END USER'S GUIDE

to

MDZ/OS

Release I

Multi-user, Multi-Processor

System Software

for use with

Teletek Enterprises, Inc.

SysteMaster™, SBC-1™ and PSIO™ Boards

P R E L I M I N A R Y

developed by

Micro Mike's, Inc.
3015 Plains Blvd.
Amarillo, Texas 79102 USA

telephone: 806-372-3633

Manual Revision 01/01

July 13, 1982

making technology uncomplicated ... for People

TABLE OF CONTENTS

INTRODUCTION

This section is designed for the end user who has little or no knowledge of his computer but who does not want to be totally dependent on his dealer or his programmer.

This manual assumes only that you know the difference between the terminal, computer and printer and someone has shown you which floppy disk drive is Number 1 or A and which is Number 2 or B.

Our first task is to convince you, the user, that magic really is not involved in making a computer perform, although your programmer has, through some deft manipulations (typed in rapidly some commands that seem unintelligible), appeared to make the computer do wonderous tasks indeed.

Installing programs on your system is a relatively simple task, so simple that a fool (we keep one around Micro Mike's, Inc. just for this purpose) can do it. This also assumes that the fool will read and take the time to study this section and follow the instructions step by step.

We assume that the hardware (the computer, terminal and printer) is in good working condition and the system software has been installed properly and is working properly.

No comment in this section is meant to imply that your programmer can be replaced. As a good accountant can keep you out of jail (or at least out of trouble with the IRS and state boys) a good programmer can keep you out of trouble and save you literally hours of work when mistakes are made. A good programmer is a jewel and, if a friend as well, to be cherished.

This manual is divided into four sections:

1. The rest of this section, covering some important concepts, a description of the various utilities available so that you can communicate with your programmer and definitions of some terms so those communications can be intelligent.

2. Technical information on the Segmentation and Logical Drive concepts.

3. The nuts and bolts of putting programs out on the hard disk, and creating menus so that you can get to those programs without a lot of hassle.

4. System utilities and how to use them.

The importance of planning cannot be overstated. You would not think of traveling to a strange city in a state you have never visited without consulting a map, calculating the expenses based on miles and time, and making provisions for your accommodations. Planning assumes a purpose, even if it is as simple as fullfilling a desire to go to a place you have never seen before.

Whether you are a small businessman struggling to keep your head above the water and retain your independence or a mogul with vast resources, you expect your computer to earn its keep. It cannot without guidance, direction, and a little hard work initially.

Planning, therefore, is essential in getting the most out of your computer system. We want to create segments or work areas (more about this in Section 1.1) large enough to serve not only your immediate needs but the foreseeable future. However, we don't want to have blank and unusable space sitting out on the hard disk forever. And, we want to be able to get to all programs quickly and easily.

Let us assume, first, that the computer will be used for record keeping, maintaining a general ledger, an accounts receivable, payroll, inventory, word processing and so on. It can do many other tasks as well, and its capabilities are limited only by your imagination and the availability of software.

Study your records. How many accounts in your general ledger and how many checks do you write in your busiest month and how many deposits in your best month; the number of employees on your payroll and how large can you reasonably expect that number to grow?

A proverb found in the Gospel According to Technology states:

> Data will fill all available space, and that space will disappear in half the time originally estimated.

Good manuals which come with your computer programs should help you estimate your data space needs. If not, ask your friend, the programmer, to help you. Also, you can consult your computer dealer. If he can't help you, find one who can.

If you are using a multi-user system, who needs access to which programs? Another way of phrasing this is, which people do you want to keep out of some of the programs?

And, start now to set up a schedule of activities or management of the system. Set out schedules so that you and/or the people in your business or organization, have the opportunity to become familiar with the system. Don't expect to do everything at once. Set out a schedule so that all reports are not due at the same instant.

A word of warning is in order at this point: NEVER press Control C, reset a slave or reboot the system or turn the system off in the middle of a program. The safest method is to get back to the program menu, or better yet, the Control Menu before shutting the system down for the day.

Always back up the segment file, found on Logical Drive 5, and the menu files on Logical Drive 4 any time changes are made in these files.

While understanding the concepts of MDZ/OS is not essential to operating your computer system, it sometimes makes things easier when you understand what's going on.

The wizardry of MDZ/OS, for the most part, is contained in two concepts, segmentation and physical drives-logical drives.

1.1 Segmentation

Segmentation is simply dividing up the disk into segments or pieces, each with a specific task or responsibility, much like a pie which is divided up into pieces to meet the appetites of a family -- a large piece for father, a slightly smaller piece for mother who is watching her weight, another large piece for older brother because he is a growing teen-ager, a piece for sister and a little bitty piece for the baby.

Another way of visualizing drive segmentation is to imagine a set of books on a shelf. Each book can have the same or a different number of pages, but every book will start with a page one and will have the same numbers to a certain place. The major difference then is the number of pages.

In addition, each book will have a table of contents at the beginning which tells the user where the major sections (chapters) can be located.

Under MDZ/OS each of these books would be a segment (drive). The page numbers correspond to sector addresses and the table of contents corresponds to the disk directory. If we want the third book on the shelf and want to look at Page 200, we would walk down the shelf, bypassing two books, until we come to the third book, where we find the page we want and have the information we need.

In a similar manner, MDZ/OS is given an address (page number) from one of the servant operating systems. MDZ/OS, utilizing the defined table, bypasses all the segments (books) not needed until it comes to the proper segment location. The address is then located in the segment (book) by reading the directory (table of contents) and the information is read or written, depending upon the action specified.

Following the analogy you should be able to understand how MDZ/OS can support many different operating systems, each with its own characteristics. By keeping track of the offsets (how many books down the row we need to go to get the right book), many segments can be created and used by a servant operating system as if it where in its normal environment.

1.2 Logical-Physical Drives

The logical drive-physical drive concept involves viewing the hard disk as the computer sees it.

Computer programs are designed to work with an amount of storage which is approximately equal to the job to be performed. Because the computer will, no doubt, have many programs working with varied amounts of space, we must divide the disk into usable areas called segments.

Also many computer programs are designed to work on Floppy Disk Drives 1 and 2, (or Drives A and B). Drive 1 (A) typically contains the programs and Drive 2 (B) typically contains the data files.

Under MDZ/OS, Physical Drives 1-4 are floppy disk drives and the hard disk is defined as Physical Drive 5 (and possibly Physical Drives 6, 7 and 8 if you have two, three or four hard disks sitting out there.) These assignments were selected arbitrarily when MDZ/OS was designed.

Using the logical drive concept, MDZ/OS makes other programs "see" the segments as any combination of drive numbers you may want.

Although the segments are located on Physical Drive 5 (the hard disk,) you tell MDZ/OS through the menu to assign a set of segments so that the program thinks it is working on the Logical Drives you assigned.

Thus, the two concepts of segmentation and logical drives merge. You can assign segments (or work areas) any size you want (smaller than a floppy disk or as large as the hard disk) and you can assign them to appear to any application programs as any drive numbers you want.

This will, in all probability, make more sense once you begin installing your programs.

1.3 Turnkey Provisions

MDZ/OS allows each slave computer to "turnkey" (auto start-up) its own particular language and program. baZic, CP/M, and many programs are fully supported upon bootup by MDZ/OS. Each slave can execute its own program by entering the appropriate command (i.e. BAZIC) in the SLVTKEY file provided on the hard disk. Multiple commands can be executed by MicroDoZ by separating each command with a back slash (\).

All I/Os and CRT configurations under MDZ/OS I are stored in files. Each I/O file is overlayed by one of two program, OLAY1 for terminal configuring or OLAY2 for input-output routines. These files are located in the SYSTEM Segment.

The baZic programs TKEYEDIT,1 and SLTKEYED,5 allow the user to turnkey the software easily. In addition, the system may be made to execute different programs in each slave by a simple program that EXAMines the slave number byte and CHAINs to the appropriate program.

1.4 Printing

MDZ/OS supports multiple printer lockouts. This feature is to prevent the printouts from two or more users combining to form a chaotic printout on any one printer. When any user is printing to a particular printer, a flag is set signifying "this printer in use." If another user attempts to print on the same printer, that user will go into a wait loop until the printer is free and the flag is cleared.

The flag is cleared any time a READY, CHAIN, BYE or APPEND is encountered in baZic from slave which has been using the printer. The flag also is cleared by a CP/M bank on a warm boot. The printer also may be cleared by sending a "printer free" code, normally (!#1, CHR$(3),) or any other user-defined code from the slave that "has" a printer. The printer-free code is defined by the user in the I/O overlays. A maximum of 26 printers may be used simultaneously by MDZ/OS.

1.5 System Utilities

System utility programs operating under MicroDoZ with which you should be familiar included the following. Operation of each of these utilities is described in detail in Section 4.

CLOCK is a baZic program which utilizes the one-second interrupts on the SysteMaster board and provides the time of day both on the screen and makes the time available, through programming, for marking reports.

COPYFILE is a copy file routine which allows the user to specify a file-of-file-names file which is used by a program to make a backup or copy of data files. Manual entry of file names is supported or entry can be via a DOSCoMmanD string from a baZic program. When this program has completed a copy, baZic and any program can be chained to automatically.

ICOPY is a machine language program designed to let the user copy between any combination of floppy disks and segments defined on the hard disk. This program is useful for programmers and end-users alike because of its "intelligent" nature and its flexibility. This program runs in Auto or Single Step Mode and accepts entry either manually or from parameters passed in a DOSCMD string. ICOPY has the ability to load baZic and any baZic program upon completion of the copy.

FORMATFD is a machine language program which formats 8-inch disks in three different formats each for MicroDoZ or CP/M.

MENU is a control menu system which provides system security as well as the ability to assign segments and passwords to users of the system. The other main function of Menu is to allow CHAINing to other programs, assigning them to the right logical drives.

SEGED is a baZic program which allows the user to control the allocation of segments. The program creates, deletes, renames, lists, and prints the segment names. All information controlled by this program is located in a file in the system segment. This program uses byte accessing to its control file so that the program can be used with any precision of baZic. The SEGMENT file should be backed up each time the file is changed.

SEGINIT is a baZic program for initializing segments on the hard disk. This program performs the same function as the MicroDoZ command of "IN" on the floppy disks (format) and will initialize either a MicroDoZ or a CP/M segment. It must be remembered that initializing will destroy all information in a segment, so this program should be used with caution.

1.6 CP/M Utilities

The CP/M supplied on the disk is a special version of CP/M that has been relocated from its normal position. SYSGEN and MOVCPM are not supported and cannot be used under MDZ/OS. Each slave computer can run a 60K version of CP/M.

If a CP/M diskette is CATaloged or LIsted from MicroDoZ, a directory will be displayed, but uDoz uses 16-byte entries and CP/M directories are displayed in two lines, while a CP/M entry is 32 bytes and is displayed in one line.

EXIT.COM - A machine language program that allows the user to return from CP/M to MicroDoZ and the Control Menu.

1.7 Definition of Terms

Control MENU - The Control MENU is the first menu which the user sees upon bootup. This program lets the user branch to other menus for specific purposes. The Control Menu also is referred to as the MDZ/OS SYSTEM MENU, MENU ONE.

DISKTABLE - The DISKTABLE is the segment allocation table which is contained in RAM in the executive software. It is the table which allows different operating systems to co-exist on the hard disk. Since the table is in RAM and is available to all users, the table can be FILLed dynamically during program execution to change the segment or logical drive assignments.

MDZ/OS or Executive - The Executive is the program which controls the entire system and actually is two executives -- the master executive and the slave executive. The master executive controls all common tasks and communicates with the slaves. The slave executive is responsible for encoding and passing commands to the master. The slave executive resides in the top 4K of memory in each slave while the master executive occupies the entire 64K in the master computer.

The executive is written in machine langauge so as to execute the programs as quickly as possibly. The master executive contains the hard and floppy disk drivers, the DISKTABLE, and the task queuing routines.

Physical Drives - Physical drives refer to the actual disk drives attached to the computer. The physical drive assignment as defined by Micro Mike's, Inc. are: 1 to 4 are floppy disk drives and 5 to 8 are hard disk drives.

Logical Drives - MDZ/OS allows the user to assign the drive number to all segments created on the hard disk and to all other drives on the system. The drive number assignment may be any number from 1 to 7 and does not have to be related to the physical drive number. The drive numbers assigned by the user through the MENU system are called the logical drive assignments.

System Utility MENU - This menu contains all of the system utilities needed to assign, initialize, test, and determine the characteristics of the segments.

SEGMENT File - This file is perhaps the most important file in the MDZ/OS system since this file stores the allocation of ALL of the Segments defined on the hard disk. **This file should be backed up on floppy disk regularly.** The SEGMENT file is used to convert a logical disk address into a physical disk address. The file contains information on the names of segments, whether it is a CP/M or MicroDoZ segment, the directory size, and the offset on the hard disk where the segment is located. **This file should be backed up every time the segment allocation is changed** (SEGED is run.)

SYSTEM Segment - This segment is defined automatically by the system software and contains the I/Os and CRT configurations as well as several boot programs and a version of baZic for the hard disk. Also in this segment is the SEGMENT file that contains the record of all segment allocations. It must be 350 blocks long and the first segment on the disk. Needless to say, this is a very important segment which should be backed up any time it is changed.

SYSTEM1 Segment - This segment also is defined automatically by the system software. It contains the set of menus which allow access to segments defined on the hard disk. MENU is the main program supported in this segment. Each menu APPENDS the MENU program. Each menu also has an menu editor associated with it. The menu items are stored in the appropriate menu file. All precisions of baZic are supported in this segment. Any time a menu is added or an allocation changed, this segment should be backed up on floppy disk.

SEGMENTATION

The ability to form segments, each with characteristics of different operating systems, is one of the most importants concept in understanding the workings of MDZ/OS.

For a single hard disk drive to support multiple operating systems, the drive must be segmented so that each segment is a separate and unique part of the disk drive. MDZ/OS implements this feature by allowing the definition of segments of any size (limited by the size of the hard disk) and allowing these segments to have a directory of any reasonable size.

Each MicroDoZ Segment may have a directory of 255 (512 byte) blocks (sectors). Each block can support 32 16-byte entries. The maximum number of entries allowed in any one segment is 8160.

Each of these areas (Segments) has its own characteristics and each is capable of communicating in the normal manner to its I/O devices. To resolve any disk accessing or common I/O confusion, all disk and common I/O requests are routed through MDZ/OS, which does all disk accesses for all slaves.

If we were to look at the physical characteristics of the hard disk drive we would see that the disk consists of one or more recording platters. Each is divided into cylinders, tracks and sectors. Each sector has a unique address and can store 512 bytes of information.

A program which runs under CP/M or MicroDoZ normally "hands" the disk operating system a command to take some action on a disk, and an address of where the operation is to be performed. The definition of a segment is therefore any area (segment) on the disk that is set up to appear to the operating systems exactly like a drive would appear if the operating systems (CP/M or MicroDoZ) were still on a floppy disk-based systems.

MDZ/OS accomplishes this task by segmenting the hard disk drive into offsets of addresses to keep each area unique. MDZ/OS most simply represented, is a table of offsets, a method of keeping track of the offsets, and a scheme for coordinating the request for service from the slave processors. The use of a table insures quick disk access since the actual (physical) disk address can be computed using a constant value for each segment in use.

When a slave operating under either CP/M or MicroDoZ requests a disk access, the slave passes the request to the MDZ/OS software which is running on the SysteMaster board. MDZ/OS queues the task immediately and "works" on the task when it has time. If the request is a disk access, MDZ/OS then looks up the table value of the particular segment in use, and adds an offset value to the disk address before accessing the data. All access to the defined segment is given the same offset so that as far as the operating systems (CP/M and MicroDoZ) know, they are communicating in their normal manner to the disk.

2.1 Segment Naming

Keeping track of the segment names, characteristics, addresses and offsets, is accomplished by a baZic program, SEGED. This program, through the SEGMENT file in the SYSTEM Segment, stores the information needed to access each segment. SEGED allows the addition and deletion of segments, and has the capability of renaming segments. Segment names can be up to eight characters in length and can contain numbers and special symbols.

To be able to use MDZ/OS you must define the segments that you will need. If you were setting up the General Ledger on the hard disk you would need one segment set up for the programs. You might name this segment GLP and define it as being 350 blocks long with a directory of 4 blocks (MicroDoZ.)

Next you would will want to set up a segment to handle the data files since data files are usually stored on a drive other than the one the programs are on. You might name this segment GLD to stand for General Ledger Data Drive 1. Depending upon the amount of information to be stored, you could create this segment 700 blocks long, also with a directory of 4 blocks (MicroDoZ.)

If your computer keeps books for more than one company, the program segment would be needed as only one segment on the hard disk. If multiple sets of books are to be used, different segments should be created for each set of books (data files). They could be named GLD2, GLD3, etc. To access each set of books a different logical drive assignment should be used. In this way you would have one set of programs which can access the data of many sets of books (through logical drive assignment) but only one set of books could be used at any time.

To use the segmentation features of MDZ/OS efficiently you must have an understanding of the concept. Proper segments must be defined for all programs to be used and for all "drives" that will contain data files. Defining segments is the same as having the ability to custom design your own disk drives, making them as large or as small as you like. The System MENU provides easy access to the programs needed to create and assign segment allocations.

For example, let us assume that several persons will be doing word processing under WordStar which operates under CP/M. One segment, called WORDSTAR is established, to hold the WordStar programs and associated utiltities. Each user needing word processing capabilities would be assigned one or more data segments, each of varying sizes that meet the needs of the user -- one who does only correspondence would require a smaller segment than one producing manuals which will be several hundred pages in length.

All users could access the WORDSTAR segment as a common segment or read only (R/O), and each user then would have his or her data segments, from which the other users would be locked out.

As a general rule, all common segments (segments which would be used by two or more persons at the same time) should be used as Read Only, unless specific provisions have been made to lock files or records.

2.2 Logical Drive Assignment

Logical drive assignment is the ability to take any of the segments you have defined on the hard disk and any of the floppy drives on your system and be able to assign the drive number you want to each segment or drive. A drive which is normally considered Drive 1 may be assigned logically to any drive number you want, say Drive 6.

The advantage of this feature is enormous. If you already have an application package running on your system and it is configured so that all the programs are on Drive 1 and all the data files are on Drive 2, and you want this package to run on the hard disk the procedure is simple under MDZ/OS. First create the segments that you want and then assign them as Drives 1 and 2 (which would act as Drives A and B under CP/M.)

Make sure that you define at least one of your floppy drives as some unused drive number, say 6 or 7 (or Drive F under CP/M). After running your drive assignment program, you can copy your programs and data files from the floppy disks to the hard disk segments you have defined. This allows backing up to floppy disks.

At this time you should be able to run your programs on the hard disk with no or few modifications.

USING THE HARD DISK

This section assumes that your hardware is fully functional and that the hard disk drive has been formatted, mapped and the system software installed in the SYSTEM and SYSTEM1 Segments. At this time you should have the CONTROL MENU on your CRT. Using the hard disk is relatively easy from this point on.

This section is a "how to." If you want to know more about the "whys" and "wheres," consult the other sections of this manual and such publications as Micro Mike's, Inc. "Beginner's Guide to baZic."

3.1 The Menu System, Your Road Map

Everything in MDZ/OS starts from Menu 1 or the Control Menu and, if the number of programs require it, submenus and even sub-submenus.

When MDZ/OS is installed, only Menu 1 or Control Menu, CP/M Segments Menu and a System Utilities Menu exist. CP/M utilities must be added. A menu create program called as an option from the System Utilities Menu makes the addition of menus as needed a very simple process.

In a relatively simple system, the Control Menu or Menu 1 could appear as follows:

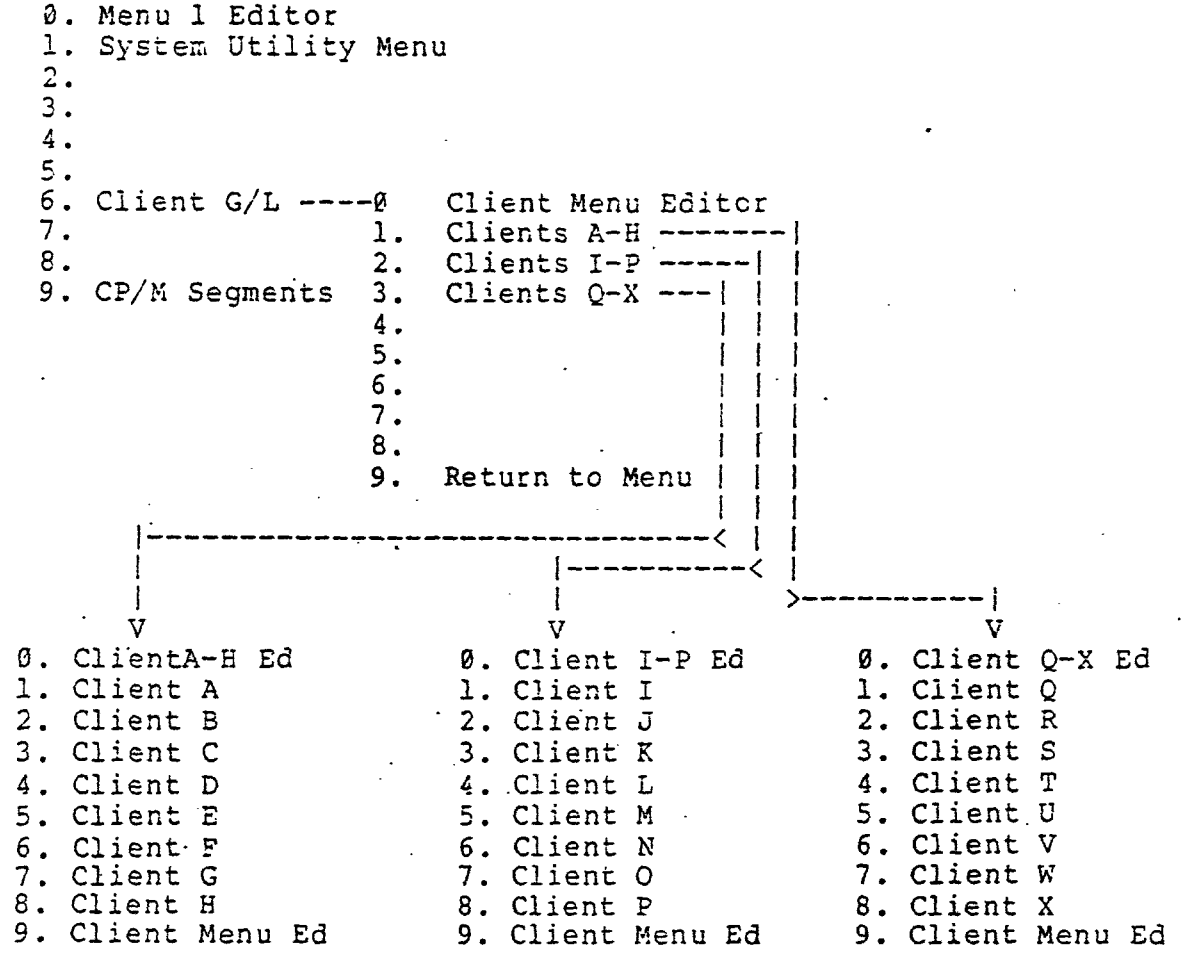
```
MDZ MENU SYSTEM MENU1
CHOOSE OPTION Ø..9
*

  Ø. MENU 1 EDITOR
  1. SYSTEM UTILITY MENU
  2. General Ledger
  3. Accounts Receivable
  4. Accounts Payable
  5. Payroll
  6. Inventory
  7. Appointment Calendar
  8. Customer Data
  9. CP/M SEGMENTS
```

This fills the Control Menu and if we have to add more MicroDoZ programs, we will need to consolidate some of the accounting functions, and then get to those programs through another menu.

With some planning, the menu system makes access to any program operating on the hard disk a relatively simple and easy process.

As an example of branching menus, let us assume that we operate an accounting service and we keep 24 sets of general ledgers for 24 clients. The number 24 is not magical; it just happens to fit the following diagram, which is just one of several ways to go about it:

Starting from the Control Menu, we could create the following menu system:



As you can see, from Item 6 on Menu 1, the Control Menu, one can branch to a Client Editor Menu, of which we have used only three of the eight available options. Working from this concept, we can add client general ledgers forever or until we run out of disk space, which ever comes first.

As we install each client's data files on the hard disk, we would need only one set of general ledger programs and by defining each client data file as a separate segment we can make MDZ/OS believe that each client is a set of two floppy disks, one for the programs and one for the data files. Two or more users can access the same general programs because nothing will be written or recorded on Drive 1 and then each can work in a different set of data files.

The beauty of this system is that data segments can be adjusted to meet the individual needs of the clients. Client A, for example, is a small business with only a few accounts within the chart of accounts and writes only a few checks per month and makes only a few deposits. You have estimated that 200 blocks will take care of his needs adequately.

Client B, on the other hand, has 200 accounts and writes several thousand checks per month and makes many deposits. Client B data files will require 1000 blocks.

3.2 Moment of Truth -- Installing the Programs

You understand the concepts presented completely, and through carefull planning, you know to the block (512 bytes in this case) just how big each segment must be. (Well, at least you think you have a fair idea of what you are doing and you have given some thoughts to program and data file sizes.)

Installing the programs and setting up data files is as simple as one through seven. Patience at this point is a virtue. Give yourself enough time to think and go through the procedures. Don't get in a hurry and don't try to get fancy. More mistakes happen in haste than for any other reason. (Another axiom of the computer profession is that trouble (mistakes) always occur at the most incpportune time. When all else fails, this axiom will not.)

Whether the software you will install operates under MicroDoZ or CP/M, the steps are the same, but with some variations. We will go through the installation of each. If you are installing under CP/M, continue reading to Section 3.3, then skip to Section 3.4.

The steps are as follows:

1. Create the segment
2. Initialize the segment
3. Put it on the menu
4. Copy the programs and data files from floppy disks
5. Re-edit the menu to add the turnkey command
6. Fix program to chain back to menu system
7. Try it out and make sure it works

For demonstration purposes, we are going to install Micro Mike's, Inc. fabulous General Ledger for our company, the Testing Company, and our in-house accountant wants to be able to reach the General Ledger from Menu 1 or the Control Menu. He has selected Option 6 as an easy number to remember. The General Ledger will be installed under MicroDoZ (uDoZ).

For the CP/M demonstration, we are going to install the CP/M utilities and because our secretary-Gal Friday likes WordStar, we will install the WordStar programs on that segment.

We will describe all of the steps involved in the installation in detail in the following sections.

3.3 Installing Programs under MicroDoZ

To install a program under MicroDoZ, follow these procedures:

3.3.1 Creating Segments

The first step will be to create drives (segments) for your programs and their associated data files. Your system has been turned on and Menu 1 or the Control Menu is on the screen as follows:

```
MDZ MENU SYSTEM MENU1
CHOOSE OPTION 0..9
*

  0. MENU 1 EDITOR
  1. SYSTEM UTILITY MENU
  2.
  3.
  4.
  5.
  6.
  7.
  8.
  9. CP/M SEGMENTS
```

From Menu 1, select Option 1, the System Utility Menu which will appear as follows:

```
MDZ MENU SYSTEM UTILITIES
CHOOSE OPTION 0..9
#

  0. SYSTEM MENU EDITOR
  1. SEGMENT EDITOR
  2. SEGMENT INITIALIZE
  3. CREATE NEW MENUS
  4. CLOCK
  5. EDIT SLAVE TURNKEY
  6. NORMAL
  7. JOETEST
  8.
  9. RETURN TO MENU 1
```

From this point select Option 1, Segment Editor. The first prompt will be:

```
                *** SEGMENT EDITOR ***
PHYSICAL DRIVE NUMBER 5-8
#
```

Press 5 and the SEGMENT EDITOR will be displayed as follows:

(The following examples show M26 disk space values.)

```
*** SEGMENT EDITOR ***                    MAX DISK SIZE = 51200
CHOOSE 'E'XIT,'L'IST,'P'RINT,'C'REATE,'D'ELETE,'R'ENAME,'N'EWDISK
*
```

SEG NAME	PHY DRV	BLOCKS	DIR SIZE	OFFSET	UNUSED	UNUSED
SYSTEM	5	350	4	0	0	0
CPM1	5	2000	CP/M	350	0	0
SYSTEM1	5	350	4	2350	0	0
MT SPACE	5	48500	4	2700	0	0

(Note: The CPM1 Segment is not created when the system software is installed. We are assuming that the dealer or programmer put this CPM1 segment on.)

You will need to create segments based upon the requirements of your software.

For our example, our Micro Mike's, Inc., General Ledger programs require a two-drive system; one drive for programs and one drive for data files. We will call the program segment GLP and the data segment GLD. To create these two segments enter a C in response to the Segment Editor's prompt.

The next prompt will be:

```
SEGMENT NAME
********
```

Enter GLP and Press Return. The next next prompt will be:

```
HOW MANY SECTORS
#####
```

Enter 350 and Press Return and the next prompt will be:

```
CP/M OR MICRODOZ SEGMENT ?
*
```

Enter M for MicroDoZ. You do not have to press Return. The next prompt will be:

```
HOW MANY SECTORS FOR DIRECTORY
###
```

Enter 4 and press Return. This creates the program segment with a directory of 4 (this is the normal situation and should be used in all cases unless you specifically need more directory space.) Note that your entries now appear next to the bottom of the display as follows:

```
*** SEGMENT EDITOR ***                    MAX DISK SIZE = 51200
CHOOSE 'E'XIT,'L'IST,'P'RINT,'C'REATE,'D'ELETE,'R'ENAME,'N'EWDISK
*
```

SEG NAME	PHY DRV	BLOCKS	DIR SIZE	OFFSET	UNUSED	UNUSED
SYSTEM	5	350	4	0	0	0
CPM1	5	2000	CP/M	350	0	0
SYSTEM1	5	350	4	2350	0	0
GLP	5	350	4	2700	0	0
MT SPACE	5	48150	4	3050	0	0

Now create the data segment by the same sequence. This time we will create the data segment 700 blocks instead of 350. Enter a "C" to create the data segment.

The next prompt will be:

```
SEGMENT NAME
********
```

Enter GLD and Press Return. The next next prompt will be:

```
HOW MANY SECTORS
#####
```

Enter 700 and Press Return and the next prompt will be:

```
CP/M OR MICRODOZ SEGMENT ?
*
```

Enter M for MicroDoZ. You do not have to press Return. The next prompt will be:

```
HOW MANY SECTORS FOR DIRECTORY
###
```

Enter 4 and press Return. The program will again display the segment allocations. The display will appear as follows:

```
*** SEGMENT EDITOR ***                    MAX DISK SIZE = 51200
CHOOSE 'E'XIT,'L'IST,'P'RINT,'C'REATE,'D'ELETE,'R'ENAME,'N'EWDISK
*
```

SEG NAME	PHY DRV	BLOCKS	DIR SIZE	OFFSET	UNUSED	UNUSED
SYSTEM	5	350	4	0	0	0
CPM1	5	2000	CP/M	350	0	0
SYSTEM1	5	350	4	2350	0	0
GLP	5	350	4	2700	0	0
GLD	5	700	4	3050	0	0
MT SPACE	5	47450	4	3750	0	0

We have now created the two segments needed. Press E for exit and return to the System Utility Menu which appears as follows:

```
MDZ MENU SYSTEM UTILITIES
CHOOSE OPTION Ø..9
#

  Ø. SYSTEM MENU EDITOR
  1. SEGMENT EDITOR
  2. SEGMENT INITIALIZE
  3. CREATE NEW MENUS
  4. CLOCK
  5. EDIT SLAVE TURNKEY
  6. NORMAL
  7. JOETEST
  8.
  9. RETURN TO MENU 1
```

3.3.2 Initializing the Segments

From the System Utility Menu select Option 2, Segment Initialize. The first prompt will ask for the physical drive number. The display will appear as follows:

```
                *** SEGMENT INITIALIZE ***
ENTER PHYSICAL DRIVE NUMBER 5...8
*
```

As for all physical drive numbers, Drive 5 is the first hard disk on your system, Drive 6 is the second and so on. Enter the drive number (5) and the next prompt will be:

```
ENTER SEGMENT NAME TO INITIALIZE
********
```

Enter the name of the first segment, GLP. The program will inform the user that the program is initializing the segment for MicroDoZ. After the segment is initialized the program will ask if you have another segment to be initialized. The display will appear as follows:

```
INITIALIZE MORE SEGMENTS Y OR N
*
```

Since you do have more, respond with a Y for Yes and then the segment name, GLD. After this segment is initialized respond with an N for No to indicate no further segments are to be initialized. The program will branch back to the System Utility Menu which will be as follows:

```
MDZ MENU SYSTEM UTILITIES
CHOOSE OPTION Ø..9
#

  Ø. SYSTEM MENU EDITOR
  1. SEGMENT EDITOR
  2. SEGMENT INITIALIZE
  3. CREATE NEW MENUS
  4. CLOCK
  5. EDIT SLAVE TURNKEY
  6. NORMAL
  7. JOETEST
  8.
  9. RETURN TO MENU 1
```

From this point, select Option 9, Return to Menu 1.

3.3.3 Assigning Logical Drives

We have created and initialized our segments. The next step will involve assigning logical drive numbers to these segments so that we can use them. The assignment of drives will be done through the MENU system.

Since the General Ledger package is normally run with the program disk in Drive 1 and the data disk in Drive 2, we will define the logical drives on the hard disk the same way. GLP will be defined as Logical Drive 1 and GLD will be defined as Logical Drive 2.

We will define the two floppy drives as Logical Drives 6 and 7. This is done so that we can copy the disks to the hard disk into the segments we have just created and initialized.

You should now have Menu 1 or the Control Menu on the screen as follows:

```
MDZ MENU SYSTEM MENU1
CHOOSE OPTION Ø..9
*

  Ø. MENU 1 EDITOR
  1. SYSTEM UTILITY MENU
  2.
  3.
  4.
  5.
  6.
  7.
  8.
  9. CP/M SEGMENTS
```

We can select any blank MENU item number we want for this next procedure and insert the name, General Ledger. We could even add another menu to branch to the General Ledger and other accounting programs. These decisions should be made by the person setting up the system. Some thought should be given to the way the entire system is to be set up to take advantage of passwords and how different persons will be accessing the different programs.

In this example we will select Option 6 and put our General Ledger program in that slot. First press Ø, Menu1 Editor and the next prompt will be:

```
SELECT OPTION (E TO EDIT)
*
```

Enter an E for Edit so that we can edit Option 6. The prompt will now read:

```
CHOOSE OPTION TO EDIT (Ø TO 9), RETURN TO SKIP
*
```

Enter 6 and the following will be displayed on the screen with the prompt:

```
MDZ MENU EDITOR MENU1 ITEM 6
CHOOSE OPTION 1..14 TO EDIT OR RETURN TO SAVE
##

1.  OPTION NAME
2.  PASSWORD
3.  USER ACCESS      11111111111111
4.  PRECISION OF baZic
                        SEGMENT NAME   PHYSICAL DRIVE    LOCK
5.  LOGICAL DRIVE 1
6.  LOGICAL DRIVE 2
7.  LOGICAL DRIVE 3
8.  LOGICAL DRIVE 4
9.  LOGICAL DRIVE 5
1Ø. LOGICAL DRIVE 6
11. LOGICAL DRIVE 7
12. MICRODOZ
13. CHAIN PROGRAM
```

We will need to enter the required data one field at a time, specifically Fields 1, 5, 6, 1Ø, and 11. If we want to use a password we also will need to edit Field 2, and if we want to limit user access, Field 3. These last two are not essential, however, and will not be included here. Instructions will be found in Section 4.1.1Ø, Menu.

Start with Field 1, Option Name. Press 1 and Return and the screen will be erased and replaced with the following prompt:

```
MDZ MENU SYSTEM
ENTER NEW OPTION NAME OR RETURN TO LEAVE SAME
**********
```

Option name is the name that you want to appear on the menu when you finish editing. The limit is 20 spaces, with each letter, number, character or space, counting as a space.

Enter the option name, in this case, General Ledger. The prompt character field will not be filled, so press Return. The display will return to the screen with option name in place as follows:

```
MDZ MENU EDITOR MENU1 ITEM 6
CHOOSE OPTION 1..14 TO EDIT OR RETURN TO SAVE
##

1.  OPTION NAME      General Ledger
2.  PASSWORD
3.  USER ACCESS      1111111111111111
4.  PRECISION OF baZic
                        SEGMENT NAME   PHYSICAL DRIVE    LOCK
5.  LOGICAL DRIVE 1
6.  LOGICAL DRIVE 2
7.  LOGICAL DRIVE 3
8.  LOGICAL DRIVE 4
9.  LOGICAL DRIVE 5
10. LOGICAL DRIVE 6
11. LOGICAL DRIVE 7
12. MICRODOZ
13. CHAIN PROGRAM
```

As each field is called for editing, the prompts will appear, and as information is entered in each field, the menu mask will reappear so that you know where you stand at all times. The following are the fields and prompts which are essential to the operation of General Ledger under MicroDoZ.

Press 4 and the following will be displayed:

```
MDZ MENU SYSTEM
ENTER BAZIC PRECISION 8, 10, 12, 14 OR 0 IF NO RESTRICTION
##
```

In this case, General Ledger is in 10-digit precision baZic. Enter 10.

Fields 5 through 11 are for the logical-physical drives. We are going to edit Line Numbers 5, 6, 9, 10, and 11. Press 5 and the following will appear with the prompt:

```
MDZ MENU SYSTEM
LOGICAL DRIVE 1 ENTER SEGMENT NAME
*********
```

Enter the segment name, in this case, GLP, and press Return. The the next prompt will be:

```
MDZ MENU SYSTEM
ENTER PHYSICAL DRIVE NUMBER 1..8
#
```

Enter 5 for the physical drive number and the next prompt will be:

```
MDZ MENU SYSTEM
CHOOSE Ø FOR NO, 1 FOR YES SEGMENT LOCKING
#
```

Since this section is holding the General Ledger programs, let's not lock it by entering a Ø. The menu editor will now look like this:

```
MDZ MENU EDITOR MENU1 ITEM 6
CHOOSE OPTION 1..14 TO EDIT OR RETURN TO SAVE
##

1.  OPTION NAME      General Ledger
2.  PASSWORD
3.  USER ACCESS      1111111111111111
4.  PRECISION OF baZic  10
                        SEGMENT NAME   PHYSICAL DRIVE     LOCK
5.  LOGICAL DRIVE 1     GLP                  5             Ø
6.  LOGICAL DRIVE 2
7.  LOGICAL DRIVE 3
8.  LOGICAL DRIVE 4
9.  LOGICAL DRIVE 5
10. LOGICAL DRIVE 6
11. LOGICAL DRIVE 7
12. MICRODOZ
13. CHAIN PROGRAM
```

Remember the dissertation on logical-physical drives in Sections 1 and 2? We are directing MDZ/OS to tell the computer that the General Ledger programs are on Drive 1, when actually, the programs are located in the GLP segment on the hard disk, Drive 5. The segment is not locked and therefore it can be accessed by other users, if need be.

Next press 6. The prompts will be repeated as follows:

```
MDZ MENU SYSTEM
LOGICAL DRIVE 2 ENTER SEGMENT NAME
*********
```

Enter the segment name, in this case, GLD, and press Return. The the next prompt will be:

```
MDZ MENU SYSTEM
ENTER PHYSICAL DRIVE NUMBER 1..8
#
```

Enter 5 for the physical drive number and the next prompt will be:

```
MDZ MENU SYSTEM
CHOOSE Ø FOR NO, 1 FOR YES SEGMENT LOCKING
#
```

Since this segment is holding the General Ledger data files, let's lock it by entering a 1. When one user is working in this general ledger, no other user can access this segment.

If we were keeping two sets of general ledgers, say one for a second division within the Testing Company, all we would need to do is fill in another menu for that division, set up a segment for that set of data files and lock it. In this way, two persons could be working in the two general ledgers at the same time, both using the same set of programs, but working in different sets of data files.

Next press 10. The prompts will be repeated as follows:

```
MDZ MENU SYSTEM
LOGICAL DRIVE 6 ENTER SEGMENT NAME
*********
```

Enter the segment name, in this case, FDMDOZ, which is a name through which MDZ/OS tells the computer that when 6 is selected as the source or destination drive, turn on Floppy Disk Drive 1. Press Return and the next prompt will be:

```
MDZ MENU SYSTEM
ENTER PHYSICAL DRIVE NUMBER 1..8
#
```

Enter 1 for the physical drive number and the next prompt will be:

```
MDZ MENU SYSTEM
CHOOSE Ø FOR NO, 1 FOR YES SEGMENT LOCKING
#
```

Let's not lock it, by entering a Ø.

Next press 11. The prompts will be repeated and the only difference will be to enter 2 in the prompt:

```
MDZ MENU SYSTEM
ENTER PHYSICAL DRIVE NUMBER 1..8
#
```

Now that all of the essential fields to this point have been entered, the display will look like this:

```
MDZ MENU EDITOR MENU1 ITEM 6
CHOOSE OPTION 1..14 TO EDIT OR RETURN TO SAVE
##

1.  OPTION NAME      General Ledger
2.  PASSWORD         Pass
3.  USER ACCESS       11011111111111111
4.  PRECISION OF baZic  10
                        SEGMENT NAME   PHYSICAL DRIVE     LOCK
5.  LOGICAL DRIVE 1     GLP                   5             0
6.  LOGICAL DRIVE 2     GLD                   5             0
7.  LOGICAL DRIVE 3
8.  LOGICAL DRIVE 4
9.  LOGICAL DRIVE 5
10. LOGICAL DRIVE 6     FDMDOZ                1             0
11. LOGICAL DRIVE 7     FDMDOZ                2             0
12. MICRODOZ
13. CHAIN PROGRAM
```

This assignment directs that floppy disks (FDMDOZ) be defined as Drives 6 and 7 and the two hard disk segments to be defined as as Drives 1 and 2. Logical Drives 3, 4 and 5 are not defined. If you try to get a CATalog of Drive 3, the program will respond with a hard disk error. Both Drives 4 and 5 can be cataloged.

NOTE: CP/M Segments may be accessed from MicroDoZ but MicroDoZ segments cannot be accessed from CP/M.

Enter a Return only to save this information in the file and return to the Control Menu which now should appear as follows:

```
MDZ MENU SYSTEM MENU1
CHOOSE OPTION 0..9
*

  0. MENU 1 EDITOR
  1. SYSTEM UTILITY MENU
  2.
  3.
  4.
  5.
  6. General Ledger
  7.
  8.
  9. CP/M SEGMENTS
```

3.3.4 Copying Your MicroDoZ Programs on the Hard Disk

Select Option 6, General Ledger, and you will be transported into the segment allocations we have just made. The program will respond with the following message to indicate that no chain program was named:

```
SEGMENTS ON
NO CHAIN PROGRAM
READY
```

We will use ICOPY (from Drive 5) to copy the programs into the proper segments. Place your two floppy disks that you want copied to the hard disk in Drives 1 and 2 as you would if you were running them on the floppies. The program disk should go into Drive 1 and the data disk in Drive 2. You will be copying the program disk from Drive 6 to Drive 1 and the data disk from Drive 7 to Drive 2. The sequence will be as follows, with this on the screen:

```
SEGMENTS ON
NO CHAIN PROGRAM
READY
```

Enter the following command:

```
DOSCMD "ICOPY,5\\BAZIC1Ø,5\CONTROL,5"
```

Press Return and the following will appear on the screen with the prompt:

```
SYSTEMASTER INTELLIGENT COPY COPYRIGHT(C)1981, MICRO MIKE'S, INC.
INSERT DISKETTES THEN PRESS ANY KEY
```

The disks are already in the drives, so press any key and the next prompt will be:

```
INPUT SOURCE DRIVE NUMBER
#
```

Enter the source drive, which in this case is 6. The next prompt will be:

```
INPUT DESTINATION DRIVE NUMBER
#
```

Enter 1 for the destination drive. The next prompt will be:

```
CHOOSE A)UTO OR S)INGLE STEP
*
```

Enter A. The intelligent copy routine will flash the name of each file as it is being copied and the display will be like this:

```
FIRSTPGM C

  etc.

LASTPGM C
```

After the copy is complete the screen will be erased and the following will be displayed:

```
SYSTEMASTER INTELLIGENT COPY COPYRIGHT(C)1981, MICRO MIKE'S, INC.
     INPUT E)XIT OR C)ONTINUE
     *
```

We still have another diskette to copy, so press C and we'll start the routine again with only minor variations:

```
SYSTEMASTER INTELLIGENT COPY COPYRIGHT(C)1981, MICRO MIKE'S, INC.
INSERT DISKETTES THEN PRESS ANY KEY
```

Press any key and the next prompt will be:

```
     INPUT SOURCE DRIVE NUMBER
     #
```

Enter the source drive, which in this case is 7. The next prompt will be:

```
     INPUT DESTINATION DRIVE NUMBER
     #
```

Enter 2 for the destination drive. The next prompt will be:

```
     CHOOSE E)XIT A)UTO OR S)INGLE STEP
     *          (enter A)
```

Again, press A. The intelligent copy routine will flash the name of each file as it is being copied and the display will be like this:

```
     FIRSTPGM C

       etc.

     LASTPGM C
```

When the second copy is completed, enter "E" to exit in response to the prompt:

```
     INPUT E)XIT OR C)ONTINUE
     *          (enter E)
```

At this point we should have the programs installed. If there are no FILLs, CALLs or IF EXAMs in the programs, the programs should now be operational.

3.3.5 Adding the Turnkey Command

Because we want a complete turnkey sequence, we should go back to the Control Menu and edit Item 6. We need to add the turnkey program name so that when Item 6, General Ledger, is selected from the menu, it will CHAIN automatically to the proper program in the General Ledger, which happens to be CSUB.

The Control Menu should be on the screen. First enter Ø to get the editor and then enter E for edit and then press 6. The menu editor will be on the screen with the prompt:

```
MDZ MENU EDITOR MENU1 ITEM 6
CHOOSE OPTION 1..14 TO EDIT OR RETURN TO SAVE
##
```

We want to edit Field 12 and the prompt will be:

```
MDZ MENU SYSTEM
ENTER CHAIN PROGRAM, OR CPMIOFILE, OR RETURN TO LEAVE UNCHANGED
**********
```

Enter CSUB and press Return. The display will appear as follows:

```
MDZ MENU EDITOR MENU1 ITEM 6
CHOOSE OPTION 1..14 TO EDIT OR RETURN TO SAVE
##

1.  OPTION NAME      General Ledger
2.  PASSWORD         Pass
3.  USER ACCESS      1111111111111111
4.  PRECISION OF baZic  10
                     SEGMENT NAME   PHYSICAL DRIVE    LOCK
5.  LOGICAL DRIVE 1  GLP                 5             Ø
6.  LOGICAL DRIVE 2  GLD                 5             Ø
7.  LOGICAL DRIVE 3
8.  LOGICAL DRIVE 4
9.  LOGICAL DRIVE 5
1Ø. LOGICAL DRIVE 6  FDMDOZ              1             Ø
11. LOGICAL DRIVE 7  FDMDOZ              2             Ø
12. MICRODOZ
13. CHAIN PROGRAM    CSUB
```

The other consideration when installing a program is a means by which the user can return to the Control Menu.

Micro Mike's, Inc. General Ledger is delivered with this provision in place. If you are installing a program which does not have this provision, get your friendly programmer to make the necessary changes. (Suggest that your friendly programmer review "Programming for MDZ/OS," found in this manual.)

In the absence of the program changes, the path of least resistance is to do what programmers call a "Control C." Any time this is tried, be sure that you are at a point where the program can be stopped. Some programs wait until going back to a certain point to write data into the files.

With one finger holding down the Control key (marked CTRL on some terminals, use another finger to press the C key.) baZic will respond something like the following:

```
STOP IN LINE XXX
READY
```

Enter the command:

```
LOAD CONTROL,5
```

Press Return and another Ready will appear:

```
LOAD CONTROL,5
READY
```

Enter RUN and press Return again and the program will return to the Control Menu.

3.4 Installing Programs under CP/M

The following sections describe in detail the procedures for installing programs under CP/M. We are going to create first a CP/M segment and install the CP/M utilities and then add WordStar to the segment.

3.4.1 Creating Segments for CP/M

The first step will be to create a drive or segment for the utilities and WordStar. We are going to create the segment large enough so that WordStar files can be created in the same segment. Your system has been turned on and Menu 1 or the Control Menu is on the screen and appears as follows:

```
MDZ MENU SYSTEM MENU1
CHOOSE OPTION Ø..9
*

  Ø. MENU 1 EDITOR
  1. SYSTEM UTILITY MENU
  2.
  3.
  4.
  5.
  6.
  7.
  8.
  9. CP/M SEGMENTS
```

From Menu 1, select Option 1, System Utility Menu, which will appear as follows:

```
MDZ MENU SYSTEM UTILITIES
CHOOSE OPTION Ø..9
#

  Ø. SYSTEM MENU EDITOR
  1. SEGMENT EDITOR
  2. SEGMENT INITIALIZE
  3. CREATE NEW MENUS
  4. CLOCK
  5. EDIT SLAVE TURNKEY
  6. NORMAL
  7. JOETEST
  8.
  9. RETURN TO MENU 1
```

From this point select Option 1, Segment Editor. The first prompt will be:

```
                    *** SEGMENT EDITOR ***
PHYSICAL DRIVE NUMBER 5-8
#
```

Press 5 and the SEGMENT EDITOR will be displayed as follows:

```
*** SEGMENT EDITOR ***                    MAX DISK SIZE = 51200
CHOOSE 'E'XIT,'L'IST,'P'RINT,'C'REATE,'D'ELETE,'R'ENAME,'N'EWDISK
*

SEG NAME  PHY DRV   BLOCKS  DIR SIZE  OFFSET  UNUSED  UNUSED
SYSTEM        5      350        4        Ø       Ø       Ø
SYSTEM1       5      350        4      350       Ø       Ø
MT SPACE      5    50500        4      700       Ø       Ø
```

Remember, you will need to create segments based upon the requirements of your software.

For our example, we are going to create two segments called CPM and CPM1 and make them large enough to handle our needs and have room for some data files. To create these segments, press C in response to the Segment Editor's prompt. The next prompt will be:

```
SEGMENT NAME
********
```

A segment name can be any group of letters, numbers, or characters eight spaces or fewer in length, with each letter, number or character counting as a space. Enter CPM and the next prompt will be:

```
HOW MANY SECTORS
#####
```

Enter 1000 and the next prompt will be:

```
CP/M OR MICRODOZ SEGMENT ?
*
```

Enter a C for CP/M. This creates the segment and returns you to the Segment Editor. Note that your entries now appear next to the bottom of the display as follows:

```
*** SEGMENT EDITOR ***                          MAX DISK SIZE = 51200
CHOOSE 'E'XIT,'L'IST,'P'RINT,'C'REATE,'D'ELETE,'R'ENAME,'N'EWDISK
*

SYSTEM        5        350       4          0        0        0
SYSTEM1       5        350       4        700        0        0
CPM           5       1000    CP/M       1700        0        0
MT SPACE      5      49500       4       1700        0        0
```

Press C again so we can create another segment following the same routine. The next prompt will be:

```
SEGMENT NAME
********
```

Enter CPM1 and the next prompt will be:

```
HOW MANY SECTORS
#####
```

Enter 1000 and the next prompt will be:

```
CP/M OR MICRODOZ SEGMENT ?
*
```

Enter a C for CP/M. This creates the segment and returns you to the Segment Editor. Note how the Segment Editor displays the second entry as follows:

```
*** SEGMENT EDITOR ***                          MAX DISK SIZE = 51200
CHOOSE 'E'XIT,'L'IST,'P'RINT,'C'REATE,'D'ELETE,'R'ENAME,'N'EWDISK
*

SYSTEM        5        350       4          0        0        0
SYSTEM1       5        350       4        700        0        0
CPM           5       1000    CP/M       1700        0        0
CPM1          5       1000    CP/M       2700        0        0
MT SPACE      5      48500       4       2700        0        0
```

At this point, press E for Exit so that you can return to the System Utility Menu which appears as follows:

```
MDZ MENU SYSTEM UTILITIES
CHOOSE OPTION Ø..9
#

  Ø. SYSTEM MENU EDITOR
  1. SEGMENT EDITOR
  2. SEGMENT INITIALIZE
  3. CREATE NEW MENUS
  4. CLOCK
  5. EDIT SLAVE TURNKEY
  6. NORMAL
  7. JOETEST
  8.
  9. RETURN TO MENU 1
```

3.4.2 Initializing the Segments

From the System Utility Menu select Option 2, Segment Initialize. The first prompt will ask for the physical drive number. The display will appear as follows:

```
                *** SEGMENT INITIALIZE ***
ENTER PHYSICAL DRIVE NUMBER 5...8
*
```

As for all physical drive numbers, Drive 5 is the first hard disk on your system, Drive 6 is the second and so on. Enter the drive number (5) and the next prompt will be:

```
ENTER SEGMENT NAME TO INITIALIZE
********
```

Enter the name of the segment, CPM. The program will inform the user that the program is initializing the segment for CP/M. After the segment is initialized the program will ask you if you have another segment to initialize. The display will appear as follows:

```
INITIALIZE MORE SEGMENTS Y OR N
*
```

We have one more segment to initialize, so press Y for Yes, and the prompt will be repeated as follows:

```
                *** SEGMENT INITIALIZE ***
ENTER PHYSICAL DRIVE NUMBER 5...8
*
```

Enter the drive number (5) and the next prompt will be:

```
ENTER SEGMENT NAME TO INITIALIZE
********
```

Enter the name of the segment, CPM1. The program will inform the user that the program is initializing the segment for CP/M. After the segment is initialized the program will ask you if you have another segment to initialize. The prompt will be:

```
INITIALIZE MORE SEGMENTS Y OR N
*
```

Since you have no more, respond with an N for No to indicate no further segments are to be initialized. The program will branch back to the System Utility Menu which will be as follows:

```
MDZ MENU SYSTEM UTILITIES
CHOOSE OPTION Ø..9
#

  Ø. SYSTEM MENU EDITOR
  1. SEGMENT EDITOR
  2. SEGMENT INITIALIZE
  3. CREATE NEW MENUS
  4. CLOCK
  5. EDIT SLAVE TURNKEY
  6. NORMAL
  7. JOETEST
  8.
  9. RETURN TO MENU 1
```

From this point, select Option 9, Return to Menu 1.

3.4.3 Assigning Logical Drives

We have created and initialized our segment, and now we will assign logical drive numbers so we can use our segment. The assignment of drives will be done through the MENU system.

We will assign a segment as Drive 1 for the WordStar programs. Drive 2 will be assigned as another segment for data storage.

We also will define one of the floppy drives as Logical Drive 3, so that we can copy the CP/M utilities and later the WordStar programs into the segment we just created and initialized.

You should now have Menu 1 or the Control Menu on the screen as follows:

```
MDZ MENU SYSTEM MENU1
CHOOSE OPTION 0..9
*

  0. MENU 1 EDITOR
  1. SYSTEM UTILITY MENU
  2.
  3.
  4.
  5.
  6.
  7.
  8.
  9. CP/M SEGMENTS
```

All programs operating under CP/M will be called from Option 9, CPM SEGMENTS. Press 9 and after a moment, we will see on the screen:

```
MDZ MENU SYSTEM CP/M
CHOOSE OPTION 0..9
*

  0. MENU 1 EDITOR
  1.
  2.
  3.
  4.
  5.
  6.
  7.
  8.
  9. RETURN TO MENU 1
```

We can select any blank MENU item number we want for this next procedure and insert WordStar into the slot. These decisions should be made by the person setting up the system. Some thought should be given to the way the entire system is to be set up to take advantage of passwords and how different persons will be accessing the system.

In this example we will select Option 1 and put WordStar into that slot. First press 0, Menu 1 Editor and the next prompt will be:

```
SELECT OPTION (E TO EDIT)
*
```

Press E for Edit so that we can add our package to the menu. The prompt will now read:

```
CHOOSE OPTION TO EDIT (0 TO 9), RETURN TO SKIP
*
```

Select Option 1 and the following will be displayed on the screen with the prompt:

```
MDZ MENU EDITOR CPM SEGMENTS ITEM 1
CHOOSE OPTION 1..14 TO EDIT OR RETURN TO SAVE
##

1.  OPTION NAME
2.  PASSWORD
3.  USER ACCESS      1111111111111111
4.  PRECISION OF baZic
                        SEGMENT NAME   PHYSICAL DRIVE    LOCK
5.  LOGICAL DRIVE 1
6.  LOGICAL DRIVE 2
7.  LOGICAL DRIVE 3
8.  LOGICAL DRIVE 4
9.  LOGICAL DRIVE 5
10. LOGICAL DRIVE 6
11. LOGICAL DRIVE 7
12. MICRODOZ
13. CHAIN PROGRAM
```

We will need to enter the required data a field at a time, specifically Fields 1, 5, 6, and 12

Start with Field 1, Option Name. Press 1 and Return and the screen will be erased and replaced with the following display and prompt:

```
MDZ MENU SYSTEM
ENTER NEW OPTION NAME OR RETURN TO LEAVE SAME
**********
```

Option name is the name that you want to appear on the menu when you finish editing. The limit is 20 spaces, using any letter, number, character or space or any combination and counting each as a space.

Enter the option name, in this case, WordStar (or WordStar for Jane) and press Return because the prompt character field is not filled. The display will return to the screen with option name in place as follows:

```
MDZ MENU EDITOR CPM SEGMENTS ITEM 1
CHOOSE OPTION 1..14 TO EDIT OR RETURN TO SAVE
##

1.  OPTION NAME      WordStar
2.  PASSWORD
3.  USER ACCESS      1111111111111111
4.  PRECISION OF baZic
                        SEGMENT NAME   PHYSICAL DRIVE    LOCK
5.  LOGICAL DRIVE 1
6.  LOGICAL DRIVE 2
7.  LOGICAL DRIVE 3
8.  LOGICAL DRIVE 4
9.  LOGICAL DRIVE 5
10. LOGICAL DRIVE 6
11. LOGICAL DRIVE 7
12. MICRODOZ
13. CHAIN PROGRAM
```

Now that we've seen how the menu editor disappears and reappears, let's move on to the next fields. Press 5 and the following prompt will appear:

```
MDZ MENU SYSTEM
LOGICAL DRIVE 1 ENTER SEGMENT NAME
*********
```

Enter the segment name, in this case, CPM, and press Return. The the next prompt will be:

```
MDZ MENU SYSTEM
ENTER PHYSICAL DRIVE NUMBER 1..8
#
```

Enter 5 for the physical drive number and the next prompt will be:

```
MDZ MENU SYSTEM
CHOOSE Ø FOR NO, 1 FOR YES SEGMENT LOCKING
#
```

We must leave it unlocked because we are going to write to it. Press Ø.

```
MDZ MENU EDITOR MENU1 ITEM 6
CHOOSE OPTION 1..14 TO EDIT OR RETURN TO SAVE
##

1.  OPTION NAME      WordStar
2.  PASSWORD
3.  USER ACCESS       1111111111111111
4.  PRECISION OF baZic
                        SEGMENT NAME   PHYSICAL DRIVE     LOCK
5.  LOGICAL DRIVE 1     CPM                   5             Ø
6.  LOGICAL DRIVE 2
7.  LOGICAL DRIVE 3
8.  LOGICAL DRIVE 4
9.  LOGICAL DRIVE 5
10. LOGICAL DRIVE 6
11. LOGICAL DRIVE 7
12. MICRODOZ
13. CHAIN PROGRAM
```

Remember the explanations on logical-physical drives in Sections 1 and 2? We are directing MDZ/OS to tell the computer that the WordStar programs are on Drive 1, when actually, the programs are located in the CPM1 segment on the hard disk, Drive 5.

Next press 6. The prompts will be repeated as follows:

```
MDZ MENU SYSTEM
LOGICAL DRIVE 2 ENTER SEGMENT NAME
*********
```

Enter the segment name, in this case, CPM1. Press Return and the the next prompt will be:

```
MDZ MENU SYSTEM
ENTER PHYSICAL DRIVE NUMBER 1..8
#
```

Enter 5 for the physical drive number and the next prompt will be:

```
MDZ MENU SYSTEM
CHOOSE Ø FOR NO, 1 FOR YES SEGMENT LOCKING
#
```

Let's not lock it, by entering a Ø.

Now press 7 so we can get a floppy disk drive operative. The following will appear:

```
MDZ MENU SYSTEM
LOGICAL DRIVE 3 ENTER SEGMENT NAME
*********
```

Enter the segment name, in this case, FDCPM, which is not a segment but an operating system for floppy drives. Press Return. The the next prompt will be:

```
MDZ MENU SYSTEM
ENTER PHYSICAL DRIVE NUMBER 1..8
#
```

Enter 1 for the physical drive number and the next prompt will be:

```
MDZ MENU SYSTEM
CHOOSE Ø FOR NO, 1 FOR YES SEGMENT LOCKING
#
```

Let's not lock it, by entering a Ø.

Field 12 is the next to select. Press 12 and the prompt will be:

```
MDZ MENU SYSTEM
Ø FOR MDOZ, 1 FOR CPM COLDBOOT, 2 FOR CPM WARMBOOT
#
```

Press 1 so that MDZ/OS will know to transfer to the CP/M operating system.

Now that all of the essential fields have been entered, the display will look like this:

```
MDZ MENU EDITOR CPM SEGMENTS ITEM 1
CHOOSE OPTION 1..14 TO EDIT OR RETURN TO SAVE
##

1.  OPTION NAME      WordStar
2.  PASSWORD
3.  USER ACCESS      1111111111111111
4.  PRECISION OF baZic
                      SEGMENT NAME   PHYSICAL DRIVE    LOCK
5.  LOGICAL DRIVE 1   CPM                  5            Ø
6.  LOGICAL DRIVE 2   CPM1                 5            Ø
7.  LOGICAL DRIVE 3   FDCPM                1            Ø
8.  LOGICAL DRIVE 4
9.  LOGICAL DRIVE 5
10. LOGICAL DRIVE 6
11. LOGICAL DRIVE 7
12. CPM COLDBOOT TURNKEY
13. CPMIO OVERLAY
14. CPM TURNKEY CMD
```

This assignment directs that floppy disk (FDCPM) to be defined as Drive C and the hard disk segments to be defined as Drives A and B. If you try to access any other drives as D, E, F or G at this point, MDZ/OS's CP/M will respond with an error message:

```
NOT A CPM SEGMENT BDOS ERROR SELECT
```

Enter a Return only to save this information in the file and return to the CP/M Segment Menu which should appear now as follows:

```
MDZ MENU SYSTEM CP/M
CHOOSE OPTION Ø..9
*

  Ø. MENU 1 EDITOR
  1. WordStar
  2.
  3.
  4.
  5.
  6.
  7.
  8.
  9. RETURN TO MENU 1
```

3.4.4 Copying Your CP/M Programs to the Hard Disk

Select Option 1, WordStar, and you will be transported into the segment allocations we have just made. The following will be displayed:

```
TURNING ON SEGMENTS
SHARED SEGMENTS SELECT TO TO CANCEL OR RETURN TO CONTINUE

DRIVE 1 TURNED ON
DRIVE 2 TURNED ON
```

Press Return and the next display will be:

```
CPM BOOT
6ØK
A>
```

We will use the CP/M utility, PIP, to copy the programs into the proper segment. Place the floppy disk containing the MDZ/OS CP/M utilities in Floppy Drive 1. Copy the program disk from Drive C to Drive A, The following should be on the screen:

```
CPM BOOT
6ØK
A>
```

Enter C: and press Return. The display now looks like this:

```
CPM BOOT
6ØK
A>C:
C>
```

Now enter the PIP command in this fashion:

```
CPM BOOT
60K
A>C:
C>PIP A:=C:*.*
```

You have just told CP/M you want all of the files on Drive C, the floppy disk drive, to be copied to Drive A. After Return is pressed, the copy program will be started and will list each file as it is being copied. When all of the files have been copied, the display will look like this:

```
CPM BOOT
60K
A>C:
C>PIP A:=C:*.*
COPYING-
PIP.COM
EXIT.COM
DDT.COM
STAT.COM
DUMP.COM
LOAD.COM
RUN.COM
BAZIC10.COM
ASM.COM
CRT.002
SUBMIT.COM
XSUB.COM
BAZIC08.COM
BAZIC12.COM
C>
```

Now, let's put WordStar on the segment. Insert the WordStar Disk in Drive C, in this case, Floppy Disk Drive 1.

We have on the screen:

```
C>
```

Enter A: and press Return. It now looks like this:

```
C>A:
A>
```

Enter the command as follows:

```
A>PIP A:=C:*.*
```

Press Return and the utility will start copying. When it is finished, the display will appear as:

```
A>PIP A:=C:*.*
COPYING-
WSMSGS.OVR
WSOVLY1.OVR
INSTALL.COM
WS.COM
WSMSGS.COM
LETTER.TXT
```

Assuming that WordStar has been installed to fit your terminal and printer, all we need to do to turnkey WordStar to come up running when selected from the CP/M Segment is add the turnkey command.

3.4.5 Adding the Turnkey Command

Because we want a complete turnkey sequence, we should go back to the CP/M Segment Menu and edit Item 1, WordStar.

The CP/M operating signal, A>, should be on the screen. Enter the following so we can get back to the Control Menu or Menu 1:

```
A>EXIT
```

Press Return and we have have the Control Menu back on the screen. select Option 9 and when the CP/M Segment Menu is on the screen, press Ø and then E and then 1 to get the Item 1 Menu Editor on the screen. The following prompt will be on the screen:

```
MDZ MENU EDITOR MENU1 ITEM 6
CHOOSE OPTION 1..14 TO EDIT OR RETURN TO SAVE
##
```

We want to edit Field 14 and the prompt will be:

```
MDZ MENU SYSTEM
ENTER CP/M TURNKEY COMMAND
**********
```

Enter WS and press Return. The display will appear as follows:

```
MDZ MENU EDITOR CPM SEGMENTS ITEM 1
CHOOSE OPTION 1..14 TO EDIT OR RETURN TO SAVE
##

1.  OPTION NAME      WordStar
2.  PASSWORD
3.  USER ACCESS      1111111111111111
4.  PRECISION OF baZic
                     .SEGMENT NAME   PHYSICAL DRIVE     LOCK
5.  LOGICAL DRIVE 1   CPM                   5             Ø
6.  LOGICAL DRIVE 2   CPM1                  5             Ø
7.  LOGICAL DRIVE 3   FDCPM                 1             Ø
8.  LOGICAL DRIVE 4
9.  LOGICAL DRIVE 5
10. LOGICAL DRIVE 6
11. LOGICAL DRIVE 7
12. CPM COLDBOOT TKEY
13. CPMIO OVERLAY
14. CPM TURNKEY CMD   WS
```

To get back to the Control Menu, exit from WordStar to get the following:

```
A>
```

Now enter EXIT as follows:

```
A>EXIT
```

Press Return and the program will be returned to the Control Menu.

3.7 Check Your Work and What to Do If It Doesn't

With Menu 1 or the Control Menu on the screen, the display shows General Ledger as Option 6. Press 6 and after a moment, if everything was done properly, either the General Ledger Menu will appear, or if the data files have not been created, the General Ledger File Create mask will appear (Micro Mike's, Inc., General Ledger.)

The following problem messages could appear:

```
SEGMENTS XXX AND XXX DON'T EXIST
READY
```

First get to the Segment Editor from the System Utility Menu and check to see how the names of the segments were entered. Little things can make the difference between success and failure. It might be best to use pencil and paper and copy the names down for comparison to that in the General Ledger Menu Editor. Capital O and Ø (zero) and lower case l and the number 1. These are not interchangable and the computer makes a distinction between them.

Maybe you forgot to create and initialize the segments?

Check the segment names in General Ledger Menu Editor. They need to agree with the names you copied out of the Segment Editor, no extra spaces before or behind the name.

```
SEGMENTS TURNED ON
NO CHAIN PROGRAM
READY
```

Either you forgot to put the chain program in the General Ledger Menu Editor or you put in the wrong program.

UTILITIES

The following are utilities which are used from day to day, usually by the system programmer or the person in charge of the system. Although these programs do not have the ability to destroy all information on the disk, they still should be used with caution and a good understanding of what the programs do.

4.1. SEGED

SEGED is a baZic program which allows the user to control the allocation of segments. The program creates, deletes, renames, lists, and prints the segment names. All information controlled by this program is located in a file in the system segment. This program uses only byte accessing to its control file so that the program can be used with any precision of baZic.

It is useful to think of this program as providing a CATalog of drives. When the user does a CAT from baZic, a list of files is displayed as well as related information about each listing. In a similar manner, when the program SEGED is RUN, the user sees a catalog of drives (segments) as well as related information about each segment.

Since this is a baZic program it must be run either from baZic or selected from the System Utility Menu. A sample run of this program would appear as follows:

```
                    *** SEGMENT EDITOR ***
PHYSICAL DRIVE NUMBER 5-8
#
```

As in all programs that request a physical drive number, 5 refers to the first hard disk, 6, 7, and 8 refer to optional drives. Enter a "5" if this is the first hard disk.

The program will now print the maximum disk size in 512 byte blocks. For a 14.5 megabyte Shugart drive this value would be 25600 and for a 29 megabyte Shugart drive this number would be 51200. For the 10 and 20 megabyte Fujitsu drives, the sizes are 20320 and 40640 respectively. The message will appear as follows:

```
*** SEGMENT EDITOR ***                    MAX DISK SIZE = 51200
```

The segment allocations will appear in the form of a table. The information will appear as follows:

```
CHOOSE 'E'XIT,'L'IST,'P'RINT,'C'REATE,'D'ELETE,'R'ENAME,'N'EWDISK

  SEG NAME   PHY DRV   BLOCKS   DIR SIZE   OFFSET   UNUSED   UNUSED

  SYSTEM        5        350        4         0        0        0
  SYSTEM1       5        350        4       350        0        0
  MT SPACE      5      50500        4       700        0        0
```

The SEG NAME is the actual name of the segment. The name is defined by the user except for the names in the preceding example since these are system segments and are named by the system itself. All other entries can have user-defined names.

The name can be up to eight characters in length and can be any combination of letters, numbers, or control characters. Care must be used in assigning control characters to names since the control characters will not print. This feature can be used for security, providing another method of keeping unauthorized persons out of a segment.

PHY DRV stands for physical drive. The physical drive will always be 5 for a system with one hard disk.

BLOCKS are the number of blocks that are assigned to the segment. The value is variable as defined by the user. Each block is 512 bytes (0.5K) of formatted storage on the hard disk. The number of blocks can be any value up to the size of the hard disk or the empty space remaining (MT SPACE) on the hard disk.

DIR SIZE refers to the number of blocks within the defined blocks that are to be used for the directory if the segment is defined as a MicroDoZ segment. In the example the directory size is defined as 4 blocks or large enough to hold 128 entries. Each directory block can hold 32 entries. The directory can be up to 255 blocks long, allowing 8160 entries in any one segment of the disk. Because CP/M doesn't have a MicroDoZ type directory, this field for a CP/M segment just states that the segment is for CP/M.

OFFSET is the next value reported. This is the physical offset of the segment from the beginning of the disk. This value should increase with every segment defined. The OFFSET is calculated by taking the previous offset value and adding the segment size. This number added to the MT SPACE should equal the maximum size of the disk.

UNUSED and UNUSED show the last two entries as zero at this time. These values are reserved for future expansion.

All segment names in the Segment File are displayed. If there are more entries than will fit in one CRT screen full the program prompts the user to:

```
PRESS RETURN TO CONTINUE
```

This will bring up the additional entries. Once all entries have been displayed the program responds with the following prompt:

CHOOSE 'E'XIT,'L'IST,'P'RINT,'C'REATE,'D'ELETE,'R'ENAME,'N'EWDISK

4.1.1 EXIT

"E" for exit will cause the program to store the present assignments in the Segment File and CHAIN to SYSTEM2,5 which brings back the SYSTEM MENU.

4.1.2 LIST

"L" for list will redisplay the segment allocation list. This command is handy when your segment allocations take up more than one CRT screen full and you want to see a previous screen.

4.1.3 PRINT

"P" for print is designed to print the segment allocations to print device number 1. You must have a Device 1 defined on your system before this option will work. This option is designed to give you a paper copy of the segment allocations. It is a good idea to keep a copy in your records for reference.

4.1.4 CREATE

"C" for create is used to create or add another segment allocation. This option responds with the following prompt:

```
SEGMENT NAME
********
```

Your response to this prompt should be a name of 8 characters or fewer. If you decide you do not want this option a RETURN only at this time will cause the program to return to the original prompt. If you should enter a duplicate name the program will print the following message and prompt:

```
SEGMENT NAME
********
DUPLICATE SEGMENT NAME
```

The program will ask again for the proper segment name. Special symbols, control characters, and numbers are allowed in the segment names.

After entering the segment name the next prompt will be:

```
HOW MANY SECTORS
#####
```

To this prompt you will respond with the number of sectors you want assigned to this segment. The value must be less than the size of the disk or of the space remaining.

The next prompt asks you which operating system the segment is to be set up under:

```
CP/M OR MICRODOZ SEGMENT ?
*
```

If you want a CP/M segment then respond with a "C." If you want a MicroDoZ segment then respond with a "M." If you respond with a "C" then no further prompts are displayed and the segment allocations are listed again.

If you have requested a MicroDoZ segment the next prompt asks for the number of sectors to be used as a directory. This value can be from 1 to 255 sectors. The prompt appears as follows:

```
HOW MANY SECTORS FOR DIRECTORY
###
```

Enter the number of sectors you want for the directory of this segment. The program will display the segment allocations and prompt again.

4.1.5 DELETE

"D" for delete will delete a segment allocation. The only prompt is:

```
NAME OF SEGMENT TO DELETE
********
```

Respond with the segment name. If you do not want to delete a segment, simply pressing RETURN will cause the program to display the segment allocations and main prompt. After deleting the segment the allocations are once again displayed.

4.1.6 RENAME

"R" for rename is used to rename a segment allocation. The prompt appears as follows:

```
NAME OF SEGMENT TO RENAME
********
```

Pressing Return will cancel the RENAME option and display the segment allocations and prompt. If you want to rename a segment, enter the name of the segment you want renamed. The program will search for the name and if it is found will respond as follows:

```
NEW NAME OF SEGMENT
********
```

At this time enter the name you want the segment to have.

If a name is entered for a segment that doesn't exist (e.g. HALTEST) the program will display a message as follows:

```
NAME OF SEGMENT TO RENAME
********
HALTEST NOT FOUND
```

4.1.7 NEWDISK

"N" for newdisk returns you to the very first prompt of the program which lets you select the physical drive number. This option is used if you have multiple hard disk drives.

4.2 SEGINIT

A baZic program, SEGINIT initializes segments on the hard disk. This program performs the same function as the MicroDoZ command of "IN" on the floppy disks. This program will initialize a MicroDoZ or a CP/M segment. It must be remembered that initializing a segment will destroy all information in that segment. This program should be used with caution.

SEGINIT is a baZic program and can be loaded and run or called from the SYSTEM MENU. The program, upon execution, responds with the prompt:

```
                *** SEGMENT INITIALIZE ***
ENTER PHYSICAL DRIVE NUMBER 5...9
*
```

As for all physical drive numbers, Drive 5 is the first hard disk on your system, Drive 6 is the second and so on. Once you enter the drive number, the program will respond with the next prompt:

```
ENTER SEGMENT NAME TO INITIALIZE
********
```

If you do not want to initialize a segment a RETURN only will take you back to the System MENU by executing a CHAIN SYSTEM2,5. If you enter a segment name that does not exist the program will respond with the message and prompt:

```
ENTER SEGMENT NAME TO INITIALIZE
********
SEGMENT NOT FOUND
```

After you enter the name of the segment you want initialized the program determines if the segment is MicroDoZ or CP/M and initializes the segment correctly. For MicroDoZ the program writes ASCII 1Ø spaces to the directory area of the segment and for CP/M the program writes hexidecimal E5 to the entire segment area. The program will respond with a prompt that says:

```
INITIALIZING SEGMENT FOR MICRODOZ (OR CP/M)
```

The program actually is performing a MicroDoZ IN command. After the program has completed the initialization the following prompt is displayed:

```
INITIALIZE MORE SEGMENTS Y OR N
*
```

A response of "N" for NO causes the program to CHAIN SYSTEM2,5 which reboots the SYSTEM MENU. A response of "Y" for YES returns the user to the first prompt of the program.

4.3 MENUCR

MENUCR is a bazic program which creates menu editors, from each of which nine additional menu items can be established. MENUCR is located on Drive 5 and can be reached from the System Utilities Menu which appears as follows:

```
MDZ MENU SYSTEM UTILITIES
CHOOSE OPTION 0..9
*

  0. SYSTEM MENU EDITOR
  1. SEGMENT EDITOR
  2. SEGMENT INITIALIZE
  3. CREATE NEW MENUS
  4. CLOCK
  5. EDIT SLAVE TURNKEY
  6. NORMAL
  7. JOETEST
  8.
  9. RETURN TO MENU 1
```

From the System Utilities Menu, press 3 and the following will be displayed with the prompt:

```
MDZMENU FILE CREATE
NAME OF FILE TO CREATE
******
```

Enter the Menu Name which must be any combination of letters, numbers or characters, six in length or less.

After the name for the menu system has been entered, the program will "work" for a short time as it creates a file to hold menu parameters. Two items (Menu Editor and Return to Menu 1) are predefined by the program.

The program then "writes" the two programs necessary for operation (the menu and menu editor.) When the program has completed its job, it will chain to the menu program just created. The display will now show the menu with the name you gave it:

```
MDZ MENU SYSTEM ABCDEF
CHOOSE OPTION Ø..9
*

  Ø. MENU EDITOR
  1.
  2.
  3.
  4.
  5.
  6.
  7.
  8.
  9. RETURN TO MENU 1
```

Select Option Ø first and the next prompt will be:

```
MDZ MENU SYSTEM ABCDEF
CHOOSE OPTION Ø..9  E TO EDIT
*

  Ø. MENU EDITOR
  1.
  2.
  3.
  4.
  5.
  6.
  7.
  8.
  9. RETURN TO MENU 1
```

From this step, the process is the same as MENU. See Section 4.4

4.4 MENU

MENU actually is a series of programs and not just one program. The MENU program and its extensions and files are stored in the SYSTEM1 segment. The MENU system provides a measure of security to the entire hard disk system through the use of passwords. Users can be restricted from particular segments by the MENU system as well as restricting the precisions of baZic that are allowed in a segment.

The MENU system is CHAINed to upon boot up by MDZ/OS. The first menu to appear is MENU ONE of the MENU SYSTEM. This menu appears as follows:

```
MDZ MENU SYSTEM MENU1
CHOOSE OPTION 0..9
*

  0. MENU 1 EDITOR
  1. SYSTEM UTILITY MENU
  2.
  3.
  4.
  5.
  6.
  7.
  8.
  9. CP/M SEGMENTS
```

Under this menu system the user can select the number of the item and the menu system provides a method of "turning on" the proper segments and CHAINing to the segments. A specific program can be specified as the "CHAIN TO" program.

Each MENU listed in the CONTROL MENU can be edited, including the CONTROL MENU. To edit this MENU enter "0." The menu will appear after a short interval and will look the same except the prompt will now read:

```
SELECT OPTION (E TO EDIT)
*
```

If you select "E" you will be allowed to edit the menu. After entering the "E" you will be asked for the item number to edit by the following prompt:

```
CHOOSE OPTION TO EDIT (0 TO 9), RETURN TO SKIP
*
```

After selecting an option number to edit, the program will respond with the following display and prompt:

```
MDZ MENU EDITOR MENU1 ITEM 2
CHOOSE OPTION 1..14 TO EDIT OR RETURN TO SAVE
##

1.  OPTION NAME
2.  PASSWORD
3.  USER ACCESS      1111111111111111
4.  PRECISION OF baZic
                        SEGMENT NAME   PHYSICAL DRIVE    LOCK
5.  LOGICAL DRIVE 1
6.  LOGICAL DRIVE 2
7.  LOGICAL DRIVE 3
8.  LOGICAL DRIVE 4
9.  LOGICAL DRIVE 5
10. LOGICAL DRIVE 6
11. LOGICAL DRIVE 7
12. MICRODOZ
13. CHAIN PROGRAM
```

Select the option number you want to edit by entering the number that corresponds to the item. When you select an option, the screen will be erased and a prompt will appear to guide in entering the data.

Start with Field 1, Option Name. Press 1 and Return and the screen will be erased and replaced with the following display and prompt:

```
MDZ MENU SYSTEM
ENTER NEW OPTION NAME OR RETURN TO LEAVE SAME
**********************
```

Option name is the name that you want to appear on the menu when you finish editing.

Enter the option name and if the prompt character field is not filled, press Return. The display will return to the screen with option name in place.

As each field is called for editing, the prompts will appear and then the menu mask will will reappear so that you know where you stand at all times. The following are the fields and prompts:

Field 2:

```
MDZ MENU SYSTEM
ENTER PASSWORD OR RETURN TO LEAVE UNCHANGED
********
```

The PASSWORD is optional. If you want a password enter any combination of letters (upper and lower case), numbers or control characters. Control characters will not display so they must be remembered. **Control characters must be used with extra caution.**

Field 3:

```
MDZ MENU SYSTEM
CHOOSE USER 1..16 TO EDIT, Ø FOR ALL TO ACCESS OR RETURN
##

 1 USER IS ON
 2 USER IS ON
 3 USER IS ON
 4 USER IS ON
 5 USER IS ON
 6 USER IS ON
 7 USER IS ON
 8 USER IS ON
 9 USER IS ON
1Ø USER IS ON
11 USER IS ON
12 USER IS ON
13 USER IS ON
14 USER IS ON
15 USER IS ON
16 USER IS ON
```

If you decide to edit, for example, to deny access to User 3, enter 3 and Return and the next prompt will be:

```
MDZ MENU SYSTEM
CHOOSE Ø TO DISABLE, 1 TO ENABLE USER 3
#
```

Enter a Ø if you don't want User 3 using this program. If later on you decide that User 3 can, go back through the process and enter a 1.

Field 4:

```
MDZ MENU SYSTEM
ENTER BAZIC PRECISION 8, 1Ø, 12, 14 OR Ø IF NO RESTRICTION
##
```

If there are no restrictions enter "Ø."

The Chain Program also is optional. If you have no Chain Program then enter a RETURN. When this option is selected the program will set up the segment allocations and "drop" you into the segments. The program will then print a message saying there is NO CHAIN PROGRAM and a READY message will be displayed indicating you are in baZic. If Control C has been disabled it will be re-enabled at this time.

Fields 5 through 11:

```
MDZ MENU SYSTEM
LOGICAL DRIVE 1 ENTER SEGMENT NAME
*********
```

Enter the segment name and the next prompt will be:

```
MDZ MENU SYSTEM
 ENTER PHYSICAL DRIVE NUMBER 1..8
#
```

Enter 5 for the physical drive number and the next prompt will be:

```
MDZ MENU SYSTEM
CHOOSE Ø FOR NO, 1 FOR YES SEGMENT LOCKING
#
```

If segment locking is called, no other user can have the segment after one user has made the call.

Field 12:

```
MDZ MENU SYSTEM
Ø FOR MICRODOZ, 1 FOR CPM COLDBOOT, OR 2 FOR CPM WARMBOOT
#
```

Field 13:

```
MDZ MENU SYSTEM
ENTER CHAIN PROGRAM, OR CPMIOFILE, OR RETURN TO LEAVE UNCHANGED
**********
```

4.5 FORMATFD

FORMATFD is an assembly language program which will format 8-inch floppy disks either for MicroDoZ or CP/M in one of three formats, 128 byte sectors, 256 or 512 byte sectors.

Starting from the MicroDoZ operating system, the following will be displayed:

```
     FLOPPY DISK FORMAT PROGRAM FOR SYSTEMASTER
COPYRIGHT (C) 1982 BY MICRO MIKE'S, INC.
CHOOSE FORMAT  A) 128 BYTE SEC  1 D
               B) 256 BYTE SEC  2 D
               C) 512 BYTE SEC  2 D
```

Select the format and the next prompt will be:

```
CHOOSE DRIVE 1, 2, 3, OR 4
 #
```

Make the drive selection and the next prompt will be:

```
CHOOSE M)ICRODOZ OR C)PM FORMAT
*
```

Select whether MicroDoZ or CP/M format and the program will be executed, displaying an asterisk for each sector formatted.

If a double-sided disk is used, the program will format both sides. When the process is completed, the program will display the prompt:

```
FORMAT ANOTHER DISK (Y OR N)
*
```

If Y for Yes is selected, the process will be repeated. If N or No is selected the program will return to the MicroDoZ operating system.

4.6 CLOCK

CLOCK is a baZic program which utilizes the a one-second interrupt on the SysteMaster board. The time established through this program can be called from Option 5 on the Utility Menu and, through programming can be called as data for reports and activities.

To set the clock, select Option 4 from the System Utility Menu.

The following will appear on the screen:

```
YEAR XX
```

Enter the last two digits of the year (82) and press Return. The display will change as follows:

```
YEAR XX
MONTH 1..12
```

Enter the month and press Return and the display will change as follows:

```
YEAR XX  82
MONTH 1..12  6
HOUR Ø..23
```

The program will ask next for minutes and then seconds. Before Return is pressed after the seconds are entered, the display will appear as follows:

```
YEAR XX  82
MONTH 1..12  6
HOUR Ø..23   14
MINUTES Ø..59  24
SECONDS Ø..59   Ø
```

Press Return and the time will be displayed as follows:

```
Press any key to return to menu
```

```
6/26/82
14:24:00
```

Press any key and the program will return to the System Utility Menu. Any time that CLOCK is selected from the menu, the time will be displayed with the prompt, "Press any key ..."

4.7 COMPACT (MicroDoZ Only)

The compact option is used to recover file space that has been lost on a drive due to the deletion of files that were not at the end of the data area. The compact option re-organizes the data on a disk so that all the files are at the beginning of the disk and all of the "blank" room is at the "end" of the disk. A copy should always be made of a disk before it is compacted.

The first prompt of this option is:

```
    COMPACT COPYRIGHT (C) 1981, MICRO MIKE'S, INC.
INPUT DRIVE TO COMPACT
#
```

If a hard disk error occurs during the compaction, the program will print the following message:

```
HARD DISK ERROR ON DRIVE X
```

When the compaction is complete or an error occurs, the following prompt is displayed:

```
INPUT E)XIT OR C)ONTINUE
```

If the user selects the "C" option to continue, the program branches to the first prompt of the compaction (INPUT DRIVE TO COMPACT). The "E" option takes the user to the DISKETTE UTILITY MENU.

Any time the user is at the INPUT E)XIT OR C)ONTINUE prompt and the user specifies the "E" option, the program will examine the default drive to determine if the proper programs are on that disk. If the user has made a copy of a disk and forgotten to place the utility disk back in the default drive, the correct programs will not be on the disk in the default drive.

When this situation occurs, the program will respond with a prompt and will wait for you to place the correct disk in the default drive and press return to continue:

4.8 CP/M Utilities

Included in this section are only the utilities developed by Micro Mike's, Inc. For information on the standard utilities supplied by Digital Research, Inc., consult the CP/M manual.

The EXIT utility is used to change the operating system from CP/M back to MicroDoZ.

4.8.1 EXIT.COM

This program is an assembly language program that allows the user to return to MicroDoZ from CP/M. Under MDZ/OS, EXIT.COM reboots to the turnkey command in the slave that executes EXIT.COM. This program has no effect on any other slave.

The program is executed like any other .COM file by typing in the name of the program in response to the CP/M prompt. The file must be resident in the current drive to operate properly. A sample execution of this program would appear as follows:

```
A>EXIT
```

Of course, Return must be entered to begin execution.

4.9 Backup Utilities

The following utilities are used for backup of segments on the hard disk.

4.9.1 ICOPY (MicroDoZ Only)

This utility copies in A)uto or S)ingle step mode. A)uto mode copies file-by-file until the entire disk is copied or will stop when the destination filespace is too small to continue or a duplicate filename is encountered. S)ingle step mode copies file-by-file but prompts the operator as to the action to be taken for each file.

ICOPY can be reached by a ^C (hold the Control key down and press C. This can also be written as CTRL C.)

At the READY signal, enter the following:

```
DOSCMD "ICOPY,5\\BAZICIO,5\CONTROL,5"
```

Press Return and the first display will be:

```
SYSTEMASTER INTELLIGENT COPY COPYRIGHT(C)1981, MICRO MIKE'S, INC.
INSERT DISKETTES THEN PRESS ANY KEY
```

Press any key and the next prompt will be:

```
SYSTEMASTER INTELLIGENT COPY COPYRIGHT(C)1981, MICRO MIKE'S, INC.
INPUT SOURCE DRIVE NUMBER
#
```

Enter the source drive number and the next prompt will be:

```
SYSTEMASTER INTELLIGENT COPY COPYRIGHT(C)1981, MICRO MIKE'S, INC.
INPUT DESTINATION DRIVE NUMBER
#
```

If the user enters the same response to both drive number requests, the program will print the following message and return to the first prompt:

```
SOURCE DRIVE CAN'T EQUAL DESTINATION DRIVE
```

The mode is selected in response to the next prompt prompt:

```
SYSTEMASTER INTELLIGENT COPY COPYRIGHT(C)1981, MICRO MIKE'S, INC.
CHOOSE A)UTO OR S)INGLE STEP
*
```

To make your selection, respond with the first letter of the name of the option you want ("A" for auto or "S" for single step). The A)UTO option begins copying from the point where it is invoked to the end of the disk. The S)INGLE STEP option allows the user to view each file name before deciding if the file is to be copied.

If the A)UTO mode is selected, the screen will appear as follows:

```
SYSTEMASTER INTELLIGENT COPY COPYRIGHT(C)1981, MICRO MIKE'S, INC.
C
     FILENAME          AF=Ø
```

The file name and the attribute will display on the fourth line of the CRT as the files are being copied. When the copy is complete, the program will respond with the following prompt:

```
SYSTEMASTER INTELLIGENT COPY COPYRIGHT(C)1981, MICRO MIKE'S, INC.
INPUT E)XIT OR C)ONTINUE
*
```

If the user selects the "E" option, the user is returned to the DISKETTE UTILITY MENU. If the user selects the "C" option, the program will branch to the initial prompt (INPUT THE SOURCE DRIVE NUMBER).

Several conditions will terminate the A)UTO mode. If space on the destination disk (the one being copied TO) is too small, the program will terminate with the message:

```
DISK IS FULL
```

If the directory space becomes full, the program will issue the following message:

```
DIRECTORY IS FULL
```

If a hard disk error occurs, the program will display the following message:

HARD DISK ERROR ON DRIVE X

If the destination disk is write-protected (write-protect tab not in place), the program will respond with the following message:

FILE IS WRITE PROTECTED

The occurrence of any of the these messages will result in the following prompt:

```
SYSTEMASTER INTELLIGENT COPY COPYRIGHT(C)1981, MICRO MIKE'S, INC.
INPUT E)XIT OR C)ONTINUE
*
```

If the user selects the "E" option the user is returned to the MicroDoZ operating system as follows:

```
1>
```

If the user selects the "C" option, the program will branch to the initial prompt (INPUT THE SOURCE DRIVE NUMBER).

If the program encounters a file that already exits on the destination disk, the program will respond with the following prompt and message:

```
CHOOSE S)KIP, R)EPLACE, N)EWNAME, OR E)XIT
*
DUPLICATE FILE NAME
```

The S)KIP option skips the present file and goes to the next file. The file is NOT copied if you specify S)KIP.

The R)EPLACE option will cause the file on the destination disk to be over-written by the file of the same name on the source disk (the disk being copied from). The program will then continue to the end or will stop if any of the terminating conditions are found again (file space too small or duplicate file name).

The N)EWNAME option will allow you to give the file a new name. In this instance you will still retain the file on the destination drive with the duplicate name but the file will be copied from the source to the destination drive and be named whatever you choose.

If the N)EWNAME option is chosen, the program will ask for the new file name with the following prompt:

```
INPUT NEW FILE NAME
********
```

If the name has fewer than eight characters, enter a RETURN after the name. If the name has eight characters the program will supply the RETURN.

The E)XIT option allows you to exit to the error condition prompt:

```
INPUT E)XIT OR C)ONTINUE
```

In S)INGLE STEP mode, the program works similarly to the auto mode except each file is displayed on the CRT one at a time, allowing the operator to make decisions based on the file name. Under S)INGLE STEP the program will display the name of the first file on the source drive and issue the following prompt:

```
CHOOSE C)OPY, S)KIP, A)UTO, OR E)XIT
*
FILENAME          AF= Ø
```

The C)OPY option performs a copy of the file and then displays the next file. If a duplicate record is encountered, the program performs exactly as stated in the A)UTO mode description of duplicate file names.

The S)KIP option skips the currently displayed file and brings up the next file for the operator to act upon.

The A)UTO option allows the user to A)UTO copy all files on the disk. A)UTO may be invoked any time under S)ingle Step to copy to the end of the disk. The previous description of A)UTO applies any time it is invoked.

The E)XIT option works exactly as described previously.

All files are compacted automatically as they are copied. Using ICOPY, files are transferred onto the "tail end" of any data already on the disk being copied to. Filenames are displayed on the CRT screen as they are being copied. ICOPY employs read-after-write verification of data being transferred.

4.9.2 COPYFILE

COPYFILE allows the user to copy a single file or multiple files from the user's choice of drives. This program normally will not be used by the end user because of its complexity. This program is "programmable," in that all the questions asked by this program can be answered by another program. In this mode, the program will answer automatically its own prompts and no operator intervention is required.

The program first prompts the user for the drive number containing the file(s) to be copied. The prompt appears as follows:

```
COPY FILE COPYRIGHT (C) 1981, MICRO MIKE'S, INC.
INPUT SOURCE DRIVE NUMBER
#
```

After entering the source drive number the program asks for the number of the drive to which to copy the file(s) to:

```
INPUT DESTINATION DRIVE NUMBER
#
```

If the source drive number is equal to the destination drive number, the program will consider the file copy a special case. This situation will be discussed later. The "normal" flow of the program is for the source and destination drives to be different. For this situation (drives different), the program begins a series of prompts requesting information on how you want the file copies to be made. The first prompt in this series is:

```
DO YOU WANT TO RENAME FILES? Y OR N
```

Often the file you want to copy from has already been created on the destination disk. The previous prompt asks if you want to rename automatically each file encountered on the destination disk having the same name as the file being copied from the source disk. Answer "Y" if you want all duplicate files automatically renamed or "N" if you do not want duplicate files renamed.

The next prompt is:

```
DO YOU WANT ALL DUPLICATE FILES REPLACED? Y OR N
```

If you answer "Y" for this prompt, any duplicate files on the destination disk will be replaced automatically by the file of the same name on the source disk. An "N" response will allow you to decide individually what action is taken for each duplicate file name encountered.

The next prompt is:

```
DO YOU WANT TO USE A FILE OF FILE NAMES? Y OR N
```

The "N" option specifies that the user will enter the names of each file to be copied. Alternately, the user may have a file established containing the names of all the files that are to be copied. Selecting the "Y" option places the file copy program in a mode allowing the program to "look" at the specified file and perform a copy of all files listed in the file of file names. If the "Y" option is selected the next prompt will be:

```
INPUT FILE NAME
********
```

The file name of the file of file names should be entered. If the file is not 8 characters long, a carriage return will have to be entered by the user, otherwise the program will supply the carriage return. After entering the file name, the program will prompt you for the drive number of the file of file names.

The next prompt will be as follows:

```
INPUT FILE NAME TO COPY
********
```

Enter the name of the file to be copied. If the file name is not 8 characters long, press Return. If the file name is 8 characters in length, the program will supply the carriage return.

If the file name selected is not on the destination drive, the program will create a file the same size as the file on the source disk and will copy the file to the destination disk. If the file already exists on the destination disk, the program will check the responses to the previous questions to determine what action it is to take.

If the answer to the auto replace prompt was positive, the file on the destination disk will be replaced automatically by the file of the same name on the source disk. The program will make the replacement and print the following message on the CRT to inform the user the auto replace mode is in effect:

```
AUTO REPLACE
```

If the new name option is selected as positive, the program will prompt the user automatically for the new name the file is to be given when it is copied to the destination disk. The prompt appears as follows:

```
INPUT NEW FILE NAME
********
```

Once the copy is completed, the program will ask the user if he wants to exit the program, copy another file using the answers to all previous questions, or start over with the program and answer all questions again. The prompt appears as follows:

```
CHOOSE E)XIT, M)ORE FILES, OR R)ESTART COPY
```

The "E" option always takes the user to the program that originally called the COPYFILE program. The "M" option allows the user to copy more files. In this mode, the user is not asked for the source and destination drive number nor the other questions pertaining to how the copy is to be made. Selecting the "R" option takes the user back to the original prompt of the program, "INPUT SOURCE DRIVE NUMBER."

If, at any time during the copy, the user encounters a duplicate file and the user specified that the program was NOT to replace automatically the destination file with the source file, the program will display the following prompt and message:

```
CHOOSE E)XIT, S)KIP, R)EPLACE, OR N)EWNAME
*
DUPLICATE FILE NAME
```

If the user selects the "E" option the program displays the following prompt:

```
INPUT E)XIT OR C)ONTINUE
```

The "E" option in this case takes the user back to the calling program. The "C" option allows the user to start over at the original prompt (INPUT SOURCE DRIVE NUMBER).

If the user selects the "S" option the file is skipped and is not copied from the source disk to the destination disk.

The "R" option specifies that the destination file is to be overwritten by the source file.

The "N" option allows the file to be copied from the source disk to the destination disk and given a new name on the destination disk. The program issues the following prompt when the "N" option is selected:

```
INPUT NEW FILE NAME
```

If responses to the initial two prompts (INPUT SOURCE AND DESTINATION DRIVE NUMBERS) result in the same drive number being input by the user, the program will continue to function but two prompts will not be issued as in the normal sequence. These two prompts are:

```
DO YOU WANT TO RENAME FILES? Y OR N
DO YOU WANT ALL DUPLICATE FILES REPLACED? Y OR N
```

The program assumes that duplicate files will be encountered since the source disk and destination disk are the same. The general procedure in this situation is to rename the file as described previously. The DUPLICATE FILE NAME messages and prompts will also be as described previously.

During the course of a file copy, several error messages may be displayed on the CRT. The following messages are essentially self-explanatory:

```
FILE NOT FOUND
DESTINATION DIRECTORY IS FULL
DESTINATION DISK IS FULL
DUPLICATE FILE NAME
```

In the automatic mode where the program is using a file of files, there are two additional messages that are displayed on the CRT simply to inform the user of proper operation of the program:

```
READING FILE NAMES FROM FILE
END OF FILE NAME FILE
```