PTS
DOS 2000 Pro

User Manual

Paragon Technology GmbH, System Programmierung

Published by:
Paragon Technology GmbH
System Programming
Pearl-Str. 1
D-79426 Buggingen
Germany

1. CONFIGURATION FILE CONFIG.PTS

1.1 GENERAL SYNTAX RULES

Any characters following '*' or ';' are regarded as comments and are ignored by the system. When processing the configuration file, the system does not distinguish between small and capital letters. All numeric values inside the configuration file may be written either in decimal or in hex modes. A hex numeric constant should always begin with a number and end with 'H' or 'h'. The feature is useful when defining memory segments addresses and determining key scan-codes in some commands.

At the beginning of any CONFIG.xxx file line the following prefixes or combinations of those may be used: CPU88, CPU86, CPU286, CPU386, CPU486, PC, AT, XT, PS1, PS2. These prefixes mean "for the xxx processor" and "for the xxx type of computer", respectively. It is possible to use '+' at the end of the CPUxxx prefixes, meaning "for the xxx processor and higher".

CPU88	- CPU486 - processors i88 - i486 and compatible
PC	- original PC
XT	- IBM PC/XT compatible
AT	- IBM PC/AT compatible
PS1	- IBM PS/1
PS2	- IBM PS/2

For example,

```
CPU386+ PS2 echo "Running on PS/2 i386/i486/P5/.. computer"
```

system, when processing the above line, will echo:

Running on PS/2 i386/i486/P5/.. computer

Some commands have optional parameters. A special symbol is used as a rule to denote such a parameter. By default, the symbol is '/'. It is useful to keep in mind that the key symbol may be defined (changed) with the help of SWITCHAR (sys) command. Once done so, the new symbol should be used instead of '/', for example:

```
SWITCHAR = -
DRIVPARM = -d:0-c
```

As the system makes several passes when processing the configuration file, the SWITCHAR command should be located at the very beginning of the file.

The following conventions are used for syntax description:

<...> - obligatory parameter;
[...] - non-obligatory (optional) parameter;
| - "or" - shows the alternatives.

For example,

```
DRIVATTR </D:dd> <[/T][/C][/S]>
```

implies that the argument of the command is a line always containing '/D:dd' and at least one of '/T' ,'/C', '/S'.

```
BREAK = <ON|OFF>
```

BREAK command should have either 'ON' or 'OFF' as an argument. Also, 'YES' and 'NO' respectively can be used in the above case.

1.2 CONFIGURATION FILE SECTIONS

The configuration file consists of commands, which may form several sections. The system itself processes only the user-chosen or predefined sections. Each section begins with a definition of its name and ends either with a next section name definition or with an end of file. Section name should be put in the '[' ']' square brackets. Comments within a section name should begin with '@'. For example,

```
[ SectionName @ This is a comment ]
```

Section name is defined by its first 16 characters. All the others, including any spaces, are ignored.

A section name should be unique, except for the special names like: COMMENT, PTSDOS, and COMMON. If more than one section has the same name, the system treats the resulting set, as a whole (does not apply to the TRACE).

The following section names have been predefined:

COMMENT - a section with this name is ignored by the system.
PTSDOS, COMMON - commands within such a section are considered global and are always executed by the system.
BOOTMGR - BOOT MANAGER parameters description section. The configuration file consists of commands, which may form several sections. The system itself processes only the user-chosen or predefined sections. Each section begins with a definition of its name and ends either with a next section name definition or with an end of file. Section name should be put in the '[' ']' square brackets. Comments within a section nam should begin with '@'. For example,

1.3 CONFIGURATION FILE PROCESSING

When booting, the system first loads the boot manager for processing sections and some commands like VIDEOMODE etc. The boot manager tries to recognize all the sections in a configuration file and, if it finds selectable sections, displays the list of these sections with a prompt to select one of them for booting. This boot menu can contain up to ten selectable sections to be user-selected. The sections COMMENT, PTSDOS, COMMON, BOOTMGR and sections with names beginning with '$' can not be selected from a boot manager prompt, but they can be enforced by FORCE command. The boot menu doesn't appear, if there are none of the selectable sections. Mode of operation of the Boot Manager can be defined with the following commands in the BOOTMGR section:

AUTOCHECK = <ON>|<OFF> ;

By default, AutoCheck = Off. When "ON", boot manager will use the user's input prerecorded into a file or CMOS memory, instead of a dialogue.

STORAGE = <FILE>|<CMOS> ;

By default, Storage = FILE. At the first call or on pressing <Esc> during booting, Boot manager will display a prompt for user selection. If a user presses '1','2',...,'0', the choice will be realized but not saved for further bootings. If any of the above keys is pressed in combination with 'Alt', then the Boot Manager will ask the permission to process each line of the configuration file (See TRACE command).

If the user inputs 'F1' - 'F10', the boot manager will save the number either in a file or in CMOS, depending on STORAGE.

FILENAME = [drive:][path]filename

Specifies the filename for storing users input. By default,
FILENAME = AUTOBOOT.SYS

CMOSADDR = <CMOS register address in decimal format>
Specifies CMOS register for storing users input. By default,
CMOSADDR = 52

FORCE = <section name>[,<section name>[, ...]]
Specifies a section or a list of sections, equivalent to those named
"COMMON".

IGNORE = <section name>[,<section name>[, ...]]
Specifies a section or a list of sections which are to be ignored by
the system.

```
COLOR  = <cc | < BLACK..WHITE [,BLACK..WHITE]>>
            [on]
            [ bb | BLACK..WHITE [,BLACK..WHITE] ]
```

colors -
BLACK, BLUE,GREEN,CYAN, RED,MAGENTA,BROWN,LIGHTGRAY
DARKGRAY, LIGHTBLUE, LIGHTGREEN, LIGHTCYAN, LIGHTRED
LIGHTMAGENTA, YELLOW, WHITE
Where, cc - foreground color, bb - background color.
Specifies boot manager prompt color.

DEFAULT = <section name>

TIMEOUT = <period in seconds>
By default, TIMEOUT = 5 (sec). DEFAULT specifies a user selection 'by default'. TIMEOUT gives a time delay before the section is selected according to the DEFAULT command.

Surely enough, the system will immediately make the right choice if the AUTOCHECK=YES and correct user information does exist. However, if such information does not exist or for any reasons (for instance, due to any modification in CPNFIG.xxx file) is unacceptable, then a standard prompt shall be given by the system, where the default selection will be marked with '+'. If no measures are taken after the TIMEOUT period or if you press [Enter], the system will load the marked section, but this choice will not be saved for further booting.

By entering the global trace mode (processing each line of the configuration file after pressing Alt-1..9 in the BootMgr prompt) it is possible, by pressing Alt-Y or Alt-N to answer all the prompts related to the global trace mode (no effect on the TRACE command itself).

If in the [BOOTMGR] section it is specified:

```
storage=FILE
autocheck=YES
```

then the date of CONFIG.xxx creation will be stored together with information on the selected section. So, after any modification of the file, the system will consider the section selection information as incorrect and will assume that it should be rewritten.

1.4 EXAMPLES OF CONFIGURATION FILE

One may, in absence of compatibility with MS-Windows and other programs, and by using non-documented DOS features, free more memory for applications by modifying system configuration. For example, let us consider the following configuration file:

```
WINDOWS = NO                  ; Enabling usage of non-standard system data
                              ; and code allocation strategy
COMAREA = STANDARD            ; Usage of memory area starting from 50:00
DOS = HIGH                    ; HMA enabling
DEVICE = HIMEM286.SYS         ; XMS driver loading
FILES = 40                    ; Maximum number of open files
```

```
BUFFERS = 20                    ; Number of input/output buffers
```

On starting-up System and running MEM, the following message will be displayed on the screen:

```
655,360 bytes total conventional memory
650,368 bytes free conventional memory
650,304 largest executable program size

3,407,872 bytes total contiguous extended memory
        0 bytes available contiguous extended memory
3,316,736 bytes total XMS memory available
3,316,736 largest XMS memory block size

PTS-DOS resident in High Memory Area
```

In this case you have 650368 bytes of free memory available for applications.

Extended command set and improved syntax of the configuration file allow a user to fine-tune the System according to own needs. For example, if several system configurations are required, then the following configuration file should be written:

```
[Common]                        ; The common section is always executed
Device = himem386.Sys           ; Loading XMS driver

[C1@Maximum Memory]             ; Section C1 - maximum free memory configuration
ComArea = Standard              ; Usage of memory area starting from 50:00
Windows = NO                    ; Enabling usage of non-standard system data and
                                ; code allocation
Force = $DosStd                 ; Linking content of the $DosStd section to
                                ; current one

[C2@Compatibility Mode]         ; Section C2 - compatibility mode configuration
ComArea = High                  ; Disable usage of memory area from 50:00 to 70:00
Windows = Yes                   ; Standard system data and code allocation
                                ; strategy
Force = $DosStd                 ; Linking content of the $DosStd section to
                                ; current one

[$DosStd]                       ; Section $DosStd - internal, used for linking to
                                ; other sections
Files           = 40
Buffers         = 20
DosDir          = C:\PTSDOS
Shell           = C:\PTSDOS\command.com C:\PTSDOS /p C:\AUTOPTS.BAT
DOS             = High Umb
Clock           = Real
IDTChk          = Off
Abort           = On
ShiftState      = Off
SoftReboot      = On
LastDrive       = k
Country         = RU,866, C:\PTSDOS\COUNTRY.SYS
Keyboard        = RU LoadOnCall ActBorder 56 DefBorder 0 HotKey 02 54
Display         = RU Text Preload High

[BootMgr]                   ; Special section BootMgr. Description and system
                            ; of applicable commands may be found in section
                            ; "CONFIGURATION FILE PROCESSING"

AutoCheck       = Yes
```

```
Color            = White on Black
Storage          = File
FileName         = AutoBoot.Sys

[C3@OldDOS]
SysBoot  = C:\oldboot.sys            ; Indicates the boot sectors address of the
                                     ; system installed before PTS-DOS on this
                                     ; computer

;<< EOF >>
```

Upon first System start-up (and subsequent start-ups, if the [Esc] key is pressed during the first loading) the following prompt shall be displayed:

```
Select section number to be active
---------------------------------------------
[F]1. C1          Maximum Memory
[F]2. C2          Compatibility Mode
[F]3. C3          ÄldDOS
---------------------------------------------
```

By pressing a corresponding numeric or function key, the System configuration shall be changed accordingly.

If a numeric key is pressed in combination with the [Alt] key, the System will step-by-step process the commands of a corresponding section. Therefore, pressing [Alt-1] in the previous example, will cause the following System prompt:

(Y/N)? >Device = himem386.Sys

Pressing <Y> will install the himem386.sys driver, <N> - will cancel the operation. TRACE command may be used for toggling the trace mode in certain lines of configuration file.

If HMA is available, the System partially allocates in it depending on the mode which is set up by command as WINDOWS or BUFFERS. Command SYSGEN may be used for direct specifying the allocation area of a system part (code, operative data, I/O buffers, file tables etc.).

1.5 SYSTEM CONFIGURATION COMMANDS:

The commands COMMENT, MULTITRACK, REM, STACKS, SWITCHES together with any their parameters are ignored by the System and have been left for the compatibility with other DOS versions.

1.5.1 ABORT

Purpose: Sets the emergency program exit option activated by the
[SysReq] key.

Syntax: ABORT = <ON|OFF>

By default: ABORT=ON

Note:

In certain cases when a program halts, it is sufficient to take measures on exiting the current process in order to resume normal system operation. If the ABORT is ON, you may do the above by pressing [SysReq]. Nevertheless, such exiting is not always correct as the program could have allocated some part of extended and/or expanded memory, and DOS can not release it automatically, or temporary files may have been created which can be deleted just on normal completion of the program. Besides that, the aborted program could have made changes to the interrupt vectors table, making further operation impossible.

1.5.2 BREAK (sys)

Purpose: Sets a check-up for pressing [Ctrl-Break].

Syntax: BREAK = <ON|OFF>

By default: BREAK=OFF

Parameters:

ON - sets BREAK = ON. The system checks if the [Ctrl-Break] keys' being pressed before executing any DOS command and breaks the execution, if those keys are pressed.

OFF - sets BREAK = OFF. In this case the system controls [Ctrl-Break]
keys' pressing only during:
- standard input operations,
- standard output operations,
- standard printing operations,
- operations with standard auxiliary devices (adapters).

Running the BREAK command without parameters leads to the BREAK current condition being displayed.

1.5.3 BUFFERS

Purpose: Sets the number of buffers allocated in RAM to buffer disk operations and I/O operations with other types of block devices.

Syntax: BUFFERS = <XX> [,YY] [/X] [/L|/H]

XX - number of buffers, from 8 to 99;
YY - not used by PTS-DOS, left for compatibility with other
DOS versions;
/X - not used by PTS-DOS, left for compatibility with other
DOS versions;
/L - instructs the System to allocate buffers only in the
lower (up to 640+) memory;
/H - enables usage of HMA for allocating buffers.

By default: BUFFERS = 8 /H

Note:

A disk buffer is a block of memory which is used for buffering I/O operations with block-type devices, that can transfer data only in blocks of fixed size.

For applications such as word processors, a number between 10 and 20 is quite sufficient for the system. If you are going to use the additional caching program, then the default setting will provide normal performance. Increasing the number of buffers by one, takes up additional 530 bytes of memory if the DOS is not loaded into the high memory, but if it is, you should look through the SYSGEN command description for explanation of the strategy of allocating DOS in memory.

You should try various settings of BUFFERS to find out an optimal I/O speed/available memory ratio.

1.5.4 CLOCK

Purpose: Specifies the type of system clock, installed in the computer.

Syntax: CLOCK = <REAL|STANDARD>

By default: CLOCK = STANDARD

Note:

In the default setting, DOS derives the current time and date using a timer available with any personal computer. The initial state is then set either by the real-time clock or equal to 00:00.00 1-1-1980. Setting CLOCK = REAL makes the system to load a code, that directly works with the real-time clock on time and date inquiries. In this mode less system memory is used and the date range extends back to the year 1553. It is recommended to set CLOCK = REAL in all cases when no compatibility with old DOS versions is required with the symbol device CLOCK$.

1.5.5.COLOR COMMAND

Purpose: Sets the DOS functions output color

Syntax: COLOR=cc|<BLACK..WHITE>[ON][bb|BLACK..WHITE]

Colors-

BLACK, BLUE, GREEN, CYAN, RED, MAGENTA, BROWN, LIGHTGRAY, DARKCRAY, LIGHTBLUE, LIGHTGREEN, LIGHTRED, LIGHTMAGENTA, YELLOW, WHITE.

Where, cc-foreground color
bb-background color

By default: COLOR=LIGHTGRAY on BLACK

Note:

The system will use COLOR when writing to the screen in text and graphic video modes. A color may be changed by using a corresponding ANSI output to console on setting SCREEN=ANSI or loading an ANSI driver.

1.5.6CDDNAME

Purpose: Defines the name of CDROM hardware device driver unit.

Syntax: CDDNAME=name
Where, name is a name of device defined with hardware

Note:

In order to use CDROM disk You usually run CDROM hardware driver and define name of unit. For example:

```
Defice=c:\sjcd.sys /D:CD1
```

CDDNAME command must define the same: CD1

1.5.7 CDDBUFFERS

Purpose: Sets the number of buffers used for CDROM device input/output and caching.

Syntax: CDDBUFFERS=<xx>[,yy]

xx - number of buffers for CDROM arranged in conventional memory, the default value is xx=2

yy - number of buffers for CDROM arranged in extended memory (XMS)

By default: CDDBUFFERS=2

Note:

A CDROM buffer is a block of memory which is used for buffering I/O operations with CDROM-type devices, that can transfer data only in blocks of fixed size.

You should try various settings of CDDBUFFERS to find out an optimal I/O speed/available memory ratio.

1.5.8 COMAREA

Purpose: Sets the DOS communication area location.

Syntax: COMAREA = <STANDARD|HIGH>

By default: COMAREA = HIGH

Note:

COMAREA = STANDARD instructs the system to use the memory area between 0050:0000 and 0070:000 for its internal purposes. As the PTS-DOS uses this memory in not completely the same way as with other DOS versions, some programs will not operate if COMAREA=STANDARD. When WINDOWS=ON, as it is set by default, the COMAREA=STANDARD is not executed and it is only permissible in case if.

```
WINDOWS=OFF
COMAREA=STANDARD
```

Allocating communication area higher than standard requires 530 bytes additionally, which are always in the lower memory area.

1.5.9 COUNTRY

Purpose: Allows PTS-DOS to use different formats of time, date, national currency symbols, thousands and decimal separators, conventions on conversion of lower case letters to upper case depending on the country specified.

Syntax: COUNTRY = <XXX>[,[YYY][,[drive:][path]filename]]
Where, XXX - country code or abbreviation
YYY - new code page
filename - file containing the required information.

By default:

If there is no COUNTRY command in the configuration file, the system uses the code page for the United States.

If no filename is specified as an argument, DOS assumes it to be COUNTRY.SYS and looks for it either in the root directory or the directory specified by DOSDIR.

Note:

This configuration command identifies to PTS-DOS which country character set is to be used. The following table lists all countries supported by PTS-DOS. The table also lists the code pages that may be used with a specific country code.

Country	Abbreviation	Code	Code pages
USA	US,USA	1	437, 850
Canada	CF	2	863, 850
Latin America	LA	3	437, 850
Russia	RU	7	866

Netherlands	NL	31	437, 850
Belgium	BE	32	437, 850
France	FR	33	437, 850
Spain	SP	34	437, 850
Italy	IT	39	437, 850
Switzerland	SG,SF	41	437, 850
United Kingdom	UK	44	437, 850
Denmark	DK	45	865, 850
Sweden	SV	46	437, 850
Norway	NO	47	865, 850
Germany	GR	49	437, 850
English (International)	-	61	437, 850
Portugal	PO	351	860, 850
Finland	SU	358	437, 850
Arabic	-	785	437
Israel	-	972	437
Japan	-	081	932,850,437
Korea	-	082	934,850,437
China	-	86	936,850,437
Taiwan	-	88	938,850,437

1.5.10 CPU

Purpose: Slightly improves the performance of a PC.

Syntax: CPU = <FAST|STANDARD>

By default: CPU = STANDARD

Note:
This command increases the memory refresh cycle, resulting in a little higher performance of a PC.

1.5.11 DEVICE

Purpose: Installs specified driver into the system list.

Syntax: DEVICE = [drive:][path] filename [arguments]
Where, arguments are all parameters of a driver.

Note:
If you purchase a new device, like a mouse or a scanner, you will receive software device driver with that device which services it. To install this driver, the DEVICE command should be used where the driver name and location should be specified.

1.5.12 DEVICEHIGH

As the DEVICE command above, but the System will try to load the specified driver to the UMB. If UMB is absent at the time of the command execution, or there is no enough memory of this type, the System will assume the command is DEVICE.

1.5.13 DISKETTE

Purpose: Sets permissible diskette types.

Syntax: DISKETTE = <EXTSIGN|DOS3>

By default: DISKETTE = DOS3

Note:

System support of a floppy disk implies that when changing a diskette, all the internal data structures associated with the specified device should be renewed, and a check on all the information possibly not recorded onto a previous diskette (for example, files which have been modified were not closed) should be carried out. Diskette format programs designed for older DOS versions didn't create a unique record in the boot sector, that helps detect a change of diskettes. Without such records the system compares diskettes by their formats. It is of course a non-reliable comparison and there is always a chance of data loss when mishandling diskettes. The DISKETTE = EXTSIGN instructs the system to work only with diskettes having an unique label.

Warning! Many format programs don't support the extended diskette boot sector format. Using DISKETTE = EXTSIGN is recommended only with disk drives without change line.

1.5.14 DISPLAY

Purpose: Sets the display font.

```
Syntax: DISPLAY = <xx|ID> [GRAPHICS|TEXT]
                          [FONTS<yy|STANDARD>]
                          [PAGES zz] [HIGH|LOW]
                          [FILE [drive:][path]filename]
                          [PRELOAD|LOADONCALL]
                          [RESIDENT|NONRESIDENT]
                          [USEHARDWARE|USEBIOS]
        Where, filename - font description file
                          zz - number of pages supported
                          xx - page code
                          ID - country identifier
```

Note: By default, the following optional parameters are assumed:

```
        GRAPHICS = FONTS STANDARD, PAGES 1, LOW \
         \ FILE Display.Cpi, LOADONCALL, RESIDENT
```

This command allows to load an alternative display font, if the videoadaptor supports this feature. As a rule, these are: EGA, VGA (SVGA), MGA, XGA and analogous. If the RESIDENT parameter is present, the system will load a code that will save the specified font during change of the videomode and make it possible to interactively switch to it by pressing Ctrl-Shift-F1...F9, where F1-F9 define active page (F1 is the BIOS supported font). For more details, refer to the external DISP command description.

PRELOAD, LOADONCALL define the initial state: in the first case a new font is loaded during processing the DISPLAY command from the first page, in the second case switching between fonts is possible either in the interactive mode, or with the help of CHCP and DISP commands. FILE sets the path and filename for the file containing binary font description. By default, the filename is Display.Cpi. HIGH and LOW set the type of memory used for allocation of font images, the former sets the usage of extended memory, while the latter concerns the lower area memory.

PAGES sets the number of various code page combinations, supported by the system. PAGES1, the default value, is sufficient for a twin-language interface. It is recommended to use higher values in combination with HIGH, which will save on system memory.

FONTS sets the number of types for each font. Normally, different matrix fonts are required to support all the available videomodes, e.g. 8x8, 8x14, 8x16. Their number is defined by the type of the videoadapter. But, if a soft font is to be used in one of the possible modes, FONTS should be specified to save the system memory.

Supported	Adapters		FONTS	
Matrices		1	2	3
8x8, 8x14, 8x 16	VGA	8x16	8x8, 8x16	8x8,8x14,8x16
8x8, 8x14	TGA	8x14	8x8, 8x14	8x8, 8x14
8x8	other	8x8	8x8	8x8

Soft font matrices

GRAPHICS, TEXT instruct the system to support a national font either both in graphic and text modes or only in text mode. When using GRAPHICS, more system memory is required, as a rule.

When processing the DISPLAY command, the System can only load fonts for a single code page (country). On specifying with PAGES a value larger than 1, loading other code pages becomes possible, but only using the DISP - external DOS command. A two-letter country identifier code may be used instead of the code page. In this case, the system will choose the corresponding page code automatically. See COUNTRY for more information on this feature.

Available code pages:

437	United States of America
850	Multilanguage interface
860	Portugal
863	Canadian-French interface
865	Norway
866	Russia

USEHARDWARE and USEBIOS define the way the videoadapter is programmed. USEHARDWARE instructs the system to program VGA/EGA adapter registers directly. With this mode there will be no display flickering when switching between fonts interactively with keys Ctrl-Shift-Fx. USEBIOS instructs the system to use the BIOS video for display fonts loading. High level of compatibility with existing hardware and software is achieved here.

1.5.15 DOS

Purpose: Makes DOS to load itself into the HMA on PCs with 286+ processors, also instructs DOS to use or not to use the upper memory area (UMB).

Syntax: DOS = < [HIGH|LOW] [UMB|NOUMB] >

By default: DOS = LOW,NOUMB

Note:
Use this command to increase the size of the free lower memory.

Options HIGH and LOW define whether DOS will use the 64K HMA or not. This option requires installation of a XMS server. The PTS-DOS package includes two XMS support drivers - Himem286.Sys and Himem386.Sys for 286 and 386+ processors, respectively.

Options UMB and NOUMB define whether DOS will use the UMB blocks or not. When DOS=UMB, a TSR or a driver may be installed into the UMB using the DEVICEHIGH=, INSTALLHIGH= commands or with the LOADHIGH DOS command. The option requires installed UMB server, e.g. EMM386, 386MAX, QEMM386.

Options may be used combined in a single line. Line:

```
DOS=HIGH,UMB
```

is equivalent to the lines:

```
DOS=HIGH
DOS=UMB
```

1.5.16 DOSDIR

Purpose: Sets an additional search path for auxiliary system files.

Syntax: DOSDIR = [drive:]path
where, path is a directory name

By default: DOSDIR =\

Note:
Some configuration file commands require corresponding special files, like display.cpi, keyboard.sys, country.sys. By default, the system will first search the root directory and in case of failure, will continue searching in the directory specified by the DOSDIR command.

1.5.17 DRIVPARM

Purpose: Amends or adds parameters set by the System for a physical disk
device at system boot.

Syntax: </D:dd>[/C][/F:ff][/H:hh][/N][/S:ss][/T:tt]

Parameters:

/D:dd - dd is a physical device number, ranging from 0 to 255.

/C - indicates that the drive with change line is used. With such a drive which detects whether a diskette has been removed, the System assumes that a diskette has not been changed and doesn't check for this state, if there is no signal from hardware. See the documentation for your disk drive to find out whether it has a change line.

/F:ff - ff specifies the block device type. By default, ff=2, possible values given below:
0 - 160/180/320/360K
1 - 1,2M
2 - 720K (3.5" disk)
5 - Hard disk
6 - Tape drive
7 - 1.44M (3.5" disk)
9 - 2.88M (3.5" disk)

/H:hh - specifies the maximum number of heads, the default is: hh=2.

/N - specifies that a disk is nonremovable.

/S:ss - sets ss as the number of sectors per track for a given device.

/T:tt - sets the number of cylinders on a disk.

1.5.18 DRIVATTR

Purpose: Sets additional physical parameters of a disk device.

Syntax: DRIVATTR =</D:dd> <[/T][/C][/S:xx]>

Parameters:

/D:dd - defines the device logical number

/T - specifies that the device controller can process input/output inquiries that require addressing several cylinders. Applicable only to IDE controllers, as a rule. WARNING! Not recommended to use when running disk cache programs, e.g. SMARTDRV.EXE and when loading MS-Windows 3.1 in enhanced mode.

/C - Disables system interference over diskette access control. With this key PTS-DOS will not support non-standard diskette formats.

/S:xx - sets the number of rotations per second for a diskette. At boot, the system sets the following values:

for a 360K drive - 5 r.p.s.
for a 1.2M drive - 6 r.p.s.
for 3,5" drives 720K/1.44M/2.88M - 5 r.p.s.
The parameter is used for calculating GPL - the filling gap between sectors on a diskette track.

1.5.19 ECHO (sys)

Purpose: Displays a message on the screen during the first pass of a configuration file.

Syntax: ECHO = "message"

Where, message is any text you like, embraced with the quote marks <">.

Note:

As the command is executed during first pass of the configuration file, there is no synchronization between the ECHO command and other commands, like DEVICE=.

1.5.20 ERREXE

Purpose: Sets the level of checking code to be loaded for its validity.

Syntax: ERREXE = <ON|OFF>

By default: ERREXE = OFF

Note:

When ERREXE = ON, the system will not load modules if:

1. The size of the part to be loaded exceeds the total file size.
2. In case of fixup overflow.

Some programs containing these errors otherwise function normally, therefore an user is left free to decide the system reaction to such occurrences.

1.5.21 EXECA20OFF

Purpose: Sets the lines state "A20" at the start of executing a freshly loaded program.

Syntax: EXECA20OFF = <YES|NO>

By default: EXECA20OFF = YES

Note:

Everything described below is applicable only when the system is using the HMA (See the DOS=HIGH command).

By default, the system tries to make the HMA area invisible before transferring the control to the loaded program's first instruction, imitating the address space of 88/86 processor PCs. It is worth mentioning that some programs designed with old versions of the Microsoft LINK, work incorrectly if loaded lower than the first 64K of memory. In such cases the error is unnoticed if the higher 64K of the first megabyte are located below (XT type). The system automatically fixes the code of such programs (See the LOADFIX command), but not always, like when, for instance, the "bad" program is packed with an utility as PKLITE etc. Therefore, if there are no old programs involved or if the "A20" line is being automatically controlled by the computer, it is better to set EXECA20OFF = NO, which will slightly accelerate loading of smaller programs.

1.5.22 FASTOPEN

Purpose: Sets the number of buffers for organizing a fast file search cache.

Syntax: FASTOPEN = xx
Where, xx - ranges from 4 to 99 and defines the number of buffers.

By default: FASTOPEN = 4

With a complex directory structure, the system has to look through a significant number of directories when searching for a file deeply located and if path to this file is not found in any of the buffers, the number of which is set by the FASTOPEN. Each buffer takes up about 80 bytes of system memory. Maximum values for the number of buffers should be used when working with file-extensive applications such as DBMS.

1.5.23 FCBS

Purpose: Sets the emulation level of FCB system calls, usual for the old system versions.

Syntax: FCBS = <xx[,yy]|ON|OFF>

By default: FCBS = ON

Note:
The yy parameter has been left for the sake of compatibility with previous DOS versions. xx values other than 0 correspond to the ON state, xx=0 - to the OFF state. FCBS=OFF saves about 1.5K of system memory. Only the Find First/Find Next/Create system calls are then available trough FCB. Other calls that are supported by the standard system interface, are possible only when FCBS = ON.

There are programs that employ undocumented FCB fields. In these cases, as a rule, messages of errors in DOS file system occur. To cure the problem, the CHKDSK external DOS command or the system independent programs as Symantec's Norton Disk Doctor, should be used.

1.5.24 FILES

Purpose: Sets the number of file open operations accessible via system calls.

Syntax: FILES =<xx>[,yy][HIGH|LOW][STANDARD|CHANGEABLE|FIXED]

Parameters:

xx	- the maximum number of concurrently open files, the default value is xx=20.
LOW	- use the standard scheme for allocating files. In this case the yy parameter is not used while xx ranges from 8 to 255.
HIGH	- instructs the system to use the extended memory for allocating file tables. yy sets the file tables number of a cache, allocated in the lower memory, while xx ranges from 8 to 60000.
STANDARD or CHANGEABLE	- specifies whether the code servicing dynamic changes of the xx value when running an application program, is to be loaded.
FIXED	- do not load the above described code.

By default: FILES = 20,8 LOW STANDARD

Note:
When allocating files in the extended memory, the value of FILES may exceed 255 with additional system call required for a program to open more than 255 files. These calls in fact set the value that is added to the argument specified with a system call. This feature is extremely useful in multitask

environments that require a large number of files concurrently opened by different tasks. Still, very large values of the parameter slow down the system performance when opening a file, since the system has to look up all the file tables in order to define the recurrence of such operation.

1.5.25 FORCE

Purpose: Makes the specified section(s) available for system processing.

Syntax: FORCE = section[,section[,...]]

Note:
A section may be defined for system processing either by user selection when running the boot manager or with the help of this command. For example,

```
[ PtsDos ]
Force = $Drv
...
[ $Drv ]
device = himem386.Sys
```

So, when processing the previously defined PTSDOS section, the system will find the FORCE = $Drv command and process the $Drv section resulting in the himem386.sys driver being installed.

As the FORCE and IGNORE commands involve only single-pass parsing, only the names of subsequent sections should be used as their parameters.

1.5.26 HMAREA

Purpose: Sets an initial HMA segment address.

Syntax: HMAREA = <xxxx|STANDARD>
Where, xxxx - segment address

By default: HMAREA = STANDARD

Note:
The STANDARD constant is equivalent to 0FFFFh. By default, the HMA is located between FFFF:0010 and FFFF:FFFF and an XMS driver is required for its support. But, if 64K of memory are available between A000 and FFFF ,it may be used as HMA with no additional driver.

It is not recommended to use this command if the HMA size is different than 64K, since some programs utilize HMA for their own purposes and may not correctly work with HMAREA.

1.5.27 IDTCHK

Purpose: Allows to detect boot viruses in certain cases.

Syntax: IDTCHK = <ON|OFF>

By default: IDTCHK = OFF

Note:
When IDTCHK = ON, the system tests the computer memory for key interrupt code receivers. In case of suspicious interrupt pointers a warning message is displayed and the user has a choice of either continuing the boot or canceling it.

1.5.28 IGNORE

Purpose: Inhibits system processing of specified sections.

Syntax: IGNORE = section[,section[,...]]

Note:
Not used, as a rule, and serves only to complement the FORCE command. As the FORCE and IGNORE commands involve only single-pass parsing, only the names of subsequent sections should be used as their parameters.

1.5.29 INSTALL

Purpose: Allows executing some programs at the stage of configuration file processing.

Syntax: INSTALL = [/e] [drive:] [path]filename [parms],

Parameters:
parms - command line that will be transferred to a program
/e - creates a program environment allocated in the higher memory.

Note:
When loading a program, the system creates an unique environment available to this program and located ,as a rule, before its beginning. Running resident programs which free the memory before exiting, leads to memory fragmentation. Using the /e key provides a program environment located after the program code. Therefore, no memory fragmentation occurs when freeing the environment block.

1.5.30 INSTALLHIGH

Analogous to the INSTALL command except that it uses the UMB for allocating program code.

1.5.31 LASTDRIVE

Purpose: Sets the maximum number of accessible logical drives.

Syntax: LASTDRIVE = ll
Where, ll ranges from 'A' to 'Z'

By default: LASTDRIVE = max('E', last installed drive)

Note:
The ll value defines the last valid logical drive that the system may access. The maximum value is derived when booting. The command is useful with a network environment because, new drives may appear in the process of work. Additional drives are also required for the SUBST external DOS command, allowing different logical drives to be used for access to parts of the physical drive.

Notice that the system allocates memory for each logical drive, so only the necessary maximum value should be specified.

1.5.32 LOCKS

Purpose: Sets the number of system file locks, accessible in system calls.

Syntax: LOCKS = xx
Where, xx is the number of locks ranging form 4 to 92, or 0

By default: LOCKS = 4

Note:
The lock principle is used primarily with networked software. In absence of such, the best memory saving choice is LOCKS = 0.

1.5.33 LOADFIX (sys)

Purpose: Corrects loading of programs designed with old versions of the LINK.EXE program.

Syntax: LOADFIX = <ON|OFF>

By default: LOADFIX = OFF

Note:

Since the old LINK.EXE versions all contained a bug, the following message is displayed when modules created with such a linker are loaded into quite low areas of memory:

Packed file is corrupt

The problem mostly occurs when DOS is loaded into the high memory and may be solved either by the LOADFIX external command or in this case, by LOADFIX = ON. The latter automatically detects incorrectly working programs. The memory located below the loaded program is free and may be used on inquiry; the system loads additional code of approximately 500 bytes.

If the message "Packed file is corrupt" is displayed even when LOADFIX=ON, the LOADFIX external DOS command may be used for solving the problem.

1.5.34 KEYBOARD

Purpose: Sets an alternative keyboard layout.

```
Syntax: KEYBOARD=[XT|AT|PC]<STANDARD|ID>
                 [SIMPLE]
                 [HOTKEY ss cc]
                 [TEMPHOTKEY cccc]
                 [PRELOAD|LOADONCALL]
                 [ACTBORDER xx]
                 [DEFBORDER yy]
                 [USEBORDER]
                 [MAPS nn]
                 [,[codepage [,[drive:][path]filename]]
```

Where, ss	- shift state
ßß	- scan code
ßßcc	- scan code

By default: KEYBOARD = STANDARD

Note:

When defining some "extended" keys it is necessary to set upper byte of the cccc as 0E1h or 0E0h. For instance, the <Ctrl> scan code should be defined as 57373 or 0E01Dh.

To change the keyboard layout from the one supported by BIOS, it is necessary to include KEYBOARD = ID, where ID is a two letter country code (See the COUNTRY command). The new layout may be substituted for another in an interactive mode by using the KEYB external DOS command. Specifying STANDARD is equivalent to stating no KEYBOARD command at all.

The XT, AT and PC prefixes specify the type of keyboard interface and are recommended only in case of faults arising from using the KEYBOARD command.

SIMPLE means absence of "dead keys". Dead keys don't lead to display of some character after being pressed separately. If a letter key follows the dead one, then the specified letter is displayed with an accent or "umlaut".

You can switch from one layout to another in interactive mode with the help of the following keys:

<Ctl-Alt-F1> - set the initial layout
<Ctl-Alt-F2> - set the loaded layout,

or with the DOS KEYB command. Also, a hot key can be specified by stating HOTKEY ss cc, where ss is the shift state and cc - the scan code. In this case you may shift between states by pressing the specified key. It is also possible to define a TEMPHOTKEY hot key used for shifting the layout only while it is held pressed.

Display border may indicate the current status with its color, if the USEBORDER parameter is specified. ACTBORDER xx defines the active state border color (white by default), DEFBORDER yy defines the initial state border color (black by default). Parameter values are treated depending on the type of display: for CGA, it is one of the 16 colors, for EGA and the like, one of the available palette colors (one of 64 or more colors).

MAPS nn defines the maximum number of layouts loaded with KEYB command. nn may range from 1 to 9, default value is 1.

NOTICE: Border color indication does not apply to MDA and Hercules displays, both not having the option.

The system automatically installs a new keyboard layout when KEYBOARD command is processed. PRELOAD is used for making an new layout active after computer loading. LOADONCALL points out that a new layout will be installed by pressing hotkeys. Codepage means exactly what it says. Filename (By default, KEYBOARD.SYS) specifies name of the file containing the keyboard layout data.

1.5.35 MEMORY

Purpose: Sets system memory allocation strategy.

Syntax: MEMORY=<[BESTFIT|FIRSTFIT|LASTFIT] [NOUMB|UMB]>

By default: MEMORY=FIRSTFIT NOUMB

Note:
This command is executed before the pass processing some commands like
INSTALL. The memory allocation strategy can be defined trough system calls.

FIRSTFIT - use the first acceptable memory block
BESTFIT - choose the block closest in size to that inquired
LASTFIT - use the last of the acceptable blocks
UMB - use UMB (only if DOS = UMB)
NOUMB - search for free memory only in the lower memory

Hence, the commands

```
MEMORY=FIRSTFIT UMB
INSTALL=filename
```

are equivalent to

```
INSTALLHIGH=filename
```

although, in the first case it is impossible to perform the simple INSTALL command in view of the global action of the MEMORY command.

1.5.36 QUIET

Purpose: Inhibits output of messages by drivers and TSRs to STDOUT when using commands DEVICExxx = ... and INTSALLxxx = ... during system start-up.

Syntax: QUIET = <ON|OFF>
By default: QUIET = OFF

Note:

The command redirects all STDOUT output to the NUL device, preventing initial loading messages made by device drivers and resident programs during system start-up. Programs, as a rule, use the STDOUT for displaying messages during normal operation and the STDERR in case of errors. But, there are programs using STDOUT for error messages. Such messages won't be seen, if QUIET= ON, this produces an impression that the system "hangs" when a program or driver waits for data input from the console. The QUIET command, therefore is recommended for use only after the system configuration has been thoroughly checked up.

QUIET=YES together with LOGO=OFF, as a rule fully suppresses all the screen output.

1.5.37 SAVENAME

Purpose: Controls the option of saving the first character of a deleted file or directory name.

Syntax: SAVENAME = <ON|OFF>
By default: SAVENAME = OFF

Note:

When SAVENAME=ON, the system saves the first character of a deleted file or directory name. The saved first character of a filename can be used by the UNERASE command.

1.5.38 SCREEN

Purpose: Sets the type of display interface.

Syntax: SCREEN=<[STANDARD|ANSI][FAST|SLOW][DIRECT]>

By default: SCREEN=STANDARD FAST

Note:

SCREEN=ANSI should be specified for supporting ANSI screen output standard. In this case, only the console write operations support ESC-sequences with no keyboard input support of the latter. ANSI.SYS (supplied with the system) should be used for the full ANSI service. When writing to the screen, the system either uses BIOS or directly writes into the video-buffer. The system is forced to write into the video-buffer when DIRECT is stated and the current video mode number is 3 or 7. The approach is recommended for the slower machines of the PC XT type. Besides, there is a possibility of deliberately processing console write system calls with the mode set by FAST or SLOW. In the first case, output rate is high while in the second, it is lower, but with no special code taking up additional memory. The difference in speed is negligible with 386 and higher machines while quite detectable with XT-type machines.

1.5.39 SHELL

Purpose: Sets an alternative command processor name.

Syntax: SHELL = [drive:][path]filename [parms]

Where, filename is a command processor name, parms is the command line to be passed to the program.

By default: SHELL = command.com /P

Note:

Refer to the description of the Command.com and the initial value of the COMSPEC environment variable. In case of failing to find the specified shell or an error while running it, the system will prompt:

"Shell ? _"

In this case, the name of the program together with its parameters should be specified, e.g. "\cmd\command.com /p/e:512", or alternatively, the [Ctrl-Break] may be used for canceling the start-up.

1.5.40 SHELLHIGH

Purpose: Sets an alternative command processor name with an attempt of loading in into the UMB.

```
Syntax: SHELLHIGH = [drive:][path]filename [parms]
        Where, filename - command processor name
                parms - command line to be passed to the program.
```

Note: As the SHELL command but tries to install the shell into the UMB - upper memory blocks.

1.5.41 SHIFTSTATE

Purpose: Sets the initial state of the trigger keys.

```
Syntax: SHIFTSTATE OFF|STANDARD|
                      |<[SCROLLLOCK][NUMLOCK][CAPSLOSK]>
                      |<[KEYRATE=rval] [KEYDELAY=dval]>

        Where,
        rval=1..32 chars./sec., or: MAX and MIN
        dval=1..4 , or: MAX and MIN
```

By default: SHIFTSTATE STANDARD

Note: STANDARD	- no action,
OFF	- resets all the keys,
SCROLLLOCK, NUMLOCK, CAPSLOCK	- sets the active state for the corresponding keys.

KEYRATE and KEYDELAY specify the auto-repeat rate of a pressed key in characters per second and the delay before the auto-repeat, respectively. Much in parallel with the MODE CON command.

1.5.42 SHUTDOWN

Purpose: Sets the system response to [Ctrl-Alt-Del].

Syntax: SHUTDOWN = <ON|OFF>
By default: SHUTDOWN = ON

Note:

When ON, the system will close all the files open for writing operations and save all the buffers to the disk in response to [Ctrl-Alt-Del]. No action is undertaken in the opposite case.

As the SOFTREBOOT=ON involves closing all the files, the SHUTDOWN=OFF cancels it.

1.5.43 SOFTREBOOT

Purpose: Quick system restart, skipping the POST BIOS.

Syntax: SOFTREBOOT=<ON|OFF|x,y>

Where, x is the trigger keys state, y is the key scan-code.

By default: SOFTREBOOT OFF

Note:
In some cases, it is convenient to quickly restart the system, e.g. when experimenting with the configuration file parameters. That's when SOFTBOOT=ON should be used. The default quick restart key combination is [Ctrl-Alt-Ins]. Another combination can be specified using the x and y parameters.

Warning! When trying to restart with the processor running in the protected mode, a sound signal warns that the action is impossible. As the SOFTREBOOT=ON involves closing all the files, the SHUTDOWN=OFF cancels it.

1.5.44 STUBA20

Purpose: Defines a special driver that increases performance of the system located in high memory.

Syntax: STUBA20=<ON|OFF>

By default: STUBA20=OFF

Note:
A special XMS support driver is required for allocating the system in high memory. Any interrupt intercepted by the system leads to a HMA access request driver call. When STUBA20=ON, the system controls the driver calls that enable or disable HMA, so that no additional calls are made. The result is a slightly improved system performance.

Warning! Not recommended for use when running programs that control the high memory, bypassing the XMS driver. STUBA20=ON is incompatible with MS-Windows.

1.5.45 SWITCHAR (sys)

Purpose: Sets the switch character.

Syntax: SWITCHAR = c

By default: SWITCHAR = /

Note:
To switch to a UNIX-like command writing style, specify:

```
SWITCHAR = -
```

All the parameters as /A/B/C should be then written as -A-B-C. '/' instead of '\' should be used in the file path. The first line of the configuration file is the best place for this command, because any other place might lead to ambiguity with parameters, since the system does several passes when processing the file.

1.5.46 SYSBOOT

Purpose: Loads another DOS-compatible system.

Syntax: SYSBOOT = <bootimagefile> [bootdrive[:]]

Parameters:

bootimagefile - file with the specified system boot sector image,
bootdrive - disk to be regarded as the initial for the specified system.

Note:

A file containing the boot sector image of the system used prior to the PTS-DOS may me created by running the SYS external DOS command with the /U key.

If a file containing the boot sector image has incorrect data or does not correspond to the initial start-up disk, the system may "hang".

1.5.47 SYSLOAD

Purpose: Loads a system from another hard disk partition.

Syntax: SYSLOAD = <partNO|?>

PartNO - (1 - 4) - number of the partition with boot sector of the system to be loaded,
'?' - question mark - the system will display all the partition numbers and information on the operating systems contained there.

Note:

An alternative UNIX-type operating system with a file structure different from that of DOS may be loaded by specifying the corresponding hard disk partition. If the partition numbers are known, they may be directly specified in the command. Specifying the question mark will make the system to display information on the partitions and wait for user input of the desired partition number. Incorrect section choice (incorrect code or a non-bootable partition) may "hang" the computer.

1.5.48 SYSGEN

Purpose: Sets the PTS-DOS component allocation strategy.

```
Syntax: SYSGEN <[/Cx][/Sx][/Lx][/Dx][/Vx][/Rx][/Fx][/Tx]>,
          Where, x is: 'L' - forces to allocate components always in
                             conventional memory,
                       'H' - enables using the HMA.
```

Note:

Please, refer to BUFFERS /L or /H - on the strategy of allocating block device I/O buffers, and to COMAREA - on specifying the communication area. The following abbreviations are used with the command:

C - system code
S - system data
L - locks data
D - disk parameters data
V - data on the additional devices
R - current directories structure
T - floppy disk parameters table
F - file tables

Setting, for example, /CH enables the system code to be allocated in the high memory when DOS=HIGH; /FL forces the system to allocate the system file table in the lower memory.

The default strategy depends on the WINDOWS command in the first place. If WINDOWS=OFF, then all data and codes are enabled for loading into the high memory, excluding additional data on the installed block device drivers.

If WINDOWS=386/V:3.0, then the file structure should be in the lower memory, while WINDOWS=386/V:3.1 enables placing the structure in the high memory.

Warning! The command is for the experienced user only. System faults may
occur with some application programs if the command parameters are
incorrect (incompatible).

1.5.49 SYSTOUMB

Purpose: Enables usage of UMB for internal system data and code.

Syntax: SYSTOUMB = <ON|OFF>
By default: SYSTOUMB = OFF

Note:
If SYSTOUMB = ON, the system will attempt to allocate first the HMA memory, then the UMB and in case of more need, the conventional memory.

With some UMB servers running on 386+ computers in protected mode and with SYSTOUMB = ON, restarting the computer by pressing [Ctrl-Alt-Del] might lead to system halt, as the server first switches off the UMB, then returns to real mode operation and jumps to a code located previously in the UMB, but not located there any more.

1.5.50 TRACE

Purpose: Sets/resets trace mode when processing configuration file.

Syntax: TRACE = <OFF>|<ON [DEFAULT=<YES|NO>][TIMEOUT=secs] >

Note:
Global trace mode may be activated by selecting a main section 'Alt-1..0' (See Configuration file processing). All sections to be processed shall be traced then. TRACE = ON action is limited to the beginning of the following section.

DEFAULT defines the default reply on the system request on the expiry of the TIMEOUT time (in seconds) or on pressing the [Enter] key. With DEFAULT specified, the request form shall be:

(Y/n/E) > or (y/N/E) >

depending on the parameter value: in the first case (as well as in the case when TIMEOUT is specified with no DEFAULT parameter), the DEFAULT=YES, while in the second case, DEFAULT=NO. In rest of the cases the prompt is:

(Y/N/E) >

If TIMEOUT is defined, the first triangle bracket of the prompt will be bright white and blinking, otherwise, its color will be identical to the color of other prompt characters.

If the global poll mode is selected (ALT-1..9 is pressed) then all the user inputs can be programmed by pressing ALT-Y or ALT-N:

[ALT-Y] - always "Yes"
[ALT-N] - always "No"

The automatic input is not used with the TRACE command.

On pressing the [E] key in reaction to the system prompt, the current line may be edited. It is worth mentioning that new line length may not exceed that of the original line. At the start of editing, [F3] may be pressed to restore the initial line. To insert a line, the following new configuration file command may be specified:

TEMPLATE

The purpose of this command, which is itself ignored by the system, is for a user to fill the line it occupies with something useful.

For example:

```
        [Section1]
        ........    No trace mode here

        Trace = On
        .......     User confirmation of each line

        Trace = Off
        ........    No trace mode once again

        [Section2]
```

1.5.51 VGASCREEN

Purpose: Controls the mode of a VGA-compatible display

Syntax: VGASCREEN = <<ENABLE|DISABLE>|<ON|OFF>>

Parameters:

ON - activate the screen
OFF - disactivate the screen
ENABLE - enable writing to the video RAM
DISABLE - disable writing to the video RAM

Note:
It worth to remember, that when OFF, the ENABLE and DISABLE commands are ignored.

The command allows to disactivate the screen or to organize video ROM with VGA-compatible displays. Fully corresponds to the analogous BIOS calls.

1.5.52 VIDEOMODE

Purpose: Sets the current video mode during system start-up.

Syntax: VIDEOMODE =[SOFT]<mm|MONO|CO80|BW80|CO40|BW40>
Where, mm is the decimal number of the current mode

Note:
The SOFT prefix adds 128 to the video mode number set by the user in order to save the contents of the video buffer. This operation is only applicable to EGA video adapters or compatibles.

1.5..53 VIRTDMA

Purpose: Defines the server ability to support the virtual DMA service.

Syntax: VIRTDMA = <ON|OFF>
By default: VIRTDMA = OFF

Note:
By default, the system intercepts the interrupt 13h (disk I/O) and executes I/O to the high memory - A:000:000 and above, by using an internal driver located in the lower memory. If the UMB supports the virtual DMA service, a condition almost always fulfilled, then specifying VIRTDMA=ON makes I/O to higher memory run faster, e.g. when loading a program using the INSTALLHIGH command. This is also recommended for usage with a UMB-server that operates on hardware level by reprogramming the chipset, for example, LASTBYTE.

1.5.54 VXD

Purpose: Sets the alternative path and name for a virtual device.

Syntax: VXD=<STADARD|ON|OFF|[drive:][path]filename>
Where, 'filename' is a virtual device name.

By default: VXD=STANDARD or VXD=ON

Note:

The system assumes that the virtual device is located in the root directory and bears the WINA20.386 name. The device is required for running MS-Windows 3.0 in the enhanced mode. If such a device is absent, the system located in high memory will display an error message when starting MS-Windows 3.0 in the enhanced mode. VXD=OFF may be specified, although it might lead to system faults when working under MS-Windows 3.0.

No virtual device is required with MS-Windows 3.1 or higher. The WINA20.386 is supplied together with the system.

1.5.55 WINDOWS

Purpose: Sets the system compatibility level when working with MS-Windows.

Syntax: WINDOWS <ON|OFF|STANDARD|286|ENHANCED|386> [/V:version]
Where 'version' sets the Windows version number, e.g., 3.0 or 3.1

By default: WINDOWS ON /V:3.0

Note:

MS-Windows requires additional DOS code and system data located in the lower areas of the memory, depending on its version and mode (real/standard/enhanced). To achieve the best results, both mode and version number should be specified. For example:

WINDOWS STANDARD WINDOWS STANDARD /V:3.0	Assumes that MS-WINDOWS 3.0 is to be used in standard mode.

STANDARD and 286 are synonyms, just as the ENHANCED and 386.

WINDOWS 286 /V:3.1	MS-Windows 3.1 in standard mode
WINDOWS 386 /V:3.0	MS-Windows 3.0 in enhanced mode

WINDOWS=ON assumes the version is 3.0 with the mode set maximum according to the type of processor used. For the 286-class machines, it is STANDARD, while for 386 and up - ENHANCED. It is not recommended to use the SYSGEN command together with the WINDOWS command, as the system generation may be incompatible with that required by MS-Windows. WINA20.386 (a driver supplied with the system) is required for running MS-Windows 3.0 in the enhanced mode (see the VXD command description).

1.5.56 ZOMBIEDRV

Purpose: Allows bufferization for file deletion on the specified disks.

Syntax: ZOMBIEDRV = drv[:][,drv[:][, ...]]

Note:

The main purpose of this command is to create a queue of files to be deleted. Due to this queue, it is now possible to restore recently deleted files with no data loss. The procedure is the following: when a file is deleted, the system moves it to the \ZOMBIE directory, created before-hand by the MKZOMBIE external DOS command. Directory size is specified on its creating and it defines the queue size. When the directory is full, the system will delete an old file and will write a new one in its place. Files from this directory are also deleted when there is no more space on a disk for new files. As the path is not saved when moving a file, the filenames may be recurring, as could be seen with DIR command. Only the first file is processed. Error messages may occur when running disk data check-up utilities because of some files with the same names.

2 DOS COMMANDS

2.1 INTERNAL DOS COMMAND ARGUMENTS

PTS-DOS operating system commands are divided in two groups:

- resident, or internal commands,
- transient, or external commands.

Internal commands are a part of the operating system and are loaded together with it. Running internal commands doesn't require the presence of a system disk in a drive.

External commands are files, recorded on a system disk. As the procedures of these commands are not included in the standard DOS shell, the COMMAND.COM program, they should be loaded before running. This means that the diskette containing the external commands should be inserted into the drive. There is no need to specify the type of an operating system command when invoking it.

The terms LIST, FILELIST, DIRLIST are used in many internal DOS commands as arguments. Semantically the term LIST may be described as follows:

```
LIST::=[ctrl_params]
        {
        character_string|
        (LIST[,|+| ]...)|
        @reference_file
        }
```

Attention! Character "+" in source part of COPY command means merging
of several files into one file.

The semantic notation will be described below line by line.

Control parameters

Control parameters of the list define the way of processing its elements. Control parameters do not change the order of processing. They are always put before the list elements which they influence, and they influence ALL the elements located in string after them.

The control parameters for FILELIST or DIRLIST may be only the following:

/S - to include all subdirectories of the given directory in the search by pattern;

/O[:order] - sets the directory element sorting criteria, where
"order" is a set of the sorting criteria control symbols:
G - sort on the attributes
N - sort on the file names
E - sort on the file extensions
S - sort on the file size
D - sort on the date of creation
Minus sign before a control symbol changes the sorting criteria to a complementary one. If no order parameter is present, the following sorting criteria is used: on attributes, then on names and extensions.

/A[:attrs] - sets the attribute mask to search, where "attrs" may
be the following:
R - sets the "read-only" attribute
S - sets the "system" attribute
H - sets the "hidden" attribute
A - sets the "archive" attribute
Minus sign before a control symbol means searching for files
without the specified attribute. If there are no "attrs"
after the /A switch, the files with arbitrary attributes will
be searched for.

Character string

Character_string cannot contain any delimiter characters (space, tab, comma, semicolon, plus sign) or a restrictor characters (opening and closing parentheses, quotes, double quotes) and a special symbol "@" as first one. The examples of character strings:

ABC
Abcdef - right strings
12ab

AB C,C
AB(C) - wrong strings
A"BC"
@123

(LIST[,|+|]...)

Any LIST element can represent a complex structure itself and may contain character strings, parentheses, and reference file specifications. One should be extremely careful when writing LIST, and be sure that every opening parentheses corresponds to the closing one.

The examples of complex arguments:

(AAA,BBB,CCC) - a list of character strings as elements;

((a:\1,a:\2),b:\) - a list, containing sublist as the first element
(a:\1,a:\2) and character string b:\ as the second one.

((/s c:,d:),(mail.txt, reply.txt)) - a list, containing two subl as arguments.

Control parameters before command arguments influence all the following list of elements (character strings and contents of reference files), with no regard to the recursive level of the element. For some internal commands (COPY, REN, etc.) the thing of great importance is the first recursive level (so-called bundles, that compose the so-called basket). Basket here is a set of bundles, following one another and enclosed between the first opening and the last closing parentheses. So, in the examples above the bundles are:

for the first example - (a:\1, a:\2) and b:\
for the second example - (/s c:,d:) and (mail.txt, reply.txt)

The bundles may be either lists or character strings. You may use the basket, say, in COPY command.

COPY ((*.c,*.h),(*.asm,*.inc)) (a:\,b:\)

When executing this command all files from the first bundle (*.c,*.h) of source willbe copied to the corresponding bundle of the target basket (to the root of the A: disk). Similarly all the files of the secong bundle will be copied to the root of the B: disk.

Complex bundles may occur in the target basket. The following command copies files with extension. ASM to the disks A and B simultaneously:

COPY *.ASM ((A:,B:))

Note, that the command

COPY *.ASM (A:,B:)

copies files with ASM extension only to the A:, second bundle B: will be ignored (See the COPY command).

@reference_file

Parameter "reference_file" after special symbol "@" defines the name of a text file that will be used for reading list elements from. So, the reference file contains elements to be used in the command.

In one of the examples above we could substitute the target bundle for
the reference file:

COPY ((*.c,*.h),(*.asm,*.inc)) @responce.txt

where the contents of responce.txt is:

```
a:\
b:\
```

The difference between LIST and FILELIST(DIRLIST)

The main difference between LIST and FILELIST(DIRLIST) is in the treatment of the "character string" notion. In FILELIST(DIRLIST) only the filenames(dirnames) may be used in a character string. Sure, these names may contain only the symbols allowed by DOS for filenames (dirnames).

In a FILELIST (DIRLIST) one may specify a group of files(directories) in one "character string" using wildcard symbols "*" and "?" But the use of these symbols is allowed only in the last element of the composite filename (directory name) after the last backslash.

Asterisk (*) in the name (extension) of the file(directory) means an arbitrary amount of any (but allowed by DOS for names) characters. All other characters following the asterisk will be ignored. So, all the files (directories) that match the pattern, will be used in the command.

Character "?" in the name (extension) of the file (directory) means that it can be replaced by any allowed symbol, but only once. In this case any characters, following the "?" symbol, preserve their meanings.

Examples:

. or ????????.???	- means all the files in the current directory.
*a.asm or ????????.asm	- means all the files having the .ASM extension.
FILE?.txt	- means all files having the .txt extension, whose names contain five characters and first four of them are "FILE".

If FILELIST contains a name of existing directory, then "*.*" is implied after this name (all files in this directory are implied).

2.2 Internal and external commands.

2.2.1 APPEND COMMAND

Purpose: Sets appended directories for data files search.

Syntax : For first use only:

```
APPEND [/X[:ON | :OFF]] [/PATH:ON | /PATH:OFF] [/E]
```

To specify directories to be searched:

```
APPEND [drive:][path][;[drive:][path] ... ]
```

To delete appended paths:

```
APPEND;
```

Type: External command

Description:

The APPEND command allows you to specify a search path for data files and executable modules. If PTS-DOS doesn't find the needed file in the current directory, it tries to find it in directories, specified in the option of the APPEND command or in the current process environment variable.

If the APPEND command is invoked without the /X switch or with the /X:OFF switch, it will act only for the "open file" system calls - (int 21h, Functions 3Dh,6Ch).

Parameters:

/X or
/X:ON — Extends the command action on system calls "execute" (int 21h, Function 4Bh), "find first" (int 21h, Function 4Eh).

/X:OFF — Turns off the /X:ON switch.

/E — In this case PTS-DOS will use the appended search paths, specified in the variable of the APPEND environment.

/PATH:ON — Indicates that, if searched files are specified with the drive and/or with the search path, PTS-DOS should search for them in the appended directories, specified by the APPEND command, in case these files are not present in the indicated place.

/PATH:OFF — Indicates that, if searched files are specified with the drive and/or with the search path, PTS-DOS should not search for them in the appended directories, specified by the APPEND command, in case these files are not present in the indicated place.

You may specify several appended directories, separating them with a semicolon - ";". If you start the APPEND command with the option a second time, the old directory list will be replaced by the new one. If the APPEND command was started with the /E switch, you may change the directory list by redefining the variable of the APPEND environment.

If you start the command without options, the current list of the appended directories will be displayed.

You can temporarily abort the execution of the APPEND command by specifying NUL directory as an option:

APPEND ;

It means that PTS-DOS will search for files only in the current directory.

Notes:
1. You can use the APPEND command in the network to access the network files.
2. If you use the ASSIGN command, you must start APPEND before starting ASSIGN.
3. If you want to set a search path for executable modules or external commands, you should better use the PATH command.

Example:

Suppose you want to access data files in the PTSDOS directory on drive C and in the EXE directory on drive A. To do this, use the following command:

```
APPEND C:\PTSDOS;A:\EXE
```

2.2.2 ASSIGN COMMAND

Purpose: Changes the drive letter for input/output operations and so makes the system to use that other drive.

Syntax: [d:] [path\]ASSIGN [x[:]=y[:][...]]
or
[d:] [path\]ASSIGN /S[TATUS]

Type: External command

Description:

The ASSIGN command is included into the operating system to make it easier to work with software, which only use drives A and B for input/output operations. Redefining the drives by the

```
ASSIGN a=c b=c
```

command allows You to use hard drive for an application program which may work only with drives A and B.

Parameters:

[d:] [path\]	- drive and directory, where the ASSIGN.COM file is located
x	- the initial drive
y	- the drive which will be used for accessing the drive x

Notes:
1. The ASSIGN command doesn't require a colon mark after the drive letters.
2. If you use the ASSIGN command without any parameters, it will reset all previous drive assignments.
3. Using the ASSIGN command with the /S switch shows the current state of the redefinition.
4. You should change drive assignment only when it is necessary. The ASSIGN command cannot be used with BACKUP, RESTORE, LABEL, JOIN, SUBST and PRINT commands. FORMAT, DISKCOPY and DISKCOMP ignore all disk reassignments, made by the ASSIGN command.

Examples:

```
C> ASSIGN a=c
```

With this command all references to the drive A: go to the drive C:. Thus, if you enter DIR A: upon the execution of this command, information about the drive C: will be displayed.

```
C> ASSIGN
```

This command resets all previous drive reassignments.

2.2.3 ATTR (Attribute) COMMAND

Purpose: Allows to set or display file attributes.

Syntax: ATTR [+S|-S] [+H|-H] [+R|-R] [+A|-A] [/P[ROMPT]] [/M[ORE]] FILELIST

Type: Internal command

Parameters:

+S - sets the system attribute of a file.
-S - cancels the system attribute.
+H - sets the hidden attribute of a file.
-H - cancels the hidden attribute.
+R - sets the read-only attribute of a file.
-R - cancels the read-only attribute.
+A - sets the archive bit of the file.
-A - cancels the archive bit of the file.

If you enter consequently several ATTR commands with different attributes the result will be the sum of commands. For example, you enter

```
ATTR +h a.bat
```

and then, after a time you enter

```
ATTR +r +s a.bat
```

A.BAT file will have attributes "hidden", "system" and "read only".

/P[ROMPT] - prompts the user for confirmation to change the attribute of the file

/M[ORE] - pauses after each screen full of information.

```
FFILELIST ::= [parms]
        {
        [d:][path]file_name[.ext] |
        (FILELIST[,|+| ]...)      |
        @reference_file
        }
```

FILELIST - defines the files for setting attributes to.
(See INTERNAL DOS COMMAND ARGUMENTS for detailed description)

file_name - may contain wild card symbols ('?' and '*').
reference file - is the name of a text file containing the command's options. It should be specified explicitly.
parms - are any combination of the following keys:

/S - to include subdirectories in the search
/O[:order] - sets the directory element sorting criteria.
order is a set of the sorting criteria control symbols:
G - sort on the attributes
N - sort on the file names
E - sort on the file extensions
S - sort on the file size
D - sort on the date of creation
Minus sign before a control symbol changes the sorting criteria to a complementary one. If no order parameter is present, the following sorting criteria is used: on attributes, then on names and extensions.

Examples:

```
1       A:\>attr +r -a file1.txt
```

This command sets the "read-only" attribute to the FILE1.TXT file and cancels the "archive" bit. The following message will be displayed on the screen:

```
FILE1.TXT attributes are set to -R---
(the "read-only" attribute is set for the FILE1.TXT file)
```

2. A:\>attr file1.txt

This command displays attributes of a file. The screen will display:

```
 -R--- FILE1.TXT
```

3. A:\>attr -r file1.txt

This command cancels the "read only" attribute.

4. A:\>attr file1.txt,

On entering this command you may see that the "read only" attribute has been cancelled:

```
----- FILE1.TXT
```

5. A:\>attr +a /P file?.txt

If the user specifies the /P key ("with prompt") then the shell will ask for confirmation before changing the attributes:

```
FILE1.TXT change attributes for this file {Y/N/All yes/Quit}? _
```

To reply, the user should input one of the above answers and press [Enter]. Pressing [Enter] alone is equal to "Y".

"Y" - the current file attributes are set and the Shell jumps to the next file.

"N" - the current file attributes are left without changes and the Shell immediately jumps to the next file.

"A" - attributes of all the remaining files are modified with no asking for user confirmation.

"Q" - quit the command.

Note:
Modifying the archive bit affects the BACKUP/M command. If the bit is set, the corresponding file is copied; if the bit is cancelled (equal to zero), then no file copying occurs.

2.2.4 BEEP COMMAND

Purpose: Sets a TSR, that changes the simple sound signal, generated by DOS after [Ctrl-Break] has been pressed for a complex one.

Syntax: BEEP [/M:mm][/T]

Type: External command

Description:
/M switch defines the index number of a generated signal. mm ranges from 1 to 3. Having defined /M:0 as the parameter, you can switch off the signal. If you want check or choose the necessary signal, you should press the /M switch together with the /T switch. Herewith you will hear the signal while the resident won't be loaded.

2.2.5 BREAK COMMAND

Purpose: Enables or disables the operation system to break the execution of any DOS function on pressing [Ctrl-Break].

Syntax: BREAK = [ON:OFF]

Type: Internal command

Parameters:

ON - sets BREAK = ON. The system checks if the [Ctrl-Break] keys' being pressed before executing any DOS command and breaks the execution, if those keys are pressed.
OFF - sets BREAK = OFF. In this case the system controls [Ctrl-Break] keys' pressing only during:

- standard input operations,
- standard output operations,
- standard printing operations,
- operations with standard auxiliary devices (adapters).

By default: BREAK = OFF.

Running the BREAK command without parameters leads to the BREAK current condition being displayed.

Examples:
1. Setting BREAK to ON

```
A:\> BREAK ON
```

2. Displaying current BREAK setting

```
A:\> BREAK
```

The following will be displayed on the screen:

```
Break is ON
```

2.2.6 CHDIR COMMAND

Purpose: Changes current DOS directory of the mentioned or current disk drive or displays the current path of a disk directory.

Syntax: CHDIR [d:][path]
or
CD [d:][path]

Type: Internal command

Parameters:

[d:] - defines a drive, the current directory of which should be changed or viewed.
[path] - defines the path to the required directory (the path_name can not be longer than 63 characters).

If a drive name is not mentioned, the current one is assumed. The CHDIR command without parameters displays the current drive.

Examples:
1. Changing the current directory of a floppy disk to the root one:

```
A:\DOS>CHDIR \
```

2. Displaying the current directory of the A: drive:

```
A:\DOS>CHDIR
```

The screen will display: **A:\DOS**

3. Changing the current directory of the B: drive to \dir1\dir2:

```
A:\>CD b:\dir1\dir2
```

4. Changing the current directory of the B: drive to a path extended by the dir3 directory:

```
A:\>CD b:dir3
```

WARNING! The real name of a directory may be hidden by ASSIGN, SUBST and JOIN commands. The following example illustrates how one may work with the C: drive, when the A: drive is the current one:

```
A:\>JOIN C: A:\CDRIVE
A:\>CD \CDRIVE
```

2.2.7 CHKDSK (Check Disk) COMMAND

Purpose: Checks specified disks, looks for errors in the file system and corrects them.

Syntax: CHKDSK [/F/S/V /C<d1>: /U<d2>:] [[d3:] ...]

Type: External command

Description:

The CHKDSK without parameters will only check the information on the disk and display messages about the errors, without making corrections. We recommend you to use this command first without any switches. Sometimes, slight corrections manually introduced by a specialist to the file system, can restore lost clusters.

The CHKDSK command also generates a small report on the disks under check. For example:

```
CHKDSK 1.04 Copyright (C) Paragon Software1993-1996

Checking drive D:
Serial number - 0054-2587 tools
Volume in this drive - TOOLS

52,340,736 bytes total disk space.
1,937,408 bytes available on disk.
50,403,328 bytes occupied on disk .
```

The generated report and all error messages may be saved by redirecting the output to a file:

```
A:\chkdsk b: > a:\disk_a.txt
```

Do not specify a file located on a drive under check, for it may cause errors.

Parameters:

/F - During the disk check CHKDSK will correct all found errors. All corrections will be written to the disk, so it is reasonable to use the
/C parameter to save the previous condition of the disk.

/C - CHKDSK will save the correction backup information on a disk, specified after the parameter without space, to a file named:

undo_(checked_disk).chk

You should not save to the disk, which is under checking. This parameter may only be used together with the /F parameter.

If the CHKDSK command detects "lost" clusters on the disk, you are prompted with a question: Convert lost clusters to files? If you type "Yes", the CHKDSK with /F parameter will write every cluster to a file named:

FILEnnnn.PTS,

where: nnnn - is a hexadecimal number, starting with 0000, FILEnnnn.PTS files are created in the main directory of the specified disk. These files can be then viewed and used, if they contain useful information.

/S - During the disk check CHKDSK will use only the first FAT copy.

/U - With this parameter the CHKDSK will restore the initial condition of the disk if the corrections were made using the /C switch. A drive containing the backup information should be specified immediately after the /U key with no trailing spaces as:
undo_(checked_disk).chk

WARNING! If the disk has been modified after correcting the errors with the /F /C parameters, e.g. files were amended, deleted etc., then the /U parameter should not be used to avoid new errors.

/V - CHKDSK, will display all file names and their paths.

During the check of a disk the system can find subdirectories, containing invalid information. Here the user should specify, whether to reform a corrupt directory into a file or to still consider it to be a directory. It is recommended to save it as a file for this would not cause new errors if the disk checking is continued; while the files, located in a "bad" directory, will become "lost". After completion of the command, they can be found in the root directory of the checked disk, under names automatically generated by the system.

Notes:

1. The CHKDSK command assumes the floppy disk to be checked is readily inserted in a disk drive.
2. The CHKDSK command doesn't work on drives, used in the SUBST and the JOIN commands.
3. CHKDSK corrects the detected errors with the switch /F only. If you have a very important information on the disk under check, we highly recommend you to use the command with /C parameter. It saves the previous condition of the disk, so that you will be able to restore the information corrected by CHKDSK. Having saved the information about corrections on the other disk, you should not then write or change anything on it. Otherwise, new mistakes can appear or the information on the disk can be corrupted, when you restore it with the /U parameter.

5. CHKDSK should not be used in multitask system environment, like WINDOWS, DESKview etc. Otherwise, the information on the disk can be corrupted.

Examples:

1 **A:>chkdsk a:**
or
A:>chkdsk

displays the condition of a floppy disk from the drive A:

2. **A:>chkdsk a: b: c: /f**

checks the condition of disks A:, B: and C:, corrects all mistakes found in the directory or FAT.

3. A:>chkdsk b: c: /f /ca:

checks the condition of disks B: and C:, corrects all errors, found in directories or FATs and saves initial condition on the disk A: to files:

```
A:\>undo_b.chk  and
A:\>undo_c.chk
```

4. A:>chkdsk c: /ua:

restores the initial state of the disk C:

5. A>chkdsk a: /v

checks the condition of the floppy disk from the drive A: and displays all filenames and directories on this disk.

2.2.8 CLS (Clear Screen) COMMAND

Purpose: Clears the screen.

Syntax: CLS

Type: Internal command

Description:

This command clears the screen of the display, while screen attributes, set by extended codes of display control (ANSI sequences) remain without changes.

2.2.9 COMMAND (Secondary Command Processor)

Purpose: Starts the Secondary Command Processor.

Syntax: [d:][path]COMMAND [/F] [/D] [/P] [/E:xxxxx] [/Cstring]

Parameters:

[d:][path] - disk and directory, from where DOS should load Command Processor. If COMMAND.COM file is absent in the specified directory, DOS finds the path in its own environment. The COMMAND.COM loads the Command Processor from a directory, defined by the COMSPEC= sequence during system initiation.

/F - always sets Fail in the handler of critical errors.
/D - cancels the mode set by the previous parameter.
/P - new command processor should constantly stay in the memory. The usage the /P parameter means, that the control will not go back to the primary command processor. The secondary command processor may only be disactivated by restarting DOS.

[/Cstring] - allows to execute "string" and to give the control automatically back to the primary Command Processor, if the /P parameter was not specified. String is a command, which the secondary command processor should execute. The command is interpreted and executed as if it has been run from the DOS prompt. For example, typing:

```
A>command /c dir b:
```

will load a secondary command processor which executes the **dir b**: command and returns the control to the primary command processor.

Including the following line in the CONFIG.SYS configuration file:

SHELL = Command Com. /P /C AUTOCALL.BAT

will give an alternative name to the AUTOEXEC.BAT file, which is run automatically during system loading, where AUTOCALL.BAT is the name of a batch file that will be executed during initial system boot.

/E:xxxx - is a decimal integer number, that determines the length of the environment (parameters, defining the configuration of the processor). The number should be in the range from 160 to 32768 and it is rounded up to the nearest paragraph boundary (16 bytes).

Running the COMMAND command without parameters loads a new copy of the command processor. This copy gets the environment of the command processor of the previous level. Changing the environment of a secondary command processor (by the SET command) has no influence upon the environment of the primary command processor. The return of the control to the primary command processor restores the environment, which existed prior to loading the second command processor.

Example:
Suppose the initial command processor uses DOS standard prompt pg (by default). The secondary command processor inherits this prompt. If you change this prompt for another one during the work of the secondary command processor, then after return to the initial command processor the prompt will still remain ng. After loading of a secondary command processor with no parameters, the primary command processor could be re-activated with the help of the special command EXIT.

2.2.10 COPY COMMAND

Purpose: Copies one or more files to a specified disk.

```
Syntax: COPY [/A|/B] [/P[ROMPT]] [/V] [/Q[UIET]]
        source
        target

source ::= FILELIST
target ::= FILELIST
```

Type: Internal command

Parameters:

/A - copied file is considered to be a text file (ASCII), data is copied up to the end of file (Ctrl-Z), the remaining part of the file isn't copied.
/B - file is copied according to the length, specified in the directory.
/V - Switches on the file verification mode during writing into the file.
/Q - Switches off messages during copying.
/P - Asks for user's permission for file copying.

```
FILELIST ::= [parms]
        {
        [d:][path]file_name[.ext] |
        (FILELIST[,|+| ]...) |
        @reference_file
        }
```

FILELIST - defines the files to be copied.

file_name - may contain wild card symbols ('?' and '*').
reference_file - is the name of a text file containing the command' options. It should be specified explicitly.

parms - are any combination of the control parameters for FILELIST or DIRLIST (see INTERNAL DOS COMMAND ARGUMENTS for detailed description).

The COPY command may be used in three ways:

- for copying without changing file names;
- for copying with changing the file names;
- for copying and merging files.

Copying without changing of a file name

Syntax: COPY [/V] [/A|/B] [d:][path\]file_name[.ext] [d:][path]

This command is used for creating files, that have identical names, types, and contents.

Examples:

1. A> COPY C:\COMMAND.COM

copies COMMAND.COM from the root directory of C: disk to the current one, while

2.` A> COPY C:\COMMAND.COM A:\

copies COMMAND.COM file to the root directory of A: disk.

Note: If the target directory doesn't exist, the COPY command considers the second parameter to be the destination file name.

Copying with changing a file name

Used to make a copy of a file, that has a different name than that of the source.

Syntax: COPY [/V] [/A|/B] [d:][path\]file_name.ext [d:][path]file_name.ext

Examples:

1. A> COPY FILE1.TXT B:REPORT.DOC

copies the file FILE1.TXT from the current directory of the A: disk into the current directory of B: disk and names it REPORT.DOC.

It is allowed to use standard devices names while copying, e.g.:

```
A> COPY CON FILE1.TXT
A> COPY CON AUX
A> COPY CON LPT1
A> COPY AUX CON
A> COPY FILE1.TXT LPT1
```

The NUL device may also be used.

2. The following example shows how you can use the COPY command, to fill the file with data straight from the keyboard.

```
A> COPY CON FILE1.TXT
```

The command instructs the system to copy the all the data input from the keyboard to the file FILE1.TXT until the [Ctrl-Z] has been pressed. The input may also be completed by consecutively pressing [F6] and [Enter].

Note: It is assumed in the above example, that the [F6] key has not been redefined.

Copying and merging of files

```
Syntax: COPY [d:][path\]filename.ext
             [+[d:][path\]filename.ext...]
             [d:][path\]filename.ext
```

Used for merging files while copying. Files specified by the command are appended to the end of the first copied file. The time and the date of the merged file will be written into the directory, where files are copied to. The message displayed at the end of the COPY command's work, states the number of created output files.

Files merging requires a list of two or more initial files, separated by a sign '+'.

Examples:

1. A> COPY FILE1.TXT + FILE2.TXT + B:FILE3.TXT BIGFILE.TXT

This command creates the file BIGFILE.TXT in the current directory. This file contains files FILE1.TXT, FILE2.TXT from the current directory and the file FILE3.TXT from the current directory of the disk B:, one after another.

If the name of the target file is not specified, extra files are added to the first one of the list. The command

2. A> COPY FILE1.TXT + FILE2.TXT

appends the FILE2.TXT file to the FILE1.TXT.

The wild cards '*' and '?' can be used in names of merged files as well as in the name of the target file.

3. A> COPY *.H + *.C PROGRAM.PRN

This command merges all the files with the .H extension, then all the files with the .C extension and writes them into the PROGRAM.PRN file.

4. A> COPY *.H + *.C *.PRN

Merges every file with the .H extension with a corresponding file with the. C extension and creates files with .PRN extension bearing the same name. I.e., the FILE1.H file is merged with the FILE1.C file thus forming the FILE1.PRN file.

2.2.11 CTTY (Change Console) COMMAND

Purpose: Changes standard input/output from a console for an auxiliary one or sets a keyboard and a screen as a standard device of input/output.

Syntax: CTTY device's name

Type: Internal command

Parameters:
The name of a device defines the type of the device, used as a system console. Specifying AUX, COM1 or COM2, makes these devices to be the system ones. Specifying the CON device restores the keyboard and the screen as the system I/O devices.

Notes:
1. The CTTY command adopts the name of any character device, providing the possibility to use in house device servicing programs. Nevertheless, it is important to make sure that a given device may be

used both for input and output. For this reason, a printing device may not be specified, as it can't be used for input.
2. The CTTY command runs only in programs, that use standard DOS functions. Other programs, like BASIC (i.e. those, where there are no DOS functions call) can not use the CTTY command for changing the standard I/O device.

Examples:
1. The following command defines the AUX device to be the standard I/O device:

```
A>CTTY AUX
```

2. The CTTY command restores the keyboard and the screen, as the standard I/O devices:

```
A>CTTY CON
```

2.2.12 DATE COMMAND

Purpose: Allows to set or change the date. This date is written to a directory when a file is created or amended.

Syntax: DATE [/Q[UIET]] [mm-dd-yy]|[dd-mm-yy]|[yy-mm-dd]

Parameters:

mm - defines the month, ranges from 1 to 12.
dd - defines the day of the week, ranges from 1 to 31.
yy - defines a year, ranges 80 to 99 or from 1980 to 1999.
/Q[UIET] - doesn't prompt the user to input the date.

Notes:
1. If you use the DATE command without parameters, the following message will be displayed:

```
Current data is mm-dd-yy
Enter new date ( mm-dd-yy ) :_
```

The date format (mm-dd-yy, dd-mm-yy, yy-mm-dd) depends on the country. The date format can be changed by creating the CONGIG.SYS file, that contains the COUNTRY command.

2. The date, month and year may be separated by hyphen "-", point ".", slash "/". For example, the date the first of May, 1991 may be presented differently:

5-1-91 or
5.1.91 or
5/1/91

3. If a new date has been correctly input, the system will move to executing other commands. If the date is wrong, the following message will be displayed:

```
Invalid data
Enter new date /mm-dd-yy/:_
```

4. The system displays the day of the week (for example, Mon) for your information only. It is not possible to modify it.

5. You can change the date from your standard input device or from a batch file. You can include the DATE command into the AUTOEXEC.BAT, and the system will automatically display a prompt for the date, when it is started.

6. If the date remains unchanged, you should press the ENTER switch.

The input of a new date doesn't change the system time.

Examples:

1. To insert a new date, type:

```
A:\>DATE
```

The screen will show:

```
Current date is Tue 2-2-1993
Enter new date: 3/24/93
```

2. To retrieve the current date, type:

```
A:\>DATE /Q

Current date is Tue 2-2-1993
```

3. It is possible to specify the date just briefly. The following command, for instance, sets the 2-nd day of the current month and year.

```
A:\>DATE 2
```

The same brief specifying may be used when the date is requested by the system.

2.2.13 DEL (Delete) COMMAND

Purpose: Deletes the specified file(s).

Syntax: DEL [/P[ROMPT]] [/Q[UIET]] [/D[IR]] FILELIST

Type: Internal command

Parameters:

```
/P[ROMPT] - causes "del" to prompt you before a deletion.
/Q[UIET] - doesn't prompt at all when deleting files.
/D[IR] - deletes a directory, when all its files have been deleted.
FILELIST ::= [parms]
        {
        [d:][path]file_name[.ext] |
        (FILELIST[,|+| ]...)      |
        @reference_file
        }
```

FILELIST - defines the files to be deleted.

file_name - may contain wild card symbols ('?' and '*').
reference_file - is the name of a text file containing the command's arguments. It should be specified explicitly.
parms - are any combination of the control parameters for FILELIST or DIRLIST (see INTERNAL DOS COMMAND ARGUMENTS for detailed description).

Notes:

1. It is not possible to delete "read only" files.
2. One has to be extremely careful when using the DEL command after having invoked ASSIGN, JOIN or SUBST.

Examples:
1. Deleting the FILE.BAT file from a diskette in the A: drive:

```
A:\>DEL A:\FILE.BAT
```

2. Deleting all files from the GAMES directory located on the C: fixed disk:

```
A:\>DEL C:\GAMES
```

The shell will prompt on deleting all the files from the directory:

```
All files in the directory C:\GAMES\*.* will be deleted !
Are You sure {y/N/All no/Quit}? _
```

In case you aren't sure which particular files you wish to be deleted, press [N], [A] or [Q], then [Enter], or simply press [Enter]. If you press first [Y] then [Enter], the shell will delete all the files in the directory.

There will be no prompt if the /Q switch is used in combination with other switches.

WARNING! When using the DELETE command with the subdirectory deletion switch, the prompt is made only ONCE, after which all files and
subdirectories (if the /D switch has been set) are deleted.

3. Using the prompt option

```
A:\>DEL /P file?.txt
```

If the /P (prompt) switch has been set, then the Shell will prompt the user before deleting any file:

```
FILE1.TXT delete this file {Y/n/All yes/Quit}? _
```

The user would reply with one of the following symbols and press [Enter] afterwards. Pressing [Enter] alone is equal to replying "Y".

"Y" - the file is deleted and the Shell moves on to the next one.
"N" - the file is NOT deleted and the Shell moves on to the next one.
"A" - the file is deleted and all the remaining files will be deleted with no prompt whatsoever.
"Q" - quitting the command.

4. Using the subdirectories option

The following command deletes the *.bak files located an all subdirectories of the C: disk:

```
A:\>DEL /S C:\*.BAK
```

The next command deletes all the files and subdirectories of the D: disk:

```
A:\>DEL /S/D D:\
```

WARNING! Improper or mischievous usage of the DELETE command may lead to loss of data on your disks! Paragon Software bears no liability whatsoever if such a damage or loss occurs.

2.2.14 DIR (DIRECTORY) COMMAND

Purpose: Lists all files in a directory or files specified in the command.

```
Syntax: DIR [d:][path][file_neme[.ext]]
        [/P] [/M[ORE]] [/W] [/L] [/U]
        [/C[=controls]]
```

Type: Internal command

Parameters:

[d:][path][filename[.ext]]
- defines a file to be viewed.

/P - causes the directory display to pause once the screen is filled. On the bottom line of the screen you will see the message:

Strike any key when ready...

To resume scrolling the display, press any key.

/W - displays filenames and subdirectories of the specified directory only.

/M[ORE] - selects page mode.

/L - displays filenames in small letters.

/U - displays filenames in small letters, and directories - in capital letters.

/C - displays the information about files in a format, specified by the user. The following information may be displayed:

N – filename
S - file size (bytes)
T - time of last modification
H - directory name
U - drive mark

E - name extension
D - date of last modification
P - whole path to file
F - number/size of files
O - general statistics

Notes:
1. Directories are marked by <DIR> in the field of file length.
2. A file name and extension may contain global characters "?", "*".
3. DIR command can work faster, if you set more buffers (3 or more) in the BUFFERS= command of the CONFIG.SYS file.

Examples:

1. The following command lists all files and subdirectories from the default drive:

```
A>dir
```

The result can be the following:

```
Volume in drive A is MYDISK
Directory of *.*

FILE1.A              10368           7-20-83  12:13p
FILE2.A              1613            4-16-86  12:40p
9X            31             8-17-85  10:59a
[LEVEL2   ]   <DIR>          9-09-85  12:10p
FILE1         2287           3-03-86  11:10a
5 File (s)    141512 bytes free
```

2. The following command lists the contents of the current directory of the drive C:

```
A>dir C:
```

3. This command lists the contents of the directory \LEVELS2:

```
A>dir \level2
```

The following information will be displayed:

```
Volume in drive A is MYDISK
Directory of \LEVEL2\*.*

MYPROG.COM          2463     7-30-82  8:55a
1 File(s) 141312 bytes free
```

4. There are two special elements of the directory, marked by "." and "..". They are, respectively, the current directory and the directory of the higher level (in this case, the parent directory). Consequently, if the current directory is the specified directory, and you want to look through the directory of the higher level, you can type:

```
A>dir..
```

5. If you type the following command:

```
A>dir *.a
```

the following information will be displayed:

```
Volume in drive A is MYDISK
Volume Serial Number is 0320-E338

Directory of *.a
FILE1.A          1580 08-12-1993 13:44
FILE3.A          9954 08-14-1993 16:30

11534 bytes in   2 file(s)
751616 bytes free
```

2.2.15 DISKCOPY (COPY DISKETTES ONLY) COMMAND

Purpose: Copies the contents of the disk in the source drive to a disk in the target drive. If necessary, you can format the target disk, when copying.

Syntax: [d:] [path] DISKCOPY <d1:> <d2:> [options]

Type: External command

Parameters:

[d:] [path] before the DISKCOPY command defines the drive and the path, that contain this command.
<d1:> - defines the source drive,
<d2:> - defines the target drive.
/?/h - help (all the other options are ignored).
/g - ignore data read errors.
/v - verification when writing.
/f - format the target disk.
/o:<interleave> /i:<displacement>
- analogous to those of the FORMAT command and are ignored if the "/f" is not specified.
/d<[:c] | <:disk>>
- enable swap-out to a local disk.

In the DISKCOPY command you can specify similar or different drives. If you specify similar drives, copying will be performed using one drive. The system will prompt you to change a diskette. To continue copying, you would press any key.

Upon completion, the DISKCOPY command prompts you with the following question:

Copy another disk (Y/N)?

If you type "Y", the system will prompt you to insert new diskettes.
If you type "N", the copying will be completed.

Notes:
1. The given command may be used only for copying diskettes. An error message will be displayed if a fixed disk is specified.
2. If the format of the target disk differs from the format of the source disk, the DISKCOPY command prompts to format it anew.
3. If the source drive contains errors and the "/g" option has not been set, the following message appears on the screen:

Disk read error !
Abort? Retry? Ignore? iGnore up to the end?

Then, the user has a choice of specifying:
A - cancel copying;
R - try again;
I - ignore the error;
G - ignore all the errors right until the end of the diskette
4. In case of target drive write error, the following is displayed:

Disk write error !
Abort? Retry? Ignore? Format track and retry? format up to the end?

Possible answers:
A - cancel copying;
R - try again;
I - ignore the error;
F - format the faulty track then write again;
O - format all the remaining tracks.
5. In cases 2. and 3. the target drive may prove unfit for subsequent usage.
6. Source and target drives may not be virtual disks, like those created by the SUBST command.
7. DISKCOPY doesn't allow redefining the drives.
8. If the double-sided disks were recorded in a high capacity drive, some problems may arise with their reading in standard capacity single- and double-sided drives.
9. DISKCOPY can't be used with the JOIN command.
10. DISKCOPY doesn't work with network drives.
11. Non-standard disk formats aren't recognized by MS-DOS.
12. In case when the copying is done on a single physical drive and the amount of free memory is not sufficient for reading the whole diskette, the system attempts to organize swapping to the XMS-memory or to a specified local device (if the /d switch has been set) such as fixed disk, RamDisk etc. Any attempt of specifying a network or removable device is regarded as an error. The /d switch is ignored if there isn't enough conventional or XMS memory.

Examples:

1. diskcopy a: b:

copies a diskette from the drive A: to a diskette in drive B:

2. diskcopy a: a: /v /d

copies the source diskette from the drive A: to the target diskette on the same drive with verification and swapping to a local disk (C:).

3. diskcopy b: b: /f /i:1 /o:1 /d:e

copies diskettes form the drive B: while formatting the target diskette and swapping to drive E:

4. diskcopy b: a: /g

copies a diskette from drive B: to a diskette in drive A:, ignoring read errors.

2.2.16 DISP COMMAND

Purpose: Sets the screen fonts driver.

```
Syntax: DISP [<xx|yyy>[,[zzz][,[drive:][path]filename]]
        or
        DISP
```

Where,

Xx	defines double-character identifier of the required country,
Yyy	defines the character set,
Zzz	defines the number of a buffer for loading,
Filename	defines a file, containing bitmap fonts for display.

Type: External command

Note:
1. The command will work, if you load the support driver, built in the operating system kernel. It is defined by the DISPLAY configuration file comand.
2. Running the command without options will show the current condition of the driver.

Description:

If you aren't sure which character set corresponds to your country, you can specify the identifier of the country instead. The System then will choose the character set according to the relationship given below:

Identifiers of countries and code pages numbers:

NO – Norway	865 or 850	
BE – Belgium	437	850
IT – Italy	437	850
LA - Latin America	437	850
CF - Canada (French)	863	850
DK – Denmark	865	850
GR – Germany	437	850
HU – Hungary	852	850
FR – France	437	850
NL – Netherlands	437	850
PO – Portugal	860	850
US – USA	437 or 850	
RU – Russia	866	850
TF - Turkey (French)	857	850
TQ – Turkey	857	850
SF - Switzerland (French)	437	850
SG - Switzerland	437	850
SV – Sweden	437	850
SU – Finland	437	850
SP – Spain	437	850
UK - United Kingdom	437	850

The system can support up to 9 loaded fonts simultaneously. Switching from one font to another is made by pressing [Ctrl-Shift-Fxx], where xx corresponds with the buffer's number. Hence, [Ctrl-Shift-F2] activates the first buffer, [Ctrl-Shift-F10] - the last one. Pressing [Ctrl-Shift-F1] combination, you specify ROM fonts to be active. The number of buffers and their location is set by the DISPLAY configuration file command. Nevertheless, the system can load only one buffer during start-up. The given command may be used for downloading all the remaining or to reset the current ones.

2.2.17 EXE2BIN COMMAND

Purpose: Converts .EXE files to .COM or .BIN files

Syntax: EXE2BIN [drv:][path]name1[.EXE] [[drv:][path]name2[.BIN]]

Type: External command

Description:
This command converts .EXE (executable) files to binary format. If you do not specify an extension for the first, input file, it defaults to. EXE. The input file is converted to a .BIN file format (a memory image of the program) and places it in the output file name2.

If you do not specify a disk or a name for the output file, the system will use the disk and the path of the input file. If you do not specify a filename extension in the output file name, the system gives the new file the extension. BIN.

Some restrictions do apply when you use this command: the input file must be in valid .EXE format; the actual code and data part of the file must be less than 64K bytes; there must be no STACK segment definition for the converted program.

Two kinds of conversion are possible, depending on the initial meaning of the CS:IP.

If the CS:IP is not specified in the program (.EXE file contains 0:0), a binary conversion takes place. If segment fixup is necessary, the system asks for the absolute segment at which the program is to be loaded. The resulting program will be usable only when loaded at the absolute memory address specified by your application. The command interpreter COMMAND.COM will not be able to load the resulting program.

If the CS:IP is 0000:0100h, the system will convert the input file to. COM format. In this instance no segment fixup is allowed, since programs of this type must be segment independent, when are being loaded. You can load the converted program with the help of the command interpreter COMMAND.COM.

2.2.18 ERASE COMMAND

Purpose: Deletes the specified file. This command performs the same operations as the DEL command.

2.2.19 FDISK COMMAND

Purpose: FDISK is used for:
- creating DOS partitions (logical disks);
- changing the active partition;
- deleting DOS partitions (logical drives) as well as partitions of other operating systems;
- displaying fixed disk partition and logical disks information;
- selecting the next fixed disk (if the PC is equipped with several fixed disks).

Syntax: FDISK [parameters]

Parameters:

```
/?, /H[ELP]         - help,
/B[LACK]            - display unused partitions,
/C[OMPATIBLE] - create MS-DOS compatible logical drives only,
/I[GNORE]           - skip error messages on start,
/M[BR]              - update MBR (master boot record) and quit,
/S[ECTOR]           - create logical drives aligned by sector
```

Running FDISK involves consecutive execution of the following operations:

1. Inserting a DOS diskette into the drive A: and restarting the system.
Once the system prompt has appeared, type:

```
a>fdisk
```

2. The following main menu appears on the screen after pressing [Enter]:

```
+-----------------------------------------------------------------------------------------+
|FDisk Setup Utility for PTS-DOS (C) 1995 Paragon Software |
|                                                                                         |
|        Fixed disk drives total:     1                                                   |
|        Current fixed disk drive:    1                                                   |
|        Logical disk drives total:   3                                                   |
|                                                                                         |
|        Total disk size (MBytes):                  520.7                                 |
|        Total disk free space (MBytes):            11.7                                  |
|        Max contiguous free space (MBytes):        11.7                                  |
|                                                                                         |
|        Main menu                                                                        |
|                                                                                         |
|1. Create logical drive.                                                                 |
|2. Set active partition.                                                                 |
|3. View/Delete partitions.                                                               |
|4. Change current fixed disk drive.                                                      |
|5. Undo changes.                                                                         |
|6. Quit FDISK.                                                                           |
|                                                                                         |
|Enter your choice [6]                                                                    |
+-----------------------------------------------------------------------------------------+
```

You should view information about partitions first. To do this, press
[3], then [Enter]. The View/Delete menu will apear on the screen.

View/Delete menu

This menu contains information about all partitions on the fixed disk.
The following information about each partition is displayed:

"Partition" the number of partition; the word "Primary", if information about the partition is in MBR;
"Drv" a drive letter, which MS-DOS would assign to the logical drive;
"Status" the word "Adwanc", if the partition is advanced active, the word "Active", if the partition is standard active.
"Type" partition type ("DOS-12", DOS-16", "BigDOS" are logical DOS drives, "Extend" is an extended DOS partition containing other partitions etc.);
"MBytes" partition size in MBytes;
"Notes" error messages if there are any.

```
+--------------------------------------------------------------------------------------------------------------+
|         Partition        Drv      Status           Type             Mbytes           Notes   |
| 1       Primary          C        Advanc           BigDOS           250.9                    |
| 2       Primary          E        BigDOS           252.4                                     |
| 3       Primary                                    Extend                                    |
| 4        Primary                                  - -
| 5                                                 HPFS       1.8                    |
| 6                                                 Extend                            |
| 7                  D                              DOS-12     2.8                    |
|                                                                                   |
| 1. Next Page.                                                                     |
| 2. Previous page.                                                                 |
| 3. Set advanced active partition.                                                 |
| 4. Delete partition.                                                              |
| 5. Delete all partitions.                                                         |
| 6. Return to Main Menu.                                                           |
|                                                                                   |
|Enter your choice: [6]                                                             |
+-----------------------------------------------------------------------------------+
```

If there is not enough place for displaying information about all the partitions, it may be divided into pages. You can scroll them using items [1] and [2] in this menu.

After viewing the information about disk partitions, you can execute the needed operations:

A. Create logical drive

To create a logical drive select item [1] of the Main Menu. The screen will show a prompt to select logical disk type to be created:

```
+-----------------------------------------------------------------------------------+
|          Partition    Drv     Status         Type        Mbyte   Notes             |
| 1        Primary      C       Advanc         BigDOS      250.9                     |
| 2        Primary      E                      BigDOS      252.4                     |
| 3        Primary                             Extend                                |
| 4        Primary                              - -                                  |
|                                                                                    |
|WARNING! You have only four records to create primary DOS, extended DOS             |
|          or non-DOS partitions.                                                    |
|                                                                                    |
|          Total disk size (MBytes):                          520.7                  |
|          Total disk free space (MBytes):                    11.7                   |
|          Max contiguous free space (MBytes):                11.7                   |
|                                                                                    |
| 1. Create primary DOS logical drive.                                               |
| 2. Create extended DOS logical drive.                                              |
| 3. Create primary HPFS logical drive.                                              |
| 4. Create extended HPFS logical drive.                                             |
| 5. Return to Main Menu.                                                            |
|                                                                                    |
|Enter your choice: [5]                                                              |
+------------------------------------------------------------------------------------+
```

This menu allows you to create the so called primary logical drives, i.e. partitions with descriptions located in the master boot record (MBR) of the fixed disk, and also the so called extended logical disks with descriptions located in extended partitions.

It is worth mentioning that the MBR can hold only up to four partition descriptions. Therefore, the total number of primary DOS partitions, other OS partitions can't exceed four partitions. As a rule, with DOS, it is sufficient to create only the primary partitions as rarely more than four logical disks are required. When using more than one operating system with one fixed disk, it is recommended to create a single primary partition, which will be the system one for DOS, while all the other partitions could be extended. The number of the latter is limited by the number of letters in the alphabet. An extended partition will occupy one track more than primary of the same working size.

After selecting the partition type, the screen will show something like:

```
+------------------------------------------------------------------------------------------------------------+
|       Partition       Drv     Status          Type            Mbytes  Notes                 |
| 1     Primary         C       Advanc          BigDOS          250.9                         |
| 2     Primary         E                       BigDOS          252.4                         |
| 3     Primary                                 Extend                                        |
| 4     Primary                                 - -                                           |
| 5                                             HPFS            1.8                           |
| 6                                             Extend                                        |
| 7                     D                       DOS-12          2.8                           |
|                                                                                             |
|       Total disk size (MBytes):                               520.7                         |
|       Total disk free space (MBytes):                         11.7                          |
|       Max contiguous free space (MBytes):                     11.7                          |
|                                                                                             |
| Enter primary logical drive size (MBytes):                    11.7                          |
+------------------------------------------------------------------------------------------------------------+
```

The partition size can then be entered in megabytes (up to two decimal points of accuracy). By default, it is assumed that the partition size is equal to the size of the largest contiguous block on the disk or to the total disk size, if the disk has been totally empty. If this is what you need, then simply press [Enter]. Otherwise, input the size and then press [Enter].

B. Set active partition

To make active one of the primary partitions, select item [2] in the Main Menu. The information panel will look like follows:

```
+------------------------------------------------------------------------------------------------------------+
|       Partition       Drv     Status          Type            Mbytes          Notes         |
| 1     Primary         C       Advanc          BigDOS          250.9                         |
| 2     Primary         E                       BigDOS          252.4                         |
| 3     Primary                                 Extend                                        |
| 4     Primary                                  - -                                          |
|                                                                                             |
| Which partition do you want to make active?                                                 |
+------------------------------------------------------------------------------------------------------------+
```

Actions:

1. Enter the number of partition to be active.
   An extended or unused partition can not be active.
2. Press [Enter]. The selected partition is now (standard) active. The status of the other partitions is cleared.

There are advanced active partitions besides standard active ones. An advanced active partition can be set by selecting item [3] in the View/Delete menu. The screen will show disk partition status data. Enter the number of the partition the status of which you would like to change and then press [Enter].

```
+------------------------------------------------------------------------------------------------------------+
|       Partition       Drv     Status          Type            Mbytes          Notes         |
| 1     Primary         C       Advanc          BigDOS          250.9                         |
| 2     Primary         E                       BigDOS          252.4                         |
| 3     Primary                                 Extend                                        |
| 4     Primary                                  - -                                          |
| 5                                             HPFS            1.8                           |
| 6                                             Extend                                        |
| 7                     D                       DOS-12          2.8                           |
|                                                                                             |
|Which partition do you want to make advanced active (0-none)?                                |
+---------------------------------------------------------------------------------------------+
```

In order to understand the difference between the standard active and the advanced active partition we should consider, what proceeds at the beginning of system booting. The MBR is loaded and run first. It checks up the presence of the advanced active partition in the system (there can be only one advanced active partition and it can be placed both in MBR and in an extended partition and even not on the first fixed disk). If the advanced active partition is found, the load sector is loaded from it. If there is no advanced active partition, a standard active partition is seated for, and the load sector is loaded from that partition then.

There are some restrictions in using active partitions with different operating systems.

PTS-DOS is always loaded from the first logical DOS drive which can be both standard active and advanced active. MS-DOS can be loaded only from a standard active partition. So, you should better make advanced active the partition with PTS-DOS and standard active the partition with MS-DOS. In this case you will be able to load MS-DOS from PTS-DOS using SYSLOAD configuration command.

C. Delete partition

To delete a DOS partition or partitions of other systems, choose item [3] ("View/Delete partitions") in the Main Menu and then choose item [5] in the View/Delete menu, if you want to delete all the partitions. To delete a selected partition, choose item [4]. Now, select a partition to be deleted:

```
+---------------------------------------------------------------------------------------------+
|       Partition     Drv     Status          Type          Mbyte          Notes   |
| 1     Primary       C       Advanc          BigDOS        250.9                  |
| 2     Primary       E                       BigDOS        252.4                  |
| 3     Primary                               Extend                               |
| 4     Primary                                - -                                 |
| 5                                           HPFS          1.8                    |
| 6                                           Extend                               |
| 7                   D                       DOS-12        2.8                    |
|                                                                                  |
|Which partition do you want to delete?                                            |
+---------------------------------------------------------------------------------------------+
```

Warning! Deleting an extended partition deletes all the partitions it once contained.

D. Change current fixed disk drive

If the computer has more than one fixed disk and the next disk needs to be prepared, select item [4] in the main menu, specify the number of the next fixed disk and press [Enter].

E. Undo changes

FDISK has an option for protection from erroneous actions. If you change something on fixed disks, a special menu is displayed on the screen:

```
+-----------------------------------------------------------+
|WARNING! Writing to disk.                                  |
|                                                           |
| 1. Create Undo file on disk A:                            |
| 2. Create Undo file on disk B:                            |
| 3. Not create Undo file.                                  |
| 4. Quit FDISK.                                            |
|                                                           |
|Enter your choice: [1]                                     |
+-----------------------------------------------------------+
```

If you select item [1] or [2] in this menu, a special file FDISUNDO.DAT will be created on the corresponding drive. This file contains all the changed fixed disk sectors. If you made a mistake while deleting or creating a partition, you can undo changes using item [5] in the main menu:

```
+----------------------------------------------------------+
|                                                          |
|1. Undo from disk A:                                      |
|2. Undo from disk B:                                      |
|3. Return to Main Menu.                                   |
|                                                          |
|Enter your choice: [3]                                    |
+----------------------------------------------------------+
```

Now you can just select the drive containing the file FDISUNDO.DAT, to undo all the changes, saved in this file.

F. Quit FDISK

To quit FDISK, select item [6] in the main menu. If changes were introduced to the fixed disk configuration, the screen will display:

```
+----------------------------------------------------------+
|WARNING! Partition tables were changed.                   |
|                                                          |
|          Quit Menu                                       |
|                                                          |
|1. Reboot.                                                |
|2. Quit FDISK.                                            |
|3. Return to Main Menu.                                   |
|                                                          |
|Enter your choice: [3]                                    |
+----------------------------------------------------------+
```

Select item [1] to reboot considering changes on fixed disks.

2.2.20 FIND COMMAND

Purpose: Searches for a specific text string in file(s).

Syntax: FIND [/C/D/I/L/N/R/U/V/W/Z] string
          [drive:][path]filename [[drive:][path]filename ...]

Type: External command

Parameters:

The FIND parameters are one or several single characters, preceded by the (/) character. Each character is a switch, that can be "switched on" and "switched off". "Switching on" is done by the (+) character, "switching off" is done by the (-) character. These characters follow the parameter of the command line (for example, /R+ or /R-). If you do not specify the action with the switch, it defaults to (+).

/C - is a counter: it displays the number of found strings. For each file, in which the command finds the string, the filename and the number of strings will be typed. The strings themselves won't be typed.

/D - directories: For each filename, specified in the command line, the FIND utility looks for all files with the same name both in the specified directory and in all subdirectories of this directory. If you specified the filename without the path, the FIND utility looks for it in the current directory.

/I - ignore case.

/L - look through files that contain the string; FIND displays only names of these files.

/N - display the found lines, together with their numbers.

/R - search for regular expressions. The text, defined as a string of search, is considered to be a regular expression.

/U - the option of change: FIND joins options specified in the command line with the default parameters and writes them all into the FIND.COM file as new parameters values, used by default.

/V - Type only those strings, that do not contain the string of search.

/W - Word search: a text will be found, if the meaning of an expression will match, and if the preceding and the following letters won't be parts of a word. The set of characters for words contains A-Z, 0-9 and (_).

/Z - FIND types the filename for every found file. The number of a string precedes every string. There is also a register with the number of found strings in every file.

Description:

Please mind that, every parameter of the FIND utility is a switch. At any time any parameter can be switched on or off. When the given parameter appears in the command line, it cancels its former meaning. For example:

```
find /r+ /i /d- /i /r- main  my *.asm
```

After entering this command line the FIND utility will be invoked with the following parameters (/d - switched off, /i - switched on, /r - switched off).

With the help of the /u parameter you can build in parameters, used by default, in the FIND utility.

Search string:

The meaning of the search string defines a pattern, using which the FIND utility makes its search. The search string can be either a regular expression or a string of characters. A regular expression is a separate character or a set of characters. Concatenation of regular expressions is a regular expression. In regular expressions some characters have a special purpose - they control the search. The string of characters has no operators - every symbol is considered to be a character. The search string is considered to be a regular expression when the FIND command is used with /r parameter.

You should put the search string in quotation marks if it contains spaces and tabulators, to avoid taking the letters as separators.

Operators in regular expressions

When you use the /r parameter, the search string is considered to be a regular expression and the following characters have special purposes:

^ - at the beginning of the expression this character corresponds to the beginning of the string.
$ - at the end of the expression this character corresponds to the end of the string.
. - this character corresponds to any character.
* - the expression, after which this separator follows corresponds to several occurrences of this expression. For example, in <fo*> it corresponds to <f>, <fo>, <foo> and etc., but it doesn't correspond to <fa>.
+ - the expression after which this separator follows, corresponds to one or more occurrences of this expression. For example: <fo+> corresponds to <fo>, <foo>, etc., but not to <f>.
\ - this symbol precedes the character, that FIND should look for.

File specification

The third part of the FIND command line is the definition of a file. It tells the FIND utility in which files (groups of files) it should look for the string. The definition of a file can be either the specified filename or wildcard ("?" and "*" symbols). You can also specify the path, as the part of the file definition. If the path is not specified, FIND looks for a string in the current directory only.

Examples:

It is assumed that all the FIND parameters are "switched off"in the following examples:

Example 1:

Command line:	find start: *.asm
It will search for:	start: restart:
It won't search for:	restarted: ClockStart:
Where:	In files with .ASM extension in the current directory.
Note:	Small and capital letters differ by default.

Example 2:

Command line:	find /i /w word *.doc
It will search for:	every new word must be on a new line However, MY WORD ! word: smallest unit of speech which conveys meaning In the beginning there was WORD, and the WORD
It won't search for:	Each file has at least 2000 words. He seems to continually misspell "toward" as "toword."
Where:	In *.DOS files in the current directory.
Note:	This format defines global search of "words".

Example 3:

Command line:	find "search string with spaces" *.doc *.asm a:\work\myfile.*
It will search for:	This is a search string with spaces in it
It won't search for:	THIS IS A SEARCH STRING WITH SPACES IN IT This is a search string with many spaces in it.
Where:	In *.DOS and *.ASM files in the current directory and in MYFILE.* files in WORK directory in the drive A.
Note:	This is an example how you can search for a string with spaces.

Example 4:

Command line:	find /i /l /d " the " *.doc
It will search for:	Anyway, this is the time we have do you think? The main reason we are
It won't search for:	He said "Hi" to me just when I Where are you going ? I'll bet you're headed to
Where:	In *.DOS files in the current directory and in all subdirectories on the current disk.

Note:	Case-insensitive search. Any filenames, containing at least one search string are displayed.

Example 5:

Command line:	find /r "^1.*" *.*
It will search for:	All strings that have "1" as the first symbol
It won't search for:	All strings, that have any symbol, except "1" as the first one.
Where:	In all *.DOS files in the current directory.

2.2.21 FORMAT COMMAND

Purpose: Formats a disk in the specified drive, creating a structure, that can be perceived by the DOS. The command looks through all paths, and blocks all damaged sectors for further usage. Initializes the Root directory.

WARNING:

Formatting destroys any previously existing data on a disk. So we recommend you make sure that your decision to run FORMAT is correct. Formatting a hard disk destroys previously created DOS segment, directories and their contents. A hard disk must have a DOS segment. If it is absent, the FORMAT command will not recognize this disk. If the letter of the drive is omitted, the following message will be displayed:

Drive not specified

Syntax: [d:][path]FORMAT d: [parameters]

Type: External command

Parameters:

[d:] [path] before the command, defines the drive and the path of the FORMAT command.

[d:] - specifies the drive containing the disk to be formatted.

/?/H - help.

/S - the following files of the operating system should be copied from the DOS system disk to the new disk:
PTSBIO.SYS
PTSDOS.SYS
COMMAND.COM

/B - the following files should be copied from the DOS system disk of DOS to a new one:
PTSBIO.SYS
PTSDOS.SYS

/U - may be useful for nonstandard format (blocks check-ups of certain parameters).

/E - "verify" mode: global disk testing (without formatting), resetting logical parameters.

/Q - "quick" mode: cleaning the Root directory and FAT only, reloading of the initial sector.

/F:<size> - choose one of the standard disk formats (sizes: 160, 180, 320, 360, 720, 800, 1200, 1440, 2880).

/V:[volume label]
- gives an opportunity to label a disk. It is convenient to use this parameter, as label identifies every disk individually. A label cannot be used instead of a filename, because the latter is an input parameter of a system command.

/N: <sectors per track>;
/T: <number of tracks>

- allows to create a disk with nonstandard physical parameters.

/C:<cluster size>

- number of sectors per cluster.

/R:<root dir size>

- max. number of files in the root directory.

/A:<number of FATs>

/O:<interleave>

- provides extra space between consecutively readable sectors on the track; "interleave" is the space size in sectors; this option is useful for format of the highest recording density for a drive; recommended interleave is 1 or 2.

/I:<intertrack optimization offset>

- provides additional interleave between the last sector on the previous track being read and the first one on the next track; it is useful for many types of floppy drives; recommended value is 1 (rarely 2).

Notes:

1. Every new floppy or hard disk must be formatted before use.
2. If the length of the DOS segment is changed with the FDISK command, a hard disk should be formatted again.
3. Formatting destroys any previously existing data on a disk.
4. All the damaged clusters get "BAD" attribute during formatting. Therefore, these clusters won't be later used for holding data.
5. PTSBIO.SYS and PTSDOS.SYS files are hidden, that's why they are invisible, when you look through directory.
6. The FORMAT command gives the following information about the disk:
- full capacity of the current disk
- the capacity of the damaged disk space
- disk space for users' files.

7. Hard disks are formatted physically during their manufacturing. The FORMAT command only checks sectors of DOS segments.
8. If you want to write all the three system modules using the FORMAT command together with the /S option and there is a lack of RAM, the system will write as much data as possible, format a target disk and write modules from memory. Then it will load the rest part of the system modules in the memory and write it on the target disk. If the source diskette is removed from the drive, the system will prompt you with the corresponding message. The execution of the command will continue, once you insert the source diskette back in the drive.
9. The FORMAT command ignores any redefinition of drives (ASSIGN).
10.You should not use the FORMAT command with drives, redefined by JOIN and SUBST commands.
11.You cannot use the FORMAT command with a network drive.
12 Hard disks have standard physical parameters, known by DOS. If you try to use /F, /T, /N options for them, it will be considered to be a mistake. By default, the max. standard size, possible for the given drive is assumed.
13.If the /C, /R, /A options are not specified, the system uses the standard parameters for the given size of a disk.
14.The /I, /O options can be used for diskettes only and serve to improve their read-write performance.
15.The following options are ignored in "verify" mode: /F, /T, /N, /I, /O; all options, except /V, /S, /B are ignored in "quick" mode.

16.The disk, that has the number of FATs different from two, won't be read correctly under MS-DOS.

17.If the /V option isn't specified, the user will be prompted to enter the volume label; if you want to get the disk without the label, you can specify /V: without the name or simply press "Enter".

18.When using the command with /S or /B options, the system files (and corresponding BOOT-code) are searched for on the disk, from which the system was last loaded; if they are not found, you should enter DOS filenames with corresponding pathes from the console; if you are trying to format the disk, from which you loaded last time, you will be prompted to insert the system disk into the drive A:.

Examples:

1.

```
A>format b:/s
```

The command formats the floppy disk inserted in the B: drive and copies the system files. Before the execution of this command the following message is displayed:

```
Insert disk into drive and press <Enter> ...
```

When you insert the corresponding disk and press the "Enter" switch, the following message will be displayed:

```
Formatting     t tracks       h heads        s sec/trk
track:         t head:        h
```

When the command is completed, the following information will be displayed:

```
Format complete
Input volume label (11 characters) : mydisk
System transferred
Volume serial number is  XXXX-XXXX

xxxxxx bytes total disk space
xxxxxx bytes available on disk
xxxx bytes per allocation unit

Format another (Y/N)?
```

In this example MYDISK is the volume label, entered by the user. If you want to continue formatting, press "Y". Otherwise, press "N".

If you are formatting a hard disk, then instead of the prompt for the floppy disk the following message will be displayed:

```
This is hard disk !!!
All information will be lost !
Proceed formatting ? (Y/N)
```

If you want to format the hard disk, press "Y". Otherwise, press "N".

It takes a few minutes to format a hard disk, depending on the capacity given to the DOS system. In other words, the FORMAT command works while the indicator on the disk panel burns.

2.2.22 HISTORY COMMAND

Purpose: Supports the "history" of commands entered from the keyboard.

Syntax: HISTORY [/H|/L] [/X|/C] [size]

Type: External command

Note: Keeps a resident part, sometimes in conventional memory.

Description:

Some commands perform the input from the keyboard using the DOS function. One of the examples of such a program is COMMAND.COM. Working with this program it is very handy to have an opportunity to recall one of the recent command lines. Exactly, what this command does. You will be able to look through all recent commands by pressing arrow keys [Up] and [Down]. Having chosen the required string, you can continue its editing, or input it by pressing [Enter].

Parameters:

/H - permission to use HMA;
/L - contrary to the previous one, the resident code will be situated
in the conventional memory;
/X - use the extended memory for data;
/C - use the conventional memory for data;
size - defines the size of data buffer (When you use the extended memory
for data, it is rounded to kilobytes)

Running the command without arguments is equal to:

```
HISTORY /H/X 512
```

2.2.23 JOIN COMMAND

Purpose: Join logically a disk to a directory or to another disk. Enables creating one directory instead of two separate ones.

```
Syntax: [d:] [path] JOIN
        or
        [d:] [path] JOIN d:  d:\directory
        or
        [d:] [path] JOIN d:  /D
```

Type: External command

Parameters:

[d:] [path] defines the drive and the directory, containing the JOIN command, only if the command is not on the current directory of the default disk.

d: defines the disk, that should be joined to the directory on the other disk.

d:\directory defines the directory, to which the disk should be joined. It should be a subdirectory of the root directory. If the specified directory does not exist, it will be created on the specified disk by the JOIN command. If it exists, it should be empty.

/D cancels joining structures. The disk to be cut off should be specified. For example, if the disk B: was joined to the C:\JOINB, the following command will cancel the previous JOIN: A>join b: /d

Notes:

1. If the directory to be joined is not empty, the following message will be displayed: "Directory not Empty".

2. The directory you want to join the disk to should not be current. The message "Invalid parameter" means, that the current directory was specified in the JOIN command.

3. The message "Invalid drive specification" accures when the user tries to use a disk which does not exist or which has already been joined. For example, if the drive A is a joined one, you cannot use "A" letter as the name of the disk to be joined, until it is cut off.

4. A directory, specified in the command, cannot be the root one (\).

5. Running the JOIN command without parameters allows you to get the list of joined disks and directories. For example, if the drive A is joined to the C:\LEVEL1 directory, the following message will be displayed:

```
A:  =>  C:\LEVEL1
```

6. When joining a disk to a directory, the whole directory tree of the disk (beginning with the root directory) joins the directory.

7. The disk, joined to a network cannot be the parameter of the JOIN command. The system displays "Cannot JOIN a network drive" error message.

8. The results of the disk join, redefined by the SUBST and ASSIGN command are unpredictable.

9. The following commands do not work on drives used in the JOIN command: BACKUP, RESTORE, FORMAT, DISKCOPY + DISKCOMP.

Example:

Let's assume, that
- drive B is a default drive,
- the current directory on the drive C is C:\.

Joining the disk to the directory:

Suppose C has the following directory tree:

```
                +--------+
                |C:\        |
                +----+---+
                     |
        +---------------+---------------------+
    +----+--+                        +------+-----+
    | FILES |                        | REPORTS |
    +--------+                       +-------------+
```

Drive A joins the directory C:\DRIVEA after the following command:

```
B>join a:\drivea
```

As the result, we get the following directory tree of the drive C:

```
                    +---------+
                    |   C:\     |
                    +----+---+
          +---------------------+-------------------+
        +----+--+         +------+------+     +-----+----+
        | FILES |         | REPORTS |         | DRIVEA |
        +--------+        +-------------+     +-----------+
```

Note: The directory DRIVEA was created on the drive C by the JOIN command.

As the result of the DIR C: command we get:

```
Volume in drive C is FIXEDS
Directory for C:\

FILES           <DIR>           8-21-85         2:10p
REPORTS         <DIR>           9-23-85         5:55p
DRIVEA          <DIR>           9-30-85         12:03a
3 Files(s)      1048956 bytes free
```

After the command DIR A: the message "Invalid drive specification" will be displayed, because the access to the joined disk is disabled. After the command DIR C:\DRIVEA the directory of the drive A is displayed.

Displaying the joined drives state:

The JOIN command without parameters displays the joined disks and their location:

```
B>join
```

On the screen we have:

```
A:  => C:\DRIVEA
```

Cancel of JOIN:

To cancel the join in the last example, use the command:

```
B>join a: /d
```

Now you get an access to the drive A, because it is not joined any longer.

How to use the JOIN command:

The JOIN command gives an opportunity to use all advantages of hard and floppy disks. The following example shows how you can get an access to files of the drive A through the drive C.

```
D>join a: c:\datafils.dir
D>cd c:\datafils.dir
D>path=c:\
D>dataprog c:
```

2.2.24 KEYB COMMAND

Purpose: Sets the keyboard layout for a required language.

Type: External command

Syntax: KEYB /B:N /F:M [xx[,[yyy][,[drive:][path]filename]]]
or
KEYB

Parameters:

xx defines the double-character identifier of the required country
yyy defines the character set
filename defines the file, containing the keyboard layout
/B:N defines the background color in using keyboard layout.
'N' is background color number.
/F:M defines the keys combination to access keyboard layout. 'M' may range from 1 to 9 and sets the number of functional key (M+1).
Having specified this parameter you may change current layout by pressing Ctrl-Alt_Fn, where Fn is F2...F8. The combination Ctrl-Alt-F1 is used to call up the default layout.

For example, after you enter the commands:

```
KEYB /B:1 /F:1 RU
KEYB /B:2 /F:2 GR
```

pressing [Ctrl-Alt-F1], you call up the default layout (english),
pressing [Ctrl-Alt-F2], you call up the RU (russian) layout,
pressing [Ctrl-Alt-F3], you call up the GR (german) layout.

Notes:

For the execution of the given command the built-in driver must be loaded with the help of KEYBOARD configuration command.

Running the command without arguments will show the current condition of the driver.

Description:

The KEYBOARD driver lets you use those characters that are not standard for this computer (as a rule, the standard character set allows to type symbols of the English ABC). Note, that the characters that appear on your screen when you type do not always match the label on the key. To produce some characters, it is required to press "dead" keys. Dead keys do not display a character when used alone, but when followed by a letter, display that letter with a change. This can be used, for example to type umlauted character (German character set). If a letter typed after a dead key does not have the altered state, a sound signal could be heard (see the BEEP command). After that two letters will be displayed: the first one - dead key, and the second - the pressed letter without change.

To activate the driver press [Ctrl-Alt-F2]. You can switch to the default keyboard format by pressing [Ctrl-Alt-F1]. For this purpose you can use also "hot" key (see the KEYBOARD command in config. sys.)

Countries' identifiers and character sets:

NO – Norway	865 or 850	
BE – Belgium	437	850
IT – Italy	437	850
LA - Latin America	437	850
CF - Canada (French)	863	850
DK – Denmark	865	850
GR – Germany	437	850
HU – Hungary	852	850
FR – France	437	850
NL – Netherlands	437	850
PO – Portugal	860	850
US – USA	437 or 850	
RU – Russia	866	850
TF - Turkey (FGGIOD)	857	850
TQ - Turkey (QWERTY)	857	850
SF - Switzerland (French)	437	850
SG - Switzerland (Ger.)	437	850
SV – Sweden	437	850
SU – Finland	437	850
SP – Spain	437	850
UK - United Kingdom	437	850

2.2.25 LABEL (Volume Label) COMMAND

Purpose: Allows to create, change and delete the volume label on a disk.

Syntax: [d:] [path] LABEL [d:] [label]

Type: External command

Parameters:

[d:] [path] before the command, defines the drive and the path to the LABEL command.

[d:] defines the drive, with which the LABEL command will work. If the drive is not specified, the command uses the default drive.

[label] is a name you can specify for a disk. A volume label may be up to 11 characters. The volume label can be also created by the FORMAT/V command. If you do not specify the label in the LABEL command, the following message will be displayed:

```
Volume in drive x is xxxxxxxxxxx
Volume label /11 characters; Enter for none/?
```

If you want to change the existing volume label, you should type a new one and press [Enter]. A new volume label will be written on the place of the previous one.

To delete the volume label you should press [Enter]. The command will display this message:

```
Delete current volume label (Y/N)?
```

If you type [Y] (for 'yes'), the command deletes the volume label on the disk.

If a volume label is longer then 11 characters, the rest characters will be ignored.

Examples:

1. How to create a volume label on a disk:
To label a floppy disk in the drive A with the name ADDRESS, you might type:

```
A>label a:address
```

2. To label a hard disk with the name FIXEDISKC, type:

```
A>label c:fixediskc
```

3. To change the volume label ADDRESS by the label PROGRAMS, type:

```
A>label a:

Volume in drive A is ADDRESS
Volume label /11 characters, Enter for none/?
```

Type "PROGRAMS" and press [Enter].

4. The following command deletes volume label of a diskette in drive A:

```
A>label a:

Volume in drive A is PROGRAMS
Volume label /11 characters, Enter for none/?
```

Press [Enter]. The following message will be displayed:

```
Delete current volume label (Y/N)?
```

If you press [Y] and [Enter], the command will delete the volume label on a disk.

Warning: The LABEL command doesn't work on drives used in SUBST, ASSIGN commands and on drives over a network.

2.2.26 LOADFIX COMMAND

Purpose: Loads the program higher than the 64k of memory and runs it.

Syntax: LOADFIX filename [arg_list]

Type: External command

Parameters:

filename - the name of loaded program
arg_list - the list of arguments of the loaded program

Note:

See the description of the configuration command LOADFIX (sys).

Description:

As the old versions of the link-program LINK.EXE contained an error, the following message may appear when loading modules created by this program into a relatively low memory area:

Packed file is corrupt

It occurs mainly when DOS is loaded in High Memory Area (DOS=HIGH command in config.sys file). In order to solve this problem use LOADFIX=ON command in config.sys file or external DOS command LOADFIX.

2.2.27 MEM COMMAND

Purpose: Displays information on the system memory usage and device drivers or loaded programs.

Syntax: MEM [<switch>[+|-]...]

```
where
"+"    - switch is on
"-"    - switch is off
```

By default: MEM /S/E/X/Y

Type: External command

Switches:

/D	Display the Memory Control Block (MCB) list
/I	Display the installed Device Drivers list
/C	Display the installed Memory Resident Programs list (TSR)
/S	Display the summary using of system memory
/E	Display the information about Expanded Memory (EMS)
/X	Display the information about Extended Memory (XMS)
/Y	Display the information about System location
/H	Display the list of EMS blocks
/B	Display the list of XMS blocks

/L	Display the list of file locks
/F	Display the information about System Data location
/K	Pause between information blocks
/A	Display all information

For example, running MEM command without any parameters, you will get the following information:

```
655,360 bytes total conventional memory
591,472 bytes free conventional memory
590,608 largest executable program size

3,407,872 bytes total contiguous extended memory
        0 bytes available contiguous extended memory
10,240 bytes total XMS memory available
10,240 largest XMS memory block size

PTS-DOS resident in High Memory Area
```

2.2.28 MKDIR (Make Directory) COMMAND

Purpose: Makes a directory on the specified disk (or subdirectory in specified or current directory).

Syntax: MKDIR [d:] [path] subdirectory_ name
or
MD [d:] [path] subdirectory_ name

Type: Internal command

Parameters:

[d:] [path] defines the drive, where the subdirectory will be created and the path to this subdirectory. If the drive is not specified, the command uses the default drive.

Subdirectory's name is specified like filename.

Notes:

The number of created subdirectories is limited by the available space and the length of path. The max. length of any path from the root directory to the desired subdirectory cannot be more than 63 characters, including obligatory backslashes "\".

Each directory may contain files and subdirectories, names of which can be met in other directories. Consequently, files and subdirectories may have similar names, if they are defined in different directories.

Examples:

1. The command:

```
A>md \level1
```

creates the subdirectory LEVEL1 in the root directory.

2. The command:

```
A>md \level1\level2
```

creates the subdirectory LEVEL2 in the subdirectory LEVEL1. If the directory LEVEL1 is the current one, the last example can be written as following:

A>md level2

The backslash before the path means that DOS will begin search of the specified subdirectory from the root directory.

Warning: Be careful creating a new directory, when ASSIGN, JOIN or SUBST commands work.

2.2.29 MKZOMBIE COMMAND

Purpose: Creates a ZOMBIE directory on specified disk.

Syntax: MKZOMBIE drv[:] [cnt]

drv - is a disk, where the directory will be created,
cnt - the length of the directory in entries (the length of the
directory by default is 1 cluster)

Type: External command

Note:
See the description of the configuration command ZOMBIEDRV.

Description:
The main purpose of the command is to create a directory for files to be deleted. The command works in the following way:

1. A new directory is created on a disk with the help of this command. Here you can specify the length of a directory in entries, available in it. It defines size of deleted files queue. It is not recommended to specify too great a number, for it may lead to significant delays in System work. The default size - 1 cluster is enough.

2. The directive ZOMBIEDRV dsk: is written to the CONFIG.SYS file, where "dsk" is a disk where you have created ZOMBIE directory.

Instead of deleting a file, the System moves it into the directory \ZOMBIE, previously created by the given command. When the directory is completely full, the System will delete an old file and write a new one instead. In case of lack of space for new data, old files will be also deleted in the directory. Names in the directory can be duplicated, because the path is not saved during file transition. This can be seen by running the DIR command. If you work with files, the first one will be processed in this case. When running utilities, that check integrity of data on a disk, error messages are likely to appear.

2.2.30 MODE COMMAND

Purpose: Sets the mode of a printing device, color graphic display or asynchronous transmission adapter.

```
Syntax: [d:] [path] MODE <arguments> ----+
                                         |
```

1. Redirection of standard printer output
PRN[:] = <DeviceName>[:]

2. Setting of the printer mode
LPTn[:] [[COLS=]c] [[LINES=]l] [TINY] [[RETRY=]r]
[[CHARS=]c] [QUALITY= DRAFT|NLQ]

3. Setting of the serial port mode

```
            COMn[:] [BAUD=b] [DATA=d] [STOP=s] [PARITY=<ODD|EVEN|NONE>]
```

4. Setting of the display mode

```
            [CON[:]] MONO|CO80|BW80|CO40|BW40| BiosModeNumber
                        COLS  = m (40,80)
                        LINES = n (25,30,43/50)
                        [/B][/V][/N]
```

Setting of the keyboard mode

```
            [CON[:]] RATE=r DELAY=d
```

Type: External command

Description:

Variant 1. Redirects Input/Output from PRN device to the device, specified by the user.

Variant 2. Sets the mode of IBM-EPSON compatible printers, connected to the parallel ports (LPT1 - LPT3).

Parameters:

n Specifies the parallel port number: 1,2,3
c Specifies the character per line: 40,80 or 132
l Specifies the character per inch: 6 or 8
r Specifies the retry action to take when time-out error occurs.
'E' - returns the error, when the device is busy
'B' - returns 'busy', when the device is busy
'R' - returns 'ready', when the device is busy
TINY Sets the minimal size characters printing mode
QUALITY Sets the ordinary print mode - DRAFT or quality print mode - NLQ

Variant 3. Controls the configuration of the serial port adapter.

Parameters:

n Specifies the number of the serial port;1,2,3,4.
b Specifies the transmission rate:
110,150,300,600,1200,2400,4800,9600 bits per second
p Specifies the parity: NONE, ODD, EVEN.
D Specifies the number of data bit: 5, 6, 7 or 8
s Specifies the number of stop bits: 1, 1.5 or 2

The default value is 7 data bit and EVEN.

Variant 4. Specifies the active video adapter and its display mode or resets the current mode of adapter display.

Parameters:

BiosModeNumber Specifies the number of the display mode:
BW40, BW80, CO40, CO80 or MONO.
m Specifies the number of characters per line: 40 or 80
n Specifies the number of lines on the display:
25,30 or 43/50
/B Using default BIOS interrupt
/V Using default VGA interrupt
/N Allows you not to check the display type

BW40 switches a color graphics adapter and specifies a monochrome

display adapter with a display of 40 characters per line.
BW80 switches a color graphics adapter and specifies a monochrome display adapter with a display of 80 characters per line.
CO40 switches a color graphics adapter and specifies color display adapter with a display of 40 characters per line.
CO80 switches a color graphics adapter and specifies color display adapter with a display of 80 characters per line.
MONO initialize a monochrome display adapter with a constant display of 80 characters per line.

Variant 5. Sets keyboard typematic rate.

Parameters:

r - Specifies the typematic interval time. The range for possible values is 1-32.
d - Specifies the auto-repeat start delay time. Possible values are 1-4. (Respectively .25, .50, .75 and 1 second).

2.2.31 MORE COMMAND

Purpose: Reads the data from the standard input device and routes them to the standard output device one screen at a time. Displays the message ---<< MORE >>--- at the bottom of each block and waits for you to press a key.

Syntax: [d:] [path]MORE

Type: External command

Parameters:

[d:] [path] before the command, defines the drive and the path to the MORE command.

Press any key to display the next block of information. Then, repeat pressing the key when you want to read more.

Examples:

1. To transfer the contents of the TEST.ASM file to the standard output device in blocks, type:

```
A>more test.asm
```

When the screen is full, you will see the message: ---MORE--- at the bottom of it. Press any key to display the next screenful of information.

2. With the help of the MORE command the output data of the TREE command will be displayed:

```
A>tree|more
```

2.2.32 NLSFUNC COMMAND

Purpose: Loads country-specific information "on the fly".

Syntax: NLSFUNC [[drive:][path]filename] [/L]

filename - specifies the file containing country-specific information,
/L - loads the resident code into the low memory not using HMA.

Type: External command

Note:

The default name of the file, containing country-specific information is "\COUNTRY.SYS". It is defined by the COUNTRY configuration command.

Description:
The command specifies the resident code, that allows to change dynamically the code page, for example with the help of the internal DOS command CHCP (Change Code Page).

2.2.33 PATH (Set Search Directory) COMMAND

Purpose: Looks through specified directories (subdirectories) to find commands or batch files, absent in the current directory.

Syntax: PATH [[d:]path[[;[d:]path]]]
or
PATH =

Type: Internal command

Description:
In the command you can specify several full paths, separated by semicolons. If you call the command, absent in the specified directory of the specified or the current drive after the execution of the PATH command, all directories (subdirectories), specified in the command will be looked through. DOS looks through directories consequently, as they were specified in the PATH command. The current directory isn't changed.

If you do not specify any options in the PATH command, the current paths will be displayed.

If you type the PATH with an equal sign only (PATH=), it will delete all paths typed before. It is the standard setting, specified when initiating the operating system. In this case DOS looks through only the current directory.

To add a new path to already existing ones you should type:

```
PATH=%PATH%;new_path
```

It will preserve your paths already existing.

Notes:
1. You should set your most often used commands search paths in your Autoexec.Bat file.
2. Errors in the path specification, such as specification of the unexacting drive or extra semicolon will be found just when looking through the given path.
3. If the directory specified in the list doesn't exist, the System ignores it and continues looking through other directories in the list.
4. The command searches only for the executable files: .COM, EXE, BAT.
5. Operating system configuration parameters (environment) are loaded together with resident programs. Resident programs (MODE, PRINT, GRAPHICS) should be executed before the setting of the full path.

Examples:
Lets assume, that the MYPROG.COM command is situated in the MYDIR directory of the drive B. The current drive is the drive A.

The command

```
A>path a:\level2\level3;b:\mydir
```

specifies the paths where the System will search for the programs: first to look through the current directory of the specified drive and then to look through the following directories: A:\LEVEL2\LEVEL3 and B:\MYDIR. If you call the program which cannot be found through the paths specified, the following message will be displayed:

```
Bad command or file name
```

And if you use the command:

```
A>myprog
```

the system will look through 3 specified directories and find the MYPROG program in the B:\MYPROG directory.

If you want to know the current parameters of the command, type:

```
A>path
```

The following information will be displayed:

```
PATH=A:\LEVEL2\LEVEL3;B:\MYDIR
```

After the command

```
A>path c:.;d:\tools
```

DOS will make search in the current directory of the drive C and in the D:\TOOLS directory.

If in a command line a name of a disk is given without specifying the directory (like, "somewhere on a disk"), then the search will involve those PATH string elements, that refer to the disk or have no references to a disk whatsoever.

After the command

```
A:\>D:ONETOOL.EXE
```

the ONETOOL.EXE file will be searched for in the D:\TOOLS directory.

2.2.34 PRINT COMMAND

Purpose: Prints the contents of text and binary files in the background mode.

Syntax: PRINT [file specification] [option list]

Type: External command

Parameters:

/Q - Specifies the number of files allowed in the print queue. The format of the menu string:

```
+------------------------------------------------------------+
|          num. Filespec     options             size        |
+------------------------------------------------------------+
```

num - the queue number of the file;
filespec - file specification with the full path;
options - options list;
size - the number of bytes to be typed.

If you want to delete the file from the queue in this mode, you should move the menu selection bar to the corresponding string and press [Enter].

/D - File printing with suppression of nonprintable characters. Characters which codes correspond to the control characters, are substituted by ".".

/B - To print a file as a hexadecimal dump. The format of the printed information will be the following:

a colon of hexadecimal codes (16 bytes per string)	a colon of the corresponding text characters

/P{first byte}[-last byte] or /L{first string}[-last string]
 - printing of the specified file fragment, from line to line or from position to position.

/U - readdress the output. Here before processing other parameters, the system prompts you with the inquiry:

Input device name [PRN]:

after that you should specify the printing device name. The default device (press [Enter]) is a standard printing device PRN. For input yo can designate both character device and disk file. If you use PRN for printing, initial setup is not needed.

/T - sets the mode of printing termination. All files in the printing queue are deleted. If some file is being printed at this moment, the printing will be stopped. The shift to a new page will take place.

/C - deletes the specified file in the printing queue.

Each parameter has effect on file after which it is situated and then on all the following files, until the next parameter appears in the command line.

The module will stay resident in the memory only if the file for printing will be specified. In this case the following message will be displayed:

Print spooler installation ...

After the normal installation of PRINT.COM resident part the following message is displayed:

Print spooler is installed

after that the file, specified in the command line, is put in the queue. For further spooling control (see the list of options), you should execute PRINT.COM program with appropriate command line.

After printing of every file, the printing device paper moves to the beginning of a new page.

If the system detects an error while reading the file to be typed, the PRINT command will:
- delete this file from the printing queue;
- move paper to a new page;
- type other files from the printing queue.

If /T and /C parameters are used in file deleting, the following actions are performed:
- shift of paper to a new page,
- if not all files were deleted from the queue, the printing process goes on, beginning with the first file in the queue.

Notes:
1. The disk with files in the printing queue should stay in the specified drive till the end of the printing.
2. You should not use the printing device for another task fulfillment, until the PRINT command finished to print all files in the printing queue. Any attempt to use the printing device (Shift-PrtSc,

LLIST, LPRINT, Ctrl-PrtSc and etc.) will cause the system to prompt you with the following message: "out-of-paper", until all files are printed or the PRINT command with the parameter /T is executed.
3. You cannot use the PRINT command in a computer that serves for a network.
4. The PRINT command changes global characters in a filename and gives the full filename, including the path and the drive. The result string can contain up to 63 characters.

Examples:

1. The PRINT command is used for the first time after the operational system run:

```
A>print a:temp1.tst
```

DOS prompts:

```
Name of list device [PRN]:
```

If you press the [Enter] key, the data to be typed will be aimed to the printing device. Then the file TEMP1.TST from the drive A: is set in the printing queue.

2. The command:

```
A>print /t
```

terminates printing. All the files are deleted from the queue. Any other information in the string will be ignored.

3. The command:

```
A>print temp.*/c
```

flushes all files TEMP.??? from the printing queue, if they are on the disk in the default drive.

4. The command:

```
A> print a:temp1.tst/c a:temp2.tst a:temp3.tst
```

deletes three files from the printing queue: TEMP1.TST TEMP2.TST and TEMP3.TST on the disk in the drive A:.

5. The command:

```
A>print temp1.tst/c temp2.tst/p temp3.tst
```

deletes TEMP1.TST from the printing queue and sets TEMP2.TST, TEMP3.TST in the printing queue.

6. The command:

```
A>print temp1.tst temp2.tst temp3.tst/c
```

sets TEMP1.TST, TEMP2.TST in the printing queue and then deletes TEMP3.TST from the printing queue.

2.2.35 PROMPT (Set System Prompt) COMMAND

Purpose: Sets a new system prompt.

Syntax: PROMPT [text of a new prompt]

Type: Internal command

Description:
After the command you can specify the text of a new prompt. This text can contain special sequences, marked by $c, where "c" may be any of the following:

$	dollar sign
t	time
d	date
p	current directory of the default drive
v	version number
n	default drive
g	">" character
l	"<" character
b	":" character
q	"=" character
h	backspace
e	ESC code
-	CR/LF sequence (carriage return/line feed)

If you do not specify any parameters in the PROMPT command, the current prompt will be displayed.

Examples:

1. The system prompt will be the default drive letter and the ">" character:

```
A>promt $n$g
```

2. A new system prompt will be the sequence "HELLO":

```
A>prompt HELLO
```

3. The system prompt changes for the current directory name in the default drive and the ">" character:

```
A>prompt $p$g
```

Note:
Including this command in the AUTOEXEC.BAT file allows to get the information about the current directory (every time the System is started). If the current directory of the drive A is the LEVEL1 directory, the system prompts you with the following:

```
A>\level1>
```

4. The command may set a prompt as following:

```
Time = (current time)
Date = (current date)
```

```
A>prompt Time = $t$_Date = $d
```

5. From the PROMPT command you can use the control sequences of the ANSI.SY file. If the CONFIG.SYS file contains the DEVICE=ANSI.SYS command during the start-up of the system, then, after executing:

```
A>prompt $e[s$e[24A$p$e[u-
```

the current directory will be displayed at the first line of a screen, and the "-" character - at the next line. The "A" character should be a capital letter in this sequence.

Notes:

1. Control sequences will work properly only if the ANSI screen driver has been installed (SCREEN=ANSI in the CONFIG.SYS or CONFIG.PTS configuration files)
2. The environment of the operating system is loaded into the memory together with the resident programs. To economize memory you should execute the resident programs (MODE, PRINT, GRAPHICS) before the long prompts string setting (see also SET command description).

2.2.36 RDZOMBIE COMMAND

Purpose: Deletes the ZOMBIE directory in the specified drive.

Syntax: RDZOMBIE drv[:]

where 'drv' is the drive where you want to delete the ZOMBIE directory.

Type: external command

Note:
See the description of the external DOS command MKZOMBIE and the configuration command ZOMBIEDRV.

Description:
To delete the ZOMBIE directory perform the following:

1. If the command ZOMBIEDRV is specified in CONFI.SYS for the desired disk, you should cancel this command in config.sys file or change its argument to another disk, reboot system to force the system to recognize the new config.sys file.

2. After that you should run the RDZOMBIE command, specifying the disk where ZOMBIE directory is to be deleted.

2.2.37 RMDIR (Remove Directory) COMMAND

Purpose: Deletes directory on a specified disk.

Syntax: RMDIR [d:] path
or
RD [d:] path

Type: Internal command

Description:
Only an empty directory can be deleted. Directories that contain hidden files may not be deleted. Only the last directory in a path can be deleted.

Example:

```
A>rd b:\level1\level2
```

This command deletes the subdirectory LEVEL2 from the directory LEVEL1 of the drive B:.

Note:
Be careful using the RD command on drives used in JOIN and ASSIGN commands. You cannot delete the directory substituted via the SUBST command.

2.2.38 RENAME (or REN)

Purpose: Changes names of files and directories specified by the first argument for names specified by the second argument.

Syntax: REN[/s] [d:][path1][filename1] [path2][filename2]

Type: Internal command

Note: Source and target disk mast be the same.

Description:
Instead of RENAME you can use short variant REN. It is allowed to use characters “*” and “?” in argument.

The/S switch specifies the work with subdirectories.

Examples:
The command: A>ren b: abode home
renames the ABODE file in the drive B: to HOME

The examples: A>ren b: abode *.xy
renames the ABODE file in the drive B: th ABODE. XY

The command: A>ren b:\level1 b:\back\level2 renames the directory \level1 with subdirectories to the directory \back\level2 in the drive B.

2.2.39 REPLACE COMMAND

Purpose: Replaces specified file(s).

```
Syntax: REPLACE [/U[PDATE]] [/N[EW]] [/P[ROMPT]] [/Q[UIET]]
        [/R[EVERSE]]
        source
        target
```

```
source ::= FILELIST
target ::= FILELIST
```

Type: Internal command

Parameters:

/P[ROMPT] - prompts for users confirmation for replacing a file.
/Q[UIET] - suppresses output of diagnostic messages to the screen.
/U[PDATE] - replaces only those files, that were are created later than target files.
/R[EVERSE] - reverses the action of the /UPDATE switch and is ignored if /UPDATE is absent.
/N[EW] - copies only the files which are absent in the target.

If neither /U nor /N switch is included in the command line, the system assumes they are both present.

```
FILELIST ::= [parms]
        {
        [d:][path]file_name[.ext] |
        (FILELIST[,|+| ]...) |
        @reference_file
        }
```

FILELIST - defines the files to be replaced.
(See INTERNAL DOS COMMAND ARGUMENTS for detailed description)

file_name may contain wild card symbols ("?" and "*").
reference_file is the name of a text file containing the command's parameters. It should be specified explicitly.
Parms are any combination of the control parameters for FILELIST or DIRLIST (see "INTERNAL DOS COMMAND ARGUMENTS").

Examples:
1. Deleting the FILE.BAT file from a diskette in the A: drive:

```
A:\>REPLACE A:\FILE.BAT B:\
```

2. Using the prompt option:

```
A:\>REPLACE /P file?.txt B:\
```

If the /P (prompt) switch has been set, then the user will be prompted before replacing any file:

```
FILE1.TXT replace this file {Y/n/All yes/Quit}? _
```

The user should reply with one of the following symbols and press [Enter] afterwards. Pressing [Enter] alone is equal to replying "Y".

"Y" - the file is replaced and the interpreter moves on to the next one.
"N" - the file is NOT replaced and the interpreter moves on to the next one.
"A" - the file is replaced and all the remaining files will be replaced with no prompt whatsoever.
"Q" - quitting the command.

3. Using the subdirectories option:
The following command replaces all the *.asm files located an all subdirectories of the C: disk:

```
A:\>REPLACE /S C:\*.ASM A:\
```

WARNING! Improper or mischievous usage of the DELETE command may lead to loss of data on your disks! Paragon Software bears no liability whatsoever if such a damage or loss occurs.

2.2.40 SET COMMAND

Purpose: Allows to add a new character string to the environment of the COMMAND.COM. All strings of the COMMAND.COM environment become available to all commands and users' programs.

Syntax: SET [name=[parameter]]

Type: Internal command

Description:
The full string (beginning with a name) is written into the memory block, reserved for the strings of the environment. During writing into the environment all small letters of the string are changed into capital letters; the remaining part of the string is copied without changes. If the name already exists in the environment, then its value changes to a new value of the parameter.

If the SET command is specified without a name, then all strings of characters, being in the environment at the given moment will be displayed.

If you specify the name in the SET command but without parameter, the last occurrence of the string "name=parameter" will be deleted from the environment.

The environment (a set of names with parameters) is available for all DOS commands and users' programs.

Notes:
1. Having found any PROMPT or PATH command, DOS writes them automatically into the environment.
2. One of the strings of the environment (written there after the DOS system initialization) is the string COMSPEC=argument. This parameter describes the path used by DOS for the second copy of COMMAND.COM, if necessary.

3. A copy of the environment is stored after the execution of any resident program. Resident programs (MODE, PRINT, GRAPHICS) should be called before writing of long character strings of the environment.

Examples:

1. A>set pgms=\level1

This example sets the string PGMS=\LEVEL1 in the environment. Having got the control, the application can find the name PGMS in the environment and use the parameter as the directory of its files.

2. A>set pgms=

deletes the string PGMS=\LEVEL1 from the environment.

3. A>set abc=xyz

sets the string ABC=xyz to other existing in the environment strings. Pay more attention to the change of small letters by capital ones. In this way you can specify keywords and parameters, that have no meaning for DOS, but can be used by application.

2.2.41 SETDRV COMMAND

Purpose: Installs a driver specified by the user into the system.

Syntax: SETDRV [/U] DriverName DriverParams

Type: External command

Parameters:

DriverName - the full driver name,
DriverParams - parameters to be passed to driver,
/U - instruction to try to load driver in UMB (if it is possible).

2.2.42 SETVER COMMAND

Purpose:

Displays resident database table of the faked DOS version versus executable filename to provide a definite version required by old-executable program.

Syntax:

```
[d:][path] SETVER [.EXE] [/?]
```

Type:

External command

Description:
SETVOR utility is designed for purpose of overriding come old-style programs that behaviour is tightly closed to the version number. These programs require a definite DOS version only though they can work under upper version of DOS. This utility fakes the DOS version to be comliant with such ill-behaved program requirements. Mostly in PTS-DOS SETVER utility to be used to obtain apropriate DOS versions for drivers cause for usual executables one can change the DOS version by using VER command of PTS-DOS COMMAND.COM shell. Suppose we have a program PROG.EXE that requires DOS version to be 5.0 only and refuses to execute under another DOS version. To fake the DOS version one can type in the following lines:

Command	Comment
>VER 5.0	This fakes DOS version to be 5.0
>PROG.EXE [params]	Run our program
>VER /R	Restore the true DOS version

This method is perfect for one-time run of the program DROG.EXE. For the running PROG.EXE every time it is more convinient to store its name to the SETVER database and load SETVER as a device driver from the CONFIG file by typing info this file the following line:

```
Devicehigh=[path] SETVER.EXE
```

Note that this is the only way to change DOS version for the device drivers loaded after the SETVER driver.

Database with the filenames version numbers is in the PTSDOS.INI FILE in the SETVER directory. Data base is to be placed in the SETVER section of this file. Every entry in this section must occupy one line only. Entry consists of two fields: identification field and information filed. The following table contains valid identificators and information types for the corresponding identificator:

IDENTICATOR	INFORMATION
VERSIOON	DOS version in the form N.MM N – major version number MM – nimor version number
PROGGRAM	Executable program or device driver filename and DOS version number (optional)

VERSION information field sets a DOS version number by default to pass it to the PROGRAM entry that does not have the latter. By default the DOS version is5.00

PROGRAM information field contains executable filename only (not including the path) and after the standard delimiter (comma or a space) optional DOS version number. If the latter didn't specified explicitly then the last VERSION statement result to use. Filename should have an extension for there is no assumption applied to the executable filename.

If any field of database entry in the PTSDOS.INI file is invalid then the whole line will be ignored.

To edit the PTSDOS.INI file you may use any text editor you want. Just be save this file when exiting text editor.

IMPORTANT: For the changes in the PTSDOS.INI file to take effect one should reboot with SETVER driver load line in the CONFIG file.

Finally, there is an example of the PTSDOS.INI file SETVER section shown below:

```
----------------------Contents of PTSDOS.INI file-------------------------------------------
[SET VERSION UTILITYY]
VERCION=6.0
PROGRAM=NETX.EXE

VERCION=5.0
PROGRAM=KERMEL.EXE
PROGRAM=NETX.COM
//There are explicit DOS version specs
PROGRAM=WINWORD.EXE      4.10
PROGRAM=EXCEL.EXE        4.10
PROGRAM=LL3.EXE          4.01
//For this line DOS version is still 5.0
PROGRAM=NET5.COM
```

```
VERCION=4.0
PROGRAM=REDIR4.EXE
```

Example:
Information will be displayed in the form of a table consisting of the filenames and associated DOS version to fake. The typical SETVER utility output looks like this:

```
The resident SETVER database contains the following entries

          VERSION FILENAME



------------------------------------------------------------------------------------------------------------------------
NETX.EXE       6.0      BNETX.EXE      6.0      EMSNETX.EXE 6.0      EMSNETX.EXE 6.0
EXTDISK.SYS    6.0      KERNEL.EXE     5.0      NETX.COM       5.0      NET5.COM       5.0
BNETX.COM      5.0      EMSNET5.EXE  5.0      XMSNET5.EXE  5.0      DOSOAD.SYS     5.0
REDIR50.EXE    5.0      REDIR5         5.0      REDIALL        5.0      REDIRNP4.EXE 5.0
EDLIN.EXE      5.0      BACKUP.EXE     5.0      JOIN.EXE       5.0      RECOVER.EXE  5.0
GRAFTABL.COM5.0      LMSETUP.EXE  5.0      STACKER.COM 5.0      NCACHE.EXE     5.0
NCACHE2.EXE  5.0      IBMCACHE.SYS5.0      XTRADRV.SYS  5.0      2XON.COM       5.0
WINWORD.EXE4.1      EXCEL.EXE      4.1      LL3.EXE        4.01     REDIR4.EXE     4.0
REDIR40.       4.0      MSREDIR.EXE  4.0      WIN200.BIN     3.4      METRO.EXE      3.31
------------------------------------------------------------------------------------------------------------------------
```

2.2.43 SHARE COMMAND

Purpose:

Sets up support for sharing files on a PC networking system. This is an external command. It is a TSR and may be loaded via INSTALL= and it may be loaded high.

Use to:

Enable network-aware software to “lock” files and/or records so that they may be accessed by only one user at a time.

Provide some protection against an invalid disk-swap on AT-, laptop, and PS/2-compatible systems.

Enable DOS to support old programs which use “FCB-style” file access on disk partitions greater than 32 Megabytes.

Syntax:

```
                    Share [/F: fbytes] [/L: locks]
          or:       INSTALL= [/F: fbytes] [/L: locks] (in the Config. Sys file)
```

Remarks:

Loads the code to handle the DOS low-level function 5cH and allocates space for keeping track of files and locks.

> /F: fbytes sots the amount of space in bytes that will be available to keep track of open files. The default is 2048 bytes. Each entry in the Shere file table uses the length of full filename plus 11 bytes.
> /L: locks sets the number of locks. The default is 20.

On diskette drives which have “change line” support, Share installs some protection against a user swapping diskettes during a critical operation (e.g., after writing to the directory, but before updating the FAT). It checks the volume label and prompts for the original diskette.

Example:

Share /F: 1024 loads network file and record locking support, and allocates space to maintain a total of 20 locks (the default) on about 30 files.

2.2.44 SORT COMMAND

Purpose: Reads data from the standard input device, sorts them alphabetically and sends this data to the standard output device.

Syntax: [d:] [path] SORT [/R] [/+n]

Type: External command

Parameters:

[d:] [path] before the command defines the drive and the path to the SORT command.
/R reverses the sort order. That is, sorts from Z to A.
/+n is the number of the column, from which the sorting should begin (the whole number). If you do not specify this parameter, the SORT command will begin sorting from the first column. The size of the file to be sorted should not exceed 63K.

The input filename must differ from the output one.

Notes:
1. Uppercase letters "a-z" are identical to lowercase ones "A-Z".
2. Characters above ASCII 127 are sorted depending on information specified by the COUNTRY command.

Examples:

1. A>sort /r <unsort.txt>sort.txt

reads the UNSORT.TXT file, sorts it in reversed order and sends it to the output device under the name SORT.TXT.

2. A>dir | sort /+14

In this example the output data of the DIR command is the input data of the SORT command. After the sorting the listing of the directory takes place starting with column 14 (it contains the file size). The output file of the SORT command is transferred to the standard output device.

2.2.45 SUBST (Substitute) COMMAND

Purpose: Allows to substitute a drive letter or a path for a one-letter drive identification.

Note: If the application doesn't recognize paths, the SUBST command allows to use drive letter instead of the path.

Syntax: [d:] [path] SUBST d: d:path

Type: External command

Parameters:

[d:] [path] before the command defines the drive and the path to th SUBST command.

d: - drive letter to be used instead of the other drive letter or the path.
d:path   - the drive letter or the path you want to access. The specified path should begin in the root directory.
/D   - canceling of the substitution. You should specify the letter of the drive you want to abort. For example, if you substitute the path C:\LEVEL1 for G, the substitution will be cancelled by the command:

```
A>subst g: /d
```

The substitution letter may range from A to Z. It is used as a petname for the disk or the long path to call. For example, if G is used to substitute the long path C:\LEVEL1\LEVEL2, you may call C:LEVEL1\LEVEL2\file1 just typing:

```
A>dir G:file1
```

instead of

```
A>dir C:\level1\level2\file1
```

Notes:
1. The letter of the drive you can use for the substitution, depends on the meaning of the LASTDRIVE in the environment. If the LASTDRIVE parameter is absent in the CONFIG.SYS file, the default value is LASTDRIVE=E, so you may use for substitution only A, B, C, D, and E letters.
2. Drive letters in the command parameters should differ.
3. The default drive letter cannot be the first parameter.
4. If you specify the path, it would begin in the root directory.
5. You would not use a network disk as a parameter. Otherwise, the following message will be displayed: "Cannot SUBST a network drive".
6. Be careful executing CHDIR, MKDIR, RMDIR and PATH commands on drives used in the SUBST command.
7. The following commands do not work on drives used in the SUBST command: ASSIGN, BACKUP, DISKCOMP, DISKCOPY, FDISK, FORMAT, JOIN, LABEL, and RESTORE.
8. The SUBST command without any parameters displays this current substitution: **G: => D:\path**
9. The Invalid Parameter message means that:
- the specified drive letter isn't valid;
- there is an attempt to substitute the default drive;
- the parameter /D was specified without the drive letter.
10.If you specify an unexacting path, the system will prompt you with the message: Invalid Path.

Example:
Lets assume that, the default drive is B, the current directory is
C:\REPORTS\FILES, LASTDRIVE=H.

```
                 +--------------------+
                 |          C:\       |
                 +---------+---------+
                           |
        +-----------------+----------------------+
        |                                        |
   +----+-------+                        +--------+--+
   | REPORTS |                           |  GAMES  |
   +----+-------+                        +------------+

   +-----+-+
   | FILES |
   +--------+
```

1. How to substitute:

```
B>subst g: c:\reports\files
```

substitutes the path C:\REPORTS\FILES for the drive letter G. After the substitution the command

```
B>dir g:
```

displays the directory C:\REPORTS\FILES. Remember, the drive letter cannot be other than any of those allowed by the LASTDRIVE command.

2. How to display the current substitution:

```
B>subst
```

This command displays the following message:

```
G: => C:\REPORTS\FILES
```

The message means, that disk G substitutes the directory C:\REPORTS\FILES, i.e. when you call the disk G the command calls the directory c:\REPORTS\FILES.

3. How to cancel the substitution:

The following command cancels the current substitution:

```
B>subst g: /d
```

How to use the SUBST command:

The substitution of the drive letter for a path is useful with applications, that do not recognize paths. For example, if you want to edit the RECIPES file from the directory C:\FOOD\FILE, you can substitute the path for the drive letter F. After this you can call the file in this way: F:RESIPES. Remember, the substitution is possible for LASTDRIVE=F at least.

2.2.46 SYS (System) COMMAND

Purpose: Transfers the operating system files: PTSDOS.SYS, PTSBIO.SYS (and COMMAND.COM) to a specified disk, making it bootable.

Syntax: SYS drive: [path][/u[:bootfile]][/p][/s][/c]

Type: External command

Note: The SYS command cannot be used with a network drive.

Parameters:

drive: defines the drive you want to transfer system files to.
[path] defines the path to the system files (and to the COMMAND.COM, if /c switch is set)
bootfile defines the filename to transfer the contents of the boot sector to before the installation of the system. This file can be used by the SYSBOOT configuration command. The default filename is .\OLDBOOT.SYS. Setting the /U switch renames an old COMMAND.COM to COMMAND.Co$ (See UNINSTALL).
/p make pauses between read/write operations. It is very important for system transition using a single drive.
/s do not copy COMMAND.COM.
/c do not use the COMSPEC= environment variable to search for the COMMAND.COM file.
/u installs PTS-DOS and saves boot-sector of the system, previousely installed on the disk. Provides the possibility to choose between several operating systems when loading, makes it possible to restore the system, installed before PTS-DOS.

2.2.47 TIME COMMAND

Purpose: Allows to enter or change the time measured by the system. If you create a new file or change the existing one, the system time is registered in the directory. The time can be changed from the keyboard or from a batch file.

Syntax: TIME [/Q[UIET]] [hh[:mm[:ss[.xx]]][A|P]

Type: Internal command

Parameters:

hh - a number defining hours (the range is 0-23),
mm - a number defining minutes (the range is 0-59),
ss - a number defining seconds (the range is 0-59),
xx - a number defining hundredths of a second (the range is 0-99).

A or P - a.m. or p.m., respectively.

/Q[UIET] - doesn't prompt the user to enter the time.

Notes:
1. If the TIME command has no specified parameters, the following message will be displayed:

```
Current time is hh:mm:ss.xx
Enter new time:_
```

If you do not want to change the time, press the [Enter] key. To change the time, type the new time and press [Enter].

2. Hours, minutes, and seconds should be separated with a semicolon (:) or a dot (.). Hundredths should be separated with a separator defined in the country-dependent information file.

3. If the time is valid, the System will adopt it. Otherwise, the following message will be displayed:

```
Invalid time
Enter new time:_
```

4. If the data you entered are not complete, for example only hours and minutes, the rest parameters will be omitted by default.

5. Every time value will be accepted by the system, if the specified numbers are in the mentioned ranges.

6. The TIME command format can be changed by the COUNTRY command in the system configuration.

Example:
The command:

```
A>time

Current time is 00:25:16.65
Enter new time: 13:55_
```

Sets the following time: 13:55:00.00.

2.2.48 TREE COMMAND

Purpose: Displays the path of each directory and subdirectory on the given drive and lists the contents of each directory.

Syntax: TREE [d:][path] [/F[ILES]] [/A[SCII]] [/M[ORE]]

Type: Internal command

Parameters:

[d:] defines the drive, whose directories should be listed. If the drive is not specified, the default drive is assumed.
[path] defines the directory, whose subdirectories should be listed. If the directory is not specified, the default directory is assumed.
/F allows to display the list of all filenames.
/A allows to use characters with < 127 code to display the subdirectory tree.
/M displays data one screen at a time.

Examples:
1. A>tree c:

displays all directories of the drive C:

```
C:
+-CD-ROM
+-DOS
+-EXE
+-NWWS
+-DOSODI
```

2. A>tree /a c:

displays all directories of the drive C, using ASCII characters only.

```
C:
+-CD-ROM
+-DOS
+-EXE
\-NWWS
\-DOSODI
```

3. A>tree b: /f > b:\tree.lst

writes the list of all directories of the drive B into the TREE.LIST file, situated in the main directory of the drive B. This file will contain names of all directories and filenames of each directory. Contents of the TREE.LST file (example):

```
There are no SubDirectories
mess
chkdsk.txt
config.txt
```

2.2.49 TYPE COMMAND

Purpose: Displays the contents of the specified file(s) on the screen.

Syntax: TYPE [/M[ORE]] [/U[NWRAP]] [/B[IN]] FILELIST

Type: Internal command

Description:
The displayed information is not formatted, except for tabs.

/B - displays a file as a dump,

/U - prevents word wrap at the screen board.
/M - displays data one screen at a time.

```
FILELIST ::= [parms]
            {
            [d:][path]file_name[.ext] |
            (FILELIST[,|+| ]...)      |
            @reference_file
            }
```

FILELIST - defines the files to be typed.

file_name - may contain wild card symbols ("?" and "*").
reference_file - is the name of a text file containing the command's parameters. It should be specified explicitly.
parms - are any combination of the control parameters for FILELIST or DIRLIST (see INTERNAL DOS COMMAND ARGUMENTS for detailed description).

Notes:
1. To print the displayed file use the redirection of the standard output printing device.
2. Printing is completed once the EOF symbol (ASCII 26) has been read from the input file; this doesn't apply for the binary dump mode (/BIN switch).

Examples:

1. A>type /b mess

displays the MESS file's contents in binary form

```
-------- Contents of MESS
00000000: 48 65 6C 6C 6F 20 57 6F 72 6C 64 21 0D 0A 1A     Hello World!...
```

The first line contains the filename, while all the others contain information from the file in the following format: hexadecimal displacement relative to the beginning of the file; hexadecimal values of the 16 bytes; the same bytes but represented as ASCII symbols (symbols lying outside the 32-127 range are substituted with dots).

2.2.50 UNINSTALL COMMAND

Purpose: Restores the booting of a system installed before PTS-DOS on the disk and restores its the boot sector.

Syntax: UNINSTALL filename [drive:]

Type: External command

Note:
This command also renames the file <drive:>COMMAND.CO$ to COMMAND.COM, where, drive is the command parameter.

Parameters:

filename - full name of a file containing the previous boot sector image prior to installing the PTS-DOS. (May be created with the SYS command).
drive - drive to be restored.

2.2.51 VER (Version) COMMAND

Purpose: Displays the operating system version number or modifies it.

Syntax: VER [version|/R]

Type: Internal command

Description:
The version number consists of a digit defining the main version number, a dot, and a two digits defining the revision number.

Parameters:

version - the version number you want to set,
/R - sets the actual DOS version number.

Examples:
1. A>ver

This command displays the data:

DOS version 5.00

2. Setting another DOS version may be achieved by typing:

A:\>VER 4.10

The change will immediately appear on the screen:

DOS Version 4.10

3. To restore the original DOS version, type

A:\>VER /R

Modifying the DOS version number is necessary for running certain programs that were originally tied to a particular version of DOS, but still can operate under later versions as well. The GetDosVersion DOS function (int 21h/AH=30h) returns the value set by the inerpreter, while the GetTrueDosVersion (int 21h/AX=3306h) returns the correct value.

2.2.52 VERIFY COMMAND

Purpose: Verifies the data on the disk.

Syntax: VERIFY [ON:OFF]

Type: Internal command

By default: VERIFY OFF

Description:
If the verify switch is ON, DOS verify the data after every write operation. The verification takes time, that's why the rate of the write operations decreases.

The VERIFY command without parameters displays the current status of program (ON or OFF).

Notice:
VERIFY does not verify the data written to the network disk.

2.2.53 VOL (Volume) COMMAND

Purpose: Displays the volume label for the specified drive.

Syntax: VOL [d:]
where d: is a drive.

Type: Internal command

If the drive is not specified, the default drive is assumed.

Examples:

1. A>vol

displays the volume label for drive A:. The following information will be on the screen:

```
Volume in drive A is MYDISK
Volume Serial Number is 3ABC-0007
A>
```

or, if there is no label on this volume:

```
Volume in drive A has no label
Volume Serial Number is 353D-17ED
```

3 BATCH FILES OPERATORS

3.1 OPERATORS

3.1.1 CALL Command

Purpose: Calls up processing of a batch file with extension different from. BAT, if it is specified explicitly. Otherwise a file with the extension .BAT is searched for.

Syntax: CALL batch-filename [arg_list]

Parameters:

batch-filename	- is the name of a bath file,
arg_list	- is the list of arguments.

The filename may contain wild card symbols ("?" and "*").

3.1.2 CHOICE Command

Purpose: Waiting for the user to choose one of the options.

Syntax: CHOICE [/C[:]options] [/N] [/S] [/T[:]c,nn] [comment] [/?]

Parameters:

/C[:]options	- defines the list of options. By default: Y, N;
/N	- doesn't display the list of options;
/S	- case sensitive search;
/T[:]c,nn	- considers the C option chosen after NN seconds.
Comment	- any users comment.

The system variable ERRORLEVEL contains the number of option chosen by the user.

3.1.3 ECHO Command

Purpose: The ECHO command is used to display users messages or to control system messages during batch file processing.

Syntax: ECHO [ON|OFF]
or
ECHO [message]
where [message] is the message desired.

Parameters:

. (dot) - line feed (if it is placed just after the ECHO command),
ON - switches on the display mode for interpreter,
OFF - switches off the display mode for interpreter,
Message - printing of this message and moving to the next line.
/? or /H or /HELP - help.

Excample:

The command:

ECHO Have a nice day!

will display "Have a nice day!".

After reading the batch file and executing all substitutions the command interpreter displays a system prompt (see PROMPT command) followed by a ready-to-execute string. This can be prevented by using the

ECHO OFF

command, that surprises the readout of system prompts and batch file commands for the given file.

To suppress the readout of any string, you may put a special symbol "@" at the beginning of this string. Although the command line isn't displayed, it doesn't have any influence on the readout of internal messages, arising during the execution of this command.

To suppress the readout of the whole batch file use

@ECHO OFF

command in the initial string of the batch file is a standard procedure.

The ECHO command without parameters will show the current ECHO status.

To restore the normal display activity, use the command:

ECHO ON

3.1.4 ECHONLF Command

Purpose: The ECHONLF command is used to display users messages or to control system messages during batch file processing. The command performs the same operations as the ECHO command but without line feed.

Syntax: ECHONLF [ON | OFF]
or
ECHONLF [message]
or
ECHONLF [/?][/H[ELP]]

Parameters:

ON - switch on the display mode for command interpreter,

OFF - switch off the display mode for command interpreter,
Message - print this message and move to the next line,
/? or /H or /HELP - help.

3.1.5 FOR Command

Purpose: Performs the FOR cycle.

```
Syntax: FOR  %var IN  (LIST)  DO  COMMAND
            or
            FOR [/?][/H[ELP]]
```

Parameters:
Var - one-letter variable (a-z or A-Z),
LIST - list of arguments with standard separators,
COMMAND - DOS command. It can contain the string "%var", that will be substituted for a value from the LIST.
/? or /H or /HELP - help.

Note: The LIST elements are considered to be files or directories first, and then a text.

3.1.6 PAUSE Command

Purpose: Makes a pause in a batch file processing, displays a message, and waits for users command to continue work.

Syntax: PAUSE [message]

Example: Pause Insert a new disk and press any key.

It is reasonable to place the PAUSE command with appropriate messages between the critical parts of the program, so that you could abort the batch file during the pause by pressing [Ctrl-Break] and make necessary corrections.

Note:
Your message will not be displayed if the readout is surprised by the ECHO OFF command. In this case you should specify your message in the string preceding to the pause by the ECHO command.

3.1.7 REM Command

All the strings which begin with REM are considered to be comments. These comments may contain, for instance, the information about the purpose of the command or its part. You also may put REM before any string to nullify its effect.

Syntax: REM [text]

The comment strings, like any strings in a batch file, are displayed on the screen during the execution. To prevent displaying the string, put the @ sign in its beginning.

```
REM this remark is displayed
@REM this remark is not displayed
```

3.1.8 GOTO Command

To change the order of processing command lines in a batch file a system of labels and conditional and unconditional transfers is used.

Batch file label is any sequence of symbols following a colon. Any lines after a colon are ignored during direct consecutive processing. For example: ... echo Hello rem (the following line will be ignored during processing): :Label ...

Unconditional jump in a batch file is used when the user wants to change the order of the command lines processing.

This is achieved by using the GOTO command containing a label name as an argument. Example:

```
...
:Label
pause Press Ctrl-C to exit, otherwise
GOTO Label
...
```

Here the execution of the batch file is looped by GOTO command. When executing PAUSE command the command interpreter asks the user to choose between suspending execution of the batch file and proceeding with the cycle.

3.1.9 IF Command

If the order of batch file processing depends on external conditions then command IF (CONDITIONAL transfer) is used.

Syntax: IF <condition> <command>,

where condition is one of circumstances described below; command is any DOS command to be executed if the condition is true. Otherwise execution of the batch file resumes starting from the line immediately following that containing IF command.

Using semantic notation we can make the following sequence:

```
condition::=[NOT] {string<rel>string|
ASK [YES:NO] <message>      |
ERRORLEVEL[<rel>]<number>   |
EXIST[{ALL:ANY}]<FILELIST>}
```

In other words, the condition is a string like one of the strings above. It can be preceded by NOT instruction which inverts its meaning.

The conditional operator used in batch files permits to use only one condition at a time. A number of conditions in logical operations can be brought together only by way of using several IF operators in a row. For example, operation AND between condition1 and condition2 can be made possible through writing a conditional operator as a command part of another. In other words, the string IF <condition1> IF <condition2> <command>

will execute AND operation.

Evidently, OR operation can be made possible by two consecutive IF operations located in different strings. The following strings do the OR operation between condition1 and condition2:

```
IF <condition1> goto cond_met
IF <condition2> goto cond_met
...... - strings to be processed if the conditions are not true.
:cond_met
...... - strings to be processed if at least one of the conditions is true.
```

See for each of the condition types in detail:

1. string1<rel>string2
2. ASK [{Y[es]|N[o]}] <message>
3. ERRORLEVEL [<rel>] <number>
4. EXIST [{ALL|ANY}] <FILELIST>

3.1 10 string1<rel>string2

Here "string1" and "string2" represent symbol sequences constituting so-called "character string". A character string can begin with any symbol except space bar, tabs and end of line. Character string ends in case blank space, tab symbol or end of line are encountered. But if the string starts with opening bracket or quotation-marks ("), then blank spaces and tab symbols can be used within the sstring which terminates with a closing bracket or quotation-marks.

E.g.:

```
string
"another string"
(last string)
```

Here the IF conditional executes symbol-by-symbol string comparison. <rel> is a relation between strings:

<rel>::={EQ:NE:LT:GT:GE:LE:==},

which means correspondingly: "EQual", "Not Equal", "first is Less Than second", "first is Greater Than second", "first is Greater than or Equal to second", "first is Less than or Equal to second", "both are equal" (the latter has been preserved for the compatibility with other DOS batch file syntax's).

A string is considered to be greater (less) than another if the initial differing symbol in this string weighs more (less) than the corresponding symbol in the other string (symbol weight is taken from DOS code table for a given code page). In other words, in sorting strings alphabetically the greater string will be located later than the less one.

Opening and closing brackets and quotation marks are considered to be value symbols participating in comparison.

If strings differ in length then for comparison the shorter one i elongated with zero weight symbols until it becomes as long as the other.

Examples:

```
IF /s==/q echo Nonsense!
IF NOT "/s" eq "/q" echo That's right!
..........
IF "greater"LT"less" echo It's O'k.
```

3.1.11 ASK [{Y[es]:N[o]}] <message>

This condition asks the user to directly interfere with the order of batch file processing. During processing a <message> is displayed on the screen containing a standard answer. If it is confirmed by the user (Yes), the condition is considered to be true, if rejected (No) - not true.

Non-obligatory parameter {Y[es]:N[o]} sets user's default answer. If the answer is "yes" then pressing [Enter] is considered to be "Yes" automatically.

You should write your <message> so that the responder could clearly understand the condition meaning.

Example:

```
IF ASK YES "Are you sure" goto1
exit
:1
.....
```

3.1.12 ERRORLEVEL [<rel>] <number>

In PTS-DOS environment any external (for the Shell) command has a so-called exit code - a number from 0 to 255. This code enables external components of the executed program to detect errors made during the program run and return error messages.

As a rule, in case of a successful run the exit code is equal to 0, and in case of an error - not equal to 0. The program terminated by the user (to prevent a critical mistake, by pressing [Ctrl-Break] or [SysReq], etc.) has the code equal to 255. But this does not mean that any program with exit code 255 has been expressly terminated by a user.

Usually first rate packages exit codes are described in detail in the accompanying documents.

Thus, ERRORLEVEL is the exit code value of the program executed last.
<Number> is a number from 0 to 255.

Non-obligatory parameter <rel> is the relation described in section "string1<rel>string2", but used here as the relation between two numerical values of <number> and ERRORLEVEL. Its absence means that the condition is true if the exit code is greater than or equal to <number> (i.e. ERRORLEVEL GE <number>).

Thus, the condition is considered to be true if the relation <rel> between the return code and <number> is true.

Example:

Suppose that PROGRAM.EXE is an executable program returning the exit code.

```
.....
RUN PROGRAM.EXE
IF ERRORLEVEL GT 5 exit
IF ERRORLEVEL 2 echo Error code greater than or equal to 2.
IF ERRORLEVEL EQ 1 echo Error code equal to 2.
.....
```

3.1.13 EXIST [{ALL:ANY}] <FILELIST>

This condition checks if there are any <FILELIST> files on the drive. In case all or any (depending on non-obligatory parameter {ALL:ANY}) such files are found on the drive (even if they are empty), the condition is considered true.

If non-obligatory parameter specifies ANY (this is assumed by default) and if the Shell comes across at least one of them then the condition is considered true.

If parameter ALL is specified the condition is true if all the files from <FILELIST> are found on the disk. The default value of this parameter is ANY.

For more information about <FILELIST> use see
"INTERNAL DOS COMMAND ARGUMENTS"

3.1.14 SHIFT Command

Changes in setting of reference parameters

Let's consider how to address the 10th, the 11th, etc. parameters of the batch file.

Syntax: SHIFT [number]

This command lets shift all reference parameters left by a number. For instance, if number=1 (or is not specified) then the old value of %0 is lost; old value %1 is changed for %0, old %2 - new %1, etc. The tenth reference parameter becomes %9. Now substitution pairs will look like that: "%0" for "parm1"; "%1" for "parm2"; ... ; "%9" for "parm10".

If a number is present and it is not equal to 1, the above procedure is repeated "number" times. If the numbered parameter is not specified in the command line, then it is substituted for a blank string. The number cannot exceed 225.

3.2 SUBSTITUTION PARAMETERS IN A BATCH FILE

A batch file may contain substitution parameters which can be substituted when executing the batch either for command line parameters or for the DOS environment variables.

Let us consider, how the substitution parameters can be substituted for the command line. Suppose there is a batch file A.BAT in the current directory which contains special symbols:

%1, ... , %n, where n is a digit from 0 to 9.

Here is one of possible command lines calling up the batch file processing:

```
a parm1 parm2 parm3 ... parm9 parm10 parm11 ...
```

where "a" - the batch file name (by default, with .BAT extension). The rest are the reference parameters which are to substitute the symbols %1. %n in a batch file A.BAT.

Character string is considered to be a parameter if it is limited on both sides by some of the following symbols: space, tab symbol, switch symbol, end of line.

The procedure of substitution is the following: while processing a batch file line the Shell comes across a special symbol "%". It then looks at the following symbol and if it is a number from 0 to 9, then these two symbols (% and number) are substituted for reference parameter value found in the command line. The name of the batch file is considered to be a zero reference parameter. The rest of the parameters are numbered according to their location in the command line. In our example, the substitution pairs will be:

"%0" will be substituted for "a";
"%1" .... "parm1";
"%2" .... "parm2";
etc. up to
"%9"....................... "parm9".

Despite the fact that the quantity of reference parameters can be limited only by the command line length you can address simultaneously only the first ten, including batch file name which is considered to be the zero parameter. To access the rest of the parameters in batch files use the SHIFT command. Of course, character strings and the length of each reference parameter are arbitrary, although the value of the initial parameter %0 can contain only symbols allowed for file names.

Example:

Suppose, the file JACK.BAT contains following lines:

```
TASM %1.asm;
TLINK %1.obj,,%2
```

Suppose you enter the command:

```
D>jack aaa eee
```

After substitution the two lines above will be:

```
TASM aaa.asm;
TLINK aaa.obj,,eee
```

A batch file also can get information from the DOS environment. For more details on DOS environment see DOS command SET description.

The procedure of substitution of environment variables in a batch file command line is the following:

Having found a special symbol "%" in a batch file, the Shell looks for another "%" symbol or the end of line. All symbols located between these two special symbols are considered to be a name of a DOS Environment variable. After that the name, including special symbols, is removed from the batch file command line and substituted for the value of the DOS Environment variable. If there are no variables bearing this name in the Environment no substitution takes place although the initial name is still removed.

There are some restrictions on the names of DOS Environment variables to be used as substitution parameters:

a) variable name cannot start with a figure (although this is not stipulated by SET command syntax), because in such a case it would provoke substitution for a batch file command line reference parameter;
b) the name cannot contain the special symbol "%".

Though, if there are two symbols "%" in a row in a batch file line, the Shell will transfer one of them to the output command. Doubling of "%" symbols seems to be the only way to place a single "%" symbol in a command file.

3.3 BATCH FILE TERMINATION

The commands of a batch file are executed in strict order, line by line, until the Shell comes across one of the following two commands: GOTO (obligatory jump) or EXIT.

The execution of a batch file is terminated in one of the following cases:
- when the Shell comes across EXIT command;
- when the batch file runs out;
- when the Shell comes across the end of file symbol (code 26 in decimal or 1Ah in hexadecimal systems;
- on user break.

NOTICE: The breaking off is done by pressing [Ctrl-Break] or [Ctrl-C] keys on the keyboard. On pressing these keys the Shell prompts:

```
Terminate batch job {Y/n}?
```

After pressing [Y] or [Enter] key the execution of your batch file will be terminated. If [N] key is pressed the batch processing will continue.

After batch job is finished the Shell resumes processing current commands which it left pending when called up a batch file.

3.4 PECULIARITIES OF BATCH FILE PROCESSING WITH PTS-DOS SHELL

Such processing has some particular features as compared to standard procedure.

First and most important among them is that the Shell launches a special process for batch file executing and terminates it when batch file processing is interrupted for some reason. This has some consequent:

1. All batch files accessed for processing from the given one, upon their termination return the control back to the given batch file.
2. Re-directing "console" device command is a local one for the batch file, i.e. after the batch file is processed the state of standard I/O devices is restored.
3. There is a possibility to use redirection of input-output flows for the batch file as a whole.

Another particularity is the use of reserved names while processing DOS environment variables. If the following "names" are used as environment variables names after % symbols:

DIR, NAME, DIRNAME, EXT;

then the line processing proceeds as follows. The parameter immediately following such a "name" is considered to be the full variable name. A part of the this variable name is to be used in the command, while the rest is to be omitted. The remaining parts of variable names are correspondingly: file directory, file name only (without extension), file directory and name (extension is omitted), filename extension only.

The other special feature is the access to the canonical full name of a batch file under processing. Reference pseudo parameter "%$" is used for this purpose.

Example:

Suppose there is TEST.BAT file on drive C: in \DIRBAT\ directory.

```
executing the following
lines of the given batch:     will cause the output:

@echo %DIR% %$                C:\DIRBAT
@echo %NAME% %$               TEST
@echo %DIRNAME% %$            C:\DIRBAT\TEST
@echo %EXT% %$                BAT
@echo %$                      C:\DIRBAT\TEST.BAT
```

4 DOS DRIVERS

4.1 HiMem?86 Driver

Purpose: HiMem386.sys driver serves to support XMS 3.0 standard with PS/2, PC/AT PCs equipped with i386 or higher main processor. HiMem 286.sys driver supports XMS 2.0 standard with PS/2, PC/AT PCs with i286 main processor. Both drivers also arrange a higher memory area (HMA).

Syntax: DEVICE = HiMem?86.Sys [/NUMHANDLES=xx][/MACHINE=yy]

Description:

/NUMHANDLES can vary from 32 to 128. It sets the maximum number of HANDLES accessible for application.

/MACHINE determines PC type: 1 - AT compatible, 2 - PS/2 compatible PC.

While using HiMem?86 driver you can instruct the configuration file to locate DOS in high memory (see DOS configuration command). HiMem?86.sys driver is needed while setting up EMS memory and UMB.

4.2 RAMDISK Driver

Purpose: RAMDISK is a virtual disk driver that permits to store the user programs or data in unused memory.

Syntax: DEVICE[HIGH] = [FullPath]RAMDisk.sys
[NNNNN[K:M]] [DDDD] [/parm1...]

Description:

This driver version makes it possible to use three sorts of memory to create the virtual disk:

- Conventional memory. The only type accessible in XT machines. We do not recommend you to create too large virtual disk on the basis of this kind of memory since it sharply reduces the amount of memory available for user's programs. Usually virtual disk in conventional memory is created on machines whiteout hard disk with the only purpose of housing Command.Com.

- Extended memory. This is a memory residing in address space above 1 Mb and accessible only when processor operates in protected mode. The memory can be rather large and is well suited for a rapid disk.

- Expanded memory. This memory is also located over 1 Mb and has an access window in accessible address space. It can be very large. Memory abundance can be used to create a virtual disk.

Description of parameters:

NNNNN - volume of required memory in Kb (if K is specified or if there are no specifications) or in Mb if letter M is added after the first digit;
DDDD - number of root directory entries (minimum 16).
param1, param2, ...
- letter parameters:
A - place operational disk area in available Expanded memory;
E - move operational disk area to Extended memory;
L - switch on disk access indication via ScrollLock LED (at AT keyboards).

The driver requires presence of Extended/Expanded memory drivers, depending on chosen memory type (HiMem.sys, EMM386.sys, etc).

If E or A parameters are not specified then the disk will be located in the conventional memory.

If only one number is specified, it is considered to be the disk capacity; the number of root directory entries in this case is equal to 64.

If no numbers are specified then the disk capacity is set, depending on the letter parameters as: - 64 Kb in conventional memory; - or all available Extended memory; - or all available Expanded memory.

The specified disk capacity includes FAT and root directory.

4.3 Driver DOSCACHE.SUS

Purpose: Driver DOSCACHE provides hard disk caching (accelerates read/write operations due to temporary storing of data in XMS memory.

Syntax: DEVICE = [drive:][path] DOSCACHE.SYS [<mem_parms>]
[/C[<cache_parms>]] [/H] [/Q[Q]] [/P] [/D]

Description:
<mem_parms> - parameters for memory size configuration:
<buf_size> - full XMS buffer size in kilobytes.
The minimum value is 128, the maximum is 14000.
By default all accessible XMS memory is occupied.

<K:keep_size> - buffer size, which the driver tries to reserve
for itself. Default value is 1/4 of full size.
<W:win_size> - memory size that will be released for using in
WINDOWS. When running WINDOWS the lack of XMS
memory can occur. To avoid it set this parameter
equal to 600.

<cache_params> - parameters for cache configuration:
U:<block_sec> - number of sectors in one cache block (1-8).
Default value is 8. Less number is recommended
if the buffer size is less than 500 KBytes;
T:<delay_ticks> - write delay period in timer ticks (1/18 sec).
Default delay is 18 (1 second);
B - sets write-back mode. In this case writing i performed in background mode;
- in case of write back mode begins writing immediately but in background mode;
A - sets read-ahead mode. By default, read-ahead is ON;
F - write stategy is set by FIFO order (First In - First Out). By default, sectors are
sorted in ascending order before writing.
L - write strategy is set by LRU order (Least Recently Used).

/H - instruction, to use HMA for table storing.

/Q - instruction, not to display installation parameters (quiet mode).

/QQ - like /Q, messages about not fatal errors are also suppressed.

DOSCACHE is managed by PTSCTL program.

PTSCTL is a manager for DOSCACHE. If you start PTSCTL without parameters it displays current cache settings and statistics, with parameters it changes cache settings.

Syntax: PTSCTL [[+|-]<number>] [<option>[+|-]]...

where,

<number> - increases or decreases memory size, used by PTSCACHE;
o - switch on caching ('o-' - switch off);
a - switch on read-ahead ('a-' - switch off);
b - switch on write caching ('b-' - switch off);
i - begin writing immediately ('i-' - with delay);
f - fix memory ('f-' cancel fixation);
z - freeze cache ('z-' - unfreeze);
r - reset all counters;
u - reset cache;
wf - write sectors in FIFO order;
ws - write sectors in sorted order;
wl - write sectors in LRU order.