories. 8
3.3.1 Directory for a Private Drive 8
3.3.2 Directory for a Public Drive. 9
3.3.3 Directory for a Public Only Drive 9

3.4 System Communication File 10

4. RECORD AND FILE LOCKING. 12

4.1 Introduction. 12

4.2 Shared-File Access. 12
4.2.1 Interleaved Updating. 12
4.2.2 Fatal Embrace 12

4.3 Control Modes 13
4.3.1 Two Fundamental Schemes of Shared-File Access . . . 13

4.4 Using Control Modes 13

4.4.1 MmmOST Control Modes - (R/O, R/W, R/L, F/L) 14

4.4.2 Control Modes with Applications Program Control - (N/L, LCK, ULK) 18

4.5 Setting Control Modes 20

4.6 Access During File/Directory Modification 21
4.6.1 Unshared File on Public Drive 21
4.6.2 Unshared File on Public Only Drive. 22

4.7 Operation with Higher-Level Programming Languages . . 22
4.7.1 Operation with CBASIC 22
4.7.2 Operation with TeleVideo COBOL. 23

5. DRIVES, DIRECTORIES, AVAILABLE SPACE, AND FILE STATUS. . . 25

5.1 Introduction. 25

5.2 Logical Drives and Logical Drive Configurations . . . 25

5.3 Directories 27

5.3.1 Directories for Public and Public Only Drives . . . 27

5.3.2 Directories for Private Drives. 27
Directories for the TS 806 Private Drive A. 28
Directories for the TS 816 Private Drive A. 28
Password Designations 29
Accessing Default Private Directories Without Passwords 29
Accessing Private Directories with Passwords. . . . 29
Access to Private Directories - Examples Using LOGON 30

5.4 Determining Available Space and File Status 34
5.4.1 DSTAT 34
5.4.2 STAT. 36

6. COMMUNICATION BETWEEN USER STATIONS. 37

6.1 Introduction. 37

6.2 Sending Messages. 37

6.3 Reading Messages. 37

6.4 Interprocessor Communications Using FIFOs 38

7. PRINTER OUTPUT AND CONTROL 41

7.1 Introduction. 41

7.2 Print Spooling. 41
7.2.1 Printer Selection 41

7.2.2 Print Spool File. 42
7.2.3 Sending Data to the Print Spool File. 43
7.2.4 Printer Breakpoint Routine. 44

7.3 Print Control and Queue Selection 44
7.3.1 Print Control and Print Control Commands. 44
7.3.2 Queue Selection 48

8. RECONFIGURING MmmOST 50

8.1 Introduction. 50

8.2 Reconfiguration Procedures. 50
8.2.1 Method 1 -- From a Service Processor Terminal . . . 51
8.2.2 Method 2 -- From a User Station 56

8.3 Running GENPARMS. 56
8.3.1 Invoking GENPARMS 57
8.3.2 Parameter/Function Descriptions 57

9. ERROR AND STATUS MESSAGES. 62

9.1 Introduction. 62

9.2 User Station Error Messages 62

9.3 Status Messages on the Service Processor Terminal . . 63

APPENDICES

A. Suggested CP/M References 68

B. Programming Examples 69

C. Memory Error Messasges 74

D. Modifying the User BIOS 76

INDEX. 79

LIST OF FIGURES

2-1 Single-User CP/M Environment. 5
2-2 Multi-User MmmOST Environment 6

3-1 Private Directory Configuration 8

LIST OF TABLES

3-1 Drives, Directories, and Files in the MmmOST Environment 10

4-1 Mode Setting with TeleVideo COBOL 24

5-1 TS 806 Standard Drive Configuration 26
5-2 Example of a Directory Configuration. 39

7-1 Print Control Commands. 46

8-1 User Station and User Station CP/M File 50
8-2 TS 806 Private Directory Configurations 58

9-1 User Station Error Messages 62
9-2 Service Processor Terminal Status Messages. 64

C-1 Memory Functions. 74

1. INTRODUCTION

MmmOST, an acronym for Multi-User, multi-tasking, multi-processor Operating System Technology, is a unique operating system controller developed by TeleVideo Systems, Inc. exclusively for use with its computers. Through it, applications programs written for a single-user CP/M environment can run on TeleVideo multi-user computers although they do not become multi-user programs. With MmmOST as the "traffic director," multiple users can access common data files simultaneously and share relatively expensive resources, such as printers and hard disks, thereby lowering the costs of computing.

This manual explains the features of Version 2.1 MmmOST, how to write or modify applications programs to take full advantage of those features, and how to change the configuration of your system.

Note!

Throughout this manual, the term **service processor** describes the central service unit which physically contains the shared disk drives and runs the program called MmmOST. Satellite user stations will be referred to as the **user stations.**

1.1 AN OVERVIEW OF MmmOST

MmmOST runs in the CP/M Transient Program Area (TPA) of the service processor and polls the service processor's RS 422 communication ports. When there is a request for use of the shared resources from a user station, MmmOST makes system calls to the CP/M BDOS and BIOS residing in the service processor.

MmmOST controls

Communications between user stations and the service processor or between individual user stations

Disk file accesses

Print spooling to shared printers

1.2 USING THE MANUAL

The concepts presented in this manual assume that you are familiar with CP/M Level 2.2 and its file implementation. If you are not already familiar with CP/M, we urge you to read the CP/M documentation published by Digital Research and furnished with

your system. Refer also to Appendix A of this manual, where additional CP/M references (suitable for new users as well as more experienced programmers) are listed.

The manual is divided into the following chapters:

2. Theory of Operation

3. Drive and File Access Under MmmOST

4. Record and File Locking

5. Drives, Directories, Available Space, and File Status

6. Communication Between User Stations

7. Printer Output and Control

8. Reconfiguring MmmOST

9. Status and Error Messages

Appendix A Suggested References

Appendix B Programming Examples

Appendix C Memory Status Messages

The theory of operation given in Chapter 2 provides the basis for understanding the organization and operation of MmmOST. Chapters 3 through 6 describe the functional features which are unique to the MmmOST multi-user environment, such as shared file access, communication between user stations, and the handling of print files for multiple users and printers. Printing features are discussed in Chapter 7 and instructions on reconfiguration of the system are provided in Chapter 8 both for user stations with 8-bit processors and user stations with 16-bit processors. Chapter 9 describes status messages which may appear on the service processor terminal and tells how to suppress certain error messages which may appear on user station consoles.

Examples are used frequently in this manual to show how statements would be written in CBASIC. There is also a general discussion on the use of higher level programming languages, including a discussion of TeleVideo COBOL, which automatically sets the control modes used by MmmOST for shared file access.

1.3 BEFORE YOU START...

BEFORE YOU USE MmmOST, BE SURE TO DUPLICATE THE MASTER SOFTWARE DISKETTE SUPPLIED WITH YOUR SYSTEM.

2. THEORY OF OPERATION

2.1 INTRODUCTION

This chapter presents an overview of the CP/M operating system, describes the single-user environment, and tells how MmmOST running on a service processor expands that environment into a multi-user environment.

2.2 AN OVERVIEW OF THE CP/M OPERATING SYSTEM

In order to get a feeling for the overall organization and operation of MmmOST, let's quickly review the features of the CP/M operating system as installed in a single-user computer.

CP/M provides communication paths between your applications programs in the Transient Program Area (TPA) and the mass storage and input/output (I/O) devices which are connected to the computer. By taking advantage of the device drivers (interfaces) built into CP/M, the applications programs can select and access disk files and I/O devices through a single entry point in CP/M. User programs making use of this entry point gain portability, since the programmer does not need to know the exact hardware configuration of the computer in use.

2.3 THE SINGLE-USER ENVIRONMENT

All CP/M file and I/O accesses are made to "logical" devices. In any particular computer configuration, logical devices are mapped to physical devices through driver programs in the CP/M Basic Input/Output System (BIOS) and Basic Disk Operating System (BDOS).

CP/M supports up to four logical I/O devices and up to 16 logical disk drives. In virtually all single-user systems using floppy disk drives, the physical disk drives correspond, one-to-one, to the logical drives. For example, selecting logical disk drive A corresponds to physical disk drive 0, logical drive B corresponds to physical drive 1, etc.

As hard disk drives have become more economical, typical systems have begun to include a combination of floppy and hard disks. Since there is more space available on the hard disk, it is usually divided into a number of logical drives: C, D, E, etc.

The four standard logical I/O devices supported by CP/M are:

CON: The system console

LST: A printer (the list device--for hard copy)

RDR: The general-purpose input device (historically named reader)

PUN: The general-purpose output device (historically named punch)

It is now common to find sophisticated peripheral devices accessed through PUN: and RDR: (i.e., a modem for communication with a remote computer, a ROM programmer, or a speech recognition/synthesis device).

Figure 2-1 schematically illustrates a single-user computer running under the CP/M operating system. User applications programs run in the TPA and access logical devices through subroutine calls to the operating system. These calls include data as well as device/file selection information. Physical devices are assigned to logical devices in accordance with the bit patterns described in the I/O Device Assignment Byte (referred to here as the IOBYTE). Any one of four physical devices may be assigned as any logical device. Device assignments can be made by the applications program or by the operator through commands from the service processor terminal (a terminal connected to the service processor).

2.4 TELEVIDEO MmmOST AND THE MULTI-USER ENVIRONMENT

The hardware organization of TeleVideo's multi-user systems and the design of MmmOST provide maximum system throughput as well as a familiar, user-friendly environment.

Each user is provided with his own microprocessor, RAM, CRT, and keyboard. The user station is loaded with its own copy of the CP/M operating system, under which the user runs his own applications programs.

Existing CP/M applications programs can take advantage of all of MmmOST's unique features once the modifications described in this manual are made. Should you wish to run applications programs as though they were still in a single-user environment, the only difference will be operational, rather than internal to the program. First, under MmmOST, print requests from user stations directed to the printers on the service processor will no longer result in immediate printed output. Second, because of the private directory feature, users can isolate files from each other.

In the minimum multi-user system (as diagrammed in Figure 2-2), all disk access and I/O requests generated by the user stations are handled by the service processor. User stations can be enhanced by the addition of a printer, modem, or other serial device (depending upon the system environment) or additional disk drives. For example, a user developing a program may need his own line printer while other users share a letter-quality printer. Each user has the opportunity to access all shared resources and yet still have his own set of private devices.

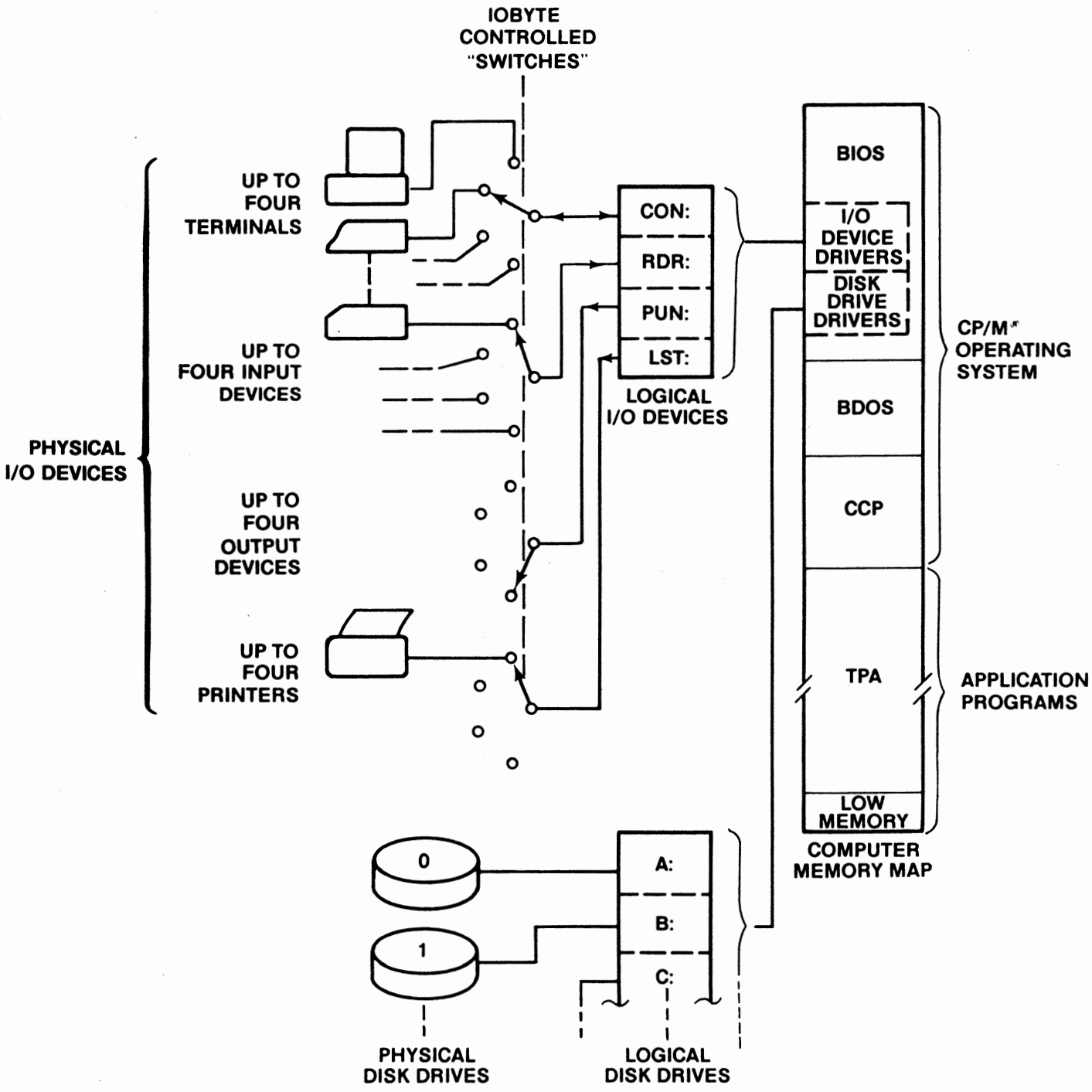


Figure 2-1 Single-User CP/M Environment

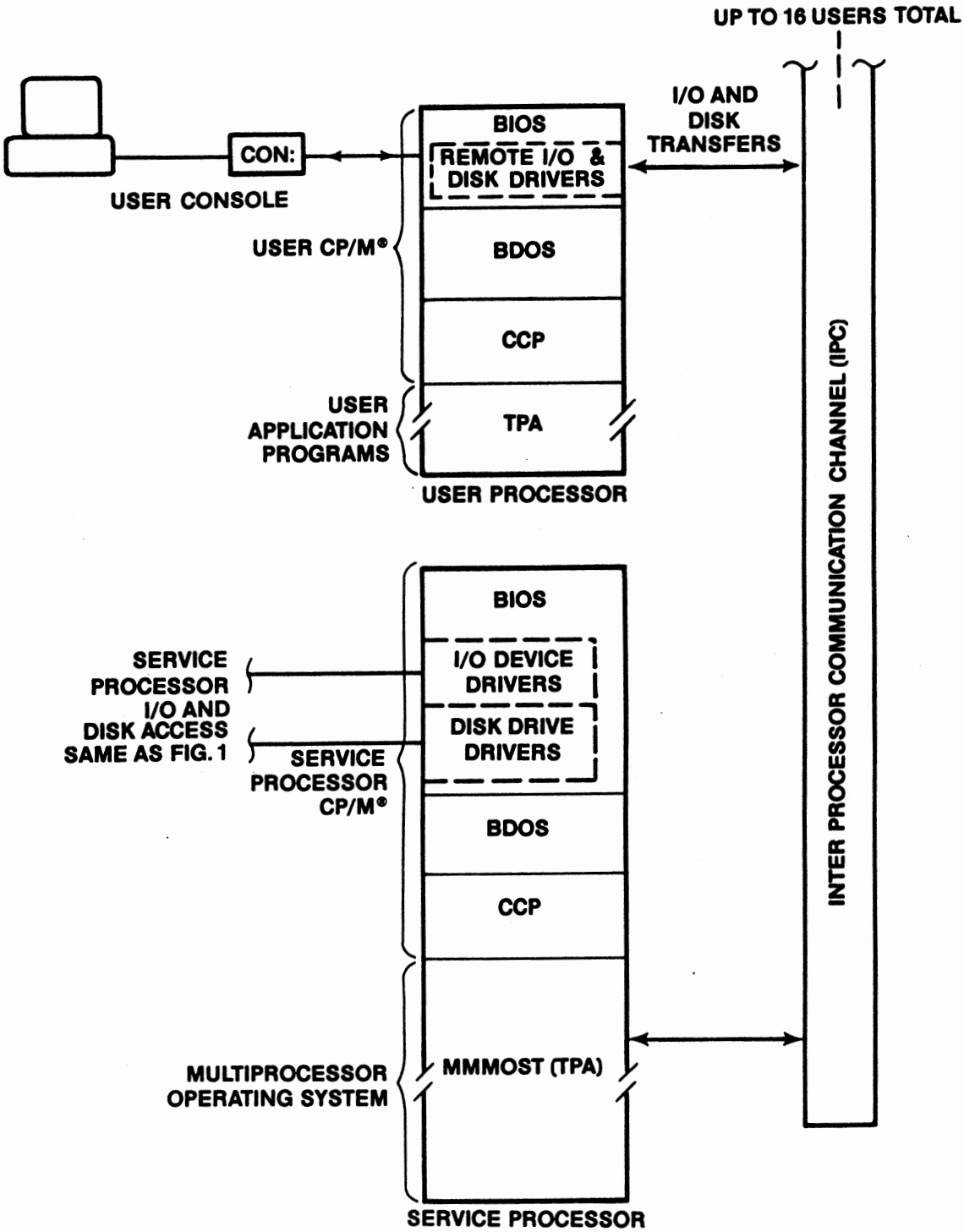


Figure 2-2 Multi-User MmmOST Environment

2.4.1 Communication Between User Stations and the Service Processor

Disk accesses and I/O functions from user stations to the service processor is performed by an Inter-Processor Communications Channel (IPC) and the MmmOST program in the service processor. The IPC controls data transfers through an EIA RS 422 serial port. The driver programs are installed in the user station BIOS and in the MmmOST program.

2.4.2 What Happens at Power On/Reset?

When the service processor is powered on or reset, MmmOST is loaded into the TPA. When the user stations are powered on, a customized version of CP/M is loaded into each station and the following sign-on message and standard CP/M operator prompt are displayed on the user station screen:

```
TeleVideo CP/M Vx.x of mm/yy
DISTRIBUTED PROCESSOR Vx.y
Proc. ID XX <=> Y

A>
```

If a user station gets "hung-up" (for instance, during assembly language debugging routines), the operator can reboot his user station without interfering with any of the other users. When the service processor is booted (warm or cold), any operating user stations must be warm booted.

2.4.3 Operating the Service Processor as a Stand-Alone

For system development, reconfiguration, maintenance, or troubleshooting, a conversational terminal can be connected to the service processor (referred to here as a **service processor terminal**). Since MmmOST runs as a transient program under CP/M in the service processor, a ^C typed on the service processor terminal will abort MmmOST. Subsequently, the service processor's CP/M system will issue the usual operator prompt and await conventional CP/M commands. With MmmOST no longer running in the service processor's TPA, the user stations will no longer have access to the shared resources, but can function in all other respects as normal single-user computers with CP/M running in their local drive. During this time, the service processor will not service user stations; thus programs running on user stations will continue until finished or hang up if remote disk or printer I/O is required. Once MmmOST is restarted, the first request from a user station must be either a warm or cold boot.

NOTE!

Since MmmOST permits the same file name to be used on different directories, you may see duplicate file names if you change your service processor from a multi-user system to a stand-alone.

3. DRIVE AND FILE ACCESS UNDER MmmOST

3.1 INTRODUCTION

This chapter defines the types of drives, files, and directories allowed in the MmmOST environment.

3.2 TYPES OF DRIVES

There are three types of drives in the MmmOST environment: private, public and public only.

When a logical disk drive is designated as "private," the directory area on the disk is segmented into a number of smaller directories. Accessing the different directories, and the files on these directories, is done through a password. One user at a time may occupy any single private directory.

Public drives contain a single directory that can be accessed by multiple users simultaneously. Both read and write operations are allowed by a number of users at the same time.

A public only drive also contains a single directory that is accessible by all users. This directory and the included files can be read by multiple users simultaneously; however, only one user can write to the drive at one time.

3.3 TYPES OF DIRECTORIES

Different drives have different types of directories. Although there is no physical difference in the types of directories, there are logical differences. The way the drives are logically defined in the MmmOST environment specifies the various features for each directory.

3.3.1 Directory for a Private Drive

The directory on a private drive is divided into a set of smaller directories. These are called private directories. Private directories are defined by MmmOST (not CP/M).

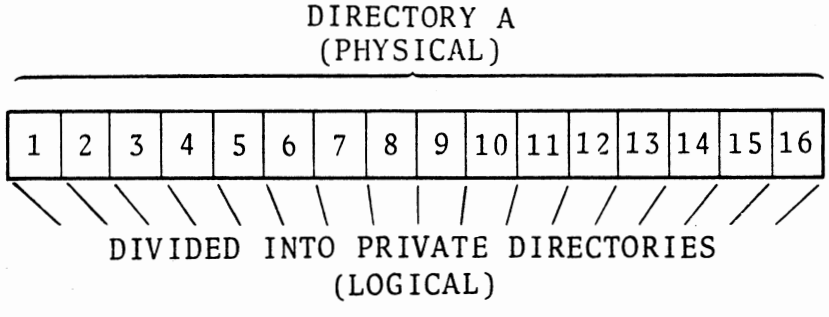


Figure 3-1 Private Directory Configuration

Only one user at a time can occupy a single private directory. Users can occupy different private directories simultaneously.

One private directory is automatically assigned to each user station port (6 for the TS 806, 16 for the TS 816). These assigned private directories are called **default directories.** They are accessed automatically when the user station boots up. The RS 422 port to which the user station is connected to determines which default directory is assigned to the user station.

The other type of private directory is the **password directory.** As the name implies, a password must be used to gain access to this type of private directory.

The files contained in these private directories are referred to as **private files.**

NOTE!

Refer to Chapter 5 for more information on private directories.

3.3.2 Directory for a Public Drive

MmmOST defines the directory on a public drive in the following way:

1. the directory appears identical to each user station
2. multiple users can read and write to this directory simultaneously
3. the system communication file, MULTI.SYS, resides on this drive and therefore is listed in this directory (see 3.4)
4. files listed on this directory can be defined as **shared files** or **unshared files.** Shared files can be updated (read, modified and written to) by several users concurrently. Unshared files cannot be updated by several users concurrently.

Note!

The file-sharing process can only occur on a public drive.

3.3.3 Directory for a Public Only Drive

MmmOST defines the directory on a public only drive in the following way:

1. one directory for all users

2. all users can read from the directory at the same time, but only one user can write to the disk at a time. In order for a second user to write to the disk, the first user must warm boot; this occurs when a CP/M applications progam terminates.

3. all files on public only drives are unshared

Table 3-1 summarizes drive, directory, and file information.

Table 3-1

Drives, Directories, and Files in the MmmOST Environment

Type of Drive	Directories Allowed on Drive	Files Allowed on Drive
Private	Default Password	Private
Public	Public	Shared Unshared
Public Only	Public	Unshared

3.4 SYSTEM COMMUNICATION FILE

The system communication file, MULTI.SYS, is a special file. Although disk space is reserved for the file, the contents of the file actually reside in the memory of the service processor. MULTI.SYS is a memory-resident file. Whenever a read or write to this file occurs, MmmOST intercepts the request and either reads or writes to the service processor memory.

Accessing MULTI.SYS allows a user or an applications program to communicate directly with MmmOST. Such communication with MmmOST is necessary when setting control modes (Chapter 4) and when reading and writing mailboxes or FIFOs (Chapter 6).

The format of the MULTI.SYS file is a series of 128-byte records. The first record of MULTI.SYS is called the status record. The structure of this record is as follows:

```
USERID:          XX  MESSAGEbCOUNT:     YYY    STATUS     Z

1                13  16                 36     40         48
```

Positions 49 through 126 contain blanks (20H).
Position 127 contains a carriage return (0DH).
Position 128 contains a line feed (0AH).

Control modes for shared file access are set by writing to the this record; refer to 4.5.

Other records in MULTI.SYS will contain 20H, except when (1) the record is used as an entry to a user function subroutine (2) the record is being used as a mailbox or (3) the mailboxes are being used as FIFOs. Mailboxes and FIFOs are discussed in Chapter 6, Communication Between User Stations.

4. RECORD AND FILE LOCKING

4.1 INTRODUCTION

MmmOST allows users to use shared files simultaneously through record and file locking without damaging data integrity. This chapter describes the control modes used to set locks, explains shared file access, and gives examples in different programming languages.

4.2 SHARED FILE ACCESS

Under certain circumstances, shared file access can damage data integrity. MmmOST, through various record and file locking techniques, does not allow data damage to occur; it prevents two common destructive simultaneous writes: Interleaved Updating and Fatal Embrace.

4.2.1 Interleaved updating--can best be illustrated by an example. Suppose Jim and Mary are both updating the restaurant inventory. Jim's program reads the sugar inventory and sees 10 pounds. Mary's program reads the same record and also sees 10 pounds. Jim adds 6 pounds purchased, writing out the total 10+6=16 pounds. Then Mary removes 3 pounds consumed and incorrectly writes back 10-3=7 pounds instead of the correct 10+6-3=13 pounds.

MmmOST prevents interleaved updating by denying either read or write access to a record when it is being written. The record is said to be locked. In the example above, the interleaved update is avoided by making Mary wait to read the sugar record until Jim has completed his update to 16 pounds.

4.2.2 Fatal embrace--is the condition which arises when Mary is waiting to read the sugar record from the inventory file which Jim has read and locked but not yet updated. At the same time, Jim is waiting to read the grocery record from the suppliers file which Mary has read and locked but not yet updated. Jim and Mary are thus in a situation in which neither is able to proceed because each is waiting for the other.

MmmOST prevents this from happening by requiring that files be accessed in the hierarchical order established when the program GENPARMS was run. GENPARMS is explained in Chapter 8. Requesting multiple record locks from files in other than this sequence causes automatic release of previously-set locks.

4.3 CONTROL MODES

There are seven distinct locking commands, or control modes, that are divided into two fundamental schemes of shared file access.

4.3.1 Two Fundamental Schemes of Shared-File Access

The two schemes for shared-file access are **MmmOST Control** and **Applications Program Control.** Under the MmmOST Control scheme, file access problems (read/write contention) are resolved by MmmOST. Under the Application Program Control scheme, file access problems must be resolved by the application program using status information supplied by MmmOST.

The four control modes used in conjunction with the MmmOST Control scheme are:

R/O	Read Only
R/W	Read record, Wait for lock
R/L	Read record, test for Lock
F/L	File Lock

These control modes are automatically and directly controlled by MmmOST.

The three control modes used in conjunction with the Applications Program Control scheme are:

N/L	Non-automatic Lock
LCK	LoCK
ULK	UnLocK

These control modes are not automatically controlled by MmmOST. Maintenance of data integrity is totally dependent upon the applications programs. MmmOST will only inform the applications program whether or not a record is locked; at that point, the program decides whether or not to write to that record.

Each of these control modes is described in the following two sections. Section 4.5 explains how to set a control mode.

4.4 Using Control Modes

Different applications programs accessing the same files should use either:

1. control modes from the MmmOST Control scheme or
2. control modes from the Applications Program Control scheme

Do not use both types of control mode schemes in different applications programs when they access the same files. You can use different control modes as long as they are in the same scheme. For example, User A runs program A on a file. User B runs program B on that same file. Programs A and B should use either the MmmOST control mode scheme or the applications control mode scheme. Do not mix the two control mode schemes in programs that access the same files.

4.4.1 MmmOST Control Modes - R/O, R/W, R/L, F/L

4.4.1.1 R/O (Read Only) Mode

a. This is the default mode. Warm/cold booting the service processor sets all shared files for each user station to this mode.

b. If this mode is set for a file by all user stations (LCK or ULK, applications programs control modes, may also be simultaneously set by any of those user stations), then the first user to write to the file will get exclusive write access to the file until that user warm/cold boots.

d. If this mode has been set for a file by any user station and the user station attempts to write to the file, write access will be denied and the following error message will be displayed:

BDOS ERR on [drive:] FILE/DRIVE R/O

e. This mode is retained when the user station is warm/cold booted.

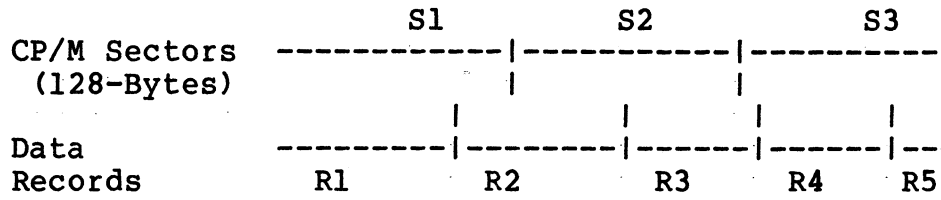
f. All programs should set the control mode to R/O before terminating--or the current control mode will remain in effect until the service processor is warm/cold booted.

4.4.1.2 R/W (Read, Wait for lock) Mode

a. If this mode is set for a file, then MmmOST will lock any data record that is read. If no lock exists for the record, the read request will be granted and the record will be automatically returned. If a lock already exists, the record will not be returned and the program will have to wait until the lock is removed. When the lock is removed, the request will be granted and the record will be returned.

b. Only one data record may be locked in a file by a program at the same time using this mode--with one exception; see item d below.

c. If you lock a record that spans more than one 128-byte CP/M sector, you will lock all 128-byte CP/M sectors in which the record lies. Suppose we have:



Locking R2 will lock S1 and S2 and locking R3 will lock S2 and S3.

Note!

The maximum number of CP/M sectors that can be locked at the same time is set in **GENPARMS** (see Chapter 8). The default maximum number is 4.

d. The Exception--Locking Consecutive Records in a File: If consecutive records in a file are read in sequence (say record 4, 5, 6 and 7) and with no intervening write operations, then the consecutive records will all be locked--assuming that the maximum number of CP/M sectors that can be locked simultaneously has not been exceeded. To retain the record locks, subsequent writing to these records must be done in the same order (4, 5, 6, and 7) or all locks will be lost.

e. Single records from different files may be simultaneously locked if they are read in the same order in which files were declared in the shared file list. To maintain locks, writing must be done in the reverse order. See the example below.

Example:

If the sequence of the files as listed in the shared file list is

1 SAMPLE.DOC

2 EXAMPLE.DOC

3 FINAL.DOC

The following read access sequences are permitted:

1,2
2,3
1,3
1,2,3

The following read access sequences are not allowed (assuming you are in R/W or R/L modes):

1. Reading from file 3 and then from file 2 will result in losing lock on file 3.

2. Reading from file 3 and then from 1 will result in losing the lock on file 3.

3. Reading from file 2 and then from 1 will result in losing the lock on file 2.

The following write access sequences are permitted:

```
3,2,1   (if read sequence was 1,2,3)
  2,1   (if read sequence was 1,2)
  3,2   (if read sequence was 2,3)
```

The following write access sequences are not permitted (assuming files 1, 2, and 3 were read in sequence):

1. Writing to file 2 before file 3 will result in losing the lock on file 3.

2. Writing to file 1 before writing to file 2 and file 3 will result in losing the lock on files 2 and 3.

f. If a program tries to write to a record which it did not first read, the write will not occur and the following CP/M error message will appear on the terminal display:

```
BDOS ERR ON [drive:] FILE/DRIVE R/O
```

The user station will automatically boot and return to the CP/M prompt.

g. In this mode, once a record is locked, it remains locked until:

- it is rewritten

- another record in the same file is read (subject to the exception of consecutive records discussed in item d above)

- another record in another file is read and the files are not in sequence (as described in item e above)

- the user station is warm/cold booted (assuming that the utility KEEPLOCK, as discussed below, has not been executed to enable locks to be maintained)

h. If a lock is active when the user station is warm/cold booted, that lock will be lost. If for some reason it is desired to retain such a lock, execute the following command prior to the warm/cold boot:

```
KEEPLOCK E
```

To restore the default behavior, execute:

KEEPLOCK D

4.4.1.3 R/L (Read, test for Lock) Mode

a. R/L mode operates in the same manner as R/W mode with this exception: here the record is returned (i.e., there is no wait), but the user must check the status character of the status record in the system communication file (character 48 of the first record in MULTI.SYS) to determine if the lock has actually been granted.

   If the lock has been granted, the status character will be S; if the lock has not been granted, it will be U.

b. This lock is lost when the user station is warm/cold booted.

c. If the lock is not granted, and a write is attempted, the message

   FILE/DRIVE R/O

   will appear, and the user program will be aborted.

4.4.1.4 F/L (File Lock) Mode

a. This mode locks the entire file. If other users attempt to read it, and have set R/L or R/W for this file, they will be unsuccessful. All users, regardless of the mode set, will fail if they attempt to write to a file which has been set in the F/L mode.

b. The extended portion of the file will be under the control of MmmOST and the new data can be read immediately by other users, even though the user station extending the file has not been warm/cold booted;

c. Other users can access it through

   (1) the various programming languages supported by CP/M

   (2) the CP/M Built-In TYPE command

   (3) the CP/M Transient Commands PIP, DDT, and DUMP

d. A lock set with F/L mode is lost when the user resets the control mode or the service processor is warm/cold booted. A lock set with F/L mode is not lost when the user station is warm/cold booted.

e. The following error message will be displayed when one user tries to write to a file which has been assigned F/L mode by another user:

BDOS ERR ON [drive:] FILE/DRIVE R/O

4.4.2 Control Modes with Applications Program Control

4.4.2.1 N/L (No Lock) Mode

a. Setting this mode for a file allows a user to update (read, modify and then write to) any record in the file.

b. It does not require a read before a write.

c. It allows files to be extended, with the new data immediately visible to other users.

d. If no locks are currently in effect on a file and a user requests N/L mode via writing to MULTI.SYS, the request will be granted and the status character (character 48 of the first record of MULTI.SYS) will be S.

e. It is released when the mode is reset or the service processor is warm/cold booted.

f. If a file is extended in N/L mode:

   a. the extended portion of the file wil be under the control of MmmOST and the new data can be read immediately, even though the user station extending the file has not been warm/cold booted;

   b. other users can access it through

      (1) the various programming languages supported by CP/M

      (2) the CP/M Built-In TYPE command

      (3) the CP/M Transient Commands PIP, DDT, and DUMP

g. A user who only plans to read a file should set the R/O mode for that file. A user who plans to lock/unlock a file in a shared access manner should set N/L (and use the LCK/ULK mechanism to perform the locking/unlocking).

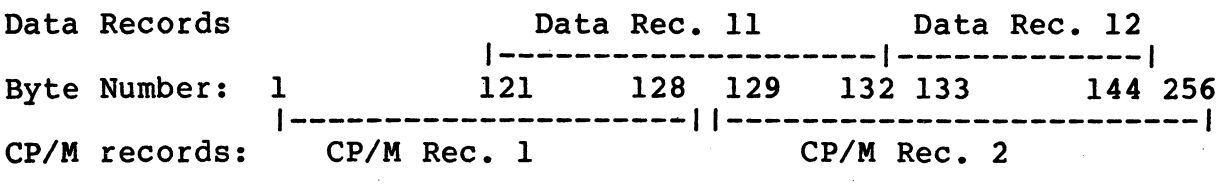
4.4.2.2 LCK (Lock), ULK (UnLocK) Modes

a. LCK and ULK, used with applications programs in a particular way, avoid the occurrence of interleaved updating and fatal embrace. LCK and ULK are used to maintain a list of locked resources in the service processor. LCK causes the "resource" to be placed on the list. ULK causes the resource to be removed.

b. Neither mode requires a read; they are effective when set. To determine if the mode has been set, the status character of the status record should be read.

c. With LCK, the user can simultaneously lock any number of records in any order (even if those records are in the same file) and maintain those locks.

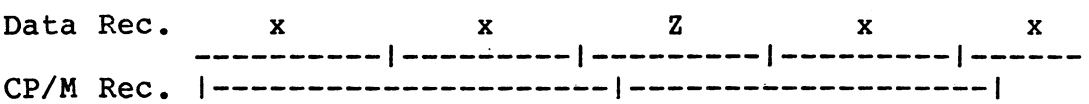
d. The (rn) in the command syntax for setting a mode (see 4.5) is optional, but if it is present, it refers to a record number within the file. Whether this number is a data record (at the applications level) or a CP/M physical record relative to the beginning of the file is a convention to be established by the applications program. But remember that CP/M will always read and write complete 128-byte records/sectors. For example, suppose your data records are 12 bytes in length. Comparing data records to CP/M records, we would have:



Here, because of the record length and the overlapping of records, we have the potential problem of interleaved data. For example, if data records 11 and 12 are read simultaneously by different applications programs, and 11 is written back, followed by 12, then the last five bytes of record 11 would be overwritten by record 12, since the entire CP/M record 2 would be written back.

Thus, if only (rn) is specified, data records should always be integer multiples of 128-byte CP/M record sectors.

If it is necessary to use records other than integer multiples of 128 bytes, then the (rn,rl) form of the command should be used. If (rn,rl) is specified for a record, MmmOST determines which surrounding records, if read/written, could cause contention problems (interleaved updating). MmmOST places them, along with the "locked" record, on the list of locked resources. For example, suppose the record Z in the next illustration is "locked".



Then the records marked x would also be "locked" (placed on the list of locked resources).

e. It is important to note that MmmOST does not enforce these types of locks; it does not prevent writing to records of

files locked with LCK. Therefore, to prevent contention problems, the applications program must read status to determine if its request for a lock has been granted.

f. ULK is used to release a lock obtained with LCK.

g. As mentioned earlier (see explanation under the N/L mode description), N/L mode is normally set by user who intends to lock/unlock a file in a shared access manner using the LCK/ULK facility.

4.5 Setting Control Modes

Control modes are set by writing to the first logical record of the system communication file MULTI.SYS. To write to the first logical record, enter

d:filename.typ(x) = control mode

where

d: = the logical drive where the file resides

filename.typ = the unambiguous file name of the file to be accessed

x = rn or rn,rl; valid only for LCK and ULK modes

rn = the record number to be locked or unlocked; valid only for LCK and ULK modes

rl = the record length (in bytes) of the record to be locked or unlocked; valid only for LCK and ULK modes

control mode = R/O, R/W, R/L, F/L, N/L, LCK, or ULK

Notes:

1. No more than one blank may appear between adjacent fields in the command message.

2. If required by the applications program language, single or double quotation marks may be placed around the command message.

3. "d:" and "(x)" are optional in the command message.

   a. If d: is not entered, the currently logged drive will be assumed.

   b. (x) may be omitted for any control mode (in which case the whole file is then specified).

c. (x) is never specified for R/O, R/L, R/W, F/L, or N/L modes (i.e., the whole file is specified).

d. (x) = (rn) can be specified for LCK and ULK modes only.

e. (x) = (rn,rl) can be specified for LCK and ULK modes only.

4. It is possible to set control modes for nonexistent files.

4.6 ACCESS DURING FILE/DIRECTORY MODIFICATION

With the exception of updating, all of the following file operations modify the directory of the file:

updating

extending

creating

renaming

erasing

The following sections discuss access to unshared files and directories during the above operations.

4.6.1 Unshared File on Public Drive

If a program

updates, extends, or creates

an unshared file on a public drive, then the program has exclusive rights to the directory of that file until it warm boots the user station (i.e., a second program cannot update, extend, create, rename, or erase the file until the first program warm boots).

If a second program attempts to update or extend the file while the first program has exclusive right to it, then the following error message will be displayed:

BDOS ERR ON [drive:] BAD SECTOR

NOTE:

The user station does not automatically boot if this occurs.

If a second program attempts to

rename or erase

the file while the first program has exclusive rights to it, then the following error message will be displayed:

BDOS ERR ON [drive:] FILE/DRIVE R/O

Note!

The user station will be automatically booted if this occurs.

4.6.2 (Unshared) File on Public Only Drive

If a program

updates, extends, creates, renames, or erases

a file on a public-only drive (hence an unshared file), then the program has exclusive rights to the entire drive until the first program warm boots the user station (i.e., a second program cannot update, extend, create, rename, or erase any file on the drive until the first program warm boots.)

If any other program attempts to update, extend, create, rename, or erase any file on the drive before the first program warm boots, the following error message will be displayed:

```
BIOS WRITE ERROR ON DRV=[DRIVE:] TRK=XXXX; SCTR=YYYY;WRT
TYPE=ZZ;RTN CODE=AA
WRITE PROTECT ERROR
BDOS ERR ON [DRIVE:] BAD SECTOR
```

4.7 OPERATION WITH HIGHER-LEVEL PROGRAMMING LANGUAGES

4.7.1 Operation with CBASIC

All control modes described in 4.4 may be used with CBASIC. Setting a control mode is accomplished by first opening the file MULTI.SYS and then writing the message

d:filename.typ(x)=control mode

For example, suppose we have a file called INVNTRY.DAT and we want to set the control mode for the entire file as R/W. Suppose also that our system communication file, MULTI.SYS, is located on drive B. We first need to open the file MULTI.SYS on drive B as file N with a record length of 128-bytes. (Let us assume N=1.) We can do this by entering

OPEN "B:MULTI-SYS" RECL 128 AS 1

Now that MULTI.SYS is open, we can write to it, setting the control mode for any file desired. The following statement sets the control mode for INVNTRY.DAT as R/W:

```
PRINT #1,1; "B:INVNTRY.DAT=R/W"
```

We can then read a record, say record 10, with the following statement (assume that INVNTRY.DAT has already been opened as file 2):

```
READ #2,10;  ITEM$QUANTITY, COST

             |  |   |----------------|
             |  |   data fields in file
             |  |
             |  |---record number
             |
             |------file number
```

At this point, the record is actually read and locked, and other users in R/W or R/L mode will not be able to read or write to it until it is rewritten. Also, users not in R/L or R/W mode are able to read the file at any time.

Suppose again we want to read record 10 in INVNTRY.DAT, but this time in R/L mode. Our PRINT statement would then say:

```
PRINT #1,1;  "B:INVNTRY.DAT=R/L"
```

and our READ statement would be the same as the one in the previous example. After the READ statement, however, we need to check the status character (character 48 of the first record of MULTI.SYS) to determine if our READ was successful (i.e., if we successfully locked the record). We can do this by entering

```
READ #1,1; STAT$
LET STATUS$ = MID(STAT$,48,1)
```

The variable STATUS$ will then contain either an S or a U. If it contains an S, then the record was successfully locked and our data is good; we can proceed.

If it contains a U, however, we must again read the record and status until valid data is read (as determined by the status).

Whenever you set the control modes R/L, F/L, N/L, and LCK, you must read the status character to determine if the mode was successfully set.

4.7.2 Operation with TeleVideo COBOL

TeleVideo COBOL provides user transparent record and file locking using three control modes: N/L, LCK, and ULK.

When using CBASIC with these control modes, a user's program must keep track of whether a file/record is actually "locked" or not, meaning before it writes to a record, it must read the status character in the status record to determine if the record/file has actually been placed on the list of lock resources. With TeleVideo COBOL, this is unnecessary; TeleVideo COBOL does this checking for the user, so that a record is not written before it is locked.

As far as the user is concerned, mode setting, and hence record/file locking and unlocking, is accomplished via OPEN and CLOSE statements and warm booting. For a file X, Table 4-1 shows how the N/L, LCK, and ULK modes may be set.

Note that when a record that is not an integer multiple of 128 bytes is locked, surrounding records that could cause contention problems (interleaved updating) are locked as well. For more information, refer to the discussion in 4.4.2.2 under LCK (LoCK), ULK (UnLocK), item d.

Table 4-1

Mode Setting with TeleVideo COBOL

Cobol Statement	Description
OPEN INPUT X	For reading File X only--sets N/L mode for entire file.
OPEN OUTPUT X	For writing File X only--using LCK mode, locks the entire file as soon as it is available after the OPEN statement occurs.
OPEN I/O X WITH LOCK	For reading and rewriting records in file X--using LCK mode, locks the entire file as soon as it is available after the OPEN statement occurs.
OPEN I/O X	For reading and rewriting records in file X--using LCK mode, locks a record as soon as it is available after a READ statement occurs; using ULK mode, automatically unlocks the record after the REWRITE operation is complete.
CLOSE X	To terminate processing of file X--if a file has been locked using the OPEN OUTPUT X or OPEN I/O with LOCK command, this statement unlocks the file using the ULK mode.

5. DRIVES, DIRECTORIES, AVAILABLE SPACE, AND FILE STATUS

5.1 INTRODUCTION

This chapter discusses logical drives, logical drive configurations, directories, and access to directories. It also discusses two utilities (DSTAT and STAT) that allow the user to determine the amount of space that a file occupies, the amount of space left on a drive, and the status of files on drives.

5.2 LOGICAL DRIVES AND LOGICAL DRIVE CONFIGURATIONS

The assignment of letter names (A, B, C, etc.) to physical disk space is called logical-to-physical disk mapping.

"Physical disk space" may be an entire disk, or it may be a subdivision of a disk--depending on how the disk has been formatted. For instance, the TeleVideo TS 806 contains two physical disk drives: 5 1/4-inch floppy disk drive and a 5 1/4-inch Winchester disk drive. The floppy drive is treated as a single drive, while the Winchester disk is subdivided, and treated as two hard disk drives.

TeleVideo systems come with logical disk drive names pre-assigned to physical disk space. For instance, the logical name for the large Winchester hard disk (subdivision) is A; and the logical name for the floppy disk is C.

Note!

TS 806 logical drive designations are dependent upon the source of the boot; the two sources are the hard disk or the system floppy diskette. The designations listed below assume the TS 806 is booted from the hard disk.

Under MmmOST, logical drives are of three types: private, public, and public only (see 3.2). The default type-of-drive assignments for a TS 806 service processor are:

Logical drive A (first portion of the drive)--Private

> Drive A is designated as a private drive through the program GENPARMS. Included on this drive are MmmOST and the system programs used by MmmOST to control the system.

Logical drive B (second portion of the drive)--Public

> Drive B is designated as a public drive through GENPARMS. The print spool file, which must be located on a public drive, is located on drive B.

Logical drive C (floppy drive)--Public Only

Drive C is designated as a public only drive through GENPARMS. All three types of drives are represented on the TS 806.

When you receive your system, you can change the default drive categories (private, public, or public only) in your system by running GENPARMS. Refer to Chapter 8, Reconfiguring MmmOST, for a description of GENPARMS.

Note!

It is recommended that drive A always remain a private drive in order to properly use the CP/M SUBMIT command. To use the CP/M SUBMIT command, you must be logged onto drive A. If drive A is not private and multiple users want to use SUBMIT, you must use the isolation capability provided by the CP/M user number. To avoid this, leave drive A as private.

Table 5-1 shows the logical-to-physical drive assignments, as well as the default type-of-drive assignments, for the TS 806 service processor.

Table 5-1

TS 806 Standard Drive Configuration*

Logical Drive Name	Physical Drive	Type of Drive
A	5 1/4-inch Winchester hard disk drive--first portion (subdivision)	Private
B	5 1/4-inch Winchester hard disk drive--second portion (subdivision)	Public
C	5 1/4-inch floppy disk	Public Only
D, E	Reserved	
F, G, H, I, J, K, L	Not used	
M, N, O, P	Local user station disk drives	Completely Private to User Station

*Drive designations valid when TS 806 is booted from the hard disk.

5.3 DIRECTORIES

Some of the space on logical drives is taken up by directories. Directories provide information about files stored on drives.

In the MmmOST environment there are two types of directories: directories for private drives and directories for public and public only drives.

5.3.1 Directories for Public and Public Only Drives

Public directories and public only directories contain information, respectively, about files on public and public only drives. All users have access to these directories/files--subject to the rules discussed in 4.4, 4.5, and 4.6.

Access to public and public only directories is through standard CP/M commands--special commands are not required.

5.3.2 Directories for Private Drives

Private directories are located on private drives and contain information about private files. In terms of access, there are three types of private directories:

default directories with no assigned passwords

default directories with assigned passwords

password directories

A default directory with no assigned password can be accessed only from the user station plugged into the proper channel on the service processor. For every user station and for every private drive, there is a default directory.

A default directory with an assigned password may be accessed by another user station, via the password, when the user station containing the desired password directory is powered on and logged onto another directory.

A password directory may be accessed only via the password.

Note that:

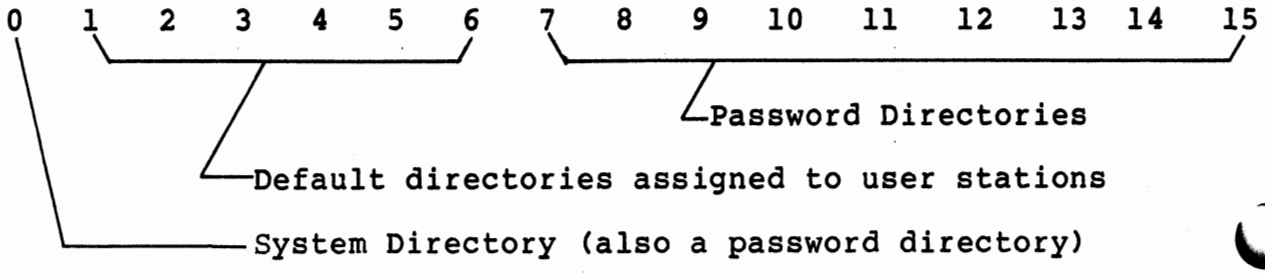
1. a user may access only one private directory per drive at a time

2. the number of private directories and the number of entries (files) per private directory are variables set in GENPARMS. For a discussion of these variables, see private drives under 8.3.2 (item 1--Private Drives).

The discussion below assumes the default values for the number of private directories to be: TS 806--16; TS 816--32.

5.3.2.1 Directories for the TS 806 Private Drive A

Directories in a TS 806 are currently designated 0 through 15 (drive A). System programs are always listed in private directory 0. Directories 1 through 6 are automatically assigned to the user stations whenever the service processor is warm/cold booted. They can also be assigned a password (see section 5.3.1.3). This makes it possible for a private directory to be accessed from the assigned user station or by a password from any user station should the assigned user station be in use on another password directory. A directory which is assigned to a specific user station without an associated password can only be displayed when a user requests that directory from the assigned user station. Password directories can be accessed only by a password.

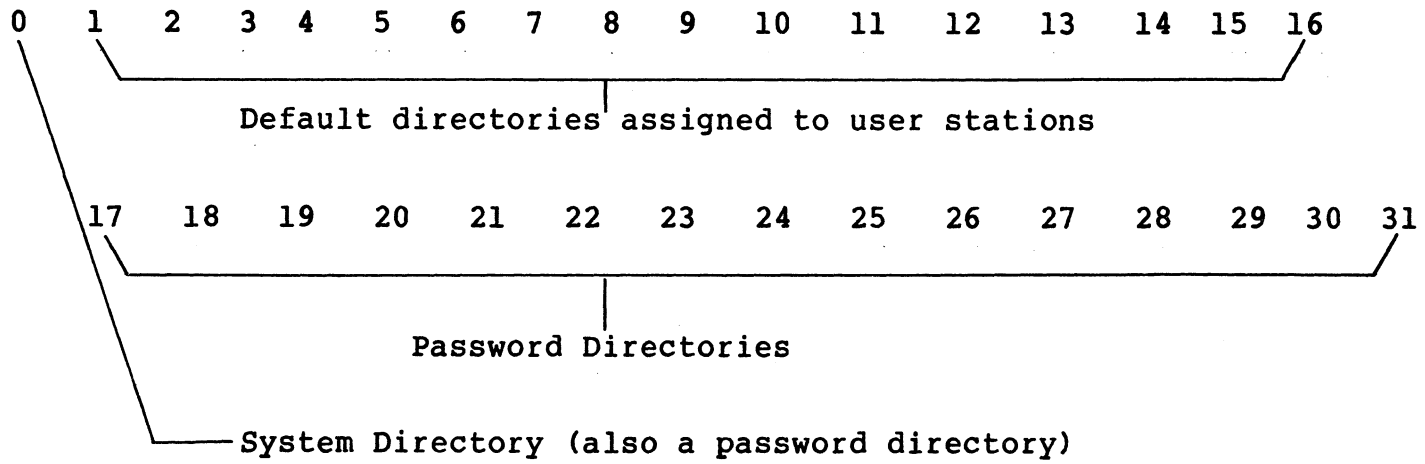


Note!

Remember not to store confidential information on default directories.

5.3.2.2 Directories for the TS 816 Private Drive A

Private directories in a TS 816 are currently designed 0 through 31 (drive A). System programs are always listed in private directory 0. Directories 1 through 16 are automatically assigned to the user stations; these directories can also be assigned passwords. This makes it possible for a private directory to be accessed from the assigned user station or by a password from any user station should the assigned user station be in use on a password directory. A directory which is assigned to a specific user station without an associated password can only be displayed when a user requests that directory from the assigned user station. Password directories can be accessed only by a password.



Note!

Remember not to share confidential information on default directories.

5.3.2.3 Password Designations

To add or change any passwords, use the program GENPARMS (see Chapter 8). Although your system arrives with the private directories assigned to the user stations, no passwords have been assigned to the default directories. The same password can access one or more directories if there are multiple private drives in the system.

5.3.2.4 Accessing Default Private Directories Without Passwords

If a default private directory (the directory automatically assigned to a user station) has not been assigned a password, then it can only be accessed by the user station plugged into the appropriate channel on the service processor. Running the LOGON program as described below is not required to access default directories without passwords.

5.3.2.5 Accessing Private Directories with Passwords

The program LOGON allows you to access private directories with passwords and to move from a password directory back to your default directory. To run LOGON, enter one of the following commands at the user station:

[drive 1:] LOGON [drive 2:] [password]<CR>

[drive 1:] LOGON [drive 2: or *:]<CR>

where

drive 1 = the drive where the LOGON program is stored. LOGON can be stored on any

drive, public or private. If no drive is specified here, CP/M will search for the LOGON program on the currently logged drive.

drive 2 = the drive on which the directory or file you want to use is stored. If you enter both drive 2 and a password, you will have access to drive 2 only if your password is valid for a directory on that drive. If you do not specify drive 2, you will have access to directories and files in private directories on all private drives for which the password is valid. (Refer to Examples 6 and 7.)

If the second form of the command is used (drive or asterisk, but no password is entered), you will be prompted with an asterisk (*) for your password. In this case, the password will not be displayed on the terminal as you type it. Rubouts are not allowed. This method allows you to keep your password secret from bystanders.

5.3.2.6 Access to Private Directories--Examples Using LOGON

Table 5-2 shows how passwords could be assigned to directories in a system which has been reconfigured with drives A and B both private drives (and hence containing only private directories).

Table 5-2

Example of a Directory Configuration

Drive	Directory No.	Password
A	0	John*
	1	Judy
	2	David
	3	Mary
	12	Sue
B	0	John*
	1	----
	2	Payroll
	3	Invntory
	4	Rceivbls

Note:

The password "John" accesses directory 0 on drives A and B.

The following examples show how to use LOGON. They assume that your directory assignment is as shown in Table 5-2, and LOGON is stored only on the default (currently logged) drive. User input is indicated in bold print, while system response is shown in normal print. Note that drive 2 is specified in examples 1 through 4, and that it is not specified for examples 5 and 6.

Example 1: Correct Entry

If you enter

LOGON A:JOHN<CR>

from user station 1, you will see the message

End of Execution

displayed on the screen, and you will be able to access directory 0 on drive A and all files on drive A listed for that directory. Since you specified drive A in your command, LOGON will only search drive A for a directory which has the password John. It will **not** search drive B even though drive B also contains a directory with the same password.

If you then enter

DIR A:<CR>

information about directory 0 will be displayed.

Example 2: Incorrect Password/Drive Association

If you enter

LOGON B:JUDY<CR>

then

Unsuccessful
End Execution

will be displayed on the screen. You will see the message Unsuccessful on the screen because password JUDY is not assigned to drive B.

Example 3: Invalid Password

If you enter

LOGON B:TOM<CR>

then

Unsuccessful
End of Execution

will be displayed on the screen. You will see the message **Unsuccessful** because you used an invalid password.

Example 4: Directory Already in Use

If you enter the following message while someone else is using directory 2 on drive B

LOGON B:PAYROLL<CR>

then

Unsuccessful
End of Execution

will be displayed on the screen and you will not have access to directory 2.

Example 5: Invalid Password

If you enter

LOGON TOM<CR>

then

Unsuccessful
End of Execution

will be displayed on the screen. You will be unsuccessful because you used an invalid password. (Here, MmmOST will search for TOM on both private drives A and B, since a drive was not specified, but will find it on neither.)

Note!

The same message ("unsuccessful") is displayed if a password is invalid or if the directory is already in use.

Example 6: Directory Security at the Individual User Station

If you enter

LOGON JOHN<CR>

this message will be displayed (assuming drives A and B are private and JOHN is a password for both drives):

A B drive(s) logged to password account

End of Execution

You will be able to access directory 0 on both drive A and B.

If you are logged onto drive A, enter

DIR<CR>

to see the directory of JOHN on that drive.

To see the directory of JOHN on drive B, enter

DIR B:<CR>

Whenever (1) you do not specify a drive in the LOGON command, (2) the password entered is valid for all private drives, and (3) the execution is successful, you will have access to all directories on all drives for which that password is valid. No other user can gain access to any of those directories until you enter the command in Example 7.

Example 7: Leaving Password Directories

When you are finished using the files in a password directory, enter

LOGON<CR>

to release your control of the password directory and return to your default directory. Another user can now access the password directory you were previously working in providing he knows the password.

When you leave your user station, always use this command to return to your default directory. This will prevent any other user who works at your user station from seeing the files on the password directory you previously accessed.

For example, if you do not enter LOGON<CR> after accessing password directory John for both drives A and B (Example 6), and a second user working at your user station now enters

LOGON MARY<CR>

The system will respond

End of Execution

When the second user enters

DIR<CR>

he will access directory 3 on drive A (MARY) or directory 0 on drive B (JOHN)--depending on which drive he is currently logged onto. Even though the second user's entry only entitles him to directory 3 (MARY) on drive A, he has access to password directory JOHN on drive B because you did not leave this directory (by entering LOGON<CR>). JOHN on drive A was not accessed because MARY on drive A was.

A user at a user station can be logged onto one and only one private directory for each private drive in a system. Thus here, when the second user does not specify a drive and logs onto MARY, access to JOHN on drive A is lost because access to MARY on drive A was gained. But, since there is no MARY on drive B, access to JOHN on drive B is retained.

5.4 DETERMINING AVAILABLE SPACE AND FILE STATUS

Through the commands DSTAT and STAT, a great deal of information can be obtained about available space on drives and file status.

5.4.1 DSTAT

For private drives, DSTAT states the amount of available free space in kilobytes. For all open files and all shared files on public drives, DSTAT provides information in the following format:

File Name	File Type	Access Type 1	Access Type 2	Access Mode 1	Access Mode 2

where

File Name = the CP/M file name.

File Type can be any of the following:

Exclusive	unshared file on a public drive
Shared	shared file on a public drive
Communications	system communication file maintained on a public drive

Access Type 1 & 2 can be any of the following:

Owned by X	X is the user station that owns the file
Opened by X	X is the user station that has opened the file
(null)	the file is not currently being accessed

Access Mode 1 & 2 can be any of the following:

Locked by X	user station X is in the F/L control mode
No Lock for X	user station X is in the N/L control mode
Update for X	user station X is in either the R/W or R/L control mode

Locks exist for X	user station X is in either the R/W or R/L control mode and is actually updating a record at the time that DSTAT is executed
(null)	none of the following control modes is being used on the file by any user station: R/W, R/L, F/L, N/L

Access modes Update for X and Locks Exist for X will both be displayed when an actual file or record is locked in R/W or R/L mode. For example, if user stations C and F are both updating the file ZIPCODE.DAT using R/W or R/L control modes, and C actually has a record locked at the time DSTAT is executed, the following entry would be displayed:

```
ZIPCODE.DAT  SHARED  OPENED BY CF  UPDATE FOR CF  LOCKS EXIST FOR C
```

If a public only drive is owned by a particular user station, DSTAT states "OWNED BY X" where X is the user station that owns the drive.

Suppose

(1) DSTAT is run on a TS 806 with the following drive assignments:

A: Private Drive

B: Public Drive

C: Public Only Drive

(2) user D is running DSTAT and has opened MULTI.SYS

(3) users C and F are both updating the file ZIPCODE.DAT (as in the previous example)

(4) user C actually has a record locked

(5) user E has gained exclusive rights to public only drive C

The following might then be displayed by running DSTAT:

```
A:  private - free space    3300 kbytes

B:  shared

    DSTAT    COM  exclusive  opened by D
    FLIGHT   RSV  shared
    AGENT    ACT  shared
    MULTI    SYS  communications  opened by D
    ZIPCODE  DAT  shared  opened by CF  update for CF  locks
                  exist for C

C:  public - owned by E
```

5.4.2 STAT

For information about the space occupied by a file or the amount of remaining space on a public or public only drive, use the command STAT. (For more information on STAT, see the CP/M user manual--An Introduction to CP/M Features and Facilities.)

6. COMMUNICATION BETWEEN USER STATIONS

6.1 INTRODUCTION

User stations may pass messages to each other through mailboxes, which are accessed by simulating reads and writes to MULTI.SYS (described in 3.4.1, System Communication File).

For a particular user station, the first record (status record) in MULTI.SYS has two functions:

1. It keeps track of the status of the last MULTI.SYS control mode request by the user station.

2. It counts the total messages received by that user station via its mailbox.

The format of that record (reproduced here) is:

```
USERID:bbbbbbnnbMESSAGEbCOUNT:bbbbbmmmbSTATUS:bx

(nn=USERID:, b=blank, mmm=MESSAGE COUNT:, x=STATUS)
```

Note that the message count is characters 36, 37, and 38 of this record.

Each remaining record in MULTI.SYS is a 128-byte mailbox associated with a particular user station. Record 2 is the mailbox for user station 1; record 3 is for user station 2; etc. The number of mailboxes available for this purpose is set in GENPARMS (see Chapter 8).

Data is passed between user stations through these 128-byte mailboxes. These records are read or written as if an actual file access were taking place. Any language that is available under CP/M and that supports file access can read and write mailboxes.

6.2 SENDING MESSAGES

To send a message, the sending user station "writes" to the record in the MULTI.SYS file which corresponds to the destination user station. Each subsequent message written from one user station to another writes over the preceeding message.

6.3 READING MESSAGES

The following CBASIC example illustrates how to read both the status record and the mailbox records.

To read the message count from the status record (in MULTI.SYS), enter

```
READ #1,1; STAT$
LET COUNT=MID$(STAT$,37,2)
```

To read the contents of the mailbox for User 1, enter

```
READ #1,2;MSG$
```

To receive a message, a user station must read the 128-byte record in MULTI.SYS that corresponds to the user station (record 2 for user station 1, record 3 for user station 2, and so on). Note that only one message is stored for each receiving user station and that each subsequent message writes over the preceding message. Thus the only way a user will know if a message has been missed is if the program compares the current message count to the previous message count. If the difference is greater than one, a message has been missed.

To avoid missing messages, each user station should read the message count frequently. If the message count has increased, the mailbox should be read before another is sent and the first message is lost.

Note!

Not all applications programs will find the mailbox feature useful. There is no requirement to use it.

6.4 INTERPROCESSOR COMMUNICATIONS USING FIFOS

MmmOST Version 2.1 provides an additional method for users to send messages to one another. This method is known as a FIFO, or a First In, First Out mechanism. In this method, multiple messages written to the FIFO mechanism do not overwrite the last written data; all messages are placed into a long "pipeline." The receiver of the data will read the data out of the FIFO in the same order that it was written, with no loss of data. Since this data is temporarily stored in the service processor memory, read/write access to a FIFO is rapid.

For interprocessor communication, you can select using FIFOs instead of mailboxes for any one user station at any one time. Note that the use of mailboxes is the default means to interprocessor communication. A command must be entered for FIFO usage. If FIFOs are chosen to be used by one user station, other user stations can use either FIFOs or mailboxes.

There is an input and output FIFO (also known as queues) for each user. Users can choose to read/write their own FIFO pair (input/output) or any other user's FIFO. FIFOs can be grouped together under one name (referred to as a **FIFO set**) and be used by different user stations communicating with each other.

A single user station can only communicate with user stations in the same FIFO set unless the user station deselects the FIFO set being used and reselects another one. Each named FIFO set is organized as 32 separate FIFOs of arbitrary length storing records of 128 bytes.

By entering the command below to the first record of MULTI.SYS, you

(1) select a set of FIFOs by name

(2) deselect the mailboxes by setting the mode to QUE

The command is

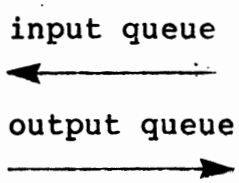
```
<NAME> = QUE
```

where <NAME> is an arbitrary string up to 8 characters and is the name of the FIFO set. All users selecting the same <NAME> will use the same FIFO set. There may be multiple FIFO sets active in the system simultaneously.

In the QUE mode, records 1, 2 and 3 of MULTI.SYS have special meanings. Record 1 is a status/control record; by writing to this record, you select the FIFO (input/output pair) that you want to communicate with. To select the input/output pair associated with your user number, write a binary zero in the first byte of record 1 (see Example 1). To select any other user's FIFO, write that user's number into the first byte of record 1 (see Example 2). This will set that user's output queue as your input and that user's input queue as your output.

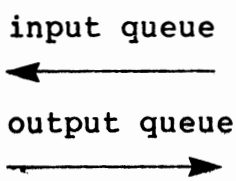
Example 1:

User 1



Unconnected Queues

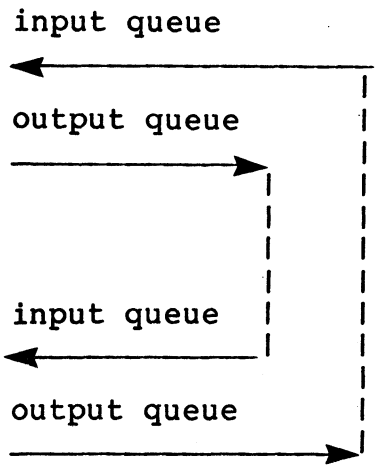
User 2



In the example above, Users 1 & 2 have opened a FIFO by writing "QUEUE1=QUE" to the first record of MULTI.SYS.

Example 2:

User 1



Connected Queues Between User Stations

In this example, User 1 set byte 1 of record 2 of MULTI.SYS to 2 (binary).

If the FIFO is empty when the user reads from it, the message

END OF MESSAGE

will be returned.

Reading record 1 of MULTI.SYS in this mode shows the number of records in each queue. Bytes 1 to 16 show the users' output queue count and 17 to 32 show their input queue count (all in binary).

Reading/writing record 2 reads/writes to the selected queue.

If the FIFO is full (signifying there is no more space in the service processor memory) when the user writes to it, the new data will be lost.

FIFO data is not deleted from service processor memory when the user stations boot. Data is only removed from the FIFO when it is read or when the service processor is warm/cold booted.

To restore access to normal mailboxes or user station functions, enter the command

x = MAIL

where x is any string.

7. PRINTER OUTPUT AND CONTROL

7.1 INTRODUCTION

To print, you can use a printer connected directly to a user station or a printer connected to the service processor (shared by all user stations). If you want to print with a printer connected directly to your user station, start the print procedure with the standard CP/M commands. The local printer is accessed as UL1:.

This chapter refers only to the use of shared printers with the exception of the information about printer selection.

7.2 PRINT SPOOLING

Data directed to a print device on a service processor from a user station is copied by MmmOST to a temporary disk file before being printed. The process, called **spooling**, allows the printer to be shared among users, and returns control of the user station to the operator while the printing is supervised in the background by the service processor.

The temporary file created to store data before it is printed is called the **print spool file**. Each printer can have many print spool files. Because there is a separate print queue for each system printer, both the serial and the parallel printer can run concurrently at the service processor. These printers are available to all users on a first-come, first-served basis.

7.2.1 Printer Selection

Printers are selected by setting the logical CP/M list device (LST:) to the desired physical list device (TTY:, CRT:, LPT:, or UL1:). The physical list devices are:

TTY: - Serial printer on TS 806 or TS 816 (first serial printer if TS 816)

CRT: - Second serial printer on TS 816 (TS 806 has only one serial printer)

LPT: - Parallel printer on TS 806 or TS 816

UL1: - Local printer (at user station)

This selection can be done in one of two ways: via the STAT command or the IOBYTE.

To select the desired printer via the STAT command, set LST: as shown below:

DESIRED PRINTER	REQUIRED STAT COMMAND
Serial printer (TTY:) of TS 806 or TS 816 (first serial printer if TS 816)	STAT LST: = TTY:
Second serial printer (CRT:) of TS 816	STAT LST: = CRT:
System parallel printer (CLT:)	STAT LST: = LPT:
Local printer (ULl:)	STAT LST: = ULl:

To select the desired printer via the IOBYTE, set the IOBYTE value (whose location in memory is 03H) as shown below:

DESIRED PRINTER	IOBYTE VALUE (HEX)	IOBYTE INDEX (HEX)
System serial printer (TTY:) of TS 806 or TS 816 (lst serial printer if TS 816)	00H	0
System serial printer (CRT:) of TS 816	40H	1
System parallel printer (LPT:)	80H	2
Local user station printer (ULl:)	C0H	3

7.2.2 Print Spool File

The temporary print spool file is stored only on a public drive. The spooling process creates a named disk file. This file is visible while it is being spooled or printed. The file name created will be:

SPLuqnnc.PRt

where

u = the alphabetic indentification (processor I.D.) of the user station. See section 2.4.1 for an example of a sign-on message.

q = the queue name -- 0, 1, 2, or 3, depending on the IOBYTE index or some other alphanumeric value set by the utility program SELQUEUE (see 7.3.2)

nn = the print priority assigned by MmmOST to the file (range is 0 to 99; 0 is highest priority, 99 is lowest).

c = the number of additional printed copies desired.

t = the type of printing where:

N = normal printing (no print form control);

H = the standard forms message followed by a pause (for paper change);

P = a pause before printing the spooled file (the PRNT M buffer is filled with 128 nulls, which are sent to the printer). Refer to 6.3.1.

If the designated system printer is available, the current contents of its print spool file will be sent to the printer whenever any of the following conditions occur:

1. An applications program or an operator warm or cold boots the user station.

2. An even-numbered (2nd, 4th, 6th, etc.) occurrence of the CP/M ^P command (as described above) is entered at the user station.

3. An applications program calls the printer breakpoint routine. This routine is described in 7.2.4.

7.2.3 Sending Data to the Print Spool File

Data is sent to the print spool file whenever any of the following conditions occur:

1. Any program, including the CP/M PIP program, sends data to the CP/M LST: device.

2. A CONTROL/P (shown hereafter as ^P) is entered at the user station. Since ^P is a toggle control function, the first ^P will begin the transfer of data to the print spool file.

The **second** ^P will end the transfer of data to the print spool file and initiate printing (despooling). The third ^P will turn on transfer of data to the print spool file again, and so on.

7.2.4 Printer Breakpoint Routine

The printer breakpoint routine allows an applications program which has generated a report to start printing as soon as possible without stopping the program to warm boot. Using this routine also minimizes the possibility of running out of space on the print spool drive. To call the routine, which resides in each user station, you must first compute its address as follows:

1. Obtain the 16-bit address whose low order byte is stored in 1B (Hexadecimal) and whose high order byte is stored in 1C (Hexadecimal). The CBASIC code would be as follows:

```
LET M%=PEEK(1BH)
LET M1%=PEEK(1CH)
```

2. To obtain the 16-bit address, multiply the higher order byte by 256, add the low order byte, and add 4 to the address. Then call the breakpoint routine at that address.

```
LET INDEX%=(256*M1%+M%)

LET INDEX%=INDEX%+4

CALL INDEX%
```

3. The printer breakpoint routine in MBASIC would be as follows:

```
10 M%=PEEK(&H1B)
20 M1%=PEEK(&H1C)
30 INDEX=(256*M1%+M%)
40 INDEX=INDEX+4
50 INDEX=INDEX-65535!
60 INDEX%=CINT(INDEX)
70 CALL INDEX%
```

4. The printer breakpoint routine in 8080 assembly language would be as follows:

```
LHLD    1BH
LXI     D,4
DAD     D
PCHL
```

7.3 PRINT CONTROL AND QUEUE SELECTION

7.3.1 Print Control and Print Control Commands

The operator at a user station can issue certain commands to control printing while the printer is operating. The commands

will apply to the printer which the user has currently selected.

Note!

These commands can be given from any user station, and the appropriate action will occur **REGARDLESS** of which user station initiated the current printout.

To initiate any of these print controls, the operator enters the following command:

PRNT <command> <CR>

where

command is one of the letters listed in Table 7-1. If an invalid letter or no letter is entered, a listing of the valid commands will be displayed at the user station.

Table 7-1

Print Control Commands

Command	Effect
T	Restarts printing at the Top of the print file.
F	Restarts printing at the most recent Form feed sent to the printer.
L	Restarts printing at the Last form feed prior to the most recent form feed sent to the printer.
1	Restarts printing at the beginning of the most recent 4K byte block of data of the print file sent to the printer. (Backs up a little.)
2	Restarts printing at the beginning of the second most recent 4K byte block of data sent to the printer. (Backs up a lot.)
X	Cancels printing currently in process by the printer and erases that file from the print spool. See example below.

EXAMPLE: Assume the printer is listing a source program, while a report from another user is waiting on the print spool file to print. An operator may cause the printer to stop printing the source listing and begin printing the report by entering the following command at any user station which has that particular printer set as its LST: device:

```
PRNT X
```

Note!

When the printer stops printing the source listing, it will be lost from the print spool file. The report, if it is next in the print queue, will then begin printing.

Table 7-1 continued

Command	Effect
S	Suspends the printer temporarily. (This could, for example, be used for forms changing.)
C	Continues printing. (Used after the Suspend command.)
D	Requests Direct printer allocation. If no other user has direct control of the printer and the user is not currently spooling, the operator using this command will see his output go directly to the printer with no disk buffering. However, no other user will be able to send data to this printer until the "P" command (below) is invoked. **STOP!** **This command can only be used when the printer is not in use. If one user has direct control of the printer when this command is invoked by a second user, there will be no effect.**
P	Releases the printer from direct allocation if the user is not printing and returns it to shared, auto-spool mode. Unlike other commands, this may not be given at any user station to release the printer from whomever had the direct allocation.
M/CTRL H	Suppresses auto form feed.
M	This command is used to customize printer action at the beginning of each printout for each user. The user will be prompted on his display to enter a string of up to 8 ASCII characters. They can include print control characters (pause, form feed, change print font, etc.) and simple messages to be printed as header information and operator prompts. If ^S is the first character of this string, the file extension character t (7.2.3) will be P. If ^S is any character except the first, t will be H.

Table 7-1 continued

To respond to the PRNT M ("enter header"), enter a string of up to 8 ASCII characters. The correct format is

1) a single quote (')

2) the expression

3) a second quote (')

4) a carriage return

Any binary print control character in the expression must be preceded by the ASCII ^ character.

Example:

If the expression

'^L^LCKFRM^S'

is entered in response to the prompt, the printer will issue two form feeds (Control-L's) to clear the last printout, print the characters CKFRM (as a reminder to the user/operator to load the printer with check forms), and then pause (Control-S). Once the new forms have been loaded, the pause can be cleared via the PRNT C command.

Notes:

1. A top-of-form command is sent to the printer before each print file (default). Use PRNT M to set a different start-of-print file message.

2. When data is sent to one of the system printers via one of the standard CP/M print commands, MmmOST creates a print spool file. The automatically named file can be renamed with a CP/M or word processing command.

3. To avoid overfilling public drive space with print spool files, use the STAT command to determine available space.

7.3.2 Queue Selection

If you wish to temporarily spool your print files but not begin printing them until you are ready, use the SELQUEUE command. This command allows you to set the queue name field of the print spool file name to any alphanumeric character. All of your

spooled print files will thereafter have the new character in the queue name instead of the (default) IOBYTE index value.

For example, if you entered

```
SELQUEUE S
```

all subsequent print spool files for this user station would be named

```
SPLuSnnc.PRt
```

Use the character Z to set the queue selection back to the (default) IOBYTE Index.

8. RECONFIGURING MmmOST

8.1 INTRODUCTION

In order to operate, MmmOST must know certain information about its environment. This information is contained in the system parameters.

TeleVideo ships the system with a standard set of system parameters, but you may wish to modify these parameters to fit your unique needs. For example, you may want to add passwords to your system or permanently enter files on the shared file list.

The process of changing system parameters is called reconfiguration.

8.2 RECONFIGURATION PROCEDURES

System parameters are changed by running the program called GENPARMS. If Parameter/Function #1, 4, or 10 in GENPARMS (section 7.3.1) is changed and the user station contains an 8-bit processor, then the program called CNFGUSER.COM must also be run. If any of the same parameters are changed and the user station contains a 16-bit processor, then the program CNFG86.COM must be run.

GENPARMS creates the file called DPCPARMS.DAT, which MmmOST uses when it is loaded into the service processor.

CNFGUSER and CNFG86.COM create a file called USERCPMx.DAT; each USERCPMx.DAT relates to a particular user station. X is set as shown in Table 8-1 below.

Table 8-1

User Station and User Station CP/M File

USER STATION	USER STATION CP/M FILE
TS 800/800A/801/802	USERCPM.DAT
TS 802H	USERCPM1.DAT
TS 802G	USERCPM4.DAT
TS 802GH	USERCPM5.DAT
TS 1600, 1603, 1603H	USERCPM3.DAT
1602G, 1602GH	USERCPM2.DAT

Reconfiguration can be accomplished in two ways. It can be done from a service processor terminal (described as **Method 1** below) or from a user station (described as **Method 2**).

If parameter #1, 4, or 10 is changed, and either CNFGUSER or CNFG86 are run, use Method 1.

8.2.1 Method 1 (Recommended) -- From a Service Processor Terminal

User: 1. Connect a terminal to the service processor.

2. Enter

   ^C

   to cancel MmmOST.

System: 3. Displays

   Enter Y to Cancel

User: 4. Enter

   Y<CR>

5. Copy the files USERCPM*.DAT and DPCPARMS.DAT from drive A to drive B by entering

   ```
   PIP B:=A:USERCPM*.DAT[V]<CR>
   PIP B:=A:DPCPARMS.DAT[V]<CR>
   PIP B:=A:MMMOST.COM[V]<CR>
   PIP B:=A:MST*.OVL[V]<CR>
   ```

6. Enter

   B:<CR>

   to log on to drive B.

7. Run GENPARMS as described in Section 8.3. Running GENPARMS will automatically create the new file DPCPARMS.DAT on drive B, as well as DPCPARMS.BAK, a backup copy of the original.

8. If you did not change Parameter/Function #1, 4 or 10 in the GENPARMS program, go to Step 9.

   If you changed Item #1, 4 or 10 in GENPARMS and your user station contains an 8-bit processor, run the program CNFGUSER as described below.

   If you changed Item #1, 4 or 10 in GENPARMS and your user station contains a 16-bit processor, run CNFG86, also described below.

9. Enter

MMMOST<CR>

to load MmmOST and test the new parameters.

System: 10. Displays

MmmOST V x.y
COPYRIGHT TeleVideo Systems, Inc. mo/da/yr

where

x.y is the actual version number (e.g., 2.1).

Note!

If MmmOST does not load, various error messages will be displayed. If that happens, re-examine the parameters in conjunction with the information provided by the error message(s).

User: 11. After the message in Step 10 is displayed, and you are satisfied with the system parameters, set them for permanent use. To do this, enter

^C

System: 12. Displays

Enter Y to cancel MmmOST.

User: 13. Enter

Y
A:<CR>

to cancel MmmOST and log on to drive A.

14. Enter

PIP A:=B:DPCPARMS.DAT[V]<CR>

to move the latest DPCPARMS.DAT file to drive A.

TO RUN CNFGUSER FOR ALL 8-BIT USER STATIONS:

User: 1. Connect a terminal to the service processor.

2. Make sure that drive A contains all of the appropriate following files:

For all user stations: CNFGUSER.COM
DCPARMS.DAT

For TS 800/802: XPDUBIOS.HEX
XPDUBOOT.HEX
CPM59.COM

For TS 802H: XPD1BIOS.HEX
XPD1BOOT.HEX
CPM58.COM

For TS 802G XPD4BIOS.HEX
XPD4BOOT.HEX
CPM58.COM
MERGE.COM
NETSYS.COM

For TS 802GH XPD5BIOS.HEX
XPD5BOOT.HEX
CPM57.COM
MERGE.COM
NETSYS.COM

3. Enter

For TS 800A/801/802:

CNFGUSER XPDU CPM59.COM<CR>

For TS 802H:

CNFGUSER XPD1 CPM58.COM<CR>

For TS 802G:

CNFGUSER XPD4 CPM58.COM<CR>

For TS 802GH:

CNFGUSER XPD5 CPM57.COM<CR>

System: 4. Displays

TELEVIDEO SYSTEMS, INC. USERCPM GENERATION

ENTER DRIVE TO OUTPUT USERCPM.DAT (A,B,C)

User: 5. Enter

B

System: 6. Displays

USERCPM.DAT FILE WRITTEN SUCCESSFULLY

User: 7. Enter

B:<CR>

to log on to drive B.

8. If you are running this program for a TS 800A/801/802/802H, go to Step 9.

Note!

If you are running this program for a TS 802G/802GH, you must enter one of the following commands at this point:

For TS 802G: **MERGE USERCPM4.DAT<CR>**

For TS 802GH: **MERGE USERCPM5.DAT<CR>**

The system will display:

MERGE COMPLETED SUCCESSFULLY

Go to Step 9.

9. Call up a directory on drive B to make sure the file USERCPM.DAT is there.

10. Enter

MMMOST<CR>

11. Cold boot each user station to load CP/M.

12. Test the area of the BIOS which was modified.

13. If everything is satisfactory, rename the original USERCPM*.DAT file (stored on drive A) to USERCPM*.BAK.

14. Enter

PIP A:=B:USERCPM*.DAT[VO]<CR>

to move the latest USERCPM.DAT from drive B to drive A.

15. Reboot the service processor. The first request from each user station must be a cold boot.

TO RUN CNFG86 FOR 16-BIT USER STATIONS

User: 1. Connect a service processor terminal.

2. Make sure that drive A contains all of the appropriate files:

For all user statons:	CPM.CMD DPCPARMS.DAT
For TS 1602G/1602GH:	XP62BIOS.H86
For TS 1600/1603/1603H:	XP63BIOS.H86

3. Enter

For TS 1602G/1602GH:

CNFG86 XP62 CPM.CMD<CR>

For TS 1600/1603/1603H:

CNFG86 XP63 CPM.CMD<CR>

System: 4. Displays

TELEVIDEO SYSTEMS USER CPM-86 GENERATION

ENTER DRIVE TO OUTPUT USERCPM FILE (A,B,C)

User: 5. Enter

B

System: 6. Displays

USERCPM.DAT FILE WRITTEN SUCCESSFULLY

7. Enter

B:<CR>

to log onto drive B.

8. To test USERCPM*.DAT, make sure DPCPARMS.DAT is on drive B.

9. Enter

MMMOST<CR>

10. Cold boot each user station to load CP/M.

11. Test the area of the BIOS which you modified.

12. If everything is satisfactory, rename the original USERCPM*.DAT (stored on drive A) to USERCPM*.BAK.

13. Enter

    PIP A:=B:USERCPM*.DAT[VO]<CR>

    to move the latest USERCPM*.DAT from drive B to drive A.

14. Reboot the service processor. The first request from each user station must be a cold boot.

8.2.2 Method 2 -- From a User Station

1. From any user station, use the program LOGON (as described in Chapter 5) to obtain access to the system directory. If the passwords have not been modified since TeleVideo shipped the system, enter

   B:LOGON SYSTEM<CR>

   (SYSTEM is the password for the system directory.)

2. Run GENPARMS as described in Section 8.3. This will automatically create a new file DPCPARMS.DAT on drive A, as well as DPCPARMS.BAK, a backup copy of the original.

3. Push the RESET button on the service processor.

4. a. If the new parameters are satisfactory and the user stations are working, you are finished.

   b. If user stations are working but the parameters are still not satisfactory, run GENPARMS again.

   c. If the user stations are not working, then attach a terminal to the service processor and use Method 1 described in Section 8.2.1.

8.3 RUNNING GENPARMS

This section describes how to run GENPARMS. GENPARMS is a menu-driven program which will write a system parameter file to disk. The name of this system parameter file is DPCPARMS.DAT. This file allows you to customize MmmOST to fit your own unique environment. MmmOST uses this file to configure the service processor to your requirements. Section 8.3.1 shows how to invoke GENPARMS and the primary menu that it displays. A description of each parameter function is given in Section 8.3.2.

8.3.1 Invoking GENPARMS

To invoke GENPARMS, enter

GENPARMS<CR>

The system will then respond by displaying its main menu:

```
TeleVideo Systems, Inc.
MmmOST system parameter generation V2.0

No. Parameter/Function
-----------------------------------------------------------------

1   Private drives (currently: A: )
2   Password (account) names
3   Repeats and delays (currently: 32, 64)
4   Drives with shared files (currently: B: )
5   Shared file names
6   Maximum sectors in lock (currently: 4)
7   Number of mail stops (currently: 6)
8   User function access (currently assigned mailboxes are: )
9   User stations which autoboot a program (currently: )
10  Print spool drive (currently: B:)
11  End input and write new DPCPARMS.DAT file to disk
12  Abandon this update

Enter number of parameter/function -
```

Note!

GENPARMS displays the values that it reads from the old version of DPCPARMS.DAT. Illustrated above are the (default) values as the TS 806 is shipped from the factory.

8.3.2 Parameter/Function Descriptions

This section describes the parameters which can be changed by your responses during the GENPARMS program.

1. Private Drives

   With this Parameter/Function, you can add, modify, or delete private drives. (You modify a private drive by entering data as though you were adding it.)

   If you add (or modify) a private drive, you will be asked for the number of entries (files) per directory (called "directory size" and "number of directory entries") and the number of directories (called "personal directory size"). If there is no password account for this drive, you will be asked if you want to (1) add passwords individually or (2) copy them from another drive.

The total number of directory entries (files) on a drive is given by the following:

total number of directory entries (files)=

number of directories X number of directory entries (files) per directory

In assigning the number of directories and the number of directory entries, note the following two rules:

1. The total number of directory entries must be less than or equal to the maximum number of allowable directory entries (files) for the particular drive.

2. The number of directory entries (files) per directory must be a multiple of the number 4.

The maximum number of directory entries (files) on drive A of a TS 806 is 2048. Table 8-2 below shows some of the possible directory configurations for the TS 806.

Table 8-2

TS 806 Private Directory Configurations

NUMBER OF DIRECTORIES	NUMBER OF ENTRIES (FILES) PER DIRECTORY	TOTAL NUMBER OF DIRECTORY ENTRIES
8	256	2048
10	200	2000
16	128	2048
20	100	2000
32	64	2048
51	40	2040
64	32	2048
128	16	2048
256	8	2048

Note!

This table represents some but not all directory configurations for a TS 806.

Defaults for the TS 806 are 16 directories with 128 entries (files) per directory. Defaults for the TS 816 are 32 directories with 128 entries (files) per directory.

2. Password (account) names

This parameter allows one to change the passwords associated with the various private directories on the private drives. GENPARMS gives you the complete list of passwords and the directories to which they are currently attached. It then prompts you for addition or deletion of passwords. In order to change an existing password, you may (1) add the new password to the same relative directory and then delete the old password or (2) delete the old password and then add the new one to the same relative directory. The order does not matter.

Up to eight alphanumeric characters may be entered for a password.

3. Repeats and delays

Repeats is the maximum number of consecutive I/O requests (disk reads/writes, printer I/O) granted to a user, assuming that the delay between requests is less than the defined variable delay. Delay is the maximum I/O delay allowed between the consecutive requests of a user; if this delay is exceeded, another user's request will be serviced. A delay value of 25 equals approximately 1 millisecond.

Higher values for repeats and for delays (up to an optimum delay value) increase system throughput. (In the case of disk reads/writes, this is due to the increased ratio of reads/writes to head arm movement.) Lower values will give quicker response to a number of different users. Either value can be set from 1 to 32767.

4. Drives with shared files

This parameter tells MmmOST which drives will contain shared files and system communication files, i.e., it tells MmmOST which drives are public drives.

5. Shared file names

Selecting this parameter updates the shared file list, in which the names of shared files and system communication files reside. GENPARMS displays the entire shared file list, showing for each file on it the drive, filename, and

category (either shared file or system communication file). Files can be added, deleted, or changed positionally in the list. Remember that the sequence in which shared files are listed is important if you intend to use either the R/W or R/L access modes (i.e., shared files must be accessed in the order they are entered on this list if locks are to be maintained).

6. Maximum sectors in lock

   This parameter specifies the maximum number of contiguous CP/M 128-byte sectors that can be locked. The largest allowable number is 127.

   Since a user can lock only one record per file (or a set of consecutive records within a file) in R/W and R/L control modes, this sets a limit on the number of sectors a record (or set of consecutive records) can span in those modes.

7. Number of mail stops

   The number of mail stops specifies the number of records in MULTI.SYS which can be used as mailboxes for data transmitted between user stations. The maximum number of mail stops is 127.

8. User function accesses

   User function accesses allow you to indicate to MmmOST which mailbox(es) act as an entry point(s) to the user function subroutine(s). There are a maximum of four accesses and four subroutines. Mailboxes 1-4 (Records 1-4 of MULTI.SYS) are used for access.

9. User stations which autoboot a program

   This parameter allows selected user stations to perform an immediate execution of a program when they are powered up. To allow autobooting of a program, the user station CBIOS must be modified.

10. Print spool drive

    MmmOST Version 2.1 requires all print spooling to be done on a public drive (not a public only drive). You can specify any public drive on the system as the spool drive. In many applications, drive B is used as the spool drive.

11. End input and write new DPCPARMS.DAT file to disk

    Select this function when you are satisfied with the values of parameters 1-10. This function will write the new DPCPARMS.DAT file to the currently logged disk (along with a backup copy of the original called DPCPARMS.BAK) and return you to CP/M.

12. Abandon this update

Select this function if you decide not to reconfigure MmmOST after you have started GENPARMS. No changes to any of the parameters will be saved, and the original version of DPCPARMS.DAT (i.e., the version that existed before you ran GENPARMS) will be kept. You will return to CP/M.

9. ERROR AND STATUS MESSAGES

9.1 INTRODUCTION

This chapter provides information on error messages that may appear at the user stations (and tells how to suppress those messages if so desired). It also provides information on the status messages that may appear on the service processor terminal.

9.2 USER STATION ERROR MESSAGES

Table 9-1 provides a complete list of the error messages that may appear at a user station. If so desired, certain messages (marked * in the table) may be suppressed by entering

ERRMSG D<CR>

These error messages may be re-enabled by entering

ERRMSG E<CR>

Note!

If the error messages are suppressed, a CP/M error code will be returned to the applications program instead of the normal error message display.

Table 9-1

User Station Error Messages

Message:

BDOS ERR ON [drive:] FILE/DRIVE R/O*

Explanation:

1. Program in R/W or R/L mode tries to write to a record which it did not first read. The write will not occur, the above error message will appear, and the user station will automatically boot and return to the CP/M prompt.

2. One user tries to write to a file which has been set to F/L mode by another.

3. One program updates, extends, or creates an unshared file on a public drive and a second program attempts to update, extend, rename, or erase file before the first program warm boots.

4. One program opens a file (Unshared or Shared) on a public drive and a second program attempts to rename or erase the file.

Note!

This may occur when certain built-in CP/M commands, such as TYPE, SAVE, and so on, are used since they open a file and never warm boot.

5. R/O mode has been assigned by a user station to a file and the user station tries to write to the file while another user station has one of the following modes assigned:

R/W
R/L
F/L
N/L

Message:

BIOS WRITE ERROR ON DRV=[drive:] TRK=XXXX; SCTR=YYYY; WRT TYPE=ZZ; RTN CODE=AA

WRITE PROTECT ERROR

BDOS ERR ON [drive:] BAD SECTOR*

Explanation:

One program updates, extends, creates, renames, or erases a file on a public only drive and another program attempts to update, extend, create, rename, or erase any file on the drive before the first program warm boots.

9.3 STATUS MESSAGES ON THE SERVICE PROCESSOR TERMINAL

Status messages may appear on the service processor terminal. These messages will be preceeded by the user station identification number. Table 8-2 describes these messages.

Table 9-2

Service Processor Terminal Status Messages

MESSAGE	MEANING
ACT	ACTive: Indicates that the user went active or that normal polling was resumed.
BTE	BooT Error: A communication error occurred when the user station attempted to boot. If this error persists, a service representative should be called.
CME	CoMmand Error: A direct command or a LOGON name was not recognized.
DPV	Disk Pointer Virtual: This is a virtual allocation error; it is internal to MmmOST. If it persists, a service representative should be called.
ENB	ENd of Boot file: The user station tried to boot and read beyond the end of the CPM.DAT file. This may occur if the user has generated a new version of CPM.DAT and has incorrectly specified the file length in XPD*BOOT.HEX.
FDR	Write Fault on DRive: Write fault on allocation of private drive. If this message occurs, a service representative should be called.
MEM	MEMory: The service processor memory has been filled. Refer to Appendix C for more information.
MNP	MiNimum Poll: Due to a request error received by MmmOST from the user station, MmmOST will poll the user station only once every thousand polls until at the receipt of a valid request from the user station, active polling is restored. (Request errors are usually caused by noise on the RS 422 line.) If this message persists, call a service representative.

Table 9-2 continued

MESSAGE	MEANING
PRQ	PRinter Queue: This message indicates that the user has created a new print spool file.
REQ	REQuest error: The first request of a user station after the service processor is booted should be a boot request. This message appears if it is not, or if the request sent is unrecognizable.
R/O	Read/Only: This is a protection violation error. It may occur when 1) A user tries to erase a system communication file (usually called MULTI.SYS). 2) A user tries to erase a shared file, that is opened by another user, while in R/O mode; 3) A user who has set R/L mode for a file writes to the file when the status character is u (i.e., no lock has been granted); 4) One user has exclusive write access to an unshared file on a public drive and a second user tries to write to the file.
SEL	SELect error: This appears if the user station tries to select a drive that is not in the system.
SPD	SPool Directory: This message indicates that the user is trying to create a print spool file on a drive, and the directory for the drive is already full.
SPF	SPool overFlow: A disk on which a print spool file is being created has overFlowed.

Table 9-2 continued

MESSAGE	MEANING
WLK	Wait on LocK: Indicates the user is waiting for a locked record.
WTB	Wait terminated or broken: User wait has been terminated or broken. This will be displayed under two conditions: (1) A user has received a record lock that he was waiting for in R/W mode. (2) The user has the printer directly assigned (via the PRNT D command) and has sent data from its (128 byte) buffer to the service processor 128 byte) buffer for printing. Data is sent to the printer in 128-byte "chunks" via these two buffers. (3) A user station which was in minimum poll mode sent a valid request.
XFC	X Function Call: Indicates an illegal BDOS function call by user.
XFI	XFer In: Indicates that a communication error has occurred. The communication was from the user station to the service processor.
XFO	XFer Out: Indicates that a communication error has occurred from the service processor to the user station.

APPENDICES

A. Suggested CP/M References

B. Programming Examples

C. Memory Error Messages

D. Modifying the User BIOS

APPENDIX A

SUGGESTED CP/M REFERENCES

The following books will be useful references.

1. Fernandez, Judi N. and Ruth Ashley, Using CP/M: A Self Teaching Guide, John Wiley and Sons, 605 Third Avenue, New York, New York, 1980, $8.95.

(This is an excellent self-teaching guide, highly recommended for the user unfamiliar with CP/M but perhaps familiar with computers. Assumes no prior knowledge about operating systems.)

2. Hogan, Thom, Osborne CP/M User Guide, Osborne/McGraw-Hill, 630 Bancroft Way, Dept. UB, Berkeley, CA 94710, $12.99.

(A complete book for the beginning computer operator. Covers all aspects of computer operation and CP/M use. Organization and presentation are outstanding.)

3. Zaks, Rodney, The CP/M Handbook with MP/M, SyBex, 2344 Sixth Street, Berkeley, CA 94710 (415/848-8233), 1980, $13.95.

(This book gives more complete detail than many beginners will want.)

4. CP/M Summary Guide for Versions 1.4 and 2.X, edited by Bruce Brigham, The Rosetta Stone, Post Office Box 35, Glastonbury, CT, 06025, 203/633-8490, $6.95 plus $1.00 postage and handling.

5. Murtha, Stephen M. and Mitchell Waite, CP/M Primer, Howard W Sams & Co., Inc, 4300 West 62nd St., Indianapolis, Indiana 46268, $14.95.

(An excellent introduction to both CP/M and microcomputers.)

6. Townsend, Carl, How to Get Started with CP/M, Dilithium Press, Post Office Box 606, Beaverton, Oregon 97075, $9.95

(A succinct discussion of CP/M. Also includes a listing of CP/M software suppliers.)

TeleVideo welcomes comments from you about these books as well as names of others which you find useful.

APPENDIX B

PROGRAMMING EXAMPLES

Example 1: MBASIC Example of Record Locking in the R/W Mode

```
10 OPEN "R",#3,"B:MULTI.SYS"
20 FIELD #3, 128 AS MSS$
30 LSET MSS$="B:LOCKTEST.FIL=R/W"
40 PUT #3,1
50 GET #3,2
60 GET #3,1
70 PRINT "STATUS OF MULTI SYS WRITE ";MSS$
80 OPEN "R",#1,"B:LOCKTEST.FIL"
90 FIELD #1, 22 AS RECORD$
100 FILE.NO%=1
110 REC.LEN%=22
120 INPUT "WHICH RECORD TO UPDATE 1 THRU 10";REC.NO%
130 IF FILE.NO%=1 THEN B1$="B:LOCKTEST.FIL=R/O":B2$="B:LOCKTEST.FIL=R/W"
140 GOSUB 600 : REM SUBROUTINE TO FLUSH MBASIC BUFFER
150 REM UPON RETURN FROM 600 IT IS CERTAIN THAT THE BUFFER DOES NOT
160 REM CONTAIN REC.NO%
170 GET # FILE.NO%,REC.NO%
180 PRINT "RECORD NUMBER  ";REC.NO%;" OF FILE ";FILE.NO%; "READS ";
190 PRINT RECORD$
200 INPUT "WHAT WOULD YOU RATHER IT BE?";NOW$
210 LSET B$=NOW$
220 PUT # FILE.NO%,REC.NO%
230 INPUT "WANT TO PLAY SOME MORE? Y OR N";ANS$
240 IF ANS$="Y" THEN GOTO 100
250 LSET MSS$="B:LOCKTEST.FIL=R/O"
260 PUT #3,1
270 STOP
280 REM ************************************************************
600 REM THIS SUBROUTINE MOMENTARILY SETS FILE.NO% TO R/O AND READS A
610 REM NEARBY RECORD IN ORDER TO ASSURE A DISK READ OF REC.NO%
620 LET INCR% = INT(128/REC.LEN%)
625 IF INCR%=0, THEN INCR%=1
630 IF REC.NO%=10 THEN DUM.REC%=9-INCR% ELSE DUM.REC%=REC.NO%+INCR%+1
635 LSET MSS$ =B1$ : PUT #3,1
640 GET # FILE.NO%,DUM.REC%
650 LSET MSS$=B2$
660 PUT #3,1
670 GET #3,2
680 GET #3,1
690 PRINT "THIS IS STATUS OF 600 SUBROUTINE ";MSS$
700 RETURN
```

In line 30, the access mode is set for the file LOCKTEST.FIL. With MBASIC, the access mode must be set before the file is opened.

Line 110 sets a variable 'REC.LEN%' to the value of the record length of the file that is being opened. This is used to calculate the record number which must be read from the file in order to flush the buffer. To flush the buffer, it is necessary to set the file temporarily to the R/O mode and read a record that occupies a different sector than the record to be locked. This is accomplished in the 600 subroutine. This subroutine calculates the record number that must be read in order to flush the buffer, sets the file temporarily to the R/O mode, reads the record, and then sets the access mode back to R/W. Upon return from this subroutine, the record to be locked is read and becomes locked. If this buffer flushing technique is not used, a "BDOS R/O" error will be received when the record to be locked is read.

Example 2: MBASIC Example of Record Locking Using R/W Mode with Two Files

```
10 OPEN "R",#3,"B:MULTI.SYS"
20 FIELD #3, 128 AS MSS$
30 LSET MSS$="B:TEST.FIL=R/W"
40 PUT #3,1
50 GET #3,2
60 GET #3,1
70 PRINT "STATUS OF FIRST WRITE TO MULTI.SYS= ";MSS$
80 LSET MSS$="B:TEST1.FIL=R/W"
90 PUT #3,1
100 GET #3,2
110 GET #3,1
120 PRINT "STATUS OF SECOND MULTI.SYS WRITE  = ";MSS$
130 REC.LEN%=22
140 OPEN "R",#1,"B:TEST.FIL",REC.LEN%
150 OPEN "R",#2,"B:TEST1.FIL",REC.LEN%
160 FIELD #1,22 AS RECORD$
170 FIELD #2,22 AS RECORD2$
180 INPUT "WHICH RECORD TO UPDATE IN FILE 1 ";REC.NO%
190 REC.NO1%=REC.NO%
200 B1$="B:TEST.FIL=R/O" : B2$="B:TEST.FIL=R/W"
210 FILE.NO%=1 : GOSUB 630
220 GET #FILE.NO%,REC.NO1%
230 PRINT "RECORD ";REC.NO%;" OF FILE 1 READS ";RECORD$
240 INPUT "WHAT WOULD YOU LIKE IT TO READ ";CHANGE1$
250 INPUT "WHICH RECORD TO UPDATE ON FILE 2 ";REC.NO%
```

```
260 REC.NO2%=REC.NO%
270 B1$="B:TEST1.FIL=R/O" : B2$="B:TEST1.FIL=R/W"
280 FILE.NO%=2 : GOSUB 630
290 GET #FILE.NO%,REC.NO2%
300 PRINT "RECORD ";REC.NO%;" OF FILE 2 READS ";RECORD2$
310 INPUT "WHAT WOULD YOU LIKE IT TO READ ";CHANGE2$
320 LSET RECORD2$=CHANGE2$
330 PUT #2,REC.NO2%
340 LSET RECORD$=CHANGE1$
350 PUT #1,REC.NO1%
360 INPUT "CONTINUE? Y OR N";ANS$
370 IF ANS$="Y" THEN GOTO  180
380 LSET MSS$="B:TEST.FIL=R/O"
390 PUT #3,1
400 LSET MSS$="B:TEST1.FIL=R/O"
410 PUT #3,1
420 STOP
600 REM ************************************************************
610 REM THIS SUBROUTINE MOMENTARILY SETS FILE.NO% TO R/O AND READS A
620 REM NEARBY RECORD IN ORDER TO ASSURE A DISK READ OF REC.NO%
625 IF INCR%=0, THEN INCR%=1
630 LET INCR% = INT(128/REC.LEN%)
640 IF REC.NO%=10 THEN DUM.REC%=9-INCR% ELSE DUM.REC%=REC.NO%+INCR%+1
650 LSET MSS$ =B1$ : PUT #3,1
660 GET #FILE.NO%,DUM.REC%
670 LSET MSS$=B2$ : PUT #3,1
680 GET #3,2
690 GET #3,1
700 PRINT "MULTI.SYS STATUS IN 600 = ";MSS$
710 RETURN
```

This example uses the same methods of buffer flushing as the first example with one exception: in this case, two files are updated at once. Since the record from TEST.FIL is read first, followed by a record from TEST1.FIL, it is necessary to write the records back to the files in reverse order. TEST1.FIL is updated with the new record first and then TEST.FIL is updated with the new record specified for that file.

Example 3: CBASIC Example of Record Locking Using the 'R/W' Mode

```
    1:          OPEN "B:MULTI.SYS" RECL 128 AS 3
    2:
    3:          PRINT #3,1;"B:DATABASE.ONE=R/W"
    4:
    5:          READ #3,1;STAT$
    6:          USER.ID% = VAL(MID$(STAT$,14,2))
    7:          STAT$=MID$(STAT$,48,1)
    8:          IF STAT$="S" THEN STAT$="SATISFACTORY"
    9:          IF STAT$="U" THEN STAT$="UNSATISFACTORY"
   10:          PRINT "STATUS IS ";STAT$
   11:
   12:          PRINT "YOUR TERMINAL ID IS ";USER.ID%
   13:
   14:          IF END #1 THEN 5
   15:          OPEN "B:DATABASE.ONE" RECL 128 AS 1
   16:          GOTO 10
   17:   5      CREATE "B:DATABASE.ONE" RECL 128 AS 1
   18:          FOR I% = 1 TO 50
   19:          RECORD$="THIS IS RECORD #"+STR$(I%)
   20:          PRINT USING "&";#1,I%;RECORD$
   21:          NEXT I%
   22:
   23:  10      INPUT "WHICH RECORD TO UPDATE (17-50)? ";REC%
   24:          IF REC% < 17 THEN \
   25:            PRINT "RECORD CHOSEN IS FOR SYSTEM USE - PLEASE RE-ENTER" :\
   26:            GOTO 10
   27:
   28:          IF REC% > 50 THEN \
   29:            PRINT "RECORD OUT OF RANGE PLEASE RE-ENTER" : GOTO 10
   30:
   31:          READ #1,REC%;LINE RECORD$
   32:
   33:          PRINT "RECORD ";REC%;" NOW READS- ";RECORD$
   34:          INPUT "WHAT WOULD YOU RATHER IT READ? ";UPDATE$
   35:          PRINT USING "&";#1,REC%;UPDATE$
   36:          READ #1,USER.ID%;DUMMY$
   37:          INPUT "PLAY SOME MORE? ";ANSS$
   38:            IF ANSS$="Y" THEN GOTO 10
   39:          CLOSE 1
   40:          PRINT #3,1;"B:DATABASE.ONE=R/O"
   41:          CLOSE 3
   42:          STOP
```

This CBASIC example uses a file that contains fifty records. The file is intiialized so that each record contains its record number within the record.

The file DATABASE.ONE is set to the R/W mode in line 3. Notice that the locking mode is set before the file is opened. The status of the write to MULTI.SYS is checked in line 5 and printed to the screen. This check of the status performs the following three functions:

(1) The read forces the CBASIC buffer, which contains the record setting the R/W status, to be written to the disk.

(2) The status of MULTI.SYS is checked to see if the file was successfully set to the R/W mode.

(3) The number of the user station is obtained from the status record.

Since a write to a file in CBASIC is buffered (i.e. it remains in memory until another access such as a read of the file, is performed), it is necessary to set up a dummy record for each user station that will be using the file in the R/W mode. After performing a write to a data record in the file, a dummy record corresponding to the user station number is then read. Once the dummy record is read, the data record is actually written to the file and the lock is released. Since a read of a record locks that record in the R/W mode, the dummy record is then locked. Because no one else will ever read it, it can stay locked until another data record is read and locked, and the lock on the dummy record is released. This process can be repeated for the next write to a data record.

Lines 35 and 36 demonstrate this process. In line 35, the actual data record is written. In line 36, the read of a dummy record is performed. The read is of record number USER.ID%, and the value of USER.ID% is the user station number. This number was obtained in line 6 when the value of the 14th and 15th position of the status record, returned from MULTI.SYS, was received.

The program will not allow a data record read of a record number less than 17 because the first 16 records are set up to be dummy records for all of the TS 816 user stations. If all of the TS 806 user stations were actively used in this process, only 6 dummy records would be needed.

APPENDIX C

MEMORY STATUS MESSAGE

If the service processor memory has been filled, you will see the message

MEM

on your service processor terminal screen. This message occurs if the dynamically memory allocated in the service processor (for the functions listed below) exceeds the available space. Available space may be determined by entering

S

on a service processor terminal. The system will display

dynamic space: xxxxx contiguous space: xxxx

where

xxxx is in 16-bit words.

Table C-1 lists various memory functions and the number of 16-bit words used by them.

Table C-1

Memory Functions

Total Number of 16-bit Words Used	Function
13	per virtual allocation block on a private drive. These are released at user warm boot.
14	per resource lock obtained with the LCK option for shared read/write files.
9	per lock obtained by read in R/W or R/L mode.
12	per open for an unshared file on a public drive.

Table C-1 continued

Total Number of 16-bit Words Used	Function
20	per open for unshared file that has been temporarily placed on the shared file list by setting one of the following control modes for it: R/O, R/L, R/W, F/L, N/L.
14	per directory entry for every shared file that has been opened on a public drive.
103	per every existing queue (FIFO)
67	per every record write using a FIFO

APPENDIX D

MODIFYING THE USER BIOS

To generate a a user station CP/M, run the submit file GENUSER for 8-bit systems or GENUSER86 for 16-bit systems. This file must be run from a service processor terminal.

GENUSER (FOR 8-BIT SYSTEMS)

User: 1. Make sure that drive A contains all of the following files:

GENUSER.SUB
SUBMIT.COM
CNFGUSER.COM
XPDUBIOS.ASM
XPDUBOOT.ASM
XPD1BIOS.ASM
XPD1BOOT.ASM
M80.COM (available from Microsoft)
L80.COM (available from Microsoft)

2. Enter

SUBMIT GENUSER<CR>

System: 3. Automatically displays a series of messages. The first one that requires a user response is

ORIGIN ABOVE LOADER MEMORY, MOVE ANYWAY (Y OR N)?

User: 4. Enter

N<CR>

System: 5. Automatically displays another series of messages. The next message that requires a user response is

ENTER DRIVE TO OUTPUT USERCPM FILE (A,B,C)

User: 6. Enter

B

System: 7. Displays

CPM-80 USERCPM FILE WRITTEN SUCCESSFULLY

8. Enter

   B:<CR>

   to log onto drive B.

9. To test USERCPM.DAT, make sure DPCPARMS.DAT, MMMOST.COM, and MST*.OVL are on drive B.

10. Enter

    MMMOST<CR>

11. Cold boot each user station to load CP/M.

12. Test the area of the BIOS which you modified.

13. If everything is satisfactory, rename the original USERCPM*.DAT, stored on drive A, to USERCPM*.BAK.

14. Enter

    PIP A:=B:USERCPM*.DAT[VO]<CR>

    to move the latest USERCPM*.DAT from drive B to drive A.

15. Reboot the service processor. The first request from each user station must be a warm/cold boot.

GENUSER86 (FOR 16-BIT SYSTEMS)

User: 1. Make sure that drive A contains all of the following files:

   GENUSER86.SUB
   SUBMIT.COM
   CNFGUSER.COM
   XPDUBIOS.ASM
   XPDUBOOT.ASM
   XPD1BIOS.ASM
   XPD1BOOT.ASM
   M80.COM (available from Microsoft)
   L80.COM (available from Microsoft)

2. Enter

   SUBMIT GENUSER86<CR>

System: 3. Automatically displays another series of messages. The next message that requires a user response is

   ENTER DRIVE TO OUTPUT USERCPM FILE (A,B,C)

User: 4. Enter

B

System: 5. Displays

CPM-86 USERCPM FILE WRITTEN SUCCESSFULLY

6. Enter

B:<CR>

to log onto drive B.

7. To test USERCPM.DAT, make sure DPCPARMS.DAT, MMMOST.COM, and MST*.OVL are on drive B.

8. Enter

MMMOST<CR>

9. Cold boot each user station to load CP/M.

10. Test the area of the BIOS which you modified.

11. If everything is satisfactory, rename the original USERCPM*.DAT, stored on drive A, to USERCPM*.BAK.

12. Enter

PIP A:=B:USERCPM*.DAT[VO]<CR>

to move the latest USERCPM*.DAT from drive B to drive A.

13. Reboot the service processor. The first request from each user station must be a warm/cold boot.

Accessing
  private directories 5.3.2
  private directories with passwords 5.3.2.5
  shared files 3.3.2, 4.2
  unshared files 3.3.2

CBASIC examples 4.7.1
CNFG86 8.2
CNFGUSER 8.2
Control modes 4.3
  using 4.4
  setting 4.5
  with applications programs 4.4.2
   LCK 4.4.2.2
   N/L 4.4.2.1
   ULK 4.4.2.2
  with MmmOST 4.4.1
   F/L (File Lock) 4.4.1.4
   R/L (Read, test for Lock) 4.4.1.3
   R/O (Read Only) 4.4.1.1
   R/W (Read, Wait for lock) 4.4.1.2
CP/M
  BDOS 2.3
  BIOS 2.3
  files for user stations 8.2
  overview 2.3

Directories 3.3, 5.3
  accessing 5.3.2
  default 3.3
  for private drives 3.3.1, 5.3.1
  for public and public only drives 3.3.2, 5.3.2
  for TS 806 5.3.1.1, 8.3.2
  for TS 816 5.3.2.2
  modifying 4.6
  password 3.3.1
Disk mapping 5.2
Drives
  private 3.2
  public 3.2
  public only 3.2
  TS 806 configuration 5.2
  TS 806 default drive assignments 5.2
DSTAT 5.4.1

Error messages 9.2

Fatal embrace 4.2.2
Files
  modifying 4.6
  private 3.3.1
  shared 3.3.2
  system communication 3.4
  unshared 3.3.2
   on public drive 4.6.1
   on public only drive 4.6.2

GENPARMS 8.2
  invoking 8.3.1
  parameter/function descriptions 8.3.2
  running 8.3

Interleaved updating 4.2.1
Inter-Processor Communications Channel (IPC) 2.4.1
IOBYTE 2.3

KEEPLOCK 4.4.1.2

Locking records and files 4.0
Logical devices 2.3
LOGON 5.3.2.5
 examples 5.3.2.6

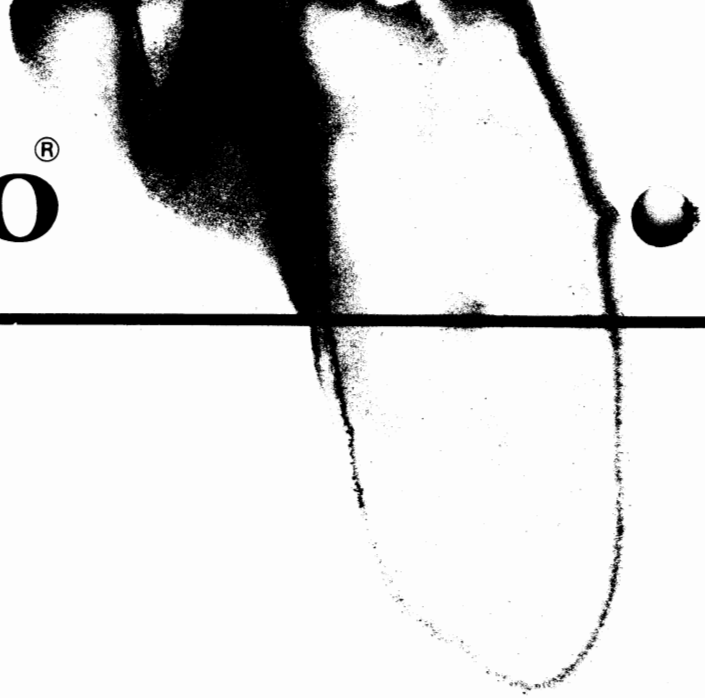
Mailboxes 6.1
Manual
  organization 1.2
Messages
  error (user station) 9.2
  memory status Appendix C
  status 9.3
MmmOST
 reconfiguring 8.2
Modes, control
  using 4.4
  setting 4.5
Modifying the User BIOS Appendix D
MULTI.SYS 3.4
Multi-user environment 2.4

Physical disk space 5.2
Power on 2.4.2
Print control 7.3.1
 commands Table 7-1
Print queue selection 7.3.2
Print spooling 7.2
Print spool file 7.2, 7.2.2
  sending data to 7.2.3
Printer breakpoint routine 7.2.4
Printer selection 7.2.1
Programming examples Appendix B
Prompt, CP/M 2.4.2
Service processor 1.0

Service processor terminal 2.4.3
Single-user environment 2.3
Stand-alone operation 2.4.3
STAT 5.4.2
Status record 3.4.1
Suggested References Appendix A

TeleVideo COBOL
  examples 4.7.2
  setting modes with 4.7.2
Transient Program Area (TPA) 2.2

User stations 1.0



TeleVideo®

TeleVideo® Systems, Inc.

Printed in U.S.A.
LMO

TeleVideo Systems, Inc., 1170 Morse Avenue, Sunnyvale, California 94086
408/745-7760 800/538-8725 (tollfree outside CA) TWX 910 338 7633