

UNI-DOS

From S. O. Software.

UNI-DOS MANUAL

Contents

Section	page
1 Introduction	2
2 Plus D Installation	3
3 DISCiPLE Installation	5
4 System Setup	7
5 File, Disk and Program Conversions	8
6 Simple Commands	9
7 Random Access Files	12
8 Printer Commands	15
9 Advanced Commands	16
10 Functions	20
11 Create Files	21
12 Snapshot Functions	23
13 Error codes	24
14 Hook codes	25
15 Directory Structure	27
16 Dos Variables	28

UNI-DOS MANUAL

Introduction

First of all thank you for buying UNI-DOS and welcome to a new level of power for your Spectrum.

I expect that you have already had many hours of use from your PLUS D or DISCiPLE, and now, with UNI-DOS, you can look forward to many more.

Before going any further check that you have everything you need to run UNI-DOS. You should have received this manual, a disc and a REPROM chip. There are two versions of the EPROM chip. Do make sure that you have the correct chip for your system. You can not use the wrong chip. It won't do any harm but it will not work.

You should also have a disk. This will be formatted for 80 tracks and can be used on both single and double sided drives. If you have the wrong chip or a disk that will not work in your drive please return the entire package to S D Software clearly stating the problem. DO NOT FORGET TO INCLUDE YOUR NAME AND ADDRESS.

Now that you are sure you have all the parts you can begin to install UNI-DOS on your machine. If you have a PLUS D then follow the instruction in the [next section](#). DISCiPLE users should go to [section three](#).

UNI-DOS MANUAL

PLUS D installation

To install UNI-DOS on your PLUS D you will need a small cross headed screwdriver, a small flat bladed screwdriver and about 30 minutes.

Read through these instructions BEFORE you begin.

With all power switched off disconnect your PLUS D completely from the Spectrum and remove all cables connected to it.

You will find two small cross head screws an each side of the PLUS D. Carefully remove these and keep them safe. Ease the top cover off the PLUS D to expose the printed circuit board.

Examine the circuit board and make note of the three large chips. The ROM is the chip nearest to the printer plug and furthest away from the snapshot button. Note also that this chip has a small notch at the end nearest to the centre of the circuit board.

Remove this ROM chip using the flat blade screwdriver by easing the blade between the chip and the socket it is plugged into. Do take the greatest of care as these chips and the circuit boards can be damaged all too easily.

Now is the time to remove the new UNI-DOS chip from its packaging. Examine the chip to make sure that none of the pins or legs aren't bent or broken. Place the new chip onto the socket making quite sure that the end with the notch is to the centre of the circuit board. **IF YOU GET IT THE WRONG WAY YOU WILL DESTROY THE CHIP WHEN YOU SWITCH ON THE POWER.** Taking care not to bend any of the legs press the new chip into the socket.

Now test that the PLUS D is working by connecting it to a Spectrum and switching on. There is no need for a disk drive to be connected at this stage. You should get the normal switch on screen. Then type RUN and press enter and you should get a message to CHECK disk in drive. If this is not the case then switch off and check that you have not got any chip legs bent under the chip or not in the socket, or that you have not plugged the chip in the wrong way.

Assuming that all is well switch off and disconnect the PLUS D from the Spectrum. Now replace the top onto the PLUS D. Take care with the red light to get it in its hole. Replace the screws and reconnect the cables and plug into your Spectrum. Switch on to make sure everything is still working.

UNI-DOS MANUAL

Store your old GDOS chip in the packaging that the UNI-DOS chip came in and take care of it. Just in case you ever want to go back to using GDOS.

Now go to [section four](#) and follow the setup procedure.

UNI-DOS MANUAL

DISCiPLE installation

To install UNI-DOS on your DISCiPLE you will need a small cross headed screwdriver, a small flat bladed screwdriver and about 45 minutes.

Read through these instructions BEFORE you begin.

With all power switched off disconnect your DISCiPLE completely from the Spectrum and remove all cables connected to it.

You will find six small screws on the underside of the DISCiPLE. Carefully remove these and keep them safe. Remove the base cover off the DISCiPLE to expose the printed circuit board. Turning the DISCiPLE over you will find a screw at either side of the Spectrum edge connector. Remove these and keep them safe. You should now be able to drop the printed circuit board out of the top cover. Do take care not to damage the Spectrum connector which will need to be helped through the slot in the top cover.

Examine the circuit board and make note of the three large chips. The ROM is the chip nearest to the Spectrum connector and furthest away from the snapshot button. Note also that this chip has a small notch at the end nearest to the centre of the circuit board.

Remove this ROM chip using the flat blade screwdriver by easing the blade between the chip and the socket it is plugged into. Do take the greatest of care as these chips and the circuit boards can be damaged all too easily

Now is the time to remove the new UNI-DOS chip from its packaging. Examine the chip to make sure that none of the pins, or legs are bent or broken. Place the new chip onto the socket making quite sure that the end with the notch is to the centre of the circuit board. **IF YOU GET IT THE WRONG WAY YOU WILL DESTROY THE CHIP WHEN YOU SWITCH ON THE POWER.** Taking care not to bend any of the legs press the new chip into the socket.

Now test that the DISCiPLE is working by connecting it to a Spectrum and switching on. There is no need for a disk drive to be connected at this stage. You should get the normal switch on screen. Then type RUN and press enter and you should get a message to CHECK disk in drive. If this is not the case then switch off and check that you have not got any chip legs bent under the chip or not in the socket, or that you have not plugged the chip in the wrong way.

UNI-DOS MANUAL

Assuming that all is well switch off and disconnect the DISCIPLE from the Spectrum. Now replace the covers onto the DISCIPLE in the reverse manner that you removed them. Replace the screws and reconnect the cables and plug into your Spectrum. Switch on to make sure everything is still working.

Store your old GDOS chip in the packaging that the UNI-DOS chip came in and take care of it. Just in case you ever want to go back to using GDOS.

You will now find that Uni-Dos does not contain any networking code. This has had to be deleted to make space for the random filing code. If there is enough demand then networking may be introduced as a create file.

Now go to section four and follow the setup procedure.

UNI-DOS MANUAL

System File Setup

Now that you have changed the ROM on your PLUS D or DISCIPLE you will need to create a customized system file. The disk already has a system file that should work with all systems but to get the most from UNI-DOS you may need to change some of the defaults. The defaults are for two 80 track drives and an EPSON compatible printer. Before making any changes you should have the drive and printer manuals to hand.

With all the cables connected and the Spectrum switched on load your UNI-DOS disk and type RUN to boot the UNI-DOS system. The first thing that you must do whether you are going to use the setup program or not is to make a copy of the UNI-DOS disk. To do this type LOAD P "backup". The program will auto run and will request that you change disks as required. Once finished the setup program will load automatically. You can load the setup program at any time by typing LOAD P "setup".

The program is menu driven and is simple to use. Often only a single key press is required to answer a question. The first part will configure the number and type of drives you have on the system. Note that the stepping rate is now different to GDOS. The value you enter will be rounded to 2, 3, 6 or 12 ms. Only the last step will use the full time entered. This makes disk use much smoother and faster but keeps the reliability of slow step rates. The default step rate is 25 ms.

The second part of the program, which you may skip, is the printer setup. If your printer is not EPSON compatible you may well have to use this section to make your printer work, otherwise only use it to set up anything special for your printer.

When the program has run a new configured UNI-DOS file will be saved onto your disk and loaded into the PLUS D or DISCIPLE.

UNI-DOS MANUAL

Conversion

If wish to use any of the new UNI-DOS features on execute or snapshot files then you will need to run the conversion program provided. The conversion only changes the directory entry to allow random access to the files. They can be loaded and run without conversion.

Microdrive files have changed completely and if any of your programs use MDRV files then they will need to be converted. This conversion is one way only and once changed the files can not be used by GDOS. It should go without saying that you should convert copies of your files not the originals.

While it is not necessary to convert your disks you will not get the full benefit of UNI-DOS unless you do. With UNI-DOS whenever a disk is formatted a magnetic mark is written onto every sector to identify different disks. This allows UNI-DOS to tell if you have put the wrong disk into a drive during a file copy or if a disk has changed when using opentype files.

To format a disk you now need to load the format program. This is done by LOAD P "format". There are six options. Two to simply format disks in drive 1 or 2. These will erase all data from the disks. The reformat options will first read the information from the disk into memory format the disk and then load the data back so no data is lost, unless the disk can not be read in the first place.

The final options will do a complete disk copy after formatting and can be used to make backup copies of your disks including your UNI-DOS disk.

The last conversion you may need to do is to any programs that use the SAVE command to copy files. There are very few of these as under GDOS a file copy ended with a system reset. The new command is MOVE and can be found in [section 6](#).

UNI-DOS MANUAL

Simple Commands

Anything included below in square brackets is optional and may be left out of the command. <channel> is a number from 0 to 16, and <drive> is 1,2 or * if you want to use the last drive accessed.

RUN

Boots the DOS from a disc in drive 1. If there is a program in memory then that will be run. If you wish to boot UNI-DOS when there is a program in memory use RUN 10000

CAT [d] <drive> ["<filename>"] [!]

CAT [#<channel>;] [d] <drive> ["<filename>"] [!]

This command first prints the disc name if it exists followed by the actual catalogue. The total free disc space available is printed at the end of the listing. The <filename> part of the command when included can cause the listing of all programs with a specified name (e.g. using wildcards). A "!" after the string will produce a catalogue of filenames only. Both the forms CAT 1 and CAT dl are valid, and the listing can be directed to a stream by specifying a channel number.

MOVE [OVER] d<drive>["<filename>"] TO d<drive>["<filename>"]

MOVE (OVER) d<drive>["<filename>"] TO "<filename>"

This command replaces SAVE TO and copies any files (except directories) with non-zero length including SNAPSHOTS, MDRV, OPENTYPES and files longer than 64K. The memory above the BASIC area and below the machine stack is used for transferring the files and so any BASIC resident in memory during this command will not be corrupted since the routine does not reset when it has completed. Wildcards are allowed in both the optional strings. If the second drive number is omitted then the routine assumes you are copying files on the same disc and no prompts are given to swap discs. If OVER is included in the command then duplicate files with the same name are automatically overwritten without asking for confirmation unless the file is protected in which case an error is given.

MOVE #<channel> TO #<channel>

MOVE d<drive>"<filename>" TO #<channel>

MOVE #<channel> TO d<drive>"<filename>"

These commands allow you to move data between channels. If required, a temporary channel to a file is created using the d<drive>"<filename>" form, but the channel will be closed and reclaimed as soon as the command finishes, even if an error occurs.

UNI-DOS MANUAL

ERASE d<drive>"<filename>"[NOT]

Erases the specified file, or files. You cannot erase a file while it has a channel open to it or if the file is protected. Wildcards are allowed and if "NOT" is used then no error is given if the command ends without actually having erased a file.

ERASE d<drive>"<filename>" TO "<filename>"

Renames the specified file or files, provided the file does not have a channel attached to it and that there is no other file that already uses the new filename. Wildcards are allowed in both filenames and even ERASE d<drive>"name" TO "NAME" is allowed. The second string must be a simple filename and not a pathname. (See section 9 [Advanced commands](#)).

SAVE [OVER] d<drive>"<filename>"[LINE <auto-run>]

Saves a BASIC program with an optional auto run line number. If the keyword OVER is specified then the command automatically overwrites duplicate files without giving a prompt unless the file is protected in which case an error is given.

SAVE [OVER] d<drive>"<filename>"DATA <var>[\$()]

Saves either a string or numeric array. It is possible to save a simple non-array string using this command, but the final brackets must still be included. It is not possible to save a string or array splice.

SAVE [OVER]d<drive>"filename>"CODE <addr>,<length>[,<auto-run>]

Saves a code block with an optional auto run address

SAVE [OVER] d<drive>"<filename>"SCREEN\$

Saves a screen image. Screen files are just a special type of code file and can be loaded as code files.

SAVE [OVER] d<drive>"<filename>"X,<addr>

Saves a single sector length execute file which will be executed within the DOS RAM when it is loaded back.

LOAD d<drive>"<filename>"

Loads a BASIC program.

LOAD d<drive>"<filename>"DATA <var>[\$()]

Loads either a simple string or a dimensioned array.

LOAD d<drive>"<filename>"CODE [<addr>[,<length>]]

Loads a code or screen file. The address takes precedence over the value the file was saved with, but the length must be equal or greater than the saved value or an error is given.

UNI-DOS MANUAL

LOAD d<drive>"<filename>"SCREENS

Loads a screen file.

LOAD d<drive>"<filename>"[;]X

Loads an execute file. The optional separator is to allow the form LOAD d<drive>;a\$,X to pass syntax.

LOAD d<drive>"<filename>"[;]S

LOAD d<drive>"<filename>"[;]K

Loads snapshot files. S loads a 48K snapshot and K a 128K snapshot.

LOAD p<file-number>

LOAD p"<filename>"

Loads any type of file from the current disk. The file may be specified by either the file's program number which is printed during a full catalogue, or by the file's pathname. Note that arrays will be loaded correctly using these forms.

VERIFY d<drive>"<filename>"

VERIFY d<drive>"<filename>"DATA <var>\${(}

VERIFY d<drive>"<filename>"DATA <var>()

VERIFY d<drive>"<filename>"CODE [<addr>[,<length>]]

VERIFY d<drive>"<filename>"SCREENS

VERIFY p<file-number>

VERIFY p"<filename>"

All of the above verify commands work. If there is no error in the verify the program continues.

MERGE d<drive>"<filename>"

MERGE p<file-number>

MERGE p"<filename>"

Merges a BASIC program with the one already in memory. Duplicate program line numbers are overwritten.

CLS #

Resets the screen colours and clears the screen. You can change the default colours using POKE @13,<colour>.

UNI-DOS MANUAL

Random Access Files

Just like GDOS, UNI-DOS allows you to create OPENTYPE files; however you can now access any part of an existing file directly using the random access capabilities of UNI-DOS. You still have to OPEN# a stream to the file but there are now three ways of opening the file. Because there are now separate bit maps for both drives, you can safely use OPENTYPE files on both drives at once.

OPEN #<channel>;d<drive>"<filename>"OUT

Creates an openout channel. The file is actually created in the disk directory when you close the channel and its size depends on how many bytes are printed to the channel. Openout channels are consecutive and write only.

OPEN #<channel>;d<drive>"<filename>"IN

OPEN #<channel>;d<drive>"<filename>"RND

Creates a random access channel attached to the specified file. Up to 16 channels can be attached to the same file at once. If IN is specified then the channel will be read only, but is still random access.

OPEN #<channel>;d<drive>"<filename>"RND <length>[,<byte>]

Creates a file of the specified length and initializes every byte to the value specified. Then a random channel is opened to that file. This can be considered as creating an empty random file.

Notes: It is important to remember that BASIC, CODE, DATA and SCREFN\$ files all start with a 9 byte header and the actual program data does not start until byte 10. Also the first byte of a 128K snapshot is a byte used for paging and not part of the program. You can switch between reading and writing to a random channel at will, providing the channel is not read only. Two or more channels can be reading exactly the same part of the file at once - if you write to a file using one channel you will be immediately able to read back the change using another. However what is in the channel buffers may not be what is presently on the disc since a sector is only written back to the disc when all the other channels have finished with it or by using the OUT command - bear this in mind if you want to use other DOS commands in conjunction with random channels.

UNI-DOS MANUAL

All channels of any type must be closed or cleared before you remove the disc from the drive and replace it with another. If you try to use a DOS command after swapping discs with a channel open an error will be given, but NEVER use PRINT, INPUT or INKEY\$ when the wrong disc is in the drive as the results will be unpredictable and may corrupt your disk

PRINT #<channel>,"<data>"

Outputs data to any write channel.

INPUT #<channel>,[I,LINE]<var1>[\$];[LINE]<var2>[\$] . . .

INKEY\$ #<channel>

Inputs data from random or read only channels. INPUT reads data up to and including the next return character (CHR\$ 13) in the file. Numerical values as well as strings may be written to and read from the disc channels. Note that when using the random channels, INPUT can actually write data to the channel as well as reading from it if the line contains strings or comma separators

When you reach the end of a random file a null string is registered as an end marker by INKEY\$ and an extra return character by INPUT. Any further attempts to read the end of the file will generate an error message

POINT <channel>,<file addr>

The point command can be performed on any random channel and allows you to move to the specified position in the file at any time. The next read or write will occur at this position and further reading and writing occurs consecutively. e.g. POINT #<channel>,1 moves to the first byte of the file the channel is attached to and PRINT #<channel>,"TEST"; overwrites the first 4 bytes of the file

You can always use the POINT command on a random channel, even if the end of the file was reached or the channel is currently in an error status, but an error is given if you try to move outside the file

CLEAR #

Removes all channels from the channel area without creating any outstanding, openout channels or saving the sector buffers of random channels.

UNI-DOS MANUAL

CLOSE #* [<channel>]

Closes one or all disc channels, actually creating openout files and saving any random sector buffers that need saving. You can also use this command to close any channels connected to streams "S", "P" and "K" (the BASIC streams) and so this DOS command can now completely replace the ZX CLOSE command. No error is given if the channel being closed was not originally open.

UNI-DOS MANUAL

Printer Commands

There are now three ways of producing a screen dump to your printer. The first two are the same as GDOS except that there is no longer any grey scale shading. Also the "all black" problem has been fixed.

SAVE SCREEENS\$ <1 or 2>

This command dumps the screen using one of two sets of preset parameters. These two modes are also the ones used by the snapshot dump facility. If required, the second set of presets can be altered using POKE @s.

SAVE SCREENS\$ #flag[,pass[,margin[,y[,x[,height[,width]]]]]]

This command is a very powerful dump facility that takes up to 7 parameters and allows you to choose from a vast number of different dump styles: 'flag' is the sum of:

128 if you want the dump printed sideways.

64 if you want to include screen colour data rather than just the pixel data.

8 times the vertical magnification (0-7 where 0 is in fact a magnification of 8).

1 times the horizontal magnification (0-7 again).

"pass" is the number of printer passes per line to allow you to produce different degrees of blackness.

"margin" is the dump's margin in character widths which is totally independent of the margin used by LPRINT.

"y"/ "x" define the top left hand corner of the image you wish to print in the same way as PRINT AT y,x;

"height" is the image's height in character squares.

"width" is the image's width in character squares.

You will also find that you can only stop the dump by pressing the break key at the end of each printed line. This is to stop the printer being left in a graphics printing mode.

LPRINT and LLIST are supported and work in the 'same way as under GDOS. You should also see section 14 as there are a number of DOS variables that effect the way the printer functions.

UNI-DOS MANUAL

Advanced Commands

One of the main features of UNI-DOS is the use of directories. Those of you used to the GDOS disk system will be aware of the disk directory which could hold up to 80 files. UNI-DOS has created a new file type which is a directory. Now when you wish to specify a file you have to provide a PATH to the file. In the simplest case this is Just the file name. In this case the file is looked for in the current directory.

The directory created when the disk is formatted is called the root directory. This is the directory you are placed into when you first boot UNI-DOS.

Most of the DOS commands require you to specify a pathname. These strings tell the DOS which directory the command is to operate on. Pathnames may have a number of forms:

"<filename>"

Specifies a file in the current directory. The filename can contain the wildcards "?" and "*".

"<dirname1>/<dirname2>/< == > /<filename>"

Specifies a file in a sub-directory which branches from the current directory. There is no limit to how many dirnames can be included, but dirnames cannot contain wildcards.

If the string starts with a "/" then the DOS first goes back to the root directory. Therefore:

"/<filename>"

Specifies a file in the root directory. The root directory is 80 files long and held on the first 4 tracks of the disc and is created when the disk is formatted.

"/<dirname1>/< == > /<filename>"

Specifies a file in a sub-directory which branches from the root directory.

A dirname of "." specifies the current directory and a dirname of ".." specifies a parent directory. The parent of the root directory is the root directory itself. Therefore:

"<dirname1>/<dirname2>/./<dirname3>/./<filename>"

Specifies a pathname of "<dirname1>/<dirname3>/<filename>".

The CAT, MOVE, IN and OPEN commands allow strings with the filename part excluded. In these cases the pathname refers to the directory itself.

UNI-DOS MANUAL

POKE @<addr>,<value>

Pokes the value to the DOS variable area. It can also be used to poke values into main memory where the address poked will be 8192 higher than the <addr>. It may be used to poke values up to 65535.

CAT [#<channel>;][d]<drive>["<pathname>"][[NOT]

CAT [#<channel>;][d]<drive>["<pathname>"]![[NOT]

This command first prints the disc name if it exists followed by the actual catalogue. If the root directory is being catalogued then the total free disc space available is printed at the end of the listing. If the catalogue is of any other directory, the dirname is printed at the top and the directory's capacity is printed at the bottom. The string part of the command when included can cause the listing of all programs with a specified name (e.g. using wildcards), or if the string ends with a "/", a directory will be catalogued. A "!" after the string will produce a catalogue of filenames only, and if you include "NOT" at the end then any hidden files will also be listed. Both the forms CAT 1 and CAT d1 are valid, and the listing can be directed to a stream by specifying a channel number.

MOVE [OVER] d<drive>["<pathname>"] TO d<drive>["<pathname>"]

MOVE [OVER] d<drive>["<pathname>"] TO "<pathname>"

This command replaces SAVE TO and copies any files (except directories) with non-zero length including SNAPSHOTS, MDRV, OPENTYPES and files longer than 64K. The memory above the BASIC area and below the machine stack is used for transferring the files and so any BASIC resident in memory during this command will be uncorrupted since the routine does not reset when it has completed. Wildcards are allowed in both the optional strings and they may both be pathnames ending in a "/" allowing you to move all the files from one directory to another. If the second drive number is omitted then the routine assumes you are copying files between directories on the same disc and no prompts are given to swap discs. If OVER is included in the command then duplicate files with the same name are automatically overwritten without asking for confirmation unless the file is protected in which case an error is given.

MOVE #<channel> TO #<channel>

MOVE d<drive>"<pathname>" TO #<channel>

MOVE #<channel> TO d<drive>"<pathname>"

These commands allow you to move data between channels. If required, a temporary channel to a file is created using the d<drive>"<pathname>" form, but the channel will be closed and reclaimed as soon as the command finishes, even if an error occurs.

UNI-DOS MANUAL

SAVE @<drive>,<track>,<sector>,<addr>

Saves the data at the specified address directly to a disc sector.

LOAD d<drive>"<pathname>"[LINE <auto-run>]

Loads a BASIC program. If LINE is specified then the auto run takes precedence over the auto run the file was saved with.

LOAD d<drive>"<pathname>"CODE [<addr>,<length>[,<Auto-run >]]

Loads a code or screen file. The address and auto run take precedence over the values the file was saved with, but, the length must be equal or greater than the saved value or an error is given.

LOAD @<drive>,<track>,<sector>,<addr>

Loads a single sector of raw data from the disc to the specified address.

MERGE d<drive>"<pathname>"

MERGE d<drive>"<pathname>"CODE [<addr>,<length>]

MERGE p<file-number>

MERGE p"<pathname>"

Only BASIC programs and code blocks may be merged. Merging a code block has the effect of disabling the auto run address.

OPEN #<channel>;d<drive>"<pathname>"IN

OPEN #<channel>;d<drive>"<pathname>"RND

Creates a random access channel attached to the specified file. Up to 16 channels can be attached to the same file at once. If the pathname ends in a "/", the file is expected to be a directory. Therefore for example

OPEN #<channel>;"/"RND opens a channel to the root directory. If IN is specified then the channel will be read only, but is still random access.

OPEN #<channel>;d<drive>;IN

OPEN #<channel>;d<drive>;RND

Opens a random channel to the entire disc.

Only channels attached to directories or to the entire disc; will let you examine and change the last two bytes of any sector since these are normally only for internal use.

OUT #[<channel>]

Saves all outstanding sector buffers of random channels back to the disc immediately. An error is given if the channel specified is not a random channel.

UNI-DOS MANUAL

IN d<drive>

Selects the current drive. A feature of the 128k Spectrum means that this command is interpreted as a calculator expression if it is typed as a single command on the edit line without a line number. The DOS command can be accessed in this case by either preceding or following the command with a colon.

IN d<drive>"<pathname>/"

Sets the, current drive and directory to those specified. The pathname must end with a "/".

LINE <line number>

Enables error trapping of all errors including 'out of memory' and all the DOS and 128K errors. When an error occurs in BASIC, instead of stopping with an error message, the program will jump to the specified line number and carry on running. The error code can be retrieved using (PEEK @99), where values greater than 127 are DOS errors. If you want to return to the command where the error occurred you may do so with CONTINUE.

The routine uses the last 16 bytes of the main RAM and so these should not be overwritten when the error trapping is on. The trapping is automatically switched off when a program finishes, on a reset, when the break key is pressed, or by using LINE 0.

The break key can also be disabled independently of the error trapping by using PORE @4,0 - this disables the break key at most times but not for example during a scroll? Prompt which can be trapped using the LINE command. Break will be re-enabled by a reset or by PORE @4, 1, but at other times be careful not to get the BASIC stuck into an endless loop.

UNI-DOS MANUAL

FUNCTIONS

A function is the part of a BASIC line that provides a number or a string result to the command. When using DOS functions, the BASIC statement can start with either BASIC or DOS commands.

e.g. LET a=(PEEK @<addr>) or POINT #<channel>,(LEN #<channel>)

All the DOS commands must be enclosed in brackets but the functions can still be included in expressions.

e.g. (PEEK @<addr1>)+(PEEK @<addr2>)*256 or (PEEK @(PEEK@<addr>))

(PEEK @<addr>)

This function peeks the DOS system variables.

(LEN #<channel>)

Returns the length of a random channel. An error is given if the channel is not random access.

(POINT #<channel>)

Returns the current file pointer of a random channel or zero if the end of the file has been reached. An error is given if the channel is not random access.

(IN #<channel>,<length>)

This function reads the specified number of bytes from any type of channel as a string and is an extremely useful way of getting around the limitations of the INPUT command.

(AT d<drive>"<pathname>")

Searches through the specified directory and returns either the file number of the file found, or zero if the file was not found.

UNI-DOS MANUAL

CREATE FILES

LOAD d<drive>"<pathname>"USR
SAVE [OVER] d<drive>"<pathname>"USR <addr>,<length>

These file types and the associated DOS commands for creating and loading them, provide an easy way of adding new commands and functions to the DOS. These extensions are loaded and stored between the channel area and the BASIC program and so will be lost if you type NEW but will leave the main DOS uncorrupted.

When you load a CREATE file, any previously loaded extensions with the same first syntax character are reclaimed so that loading a CREATE file twice has the same effect as loading it once. Apart from this, the effects of loading CREATE files is accumulative. Several CREATE files are already available:

"ext code" contains the following extensions:

FORMAT d<drive>"<discname>"

Names the disc which will then, be printed during any catalogue of this disc. To erase the disc's name, use a null string.

(LINE)

This function is designed to tell the BASIC programmer which disc is currently in use. The numeric result is the sum of:

- 128 if the current disc is write protected
- 2 if the current drive is drive 2.
- 1 if the current drive is drive 1.

The result will be negated if there is no disc in the current drive.

(STR\$ #<channel>)

This function returns as a string result either a null string if the specified channel is closed, or the stream letter that the channel uses. "d" is the openout stream and "D" the random access stream.

"dir_code" contains the following extensions:

SAVE [OVER] d<drive>"<pathname>"CAT <capacity>

Creates a directory with room for the specified number of files. The directories are created as protected files and cannot be erased using the normal DOS commands. Directories are saved on consecutive sectors and so you may not be able to create a directory of the specified length even though there appears to be enough free disc space to do so.

UNI-DOS MANUAL

ERASE d<drive>"<filename>"CAT

Erases a directory. The string must be a simple filename not a pathname, and cannot contain wildcards.

(STEP [d<drive>])

This function returns the pathname of either the last directory accessed, or the current directory on the specified drive.

UNI-DOS MANUAL

Snapshot Functions

The snapshot button is disabled whenever a DOS command is actually being processed. On these occasions, if you press the snapshot button the computer will freeze until you release it again and the snapshot request will be ignored. As a general rule you should not use the snapshot button when the disk is active.

If an error occurs during the saving of a snapshot then the DOS does not immediately return to the main program but instead goes back to the striped border as if you had just pressed the snapshot button. This is to let you know that the snapshot was unsuccessful and allow you to retry it.

The snapshot button can be pressed at any time UNI-DOS is loaded. The result will be a striped border and the Spectrum will freeze waiting for a key to be pressed. The following keys are allowed:

1. Print a small screen dump.
2. Print a large screen dump.
3. Create a SCREEN\$ file of the current screen.
4. Create a 48K snapshot file.
5. Create a 128K snapshot file.
- x. Do nothing, return to program.
- p. Send printer initialization codes.

When you are taking a 128K snapshot the computer will freeze shortly after you have pressed 5. If the main part of the screen is unchanged then press "n" to continue. However, if the main part of the screen picture has changed then press "y". This is to allow the snapshot routine to know which screen was currently in use.

All the files are saved onto the current disk. It is most important that you do not change the disk in the drive if there are any OPENTYPE files being written to that disk as any other disk will almost certainly be corrupted by the snapshot routines.

UNI-DOS MANUAL

Error Codes

0	Nonsense in Uni-Dos
1	O.K. Uni-Dos
2	Break requested
3	Corrupt sector
4	Sector missing
5	Check disc in drive
6	DOS file not found
7	Invalid filename
8	Invalid sector number
9	Invalid device/channel
10	Wrong stream type
11	Verification failed
12	Wrong file type
13	CODE parameter error
14	Directory not found
15	File has zero length
16	Reading a write file
17	Writing a read file
18	POINT outside file
19	Channel out of order
20	Illegal drive number
21	Disc write protected
22	Not enough disc space
23	Directory full
24	File not found
25	End of file
26	Filename already used
27	File still open
28	File in use
29	Channel already open
30	Protected file
31	Unavailable RST 8

The DOS RAM is checked whenever a NEW or a reset occurs. If there has been any corruption then the DOS flag will be reset and a warning message will appear on the screen. The DOS should be reloaded.

UNI-DOS MANUAL

Hook codes

The machine code interface consists of hook codes (command codes on the PLUS D and DISCiPLE). These are accessed in the usual way of RST 8 followed by a single byte containing the number of the required hook code.

27	CONS-IN	: Interface 1 hook codes
28	CONS-OUT	
29	UNAVAILABLE	
30	UNAVAILABLE	
31	PRT-OUT	
32	KBD-TEST	
33	SEL-DRIVE	
34	OP-TEMP-M	
35	CLOSE-M2	
36	ERASE	
37	READ-SEQ	
38	WR-RECD	
39	UNAVAILABLE	
40	UNAVAILABLE	
41	UNAVAILABLE	
42	UNAVAILABLE	
43	OP-TEMP-M	
44	DEL-M-BUF	
45	UNAVAILABLE	
46	UNAVAILABLE	
47	UNAVAILABLE	
48	UNAVAILABLE	
49	UNPAGE SHADOW ROM	
50	CALL SHADOW ROM ROUTINE	
51	HXFER	: GDOS COMMAND CODES
52	OPEN FILE	
53	OUTPUT HEADER	
54	SAVE CHARACTER	
55	SAVE BLOCK	
56	CLOSE FILE	
57	PRINT CHARACTER	
58	SMALL SCREEN DUMP	
59	LOAD FILE	
60	LOAD CHARACTER	
61	LOAD BLOCK	
62	SAVE SECTOR	
63	LOAD SECTOR	
64	UNAVAILABLE	
65	ERASE FILE	

UNI-DOS MANUAL

66 LARGE SCREEN DUMP
67 DISK CATALOGUE
68 LOAD SECTOR TO *HL
69 WRITE SECTOR FROM *HL
70 OPEN AND CLOSE STREAMS
71 PAGEIN

72 LOAD FILE : UNI-DOS HOOK CODES
73 LOAD FILE
74 LOAD FILE
75 SAVE FILE
76 OPEN FILE
77 POINT
78 WRITE BUFFERS TO DISK
79 CLOSE FILE
80 CLEAR CHANNELS
81 RENAME FILE
82 MOVE STREAM
83 MOVE FILE
84 SELECT DISK & DIRECTORY

UNI-DOS MANUAL

Directory Structure

Each file has a 256 byte header stored in the directory. The root directory is 40 sectors in length and can hold 80 file headers. This is always located on the first four tracks of side 1 of the disk. Subdirectories are basically the same except that they can occur anywhere on the disk. A subdirectory is always created in a single area so even though there may appear to be enough room on a disk you may not be able to create a directory.

Unlike all other files the last two bytes of a sector in a subdirectory do not point to the next sector. This is not necessary as the next sector is always the next on the disk. The last two bytes of the last sector of a subdirectory are set to #FF to indicate the end of the directory.

In the root directory the first file header may also contain the disk name which is held at the end of the file header. The meanings of each of the bytes in a file header are as follows.

byte	use
0	file type:
	0 ERASED
	1 BASIC PROGRAM
	2 NUMERIC ARRAY
	3 STRING ARRAY
	4 CODE FILE
	5 48K SNAPSHOT
	6 MDRV
	7 SCREEN\$
	8 SPECIAL
	9 128K SNAPSHOT
	10 OPENTYPE
	11 EXECUTE
	12 SUBDIRECTORY
	13 CREATE
	Add 128 for hidden files and
	add 64 for protected files
1-10	file name
11-12	number of sectors in file 0 - 65535
13	track number of first sector of file
14	sector number of first sector of file
15-209	sector bit map. One bit per sector
210-255	file specific information has different meanings depending on file type

UNI-DOS Variables

UNI-DOS MANUAL

Adr	Len	Default	Function
0	1	0	Flash border during disk operations 0 = do not flash
1	1	208	Drive 1 Capacity. Number of tracks + 128 if double sided
2	1	208	Drive 2 Capacity. Number of tracks + 128 if double sided
3	1	6	Drive stepping rate
4	1	1	Enable break key if = 1
5	1	80	Printer Line length
6	1	0	Printer control flag. 1 = transparent or binary mode 0 = normal mode
7	1	8	n/72 `s line spacing sent to printer with initialization codes and before every CR
8	1	0	Number of line feeds after CR
9	1	0	Left margin number of spaces sent before first printed character on a line.
10	1	#80	Printer flag
11	1	1	Centronics port flag When set to 1 enables the Centronics port and signals that the printer initialization codes need to be sent. see below.
12	1	0	printer column number
13	1	#38	CLS# screen colour
Adr	Len	Default	Function

UNI-DOS MANUAL

14	2	#0000	Extended syntax address
16	2	#0000	Interrupt address
18	8	#1B408080 #80808080	Printer initialize codes Sent as soon as a printer is seen to come on line. They are resent after a NEW or reset or by pressing the snapshot button followed by "P"
26	8	#1B508080 #80808080	Set character pitch
34	8	#1B418080 #80808080	Set n/72 line space
42	8	#1B2A0480 #80808080	Set UDG bit graphics density
50	8	#80808080 #80808080	2nd initialize codes
58	8	#819C8080 #80808080	Codes for
66	8	#81238080 #80808080	Codes for #
74	8	#83286329 #80808080	Codes for (c)
82	7	#92010200 #001820	SAVE SCREEN\$ 2 parameters
89	8	#1B2A0580 #80808080	set dump graphics
97	2	#681C	addr of extra error messages
99	1	0	error code
100	2	#AA34	addr of LPRINT routine
102	2	#0000	DOS error return address

UNI-DOS MANUAL

Adr	Len	Default	Function
104	20	#00536E61 #70202020 #20202003 #00000000 #0000FFFF	Snapshot workspace
124	2	#0000	Called on reset
126	2	#A421	Called on boot up
7667	1	1	set to 0 to reset DOS

Scanned, Typed, OCR-ed, and PDF by
Steve Parry-Thomas 24th August 2004.

This PDF was created to preserve
this

Newsletter for the future.

For all ZX Spectrum, Beta Basic
And www.worldofspectrum.org users
(PDF for Michael & Joshua)