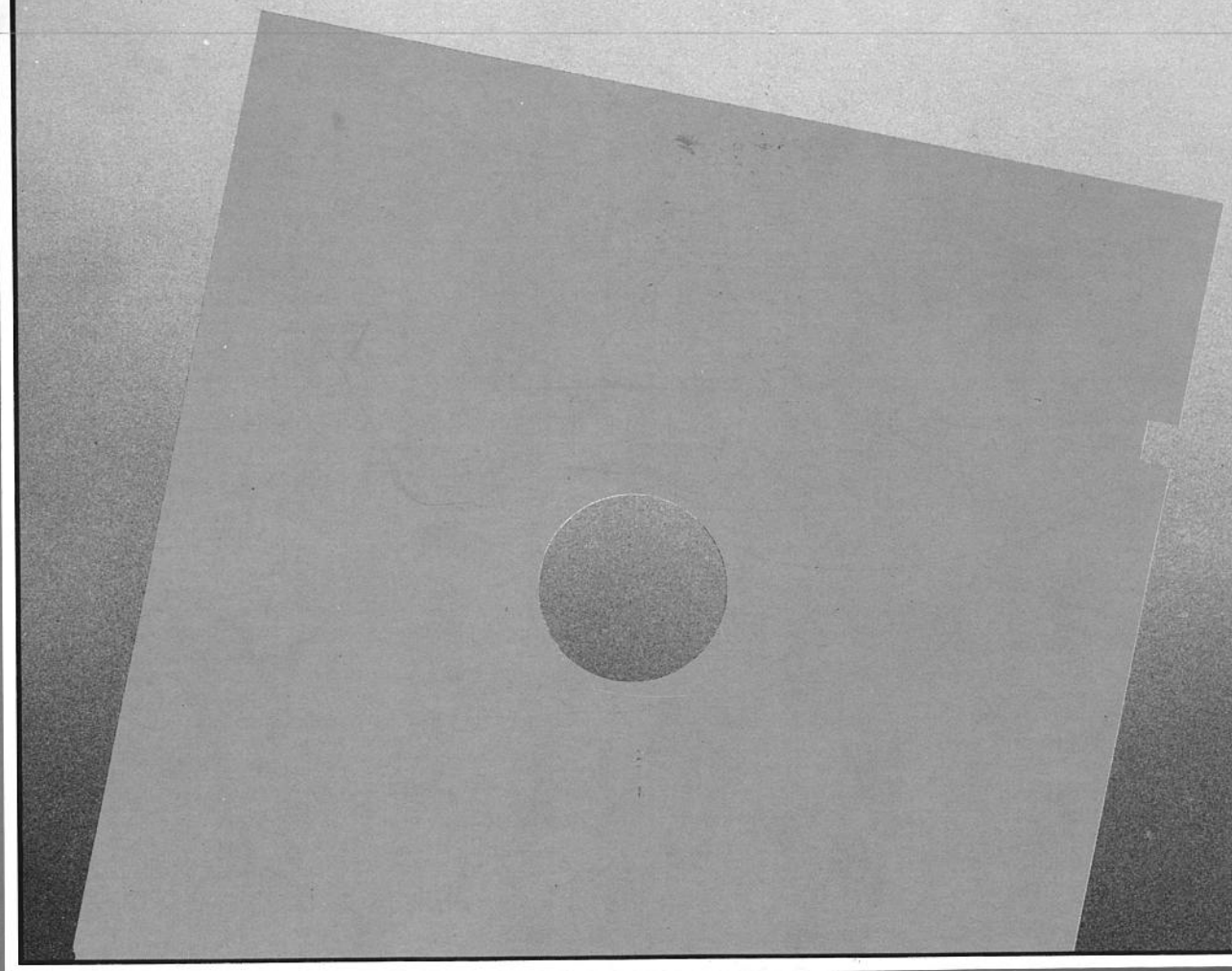
PROTON Electronics
Energiestraat 36
1411 AT Naarden NL
☎ (0)2159 - 48224*
Telex 73415 pe nl

PROTON

COMBIDOS 80

SOFTWARE



COMBIDOS tm

OPERATING SYSTEM

MANUAL

PROTON ELECTRONICS
Energiestraat 36
1411 AT NAARDEN
The NETHERLANDS

Price HFl. 25.00

COPYRIGHT

DISCLAIMER

Although care was taken with respect to the contents of this document, PROTON ELECTRONICS reserves the right to make changes at any time without notice.
No responsibility is assumed by PROTON ELECTRONICS for use of this information; nor for any infringement of patents or other rights of third parties which may result from the use of this information or this product.
Specifically any implied warranties of merchantability or fitness for any particular purpose are disclaimed by PROTON ELECTRONICS.

TRADEMARKS

COMBIDOS is a trademark of PROTON ELECTRONICS.

CONTENTS

CHAPTER 1
COMBIDOS basic operation

1.1 Introduction

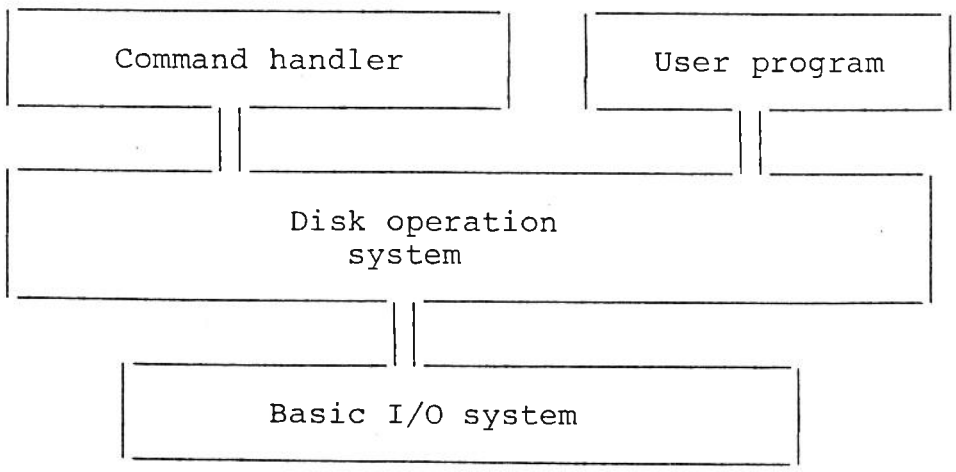
Since the introduction of the Z80 microprocessor several attempts have been made to introduce a operating system that uses the power of the Z80 microprocessor. Most of these operating systems however did not run all existing software that is marketed for the 8080 CP/M operating system (CP/M is a registered trademark of Digital Research).
Combidos is designed as a fast, userfriendly operating system which conforms to the existing industry standards of software and takes advantage of hardware features (High capacity Floppy disks and Winchester disks) which are now cost effective.
Combidos can be adapted to systems with a Z80 microprocessor and a minimum of 32 Kb RAM.
The speed of Combidos 80 can be very high when disk access is reduced by buffering of the directory and most recently used file blocks in RAM. Combidos automatically assigns part of the available RAM space to buffering when using a 128K computer.
Until now, when a user had two or more different microcomputers he could not directly access the disk files of one system with the other computer(s). Combidos solves this problem by a diskinstall feature that enables the computer to read and write disks of virtually all popular computers (including MS-DOS and PC-DOS computers). A large number of disk formats is supported as a menu selection. The user may define own formats and add them to this menu.

1.2 Functional description

1.2.1 COMBIDOS structure

The COMBIDOS 80 operating system consists of 3 main blocks.
1. Command handler.
2. Disk Operating System
3. Basic Input and Output (I/O) System

The command handler and the disk operating system do not depend on the hardware. The Basic I/O System is the link between the Disk Operating System and the hardware.
The following diagram shows the interaction between the main blocks.



1.2.2 Basic functions

The command handler is a program that reads and executes commands from the console. It operates like a 'user program'. The command handler can read and activate programs from the disk. When a user program is run, the command handler is inactive.
The disk operating system is the most important part of the COMBIDOS system. This part is responsable for the proper organisation of data on a disk and it is the link for all data transfer between programs and the outside world.
The basic I/O system is a set of functions which link the disk operation system to the hardware.

1.2.3 Commands

Upon initial start-up of the COMBIDOS system it will report itself with:

```
    COMBIDOS 80 Version x.x
(c) 1984 Proton Electronics Holland
```

The COMBIDOS functions are initialized and the system is ready to run when the prompt:

```
A%
```

is shown.

The A indicates that disk drive A is active, the % is the command-prompt of COMBIDOS. Words typed after the prompt are interpreted as commands. A command is first checked to see if it is a resident command, if not, Combidos will look for it on the active disk. If the active disk drive is not drive A and the command is not a resident command nor is it on the active disk, Combidos will look for it on drive A. This feature allows all commands to be stored on the system disk while the user disk is 100% free for user data.

Resident commands

Combidos has a set of commands which are often used, build into the system. Such commands are:

- DIR Show directory
- ERA Erase a file from disk
- REN Rename a file
- SAVE Save a block of memory in a file
- TYPE Type the contents of a textfile
- LIST Same as type, but to the list device
- GO Start program at 0100h
- USER n Select the active user area

All commands are explained in chapter 2.

User commands

The user commands invoke a user program that must be run. Any program name can be considered a user command.

Utilities

Combidos has a set of utilities that are easy to use and are very powerful. They are used to copy files and initialise new disks. The utilities are described in chapter 2.

1.2.4 Line editing

The command handler offers a set of line edit facilities while typing the command lines. The edit functions are control keys (press **ctrl** key and then a character key, just as you use the shift key).
The control characters are:

ctrl/C	Combidos warm start (when typed at the beginning of the line)
ctrl/E	The cursor performs a ´return´, the return is not entered in the command line.
ctrl/H	Erases and backspaces one character.
ctrl/J	Ends the command line, same as return (ctrl/M).
ctrl/M	Ends the command line, same as linefeed (ctrl/J).
ctrl/R	Retypes the command line.
ctrl/U	Retypes the characters of the old command line.
ctrl/X	Erases all characters from the line.
ctrl/Z	End of text files
ctrl/P	The output of the console will also be send to the LST: device until a ctrl/N is typed.
ctrl/N	Ends the output to the LST: device.
ctrl/S	Hold the output (say a listing) until a ctrl/Q is entered.
ctrl/Q	Continue the console output after ctrl/S.

1.2.5 Files

Information recorded on a disk is called a file. Files are identified by name. Together with the name, a file extension is stored, so a complete file reference consists of two names, separated by a period ".", as shown:

file name.extension

Examples of file names:

FCOPY.COM FORMAT.COM DIRF.COM

The file name may have up to 8 characters, the extension up to 3 characters. The extension may be omitted. The following characters can not be used in file names or file extensions:

[]{}^~=<>,.;:?*%()|

To select files according to a selective pattern the "?" symbol may be used to replace the unknown or variable character:

ABC?.XYZ will reference the file ABCD.XYZ but also ABCE.XYZ.

When ALL characters of the name are variable, the "*" symbol can be used:
.COM will select ALL files with a COM-extension, TEST. will select all files named TEST regardless the extension.

examples:
DIR *.COM Lists all 'COM'-files
DIR TEST.* Lists all files called TEST

1.3 DISKS

1.3.1 Multiple disk drives

The Combidos system can be used with one or more (typical 2) disk drives. Commands will operate on the selected (or active) disk. The user may switch from one drive to another by typing the drive's name (A,B,C,D....) followed by a colon ":". This will cause Combidos to activate the selected disk. The Combidos command-prompt is changed to the active disk name followed by "%".

Example to switch to drive B and back to A.

```
A% dir                          list all files of drive A
TEST    .COM

A% B:                           Switch to drive B
B%

B% dir                          List files of drive B
PROGRAM .COM

B% A:                           Switch back to drive A
A%
```

1.3.2 Changing a diskette

When you change the diskette of one of the drives, Combidos will mark that diskette to be 'read only'. A Dos error 'read only' will occur when writing to that disk is attempted. To allow writing to that diskette type ctrl/C (Warm start).

1.4 Devices

Combidos supports a set of standard Input and Output devices that depends on the hardware and the implementation of the operating system on the computer. The user can assign one of the several physical devices to one of four logical devices:

CON:	System console device, normally the keyboard-display. (Input and Output)
LST:	System list device, normally a printer. (Output only)
RDR:	Auxiliary input device, normally a communication port. (Input only)
PUN:	Auxiliary output device, normally a communication port. (Output only)

The names of these logical devices originate from the time that Paper tape punchers (PUN:) and readers (RDR:) were common use.
To distinguish between file names and device names, a device name must be followed by a colon (:).

For flexible assignment of devices, Combidos allows the installation of a number of different devices, each with a different name. These names may depend on the computer in use, common used names are:

For CON:

CRT:	Console Video/keyboard
CRT:	Console Video/keyboard
SER:	Main serial port
LPS:	Input Keyboard, Output Aux. Serial

For RDR: and PUN:

CRT:	Console Video/keyboard
SER:	Main serial port
LPS:	Input Keyboard, Output Aux. Serial
LPP:	Parallel printer port

For LST:

SER:	Main serial port
CRT:	Console Video/keyboard
LPP:	Parallel printer port
LPS:	Input Keyboard, Output Aux. Serial

The Combidos default selection is shown in boldface.

CHAPTER 2
COMBIDOS Commands

2.1 Resident Commands

Combidos has a set of commands which are always present for the user, they do not reside on disk. These commands will be described here in detail.

- DIR

  Show directory

  This command will show the disk-contents of the active disk or, when followed by a disk name, the selected disk contents is shown.

  Example:
  ```
  A% DIR
  TEST     .COM

  A% DIR B:
  PROGRAM .COM
  ```

- ERA (diskname:)filename.extension

  Erase a file from disk

  This command will erase the specified file(s) from the specified disk. When the disk name is omitted, the active disk is assumed. Combidos will search for the file(s), show them and ask permission to erase them from the disk.

  Example:
  ```
  A% ERA TEST.COM
  TEST    .COM
  OK (Y/N)Y

  A% ERA *.COM
  DEMO     .COM        DEMO2    .COM
  OK (Y/N)Y
  ```

- REN (diskname:)filename.extension=filename.extension

  Rename a file

  With this command files can be renamed. When the disk name specification is omitted, the active disk is assumed.

  Example:
  ```
  A% REN TEST1.COM=TEST.COM

  A% REN B:PROG.COM=PROGRAM.COM
  ```

- SAVE size (diskname:)filename.extension

  Save a block of memory in a file

  This command saves a block of memory to a disk file. The size of the block is given in multiples of 256 byte, the save area starts at 100h (hexadecimal). To save a program that occupies memory from 100h till 4D0h, 4 blocks of 256 byte (4 pages) must be saved:
  100h-1FFh, 200h-2FFh, 300h-3FFh, 400h-4FFh.

  Example:
  ```
  A% SAVE 4 ABC.COM

  A% SAVE 8 B:TESTFILE.TST
  ```

  When an attempt is made to overwrite an existing file, Combidos will ask permission to erase the old file.

- TYPE (diskname:)filename.extension (option)
  Type the contents of a text file
  This command will show the contents of an ASCII file on the console. Tab´s (ctrl I) are expanded, assuming tab positions at every eighth column.
  The TYPE command can show the texts in a ´page-mode´ by adding the option P. The listing is continued by typing a key. The listing can be temporary halted by ctrl/S, continued by ctrl/Q, terminated by ctrl/C.
  Example: **A% TYPE DEMO.TXT**
  This will list the file till the end.
  A% TYPE DEMO.TXT P
  This will list the file page-by-page.

- LIST (diskname:)filename.extension (option)
  Type the contents of a text file
  This command will show the contents of an ASCII file on the list device. It is operated in the same way as TYPE.

- GO
  Start program at 0100h
  The GO command allows the user to run a program that already (or still) resides in memory. It saves the time of reloading the (non resident) command or program.
  Example: **A% DIRF**
  .....

  A% GO
  .....

  Some programs cannot be re-run, they are likely to return to the command handler.

- USER usernumber
  Set the ´active user´ number
  By working under a usernumber, it is possible to save and retrieve files that are not ´seen´ by other users. On initial start, the usernumber is 0. At any time the USER command may be used to switch to another user area of the disk directory. The user number remains unaffected when disks are changed or programs are run.
  Note that ´com´-files, defined in user area 0 are accessible from other user areas.
  When the user number is not 0, it is shown in the Combidos prompt after the drivename.
  Example: **A% USER 2**
  A2%

2.2 DISK COMMANDS (Utilities)

2.2.1 DIRF

The DIRF utility shows the files of a disk in alphabetic order and in the bottom line all statistic information such as: Filesize in Kbyte, number of records and File status (R/W, R/O). For the 3.0 compatible version also Date & time of last update is shown.
To see the information, move the cursor to the desired file. The file information is shown in the bottom line.
The cursor can be directed over the screen with the standard WordStar cursor control:

cursor up ^E

cursor right ^D cursor left ^S

cursor down ^X

The command line on the screen shows:
L)ogged A:0 M)ask ´????????.???´
followed by: **A)ll_Users R)un** <Ret>|X|Quit

The Logged disk can be changed with the command ´L´. This is necessary when the diskette is changed or when the directory of an other drive is wanted.
At this time it is possible to changes the usernumber also by typing ´U´ and the usernumber.
When the files of all users must be shown, use the command ´A´. The usernumber (after the logged drive name) is changes into a ´@´ symbol.
When a program file is selected with the cursor, it can be run using the ´R´ command. This command will create a batch-file on drive A which is automatically executed by Combidos.
The DIRF command has the same command format as the DIR: It can be started by typing **DIRF drive:filename.ext.**
ex. A% **DIRF B:*.COM**
The drive and file specification is optinal. The drive specification has the same effect as the ´L´ command while the file specification sets the selection mask as with the ´M´ command.

2.2.2 FCOPY

General description

The FCOPY utility enables users to make copies of his files to other disks, to delete files or to look into a (text-)file. After starting, FCOPY will display a command line and the disk contents on the screen. The command line shows all possible commands except the cursor move commands. The cursor can be directed with the cursor keys to the desired files. When more files are on the disk than can be shown on the screen, FCOPY reports this with <<< or >>> on the bottom line of the screen. >>> Means that more files are at the right of the screen, to see the file, move the cursor to the right. <<< Means that more files are at the left of the screen.

Cursor movement with FCOPY.

The cursor can be directed over the screen with the standard wordstar cursor control:

cursor up ^E

cursor right ^D cursor left ^S

cursor down ^X

NOTE: The spacebar and return key perform a cursor down.

The FCOPY commands.

The command of FCOPY consists of two parts:
The left part contains status information about the currently logged disk name ´L´ and the directory mask ´M´.
L)ogged A: M)ask ´????????.???´

The right part of the command line shows the commands for copying (C), erase (E), select (S), unselect (U), select all files (A), inverse the selection (I), show the contents of a text file (V) and exit the FCOPY utility (Q).
C)opy E)ra S)el U)nsel A)ll I)nv V)iew Q)uit

All available commands will be explained in detail in the following section:

L Change the currently logged disk drive to another disk or reread the directory from the disk.
When the L-command is given, the statusline is replaced by the question:
Login U)ser (0) Drive (A:)
Entering <return> will reread the directory of the currently selected disk (A:), typing an A or B will read the directory of that disk.
By entering an U, the program will ask for:
Login User ?
Now a usernumber can be entered, followed by a <return>. Hereafter the drive selection must be performed (A, B or <return>).

M Enter the directory mask for unabiguous file names, a ? allows any character. The directory show by FCOPY will be according the mask after the L-command.
The mask can be given as an argument when FCOPY is started:
ex. A% **FCOPY *.COM**
FCOPY will only show the COM files of the disk and the mask in the command line will show ´???????.COM´.

S A file can be selected by typing ´S´. The filename is show in inverse video and the cursor is moved to the next file.

U A file can be deselected by typing ´U´ (Unselect). The file is show in normal video and the cursor is moved to the next file.

A The A command will select all files (show on the screen) as with the S command.

I The I command will invert all selection; selected files are deselected and unselected files are selected.

When all desired files are selected the Copy (´C´) or Erase (´E´) function can be started.

C The copy function will ask (in the command line) to which drive the selected files must be copied:

```
Copy to U)ser (0), Single, Drive (A:) ?
```

By entering an U, the program will ask for:

```
Copy to User ?
```

Now a usernumber can be entered, followed by a <return>. Hereafter the drive selection must be performed (A, B or <return>). When a <return> is entered, the default drive is used as destination drive.
An A or B will select that drive as destination drive.
When copying is started from drive A to drive A, FCOPY will ask for the source and destination diskette. You must change diskettes on request and enter <return>.

When a S is entered, single drive copy (as with A to A) is enabled, although you may select a B to A copy. This allows copying of files from another disk format (installed on drive B) to the standard combidos format on drive A.

E Will erase files from the currently logged disk. On the bottom line a message is show:

```
Erase F)ile or A)ll selected ?
```

Typing F will erase the file the cursor was on when the command was given (see bottom status line), typing A will erase all selected (inversed names) files.
An erased file entry is replaced by: ********.***
It is not allowed to operate further on erased files.

V The View command allows to look into the current text file. To display a line of the file type the space bar. On the bottom of the screen all lines are shown and the screen is scrolled up until a <return> is entered.

Q When all FCOPY operations are done you can exit to combidos by typing Q. The directory is left on the screen.

When a command is entered by accident, you can leave it by typing ctrl/C. The FCOPY command format is the same as DIRF.

2.2.3 MSCOPY

The MSCOPY utility is a menu driven file-copy program similar to FCOPY. Its function is to convert MS-dos files to or from Combidos files. This may be of advantage when documents must be exchanged to MS-dos computers.
To convert files, first install the desired disk format with the DI utility.
The MSCOPY utility will be available 2nd quarter of 1985.

2.2.4 DEV

The DEV utility is the device I/O-handler of the Combidos system. It allows the user to change input- or output-devices of the system. DEV also reports the current status of the IO-devices.
The DEV program will show a screen on which all selection possibilities are shown. The current selection is shown highlighted. The selection can be changed by selecting the Row and Column for a logical device. DEV allows temporary and permanent change of the I/O selection as a menu option.
The screen layout is shown:

Current I/O configuration

		0	1	2	3
A	CON:	CRT	CRT	SER	LPS
B	PUN:	CRT	SER	LPS	LPP
C	RDR:	CRT	SER	LPS	LPP
D	LST:	CRT	SER	LPP	LPS
		0	1	2	3

Q Quit to system, no update
U Update the current system (Not on disk)

2.2.5 FORMAT

The FORMAT utility is a program to initialise (format) blank disks. The program writes all identification information on the disk. FORMAT is able to write all required formats on the disk.
Format will ask for the name of the drive on which it must operate, it will ask if it has to perform a verify after formatting (answer Y or N) and it will wait until you acknowledge that you have inserted a blank diskette. To guarantee that a diskette is 100% readable and writable use the verify option. Errors, if any, will be shown.
The format operation will start after pressing the **return** key. FORMAT will show how many bytes per track were written and it will show the current track number.
The number of bytes written per track is a check motor speed of your drive: Double Density (250 Kbits/sec) is 6250 bytes, Single Density is 3125 bytes. The number of bytes actually written may deviate from the nominal value. When the deviation excedes 0.2 %, your drive needs service.

2.2.6 BAtch mode

Combidos can read and execute command lines from a disk file so that often used series of commands can be activated by a single command. This type of command is called BATCH mode. The file containing the commands must be prepared with a text editor and have the extension SUB (the file is submitted to batch mode processing).
The batch mode processing allows parameters to be passed to the command. Parameters are numbered and preceeded by a "$".
The form of the command is:

A% **BA filename parameter1,parameter2.....**

Example: When we want to assemble and link a program with one command, using Microsofts M80 and L80, and loading it with ZSID we must prepare a file ASM.SUB with the editor:

```
M80 $2,$1=$1
L80 $1,$1/N/E
ZSID $1.COM
```

A% **BA ASM TESTPROG LST:**

The $1 is automatically replaced by the parameter TESTPROG and $2 by LST: to given a listing on the list device.

The BA command creates a file called $$$.SUB on the active disk. Combidos will then start executing commands from this file.

2.2.7 PUTSYS

The PUTSYS command can write the Combidos system on the system tracks of a disk. Optionally systemtracks can be read from a disk.
A% **PUTSYS**

```
Combidos PUTSYS utility V1.0
Read system from drive (A/B/C/D, return = no read) ? return
Insert formatted disk on which the system has
to be written and press return > return
Write system to drive (A/B/C/D, ctrl/C = stop) ? B
```

PUTSYS is already provided with the version of combidos for your computer, so normally it is not necessary to read the system, so a return after the ´read question´ will do.

Users who need more than the standard 2 disk drives, can generate a system using **PSYS4**. This is the PUTSYS program containing a Combidos system that supports 4 drives.

2.2.8 Disk format installation DI

With the DI utility any format according to IBM3740 or IBM34 protocol can be read and written (almost any computer uses one of these standard protocols; a known exception is APPLE). The DI utility contains a list of disk formats which can be installed on drive B. The user can create new formats and add them to the list. This list is saved on disk in DINST.DAT and contains a Dpb (Disk Parameter Block) for each disk format.
The DI is started by typing:
A% DI
A menu will be shown:

```
Logged A:  Dpbfile contains XX blks  Current Dpb:  (1) '5q160+ '

S     S)elect Dpb from File

I     I)nstall current Dpb in memory

A     A)dd current Dpb to Dpbfile
R     R)eplace current Dpb in Dpbfile

G     G)et Dpb from memory
C     C)hange current Dpb

Q|X   Q)uit this program

Select a function :
```

Function 1 will show the list of currently available formats. You can select the desired format by its reference number. After selection, you can install the format in memory for drive B.
Function C allows users to change a Dpb to accomodate a specific disk format and with function A and R you can store a format in the Dpbfile.

Note: For installation of a new format permanently into the system a utility similar to DI can be purchased. This may be necessary for OEM users who want to install combidos for special types of disk drives. The utility that can do this is called SI (System Install).

The following list shows which computer is referenced with a Dpb name. The disk formats supplied are designed to operate on 80 track drives only. Users with 40 track drives cannot use the 80 track formats and will have to adapt the parameter blocks of the 40 track formats by disabling the 'double-step' feature (function selection 19).

```
 1 5q160+    = Compactboard Standard 5    Inch 2 x 80  Tracks
 2 8q154+    = Compactboard Standard 8    Inch 2 x 77  Tracks
 3 8q320     = Compactboard Standard 5(8) Inch 2 x 160 Tracks
 4 5osbod4   = Osborne 5 Inch 1 x 40 tr on 40 tr drive
 5 ibmms18   = (*) Ibm pc-dos 1.0 on 2 * 80 tr drive
 6 ibmms14   = (*) Ibm pc-dos 1.0 on 2 * 40 tr drive
 7 5bbcs8    = ($) BBC dos 1 x 40 tr Single density op 80 tr drive
 8 5bbcs4    = ($) BBC dos 1 x 40 tr Single density op 40 tr drive
 9 5ibmms2   = (*) Ibm pc-dos 2.0 on 2 * 40 tr drive
10 5big40d   = BigBoard 2 * 40 tr
11 5ibmucs   = ($) Ibm UCSD system
```

```
 12 5ex512s   = Exidy Sorcerer 1 x 40 tr 512 bytes per sector
 13 5itt30    = Itt 3030 2 x 70 tracks ( 2 x 80 tr drive )
 14 5ex512d   = Exidy Sorcerer 2 * 40 tr 512 bytes per sector
 15 5nbl60    = Newbrain 2 x 80 tracks
 16 5ibms24   = (*) Ibm pc-dos 2.0 on 2 * 40 tr drive
 17 5nb40s    = Newbrain 1 x 40 tracks
 18 5kay2     = Kaypro 2 ( 1 x 40 tracks )
 19 5p2000    = ($) Philips p2000 M ( 1 x 40 tracks ) Philips - Dos
 20 5kay44    = Kaypro 2 ( 2 x 40 tracks )
 21 5p3500    = Philips P3500 Turbo-dos
 22 5nb2004   = Newbrain 1 x 40 tracks
 23 5q35d     = Compactboard Standard 5 Inch 2 x 35 Tracks
 24 5osbod8   = Osborne Double density 1 x 40 tracks ( on 80 tr drive )
 25 5q40s     = Compactboard Standard 5 Inch 1 x 40 Tracks
 26 5osbss4   = Osborne Single density 1 x 40 tracks
 27 5p20004   = ($) Philips p2000 M 1 x 40 tracks
 28 5sup704   = Superbrain QD 2 x 35 tracks
 29 5p2000c   = Philips P2000 C 1 x 40 tracks
 30 5ta40d4   = Triumph-Adler 2 x 40 tracks
 31 5swts4    = ($) Swtpc 6800 Flex-1 Single density 1 x 40 tracks
 32 5ex2564   = Exidy Sorcerer 1 x 40 tracks 256 bytes per sector
 33 5trl60    = Trs 80 model 1 2 x 80 tracks
 34 5tshl60   = (*) Toshiba 2 x 80 tracks MS dos
 35 5bull60   = Bullet board 2 x 80 tracks
 36 5usol16   = Microsol 2 x 80 tracks
 37 d8buls    = 8 Inch Bullet board 1 x 77 tracks
 38 8rtl100   = ($) Dec pdp-11 / Minc-11 8 Inch SS/SD
 39 8cpm      = 8 Inch Cpm Ibm 3740 Standard
 40 8dj1024   = 8 Inch Disk-Jockey 1 x 77 tracks 1024 byte/sect
 41 8dj512    = 8 Inch Disk-Jockey 1 x 77 tracks  512 byte/sect
 42 8trl54s   = 8 Inch Trs-80 mod 1 2 x 77 tracks Single Density
 43 8ufx77    = ($) Uniflex 6809  8 Inch 1 x 77 tracks
 44 8ufxl54   = ($) Uniflex 6809  8 Inch 2 x 77 tracks
 45 8whight   = Whight 8 Inch Double Sided
 46 8xeroxd   = Xerox 820 8 Inch Double Sided
 47 5cs40s8   = CS 1 x 40 tracks on 80 tr drive
 48 5dig408   = Digital VT180 2 x 40 tr on 2 x 80 tr drive
 49 5rc4224   = Rc422 2 x 40 tracks
 50 5apitei   = Apitei 1 x 40 tracks
 51 8-256s    = 8 Inch (Diagnose) 256 bytes per sector
 52 5q80s     = Compactboard Standard 5 Inch 1 x 80 Tracks
 54 5sup708   = Superbrain QD 2 x 35 tracks on 80 tr drive
 55 hulpxxx   = Empty Dpb for setup of new formats
 56 5osbss8   = Osborne Single Density 1 x 40 tr on 80 tr drive
 57 5kay4 8   = Kaypro 4 2 x 40 tr on 80 tr drive
```

Note: Most 3.5 inch diskdrives are compatible with the 5.25 inch drives, so there formats are identical.

Codes marked with a (*) must be used with the MSCOPY Utility
Codes marked with a ($) are not readable by the Combidos file system because these formats are from other operating systems. By installing the Dpb for such a diskette a program may read sectors and support the file system of that operating system itself.

How to install a new format (Function C)

First choose a similar format from the Dpb file or use 'hulpxxx' (55) and change that format to the new specifications. You can add it with option A as a new entry to the Dpb file on disk (DINST.DAT)

All numbers are input in hexadecimal if prompted with a Dollar '$'. When DI does not accept a number, the number is either not a valid number or it is out of range for the prompted parameter.

Number to Choose:

18 Highest TrackNumber per Surface (if different)
- $22 = 35 tracks
- $27 = 40 tracks
- $4c = 77 tracks
- $4f = 80 tracks
- $9f = 160 tracks

12 Single = $0 / Double density = $40

The sequence of input of the sector lenght and numbers depend on whether the wanted number of sectors is less than the currently show description (take section 'A') or the number of desired sectors is more than currently shown (take section 'B').

Section A

---- Number of Physical sectors is currently more than desired ----

15 Highest Sectornumber on side 0 ($4..$1f)
13 Lowest sectornumber on side 0 ($0..$1)
14 Physical sectorsize
- 0 = 128 bytes per sector
- 1 = 256 bytes per sector
- 2 = 512 bytes per sector
- 3 = 1024 bytes per sector
- 4 = 2048 bytes per sector
- 5 = 4096 bytes per sector

Note that sectorsize 4 and 5 are not available on all types of Floppy disk controllers.

---- Skip to section C. ----

Section B

---- Number of Physical sectors is currently less tan desired -----

14 Physical sectorsize (see above)
15 Highest Sectornumber on side 0 ($4..$1f)
13 Lowest sectornumber on side 0 ($0..$1)

Section C

After entering the parameters of section A or B you can see the system has updated 0 (Number cf sectors per track).

9 Offset. This is the tracknumber on which the directory starts.

2 Block mask:

The system asks to compute the standard blocksize related variables for you (1: blockshift, 2: blockmask, 3: extend mask and 4: Number of blocks). If you answer 'Y' on this question.

Otherwise, give the Blockmask (the system then computes 1,3 and 4) and if necessary change:

3 Extend mask (Exidy Sorcerer for instance is not standard)

5 Directory size (Number of directory entries), the system puts your input-1 into the block and updates 6,7 and 8.
In non-standard cases (Kaypro) you can change:

6 Directory reservation bits first part
7 Directory reservation bits second part

10 The left nibble is the step rate of the disk drive. The real step rate depends on the chosen drive type 5.25 or 8 inch (Selection 25).

5 Inch	St.rate	8 Inch	St.rate
$Fx	2 ms	$Fx	1 ms
$Ex	4	$Ex	2
$Dx	6	$Dx	3
$Cx	8	$Cx	4
$Bx	10	$Bx	5
$Ax	12	$Ax	6
$9x	14	$9x	7
$8x	16	$8x	8
$7x	18	$7x	9
$6x	20	$6x	10
$5x	22	$5x	11
$4x	24	$4x	12
$3x	26	$3x	13
$2x	28	$2x	14
$1x	30	$1x	15
$0x	32	$0x	16

On some Fdc's these values are rounded to the nearest available value.

The right nibble is the head unload time, it can be set at $xF since normally no delay is needed.

11 Head load time (left seven bits)
5 inch: $2 = 4 ms, $FE = 508 ms
8 Inch: $2 = 2 ms, $FE = 254 ms

20 Side mode 0 = Single sided
1 = Double, first side 0 then side 1
2 = Double, alternating side 0 and 1

19 Double steps flag: 0 = normal, 1 = double stepping. To be used when you use a 40 track disk on a 80 tr drive.

23 Verify flag 0 = No verify after write,
1 = verify after write (default on)

21 Number of retries for both read and write operations.

24 Motor startup time in 10 ms steps Normally 50 (= 500 ms) An 8 Inch drive with an AC motor can be set to 1 (10 ms).

25 Drive-type Bit 0: 1 = 8 Inch, 0 = 5 Inch.
Bit 1: 1 = (Superbrain) inverted data
Bit 2: 1 = (P2000M) Track numbering offset
Bit 3: 1 = (Kaypro) side 1 has Head-ID of side 0 with higher sectornumbers.
Bit 4: 1 = Write precompensation ON (Stand. Off).

To restore a drive initialisation to the normal Combidos format type the command BOOT. This will reset the system and load the Combidos from the system disk.

2.2.9 File eXchange

The file exchange utility allows data to be transfered from any input to any output device. The program will first buffer data till either an end of file character is received or the buffer is full.
The command format is: A% **FX con:=docfile.txt**
When no command line is given, FX will explain its operation and allow entry of more than one command line. FX terminates when an empty command line is given.

2.2.10 DCOPY

The DCOPY utility is a copy program that allows the user to make a full disk backup in the shortest time possible. DCOPY will copy all used blocks of the master disk and can be used on single and multi drive systems with both drives installed to the same disk format. When a copy must be made from one format to another, use FCOPY. The backup-disk is verified for correctness to insure a 100 % reliable backup.

The DCOPY command format is:
DCOPY source drive, destination drive.

When no drive specification is given, DCOPY starts with a default setting of a copy from drive ~~A~~ to ~~B~~. the other logged

The top of the screen will show a command line:

C)opy from Drive A: to Drive B: O)k to copy <Ret>|X|Q)uit

On a two drive system the original (SOURCE) disk must be inserted in drive A and an empty (formatted disk) in drive B (DESTINATION). The command O will show the copy information:

Total diskSize = 800k. DiskSize = 790k. BlockSize = 2k.
Total Used = ...k. Free Space = ...k. Blocks to copy = ...

Press a key to confirm Start (^C Abort)

With the C command the source and destination drive can be changed. For single drive systems, assign both source and destination to drive A and give the O command. In this case, DCOPY will ask for source and destination disks when needed. The messages for changing disks and disk errors are shown in the bottom line of the screen. In the middle of the screen, DCOPY shows the current operation it is performing. The copy process may be aborted at any moment by typing ctrl/C. A warning is show that the backup is not finished.

2.2.11 COMINST (Combidos terminal install)

The combidos utilities provide the possibility to use them with any 80 collumn terminal. The video attributes codes can be changed to the code of that terminal. For this purpose the utility COMINST is used.
When COMINST is started, it has a default set of control codes configured for the Compactboard 80 (compatible with TVI 910/950). The set can be shown (selection A) and changed sequentially (selection B).
Note that for the cursor addressing lead-in 2 extra characters are needed (row and collumn characters).
The current set of control codes can be installed to any number of Combidos utilities. COMINST selection D will ask for the file name (a .COM file is assummed) and perform the installation patch. Do not try to use COMINST for other files, as the results are unpredictable.
Selection C reads the installation from a file.

2.2.12 AUTORUN

When a program is to be run as soon as the computer is switched on, the programs name must be in the command line of Combidos when it is read from disk. The utility AUTORUN will write that name to the command line of the system of disk A.
A% **AUTORUN DEMO**
Will cause the disk in drive A to start the demo program after power-on.

CHAPTER 3
Error messages

A number of errors may occur when reading or writing a disk. The operating system will 'trap' these errors and report them on the console. The operating system distinguishes four types of errors with an error description of:

Disk error
Drive not Online
Read Only.
Write Protect?

The form of the error message is:

Dos Err Disk: drivename: errordescription

The Disk error error indicates that the disk has a sector that can not be read or written properly. The reason for this can be one of the following:
- The diskette is not formatted (or formatted in a different way).
- The diskette is worn-out. Copy (or try to copy) all files to a new diskette.
- The electronics need adjustment. Call a service engineer.

Recovery from such an error is accomplished by typing ctrl/C to reboot, or a **return** which ignores the bad read/write operation. Note however that typing a return may destroy the file-system of the diskette.
It is recommended to make copies of all your files on separate disks. Despite the fact that modern disk drives are very reliable, disks may get damaged by handling.

The Drive not Online error occurs when an attempt is made to switch to an unexisting disk. Recover from this error by typing a key. Combidos will perform a warm boot as with ctrl/C.

A Read Only error occurs when an attempt is made to write to a file that is marked 'read only' by the operating system or a program. All files of a diskette are marked 'read only' when the diskette is changed without a warm boot with ctrl/C (see 1.3.2). Recover from this error by typing a key. Combidos will perform a warm boot as with ctrl/C.

A **Write Protect?** error will occur when a write is performed to a disk that is physically protected for write. Such a protection is described in the manual of the disk drive.

CHAPTER 4
How to get started

4.1 First time starting

Switch the power of your computer on and insert the Combidos system disk in drive A:. Combidos will report itself ready for operation with the system prompt **A%**.
The Combidos operating system is now loaded and if you do not have a extra copy of your system disk, you should make that copy now (see 4.2 Disk duplication).

4.2 Disk duplication

It is advised to make a backup of all your data files or disks (on which you have made changes) every week. For this purpose the command DCOPY is used.
There are two ways of creating a copy of a disk: when the disk format of the copy must be the same as the format of the master, use DCOPY (see 2.2.10), when the formats of the master and the copy must differ, use FCOPY (see 2.2.2). DCOPY will make a physical copy of the original disk while FCOPY will copy **files**. The backup disk must be a formatted disk in both cases.
Both copy programs can operate on single and multiple drive systems so a user is always capable of making a back-up disk.

4.2.1 Preparing a blank disk for use

Take a blank diskette and insert it in drive B. Format the disk with the FORMAT command (see 2.2.5). The format command will initialise the disk but does not write the system onto the system tracks. When the backup is made with FCOPY the system tracks must be written onto the disk separately with the PUTSYS program (see 2.2.7), DCOPY will automatically copy the system track from the master onto the backup disk.

4.2.2 Disk Backup

Making a disk backup with DCOPY is done in the following manner:

- start DCOPY by typing:
  A% **DCOPY**

DCOPY will report itself.

When you have a 2 drive system, place the original disk in drive A and a formatted blank disk in drive B and type O and a return. Now the contents of the disk in drive A is copied to the disk in drive B.

When you have a single drive system, place the original disk in drive A and type in a sequence: ~~C A A O~~ and a ~~return~~.
Now see to it that you follow the instructions on disk exchange very carefully. After you have exchanged a disk, enter a return.

4.2.3 Copying files to another disk

When files must be copied to another disk, use FCOPY. Insert the master disk in drive A and a blank formatted disk in drive B. Log in on drive A by typing **L A** and select files with the ´S´-key (otherwise skip with return or space) and copy with the ´C´-command by typing **C B** (see 2.2.2).
On a single drive system use the ´C´ command with a drive A to drive A copy by typing **C A.**
When converting other disk formats on a single drive system, see to it that the floppydisk drive is selected as drive A and drive B. Log in on drive B by typing **L B,** select the files as described and use with the ´C´-command the option ´S´ for single drive copy by typing **C S A.**

4.3 File printing

A text file can be listed on the console with the TYPE command. When a listing on the printer is desired, use the TYPE-command and type a ctrl/P before the return. The listing will now appear on the console and also on the printer. Halt, continue and exit will work normal (ctrl/S, ctrl/Q, ctrl/C respectively).
When the listing is finished type ctrl/N to switch off the printer (see 1.2.4).
An other way of printing a file is using the FX (File eXchange) utility: A% **FX LST:=TEXTFILE.DOC**

4.4 Using disks of other computers

If you have bought software that is not supplied in the Combidos disk format, you can configure one of your drives to read that format (see 2.2.8). If, for instance, you bought a package and received a disk in the BULLET 80 tracks format, configure drive B using the DI utility. Select from the Dpbfile the BULLET format **5BUL160** and install this format to drive B. This procedures must be followed for both single and multi drive systems.

Now, addressing drive B, you can read and write BULLET disks. Make a copy of the programs to a working diskette using FCOPY, with B to A copy (see 4.2.3).

NOTE: On single drive systems, the only way to run programs, is to have them on a combidos formatted working diskette.

In general: DO NOT TRY TO RUN PROGRAMS, NOT STORED ON COMBIDOS DISKETTES.

4.5 Direct Starting Disks

A disk can be made to start a program directly after power-on. Combidos supports this with the AUTORUN utility. When you want to have the program DEMO.COM start after power-on, copy it to your disk and also copy AUTORUN to that disk. Now type:
A% **AUTORUN DEMO**

------O-O------

CHAPTER 5

COMBIDOS system calls

The proper way to make use of system resources is to let Combidos handle the functions. This makes programs written under Combidos or CP/M portable from one system to another, without having to modify them. Combidos supports all the BDOS and BIOS calls supported by CP/M and more.

Combidos entries
The entry to combidos functions are located in low memory. The first location is called BOOT. All other entries are referenced with an offset to BOOT. Normally BOOT is located at address 0000h

BOOT:		Warm boot of the combidos system.
BOOT+0005h	BDOS:	BDOS command entry
BOOT+005Ch	FCB1:	File control block 1
BOOT+006Ch	FCB2:	File control block 2

Commands and command line support.

The command handler will accept a command, typed on the console, load the command in the program area and will prepare one or two file control blocks (FCB´s) when the command is followed by file specifications. The two optional FCB´s are located at BOOT+005Ch and BOOT+006Ch. The part of the command line after the command itself is stored at BOOT+0080h, starting with a length byte, followed by the string.

BDOS function calls

A set of functions is implemented in combidos perform I/O related operations. All functions are called at the same address (BDOS) and must have the function number in register C.
All BDOS calls are 100% compatible with the Digital Research CP/M operating system.
The functions are described in detail individually:

00 System Reset

entry: Reg. C = 00h
exit : none

Performs a warm boot in the same way as a call to BOOT would do. The command handler is reloaded and the disks are logged.

01 Console input

entry: Reg C = 01h
exit : Reg A = ASCII value

Reads a character from the console to the A register. All characters are echoed to the console. A tab (ctrl-I) moves the cursor to the next tab stop. Stop/continue (ctrl-S / ctrl-Q) and printer on/off (ctrl-P / ctrl-N) are checked and executed. This function explicitly waits until a character has been typed, so execution is suspended when no character is typed.

02 Console Output

```
entry:   Reg C  =  02h
         Reg E  =  ASCII character
exit :   none
```

The ASCII character from register E is written on the console device. Tab´s are expanded, stop/continue and printer on/off are checked as in function 1.

03 Reader input

```
entry:   Reg C  =  03h
exit :   Reg A  =  ASCII value
```

Reads the next character from the logical reader device into register A. The function suspends execution until a reader character is read.

04 Puncher Output

```
entry:   Reg C  =  04h
         Reg E  =  ASCII character
exit :   none
```

Sends the character from register E to the logical puncher device.

05 List Output

```
entry:   Reg C  =  05h
         Reg E  =  ASCII character
exit :   none
```

Sends the character from register E to the logical list device.

06 Direct Console I/O

```
entry:   Reg C  =  06h
         Reg E  =  0FFh (for input) or
                   ASCII character (for output)
exit :   Reg A  =  char or status (input only)
```

This function is for direct I/O to the console, where no control character functions of Combidos are active. The E register determines whether the function performs input (E = 0FFh) or output (E = ASCII character).
When used for input, E must be 0FFh. When a character was present the input character is return in register A , otherwise A is 00 (zero).

07 Get I/O Byte

```
entry:   Reg C  =  07h
exit :   Reg A  =  I/O Byte value
```

This function returns the information of the current I/O Byte.

08 Set I/O Byte

```
entry:   Reg C  =  08h
         Reg E  =  I/O Byte value
exit :   none
```

This function writes the contents of register E to the I/O Byte.

09 Print string

```
entry:   Reg C  =  09h
         Reg DE =  String address
exit :   none
```

The string stored at the address given by the DE register is send to the console device. The string terminator symbol is a $. As with character console output, tabs are expanded, stop/continue and printer on/off is checked.

10 Read Console buffer

```
entry:   Reg C  =  0Ah
         Reg DE =  Buffer address
exit :   console characters in buffer
```

This function reads characters from the console device into the specified buffer. Input is terminated by either buffer overflow (more than 255 characters) or a CR or LF code. Input editing is allowed using the control code described in paragraph 1.2.4 (Line editing).
The buffer holds characters with string lenght in the first position, followed by the string. The buffer is not initialised before character entry, so the lenght byte must be used for end of string calculation.

11 Get Console status

```
entry:   Reg C  =  0Bh
exit :   Reg A  =  status
```

This function checks if a console character is ready. If so, the value 0FFh is returned in register A; if not, 00h is returned.

12 Return Version number

```
entry:   Reg C  =  0Ch
exit :   Reg HL =  version number
```

This function provides information to the user program about which version of the operation system is running.
Combidos will return : H = 00, L = 22h

13 Reset Disk system

entry: Reg C = 0Dh
exit : none

This function resets the file system to a state where all disks are read/write, drive A is selected and the default DMA address is set to BOOT + 0080h. This function can be used by application programs when disks have to be changed.

14 Select Disk

entry: Reg C = 0Eh
Reg E = drive number

The specified drive is selected for subsequent file operations. FCB´s that specify a drive code zero (dr=00h) automatically reference the currently selected drive. Other drive code values explicitly select a drive.

15 Open File

entry: Reg C = 0Fh
Reg DE = FCB address
exit : Reg A = directory code

The Open file function will open a file, specified in the FCB, that currently exists on the active disk. Question marks in the file name match to any character in that position. If the file is found in the directory, all relevant information is copied into the FCB. Hereafter file access is allowed.
The open file function returns a directory code in register A in the range 0..3 for a successfull open function. If the file is not found on the selected disk, 0FFh is returned.

16 Close File

entry: Reg C = 10h
Reg DE = FCB address
exit : Reg A = directory code

Closes the file specified in the FCB. The disk directory is updated and a directory code is left in register A. The values 0..3 flag a correct close operation while 0FFh indicates that the file could not be found in the directory.

17 Search for First

```
entry:   Reg C  =  11h
         Reg DE =  FCB address
exit :   Reg A  =  directory code
```

This function scans the directory for a match with the given file name. In register A the value 0FFh is returned if the file is not found, otherwise a value in the range 0..3 is returned. When the file is found, the field at the current DMA address is filled with the record containing the directory entry. The relative starting position of the directory entry is A * 32 (rotate A left 5 bits). A question mark will cause a match on any character in the file name. If the drive specification in the FCB hold a ?, the auto-disk select function is disabled and the default disk drive is selected.

18 Search for Next

```
entry:   Reg C  =  12h
exit :   Reg A  =  directory code
```

This function is a continuation of the search made by function 17. The search is continued at the last matched entry. The directory codes left on exit are the same as in function 17.

19 Delete File

```
entry:   Reg C  =  13h
         Reg DE =  FCB address
exit :   Reg A  =  directory code
```

The file(s), specified by the given FCB is (are) removed from the file system of the selected drive. The A register returns 0FFh is no file is found, otherwise a value in the range 0..3 is returned.

20 Read sequential

```
entry:   Reg C  =  14h
         Reg DE =  FCB address
exit :   Reg A  =  directory code
```

Once a file is open for read the Read Sequential function can be used. It will read the next sequential record (128 byte) from the file into memory at the current DMA address. The record pointers in the FCB are advanced to the next record. When a logical extent overflows, the DOS will open the next extent automatically. The register A leaves 00h when the read was successfull, otherwise A will be nonzero, indicating no further data available.

21 Write sequential

```
entry:   Reg C  =  15h
         Reg DE =  FCB address
exit :   Reg A  =  directory code
```

An open file can be written to by the Write Sequential record function. The record to be written is located in memory at the current DMA address. The file is placed at the current record position in the file and the record pointer is advanced to the next record. Extent overflow is handled automatically by the DOS. A write operation into an existing file overwrites the old record. Register A is 00h for a successfull write, a nonzero indicates an unsuccessfull write caused by a full disk.

22 Make File

```
entry:   Reg C  =  16h
         Reg DE =  FCB address
exit :   Reg A  =  directory code
```

The Make File operation creates a new file on the referenced disk. The file name specified in the FCB must not exist on that disk. The return value in A is 0..3 for a successfull operation and 0FFh when the directory is full.
The created file is implicitly opened and the FCB is updated.

23 Rename File

```
entry:   Reg C  =  17h
         Reg DE =  FCB address
exit :   Reg A  =  directory code
```

This function is used to rename all occurences of a file name. The old name is specified in the first 16 byte of the FCB and the new name is in the second 16 byte. The drive code (first position) will select the drive, the drive code of the second name must be zero. If the rename was successfull the return value in register A is 0..3, when the old name could not found, A is FFh.

24 Return Log-in vector

```
entry:   Reg C  =  18h
exit :   Reg HL =  Log-in vector
```

The log-in vector indicates which drives are currently logged in. The vector is given as a bit-table, the least significant bit of L corresponds to drive A and the most significant bit of H corresponds to drive P. A 0 (bit is zero) indicates that the drive is not logged.

25 Return Current Disk

```
entry:   Reg C  =  19h
exit :   Reg A  =  Current disk
```

The currently selected disk number is returned in register A. The disk number is in the range 0..15, Zero corresponds to drive A.

26 Set DMA address

```
entry:   Reg C  =  1Ah
         Reg DE =  DMA address
exit :   none
```

The Direct Memory Access (DMA) address is used to address the 128 byte record from disk. After cold start, warm start or reset disk system (function 13) the DMA address is set to BOOT + 0080h. The user may change this address for subsequent operation.

27 Get ADDR (alloc)

```
entry:   Reg C  =  1Ah
exit :   Reg HL =  Alloc address
```

For each on-line diskdrive, an disk allocation table is kept in memory. This table can be used by system programmers to calculate the remaining disk capacity. The Get ADDR function returns the base address of the allocation table of the currently selected drive. Note that the allocation information may be invalid if the disk has been marked read-only.

28 Write protect disk

```
entry:   Reg C  =  1Ch
exit :   none
```

This function marks the currently selected drive 'write-protected' (read-only). An attempt to write to a write-protected disk will cause a BDOS Read/only error.
The write-protect status is cleared by a cold or warm start and the reset disk system function (function 13).

29 Get Read/only vector

```
entry:   Reg C  =  1Dh
exit :   Reg HL =  R/O vector value
```

The read/only vector is of the same format as the log-in vector. For each drive that is currently read/only (write-protected), the corresponding bit is set.

30 Set File Attribute

```
entry:   Reg C  =  1Eh
         Reg DE =  FCB address
exit :   Reg A  =  directory code
```

With this function, the file attributes can be changed by a user program. The attributes, set in the given FCB are copied into the directory entry with the matching file name.

31 Get ADDR (Disk Parameters)

```
entry:   Reg C  =  1Fh
exit :   Reg HL =  DPB address
```

To change the current values of the disk parameter block the base address of the DPB is returned by this function. Note that the Combidos disk parameter block is an extended DPB, so care must be taken when using this function. The Disk Install utility (DI) makes use of this function.

32 Set/Get User code

```
entry:   Reg C  =  20h
         Reg E  =  0FFh for 'get code',
                   User Code for 'set code'
exit :   Reg A  =  Current code (get code)
```

A program may interrogate or change the currently active user number. To set the user number, pass the number in register E. To get the current user number, set register E to 0FFh. On exit, register A will hold the user number (0..15).

33 Read Random

```
entry:   Reg C  =  21h
         Reg DE =  FCB address
exit :   Reg A  =  Return code
```

The Read Random function reads a record of the file as the sequential file read. In this case however, a 24 bit record pointer in position 33 (LSB), 34 and 35 (MSB) in the FCB determines the record read. The MSB indicates overflow when not zero.
When using a random access file, the file must be opened. The requested record number must be stored in the FCB record pointer and the Read Random function must be called. The error code in register A indicates:

```
00  successfull read
01  reading an unwritten part of the file
03  current extent cannot be closed
04  seeking to an unwritten extent
06  disk file overflow on seek
```

Error 01 and 04 will occur when a read is performed of a record that was not written. An unwritten record returns no valid data. If error 03 occurs, which is normally not the case, reread the record or reopen the file. The disk must not be physically write protected. Error 06 indicates a seek past the maximum file length.

34 Write Random

```
entry:   Reg C  =  22h
         Reg DE =  FCB address
exit :   Reg A  =  Return code
```

The record, located at the DMA address is written to the file on the position pointed to by the record pointer in the FCB. The write operation is very simular to the read random function. The error codes returned are the same as with function 33, with the addition of error number 05 which occurs when the directory overflows due to extent creation.

35 Compute File Size

```
entry:   Reg C  =  23h
         Reg DE =  FCB address
exit :   Record pointer set
```

This function scans the directory for the filename, given in the FCB, and leaves the record pointer pointing to the record following the end of the file. The pointer represents the virtual file size. When a file contains unwritten records, they are calculated as if they were written.
This function can be used when data must be appended to a file. First compute the filesize to set the record pointer to the first free record, then perform sequential or random writes.

36 Set Random record

```
entry:   Reg C  =  24h
         Reg DE =  FCB address
exit :   Record pointer set
```

This function is used to have the BDOS set the random record pointer from a file that has been read or written sequentially to a particular point and needs to be read or written randomly.

37 Reset Drive

```
entry:   Reg C  =  25h
         Reg DE =  Drive vector
exit :   Reg A  =  00h
```

The Reset drive function allows resetting of one or more drives. A drive that is reset is logged in first when operations to it are performed. The drive vector is a 16 bit number, the LSB represents drive A. A bit set indicates that the drive is to be reset.

40 Write random with zero fill

```
entry:   Reg C  =  28h
         Reg DE =  FCB address
exit :   Reg A  =  Return code
```

The Write random with zero fill is simular to the Write random function (34). Only, when a block (of records) is assigned to the file (allocated) it is filled with zeros before data are written.

CHAPTER 6

BIOS entry points

The entries in the Combidos BIOS are a number of jumps to routines that perform hardware dependant functions.

The routines are:

Name	Function
Boot	Cold boot entry. Resets all functions and reloads the DOS and the Command handler.
Wboot	Warm boot entry. Reloads DOS and the command handler. The jump at address 0000 points to entry.
Const	Returns the input status of the console device. If a character is received, FF is returned in A, otherwise 00 is returned.
Conin	Reads one character from the console into the A register. If no character is present, execution is suspended.
Conout	Sends the character from the C register to the console output.
List	The character in register C is send to the logical list device. If the list device is not ready to receive a character, execution is suspended.
Punch	The character in register C is send to the logical punch device. If the punch device is not ready to receive a character, execution is suspended.
Reader	Reads one character from the reader device into the A register. If no character is present, execution is suspended.
Home	Brings the read/write head of the currently selected disk to the track 00 position.
Seldsk	Selects the diskdrive given by the C register for further operation. Drivenumbers are: 0 for ´A´, 1 for ´B´ and so on. Seldsk returns the base address of the **Disk parameter header** in HL. A value 0000 is used to indicate an non-existing drivenumber.
Settrk	Register BC holds the tracknumber which is to be used in following disk operations. The head is actually moved in position by this command.

Setsec — Register BC holds the sectornumber which is to be used in the following disk operations.

Setdma — The BC register contains the **disk memory access** address for read/write operations. This address represents the start address of the logical 128 byte sector.

Read — This routine performs a sector-read of the selected sector on the selected track of the selected drive. The sector is loaded in memory at the given DMA address. Errors are marked in register A:
0 = no error, 1 = error.

Write — This routine writes the contents of the buffer, indicated by the DMA-address, to the selected sector on the selected track of the selected drive. It automatically verifies that the contents of the written sector matches the DMA-buffer contents if the verify flag in the disk parameter block is on.
The sector-size used depends on the size indicated in the ID-field of the disk. The user must use the same sectorsize.

Listst — The status of the logical list device is returned in A. If A is not zero, the list device can receive a character.

Sectrn — The translation from logical to physical sectornumber is performed here. The logical sectornumber must be passed in BC and the translation table address in DE. The sector number is used to index the translation table and return the physical number in HL.

The following is a listing of the BIOS jump table of combidos.

Note: Entries with an asterix '*' are not implemented.

```
;       Jump table 2.2 alike functions
;
jp      boot    ; 0 xx00 cold boot = reset system
jp      wboot   ; 1 xx03 warm boot = Normal end of Program
                ; Vector on Addr 0 Points to This One
                ; Reloads Dos and Command handler
jp      consti  ; 2 xx06 console (con:) In status A, FF chr ready.
jp      conin   ; 3 xx09 console (con:) char in
jp      conout  ; 4 xx0c console (con:) char out
jp      lstout  ; 5 xx0f list    (lst:) char out
jp      PunOut  ; 6 xx12 aux     (aux:) char out
jp      RdrIn   ; 7 xx15 aux     (aux:) char in
jp      home    ; 8 xx18 move head to home pos
jp      seldsk  ; 9 xx1b select disk in C
jp      settrk  ;10 xx1e set track   in C
jp      setsec  ;11 xx21 set sector  in C
jp      setdma  ;12 xx24 set dma address in BC
jp      read    ;13 xx27 read  disk ret A 0-Ok 1-Err
jp      write   ;14 xx2a write disk wrcode in C ret A 0-Ok 1-Err
jp      lststo  ;15 xx2d list    (lst:) out status
jp      Sectran ;16 xx30 sector translate returns in DE
                ;       Adress of IdTable
```

```
                ;--- Jump table 3.0 alike functions ---
jp      consto  ;17 xx33
jp      RdrSti  ;18 xx36
jp      PunSto  ;19 xx39
jp      devtbl  ;20 xx3c *              if '*' --> not used yet
jp      devini  ;21 xx3f *       entries return A:1 as error if Used
jp      drvtbl  ;22 xx42 *
jp      multio  ;23 xx45 *
jp      flush   ;24 xx48 *\
jp      move    ;25 xx4b  Implemented  Source  address  DE,  Dest
                          address HL, BC Size.
                          For  interbank  Moves  call  Xmove  before
                          calling move.
                          Note: Move Resets after completion the
                          Xmove InterBank switch.
jp      time    ;26 xx4e */
jp      selmem  ;27 xx51  in A: Bank# to Use as current Bank
jp      setbnk  ;28 xx54 *
jp      xmove   ;29 xx57  in C: SourceBank, in B: Destination bank
                          Must be called just before 'Move'
                          to do interbank move ( <= 128 bytes )
jp      userf   ;30 xx5a *
jp      resrv1  ;31 xx5d *
jp      resrv2  ;32 xx60 *
                          ;--- Extended Function table ---
jp      PunSti  ;33 xx63 Punch input status
jp      Punin   ;34 xx66 Punch input
jp      RdrSto  ;35 xx69 Reader output status
jp      RdrOut  ;36 xx6c Reader output
jp      lststi  ;37 xx6f        ;list input status
jp      lstin   ;38 xx72        ;list input
jp      readad  ;39 xx75_*      ;read next id field (depends on fdc)
jp      readid  ;40 xx78 *      ;read next 26 adress marks into dirbuf
jp      hsldsk  ;41 xx7b * select host disk in (curdsk) ret dpb in DE
jp      hread9  ;42 xx7e * read host disk sector ret A-0 inc Zflag Ok
jp      hwrit9  ;43 xx81 * write host sect curdsk-curtrk-cursec-curdma-
jp      rreadt  ;44 xx84 * read host track
jp      rwritt  ;45 xx87 * write host track
jp      wread   ;46 xx8a * read  whole track              -> not on
jp      wwrite  ;47 xx8d write whole track ( format  )  -> the 765 fdc
jp      format  ;48 xx90 * format a track on host disk !-> not on others
jp      GetMem  ;49 xx93 Get Current Memory Bank
jp      blocks  ;50 xx96  BC number of records (128 byte) for the Bios
                          to use for disk buffering. Returns in HL the
                          top of the free space in bank 1.
                          In bank 1 data can be stored from 0100h till
                          the top, given by 'blocks'.
jp      xxx     ;51 xx99 * reserved
;
 .....
```

```
 ...
;
IdTble: dw      Dos-2    ;last  2 bytes of Command processor = IdNumber
        dw      Dos      ;first 6 bytes of Dos = Idnumber
        db      BiosId,0ffh-BiosId ;this is the IdNumber of Bios
        dw      Clock    ;adress of first byte of Irq clock
        dw      SetCon   ;adress of console Acia setup
        dw      IoTbl    ;Adress of I/O Byte translation table
                         ;conin,conout,rdrin,rdrout,punin,punout,listin,lstc
                         ;each has 4 words
 ....
 ..
;
BotDsk: db      0        ;bootdisk number for warm boot
clock:  db      0,0,0,0,0,0     ;seconds, first byte is LSB
klok75: db      1        ;First time
```

---o-o-o-o---