Virtualising DOS - Part 2

Useful Links

Virtualising DOS - Part 1
Virtualising DOS - Part 3
DOS Software Index

In Part 1 I introduced three tools which can create a virtual DOS environment. They are DOSBox, vDosPlus, and VirtualBox. For the sake of simplicity, I will refer to these as DOS emulators going forward.

In this part, I will test each of them in 'real-world' scenarios and see how they fare. As I mentioned in Part 1, each emulator is really designed for a different purpose, so I fully expect their respective successes and failures to be aligned with these programs' design goals.

It's probably worth noting that there are different versions of DOSBox that are designed more for running applications instead of games. These are called *DOSBox-X* and *DOSBox-X-App*.

Typical Uses of DOS

When DOS was in active use, it was used for a broad set of consumer and business needs. Playing games was certainly not the original aim of IBM when they introduced the IBM 5150 PC in 1982, but over the years it became more and more of a gaming platform. Throughout the DOS era, the operating system was used to:

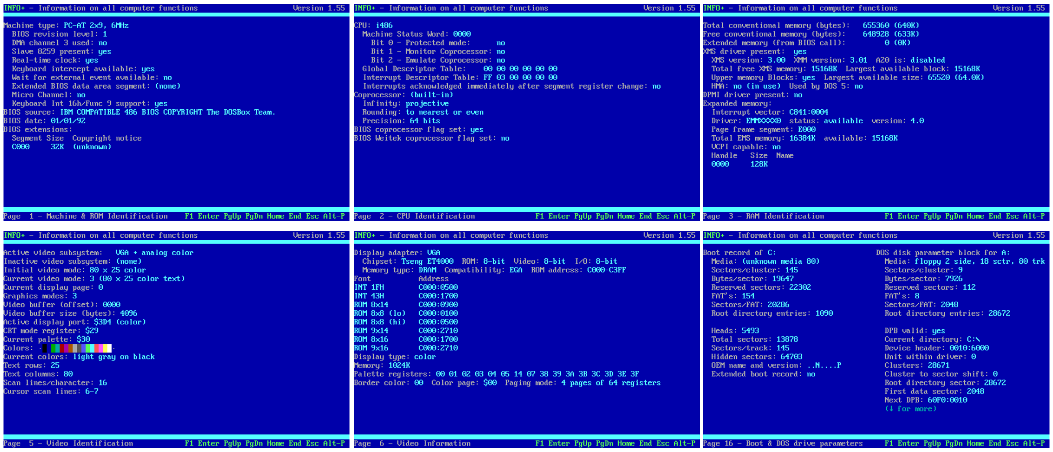
- Write documents using a word processor.
- Create spreadsheets.
- Keep track of your finances.
- Store and retrieve data in a database.
- Desktop publishing.
- Use drawing tools.
- Play games.
- Develop new software.

I will try out various DOS software applications that fall into each of these 8 categories and report on the results. While some of the above operate solely in a DOS text mode (typically 80 x 25), others require graphics modes, so it will be interesting to see how well they are supported. In the early days of DOS, a text mode of 132 columns was widely used to provide more screen 'real estate' - this too will be evaluated where possible.

So let's begin!

System Information

Before I get into testing how applications are able to function, let's take a look at what is being reported as the "guest" operating system. *NSSI (Navratil)* only worked on DOSBox - with vDosPlus and VirtualBox it crashed on startup while it was detecting SMBIOS/DMI, so I reverted to the older and more simple *InfoPlus*:



```
INFO+ - Information on all computer functions                  Version 1.55

Machine type: PC-AT 2x9, 6MHz
  BIOS revision level: 1
  DMA channel 3 used: no
  Slave 8259 present: yes
  Real-time clock: yes
  Keyboard intercept available: yes
  Wait for external event available: no
  Extended BIOS data area segment: (none)
  Micro Channel: no
  Keyboard Int 16h/Func 9 support: yes
BIOS source: IBM COMPATIBLE 486 BIOS COPYRIGHT The DOSBox Team.
BIOS date: 01/01/92
BIOS extensions:
  Segment Size  Copyright notice
  C000     32K  (unknown)

Page  1 - Machine & ROM Identification   F1 Enter PgUp PgDn Home End Esc Alt-P
```

```
INFO+ - Information on all computer functions                  Version 1.55

CPU: i486
  Machine Status Word: 0000
    Bit 0 - Protected mode:      no
    Bit 1 - Monitor Coprocessor: no
    Bit 2 - Emulate Coprocessor: no
  Global Descriptor Table:    00 00 00 00 00 00
  Interrupt Descriptor Table: FF 03 00 00 00 00
  Interrupts acknowledged immediately after segment register change: no
Coprocessor: (built-in)
  Infinity: projective
  Rounding: to nearest or even
  Precision: 64 bits
BIOS coprocessor flag set: yes
BIOS Weitek coprocessor flag set: no

Page  2 - CPU Identification             F1 Enter PgUp PgDn Home End Esc Alt-P
```

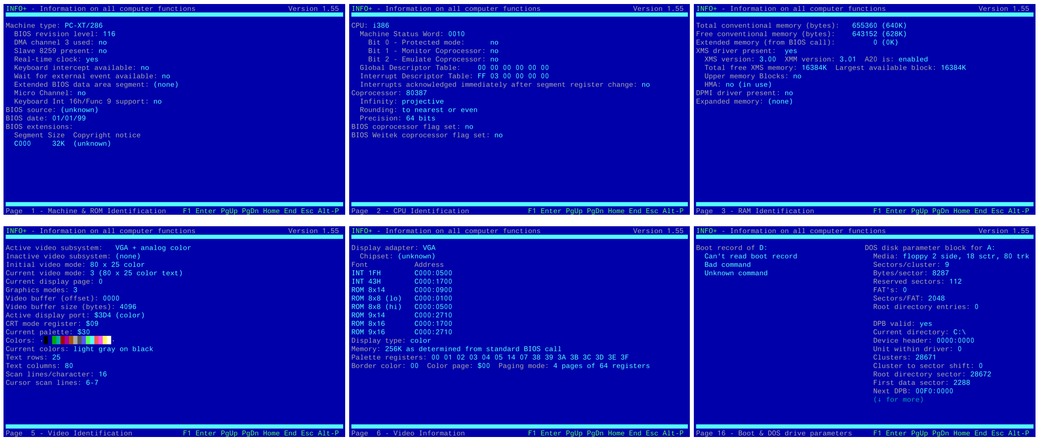
```
INFO+ - Information on all computer functions                  Version 1.55

Total conventional memory (bytes):   655360 (640K)
Free conventional memory (bytes):    648928 (633K)
Extended memory (from BIOS call):         0 (0K)
XMS driver present:  yes
  XMS version: 3.00  XMM version: 3.01  A20 is: disabled
  Total free XMS memory: 15168K  Largest available block: 15168K
  Upper memory Blocks: yes  Largest available size: 65520 (64.0K)
  HMA: no (in use)  Used by DOS 5: no
DPMI driver present: no
Expanded memory:
  Interrupt vector: C841:0004
  Driver: EMMXXXX0  status: available  version: 4.0
  Page frame segment: E000
  Total EMS memory: 16384K  available: 15168K
  VCPI capable: no
  Handle    Size  Name
  0000      128K

Page  3 - RAM Identification             F1 Enter PgUp PgDn Home End Esc Alt-P
```

```
INFO+ - Information on all computer functions                  Version 1.55

Active video subsystem:   VGA + analog color
Inactive video subsystem: (none)
Initial video mode: 80 x 25 color
Current video mode: 3 (80 x 25 color text)
Current display page: 0
Graphics modes: 3
Video buffer (offset): 0000
Video buffer size (bytes): 4096
Active display port: $3D4 (color)
CRT mode register: $29
Current palette: $30
Colors:
Current colors: light gray on black
Text rows: 25
Text columns: 80
Scan lines/character: 16
Cursor scan lines: 6-7

Page  5 - Video Identification           F1 Enter PgUp PgDn Home End Esc Alt-P
```

```
INFO+ - Information on all computer functions                  Version 1.55

Display adapter: VGA
  Chipset: Tseng ET4000  ROM: 8-bit  Video: 8-bit  I/O: 8-bit
  Memory type: DRAM  Compatibility: EGA  ROM address: C000-C3FF
Font          Address
INT 1FH       C000:0500
INT 43H       C000:1700
ROM 8x14      C000:0900
ROM 8x8 (lo)  C000:0100
ROM 8x8 (hi)  C000:0500
ROM 9x14      C000:2710
ROM 8x16      C000:1700
ROM 9x16      C000:2710
Display type: color
Memory: 1024K
Palette registers: 00 01 02 03 04 05 14 07 38 39 3A 3B 3C 3D 3E 3F
Border color: 00  Color page: $00  Paging mode: 4 pages of 64 registers

Page  6 - Video Information              F1 Enter PgUp PgDn Home End Esc Alt-P
```

```
INFO+ - Information on all computer functions                  Version 1.55

Boot record of C:                       DOS disk parameter block for A:
  Media: (unknown media 80)               Media: floppy 2 side, 18 sctr, 80 trk
  Sectors/cluster: 145                    Sectors/cluster: 9
  Bytes/sector: 19647                     Bytes/sector: 7926
  Reserved sectors: 22302                 Reserved sectors: 112
  FAT's: 154                              FAT's: 8
  Sectors/FAT: 20286                      Sectors/FAT: 2048
  Root directory entries: 1090            Root directory entries: 28672

  Heads: 5493                             DPB valid: yes
  Total sectors: 13878                    Current directory: C:\
  Sectors/track: 145                      Device header: 0010:6000
  Hidden sectors: 64703                   Unit within driver: 0
  OEM name and version: ..N....P          Clusters: 28671
  Extended boot record: no                Cluster to sector shift: 0
                                          Root directory sector: 28672
                                          First data sector: 2048
                                          Next DPB: 60F0:0010
                                          (↓ for more)

Page 16 - Boot & DOS drive parameters    F1 Enter PgUp PgDn Home End Esc Alt-P
```

DOSBox 0.74 running InfoPlus 1.55



```
INFO+ - Information on all computer functions                  Version 1.55

Machine type: PC-XT/286
  BIOS revision level: 116
  DMA channel 3 used: no
  Slave 8259 present: no
  Real-time clock: yes
  Keyboard intercept available: no
  Wait for external event available: no
  Extended BIOS data area segment: (none)
  Micro Channel: no
  Keyboard Int 16h/Func 9 support: no
BIOS source: (unknown)
BIOS date: 01/01/99
BIOS extensions:
  Segment Size  Copyright notice
  C000     32K  (unknown)

Page  1 - Machine & ROM Identification   F1 Enter PgUp PgDn Home End Esc Alt-P
```

```
INFO+ - Information on all computer functions                  Version 1.55

CPU: i386
  Machine Status Word: 0010
    Bit 0 - Protected mode:      no
    Bit 1 - Monitor Coprocessor: no
    Bit 2 - Emulate Coprocessor: no
  Global Descriptor Table:    00 00 00 00 00 00
  Interrupt Descriptor Table: FF 03 00 00 00 00
  Interrupts acknowledged immediately after segment register change: no
Coprocessor: 80387
  Infinity: projective
  Rounding: to nearest or even
  Precision: 64 bits
BIOS coprocessor flag set: no
BIOS Weitek coprocessor flag set: no

Page  2 - CPU Identification             F1 Enter PgUp PgDn Home End Esc Alt-P
```

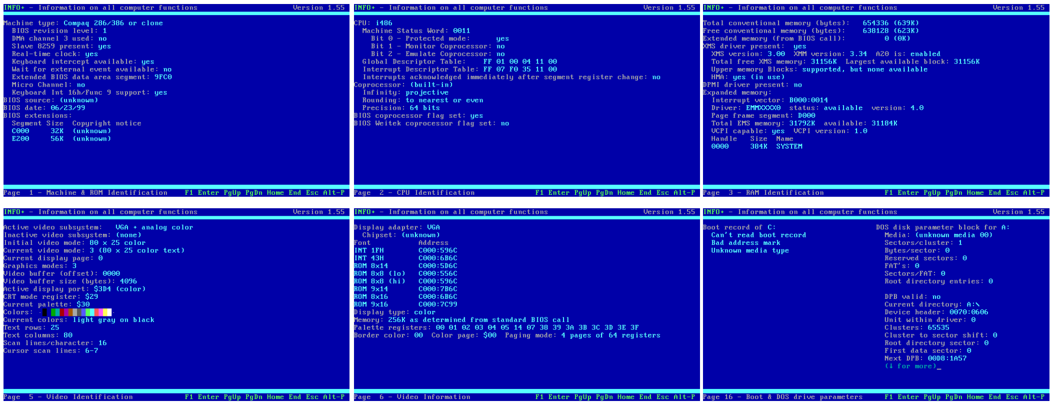
```
INFO+ - Information on all computer functions                  Version 1.55

Total conventional memory (bytes):   655360 (640K)
Free conventional memory (bytes):    643152 (628K)
Extended memory (from BIOS call):         0 (0K)
XMS driver present:  yes
  XMS version: 3.00  XMM version: 3.01  A20 is: enabled
  Total free XMS memory: 16384K  Largest available block: 16384K
  Upper memory Blocks: no
  HMA: no (in use)
DPMI driver present: no
Expanded memory: (none)

Page  3 - RAM Identification             F1 Enter PgUp PgDn Home End Esc Alt-P
```

```
INFO+ - Information on all computer functions                  Version 1.55

Active video subsystem:   VGA + analog color
Inactive video subsystem: (none)
Initial video mode: 80 x 25 color
Current video mode: 3 (80 x 25 color text)
Current display page: 0
Graphics modes: 3
Video buffer (offset): 0000
Video buffer size (bytes): 4096
Active display port: $3D4 (color)
CRT mode register: $09
Current palette: $30
Colors:
Current colors: light gray on black
Text rows: 25
Text columns: 80
Scan lines/character: 16
Cursor scan lines: 6-7

Page  5 - Video Identification           F1 Enter PgUp PgDn Home End Esc Alt-P
```

```
INFO+ - Information on all computer functions                  Version 1.55

Display adapter: VGA
  Chipset: (unknown)
Font          Address
INT 1FH       C000:0500
INT 43H       C000:1700
ROM 8x14      C000:0900
ROM 8x8 (lo)  C000:0100
ROM 8x8 (hi)  C000:0500
ROM 9x14      C000:2710
ROM 8x16      C000:1700
ROM 9x16      C000:2710
Display type: color
Memory: 256K as determined from standard BIOS call
Palette registers: 00 01 02 03 04 05 14 07 38 39 3A 3B 3C 3D 3E 3F
Border color: 00  Color page: $00  Paging mode: 4 pages of 64 registers

Page  6 - Video Information              F1 Enter PgUp PgDn Home End Esc Alt-P
```

```
INFO+ - Information on all computer functions                  Version 1.55

Boot record of D:                       DOS disk parameter block for A:
  Can't read boot record                  Media: floppy 2 side, 18 sctr, 80 trk
  Bad command                             Sectors/cluster: 9
  Unknown command                         Bytes/sector: 8287
                                          Reserved sectors: 112
                                          FAT's: 0
                                          Sectors/FAT: 2048
                                          Root directory entries: 0

                                          DPB valid: yes
                                          Current directory: C:\
                                          Device header: 0000:0000
                                          Unit within driver: 0
                                          Clusters: 28671
                                          Cluster to sector shift: 0
                                          Root directory sector: 28672
                                          First data sector: 2288
                                          Next DPB: 00F0:0000
                                          (↓ for more)

Page 16 - Boot & DOS drive parameters    F1 Enter PgUp PgDn Home End Esc Alt-P
```

vDosPlus 2017.08.01 (build 2018.03.01) running InfoPlus 1.55



```
INFO+ - Information on all computer functions                  Version 1.55

Machine type: Compaq 286/386 or clone
  BIOS revision level: 1
  DMA channel 3 used: no
  Slave 8259 present: yes
  Real-time clock: yes
  Keyboard intercept available: yes
  Wait for external event available: no
  Extended BIOS data area segment: 9FC0
  Micro Channel: no
  Keyboard Int 16h/Func 9 support: yes
BIOS source: (unknown)
BIOS date: 06/23/99
BIOS extensions:
  Segment Size  Copyright notice
  C000     32K  (unknown)
  E200     56K  (unknown)

Page  1 - Machine & ROM Identification   F1 Enter PgUp PgDn Home End Esc Alt-P
```

```
INFO+ - Information on all computer functions                  Version 1.55

CPU: i486
  Machine Status Word: 0011
    Bit 0 - Protected mode:      yes
    Bit 1 - Monitor Coprocessor: no
    Bit 2 - Emulate Coprocessor: no
  Global Descriptor Table:    FF 01 00 04 11 00
  Interrupt Descriptor Table: FF 07 F0 35 11 00
  Interrupts acknowledged immediately after segment register change: no
Coprocessor: (built-in)
  Infinity: projective
  Rounding: to nearest or even
  Precision: 64 bits
BIOS coprocessor flag set: yes
BIOS Weitek coprocessor flag set: no

Page  2 - CPU Identification             F1 Enter PgUp PgDn Home End Esc Alt-P
```

```
INFO+ - Information on all computer functions                  Version 1.55

Total conventional memory (bytes):   654336 (639K)
Free conventional memory (bytes):    638128 (623K)
Extended memory (from BIOS call):         0 (0K)
XMS driver present:  yes
  XMS version: 3.00  XMM version: 3.34  A20 is: enabled
  Total free XMS memory: 31156K  Largest available block: 31156K
  Upper memory Blocks: supported, but none available
  HMA: yes (in use)
DPMI driver present: no
Expanded memory:
  Interrupt vector: B000:0014
  Driver: EMMXXXX0  status: available  version: 4.0
  Page frame segment: D000
  Total EMS memory: 31792K  available: 31184K
  VCPI capable: yes  VCPI version: 1.0
  Handle    Size  Name
  0000      384K  SYSTEM

Page  3 - RAM Identification             F1 Enter PgUp PgDn Home End Esc Alt-P
```

```
INFO+ - Information on all computer functions                  Version 1.55

Active video subsystem:   VGA + analog color
Inactive video subsystem: (none)
Initial video mode: 80 x 25 color
Current video mode: 3 (80 x 25 color text)
Current display page: 0
Graphics modes: 3
Video buffer (offset): 0000
Video buffer size (bytes): 4096
Active display port: $3D4 (color)
CRT mode register: $29
Current palette: $30
Colors:
Current colors: light gray on black
Text rows: 25
Text columns: 80
Scan lines/character: 16
Cursor scan lines: 6-7

Page  5 - Video Identification           F1 Enter PgUp PgDn Home End Esc Alt-P
```

```
INFO+ - Information on all computer functions                  Version 1.55

Display adapter: VGA
  Chipset: (unknown)
Font          Address
INT 1FH       C000:596C
INT 43H       C000:6B6C
ROM 8x14      C000:5D6C
ROM 8x8 (lo)  C000:556C
ROM 8x8 (hi)  C000:596C
ROM 9x14      C000:7B6C
ROM 8x16      C000:6B6C
ROM 9x16      C000:7C99
Display type: color
Memory: 256K as determined from standard BIOS call
Palette registers: 00 01 02 03 04 05 14 07 38 39 3A 3B 3C 3D 3E 3F
Border color: 00  Color page: $00  Paging mode: 4 pages of 64 registers

Page  6 - Video Information              F1 Enter PgUp PgDn Home End Esc Alt-P
```

```
INFO+ - Information on all computer functions                  Version 1.55

Boot record of C:                       DOS disk parameter block for A:
  Can't read boot record                  Media: (unknown media 00)
  Bad address mark                        Sectors/cluster: 1
  Unknown media type                      Bytes/sector: 0
                                          Reserved sectors: 0
                                          FAT's: 0
                                          Sectors/FAT: 0
                                          Root directory entries: 0

                                          DPB valid: no
                                          Current directory: A:\
                                          Device header: 0070:0606
                                          Unit within driver: 0
                                          Clusters: 65535
                                          Cluster to sector shift: 0
                                          Root directory sector: 0
                                          First data sector: 0
                                          Next DPB: 00D8:1A57
                                          (↓ for more)_

Page 16 - Boot & DOS drive parameters    F1 Enter PgUp PgDn Home End Esc Alt-P
```

VirtualBox 7.0 running InfoPlus 1.51

So each virtual machine reports numerous differences, both with the machine type and CPU. Only DOSBox correctly shows the C: drive, with vDosPlus and VirtualBox unable to read the boot record.

132-Column Mode

Back in the DOS days, a lot of EGA and VGA cards supported a larger text mode of 132 columns by 50 lines. In native DOS, you would achieve this by running 'MODE CON COLS=132 LINES=50' at the DOS command-line.

In DOSBox and vDosPlus you don't have a MODE program to run. It may be that we need to run a VESA VBE driver to provide extended text modes (in this case we need 0x54 or 0x55). I also downloaded the VMODE.EXE tool from Columbia.edu, which directly manipulates INT 10h to switch to a different graphics mode, but in DOSBox and vDosPlus this simply results in a blank screen with a flashing cursor. Any programs you run will revert to 80x25 text mode. I also attempted to get both emulators to install a VESA driver (UNIVBE), but under all the 'machine' options in the DOSBox.conf file, they all returned '*No supported Super VGA* *chipset detected!* when I ran UNIVBE.EXE. I also tried the Paradise SVGA VESA drivers which appeared to load ok, but had the same outcome when running any software - a blank screen with a flashing cursor.

I mentioned *DOSBox-X* and *DOSBox-X-App* earlier - these apparently do have text mode commands that mimic what the DOS 'MODE' utility would do, such as typing '80x25' to set the text mode to 80x25. However, these max out at 100x43 (by typing '100x43'), so the common 132-column mode is not supported even in these versions.

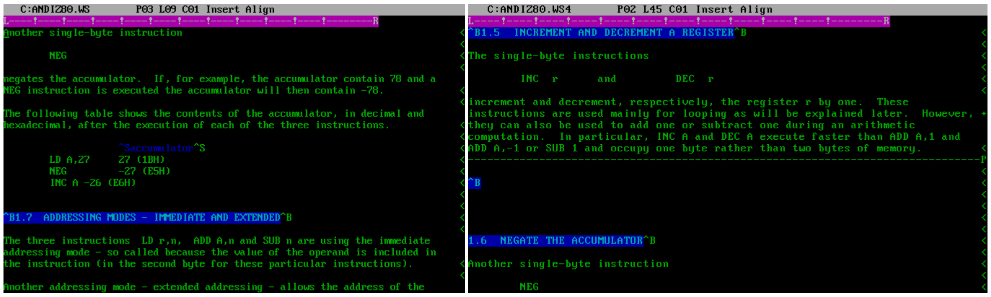
Word Processing

The most popular word processors in the DOS era were WordStar and WordPerfect. There are many others, but for this test I'll just use these two:

WordStar 4

DOSBox ran WordStar 4 just fine. I didn't use WordStar much back in the 80s, so I used the WordPerfect 'CONVERT' utility to convert one of my old documents into WordStar 3.3 format (the best it could do). The document wasn't perfectly converted, but making changes in the document and saving went without a hitch. VirtualBox also did a good job with WordStar 4 - I could open documents, change their content and save without issue. While I was hopeful to be able to highlight text with the mouse to copy some text to the host machine (my Windows 10 PC), I was unable to get this to work.

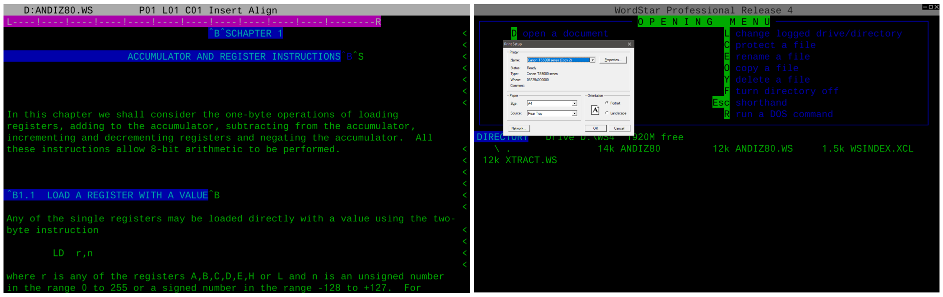
Both *DOSBox* and *VirtualBox* however, **do not** provide support to print to the host machine's attached [or network] printers. This is where vDosPlus gets an extra point - it detects output being sent to LPT1: and other ports, and sends them to the host operating system. It too behaved very well with WordStar 4 in terms of editing, saving, etc.



```
   C:ANDIZ80.WS         P03 L09 C01 Insert Align
L----!----!----!----!----!----!----!----!----!----!----!--------R
Another single-byte instruction

        NEG

negates the accumulator.  If, for example, the accumulator contain 78 and a
NEG instruction is executed the accumulator will then contain -78.

The following table shows the contents of the accumulator, in decimal and
hexadecimal, after the execution of each of the three instructions.

                    ^Saccumulator^S
        LD A,27     27 (1BH)
        NEG         -27 (E5H)
        INC A -26 (E6H)


^B1.7  ADDRESSING MODES - IMMEDIATE AND EXTENDED^B

The three instructions  LD r,n,  ADD A,n and SUB n are using the immediate
addressing mode - so called because the value of the operand is included in
the instruction (in the second byte for these particular instructions).

Another addressing mode - extended addressing - allows the address of the
```

```
   C:ANDIZ80.WS4        P02 L45 C01 Insert Align
L----!----!----!----!----!----!----!----!----!----!----!--------R
^B1.5  INCREMENT AND DECREMENT A REGISTER^B

The single-byte instructions

        INC  r      and         DEC  r

increment and decrement, respectively, the register r by one.  These
instructions are used mainly for looping as will be explained later.  However, +
they can also be used to add one or subtract one during an arithmetic
computation.  In particular, INC A and DEC A execute faster than ADD A,1 and
ADD A,-1 or SUB 1 and occupy one byte rather than two bytes of memory.
-------------------------------------------------------------------------------P

^B




1.6  NEGATE THE ACCUMULATOR^B

Another single-byte instruction

        NEG
```

From left to right: Running WordStar 4 in DOSBox 0.74 and VirtualBox 7.0



```
   D:ANDIZ80.WS         P01 L01 C01 Insert Align
L----!----!----!----!----!----!----!----!----!----!----!--------R
                                     ^B^SCHAPTER 1

                      ACCUMULATOR AND REGISTER INSTRUCTIONS^B^S




In this chapter we shall consider the one-byte operations of loading
registers, adding to the accumulator, subtracting from the accumulator,
incrementing and decrementing registers and negating the accumulator.  All
these instructions allow 8-bit arithmetic to be performed.



^B1.1  LOAD A REGISTER WITH A VALUE^B

Any of the single registers may be loaded directly with a value using the two-
byte instruction

        LD  r,n

where r is any of the registers A,B,C,D,E,H or L and n is an unsigned number
in the range 0 to 255 or a signed number in the range -128 to +127.  For
```

```
                    WordStar Professional Release 4
                          O P E N I N G   M E N U
     D open a document                      L change logged drive/directory
                                            C protect a file
                                            E rename a file
                                            O copy a file
                                            Y delete a file
                                            F turn directory off
                                          Esc shorthand
                                            R run a DOS command

DIRECTORY   Drive D:\WS4  1920M free
  \ .              14k ANDIZ80        12k ANDIZ80.WS     1.5k WSINDEX.XCL
 12k XTRACT.WS
```

Print Setup

Printer
Name: Canon TS5000 series (Copy 2)
Status: Ready
Type: Canon TS5000 series
Where: 08F254000000
Comment:

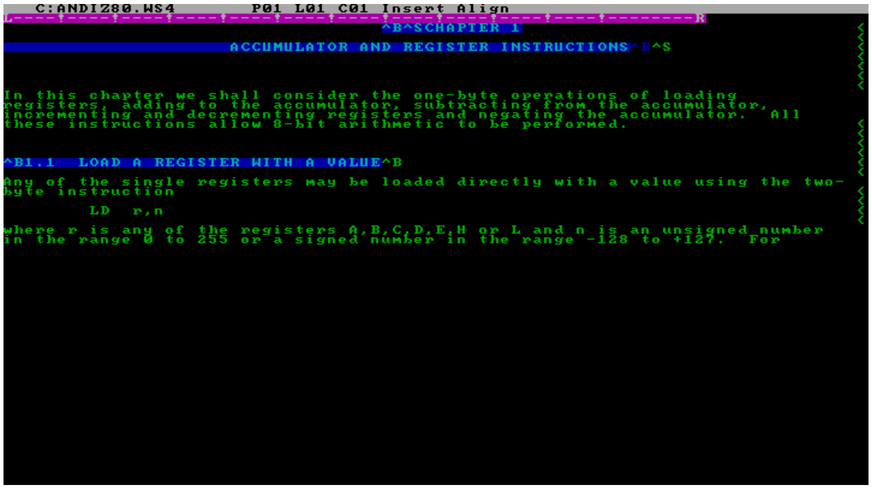
Paper
Size: A4
Source: Rear Tray

Orientation
● Portrait
○ Landscape

WordStar 4 in vDosPlus and after choosing the 'P' "Print Document" option

Unfortunately, I need to do some more research on how to properly support modern printers, as for me it just printed garbage after attempts with various printer types in WordStar - I tried Epson FX80, IBM Proprinter, or HPLJET. I downloaded vDOSPCLPS.zip and put those two files into my vDosPlus folder, and also downloaded and installed a program called Printfil, but I still could not get readable printed text from vDosPlus (Printfil's *Print Preview* screen showed the garbage chars). It's great news that vDosPlus came up with the Windows print dialog, which assures me this functionality is in place. I know it's simply lack of education on my part concerning PCL or PostScript (printing languages, basically), and will continue to work on it. I did successfully print a raw text file from the DOS command line by redirecting to LPT1:, so top marks for vDosPlus here!

With VirtualBox 7.0, I tried 132-column mode but it didn't work quite right. It might be that WordStar 4 doesn't actually support more than 25 lines, because while it ran the application with 132 columns, it restricted the number of rows to 25:

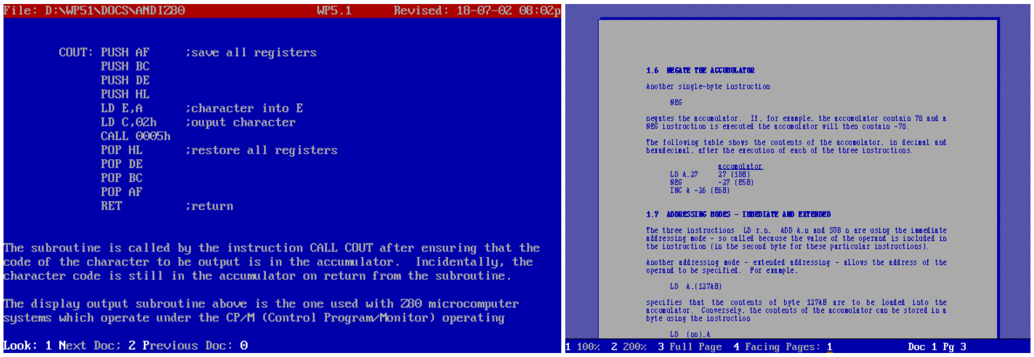


```
   C:ANDIZ80.WS4        P01 L01 C01 Insert Align
L----!----!----!----!----!----!----!----!----!----!--------R
                          ^B^SCHAPTER 1
                 ACCUMULATOR AND REGISTER INSTRUCTIONS^B^S


In this chapter we shall consider the one-byte operations of loading
registers, adding to the accumulator, subtracting from the accumulator,
incrementing and decrementing registers and negating the accumulator.  All
these instructions allow 8-bit arithmetic to be performed.


^B1.1  LOAD A REGISTER WITH A VALUE^B

Any of the single registers may be loaded directly with a value using the two-
byte instruction

        LD  r,n

where r is any of the registers A,B,C,D,E,H or L and n is an unsigned number
in the range 0 to 255 or a signed number in the range -128 to +127.  For
```

WordStar 4 running under VirtualBox 7.0 after putting the guest system in 132x50 text mode

WordPerfect 5.1

DOSBox ran WordPerfect 5.1 without any problems, aside from the fact that there is no facility to send the print job to host machine for printing on a modern printer:

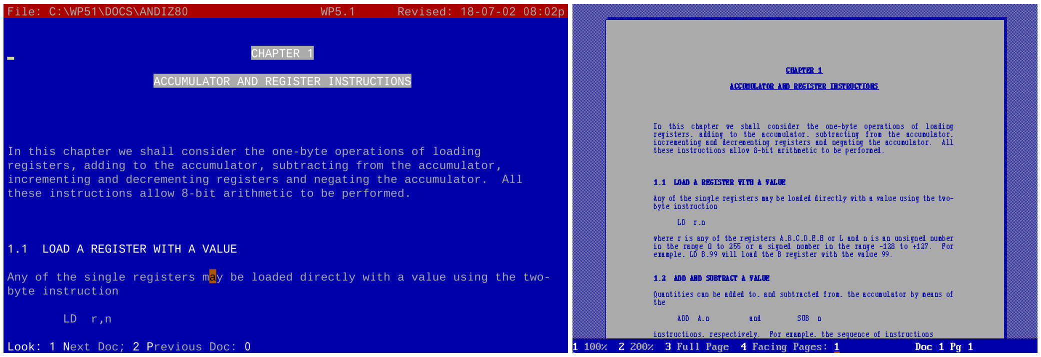


```
File: D:\WP51\DOCS\ANDIZ80                WP5.1      Revised: 18-07-02 08:02p


       COUT: PUSH AF      ;save all registers
             PUSH BC
             PUSH DE
             PUSH HL
             LD E,A       ;character into E
             LD C,02h     ;ouput character
             CALL 0005h
             POP HL       ;restore all registers
             POP DE
             POP BC
             POP AF
             RET          ;return


The subroutine is called by the instruction CALL COUT after ensuring that the
code of the character to be output is in the accumulator.  Incidentally, the
character code is still in the accumulator on return from the subroutine.

The display output subroutine above is the one used with Z80 microcomputer
systems which operate under the CP/M (Control Program/Monitor) operating

Look: 1 Next Doc; 2 Previous Doc: 0
```

```
1.6  NEGATE THE ACCUMULATOR

Another single-byte instruction

      NEG

negates the accumulator.  If, for example, the accumulator contain 78 and a
NEG instruction is executed the accumulator will then contain -78.

The following table shows the contents of the accumulator, in decimal and
hexadecimal, after the execution of each of the three instructions.

                  accumulator
      LD A,27     27 (1BH)
      NEG         -27 (E5H)
      INC A -26 (E6H)

1.7  ADDRESSING MODES - IMMEDIATE AND EXTENDED

The three instructions  LD r,n,  ADD A,n and SUB n are using the immediate
addressing mode - so called because the value of the operand is included in
the instruction (in the second byte for these particular instructions).

Another addressing mode - extended addressing - allows the address of the
operand to be specified.  For example,

      LD  A,(127AH)

specifies that the contents of byte 127AH are to be loaded into the
accumulator.  Conversely, the contents of the accumulator can be stored in a
byte using the instruction

      LD  (nn),A

1 100%  2 200%  3 Full Page  4 Facing Pages: 1            Doc 1 Pg 3
```

WordPerfect 5.1 running in DOSBox 0.74: (Left): A document opened, and (Right): The print preview screen

Next up was vDosPlus, though I really should evaluate a dedicated version of vDos that's been developed specifically for WordPerfect for DOS users - that product is called WPDOS, and is maintained by *Edward Mendelson* of Columbia State University. So if you are only virtualising DOS primarily to use WordPerfect 5.1, 6.1 or 6.2, this is more what you're after. Likewise, a dedicated fork from vDosPlus for XyWrite exists, and is maintained by *Kari Eveli* and *Lexitec*. This supports XyWrite III+, XyWrite IV, and Nota Bene.

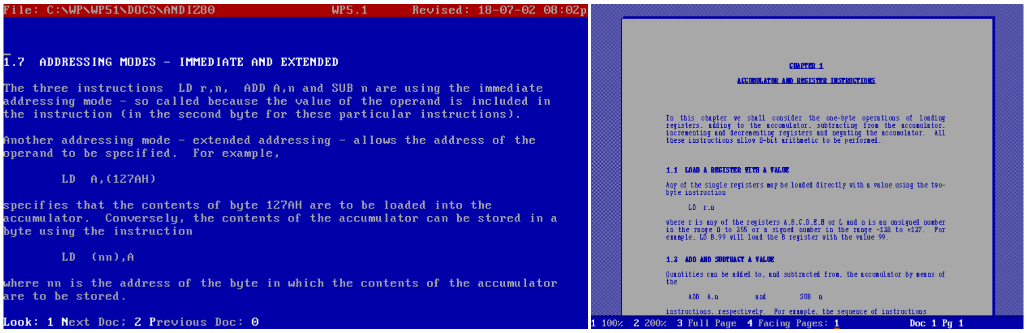


```
File: C:\WP51\DOCS\ANDIZ80                WP5.1      Revised: 18-07-02 08:02p


                              CHAPTER 1

                  ACCUMULATOR AND REGISTER INSTRUCTIONS



In this chapter we shall consider the one-byte operations of loading
registers, adding to the accumulator, subtracting from the accumulator,
incrementing and decrementing registers and negating the accumulator.  All
these instructions allow 8-bit arithmetic to be performed.



1.1   LOAD A REGISTER WITH A VALUE

Any of the single registers may be loaded directly with a value using the two-
byte instruction

        LD  r,n

Look: 1 Next Doc; 2 Previous Doc: 0
```

```
                     CHAPTER 1

          ACCUMULATOR AND REGISTER INSTRUCTIONS



In this chapter we shall consider the one-byte operations of loading
registers, adding to the accumulator, subtracting from the accumulator,
incrementing and decrementing registers and negating the accumulator.  All
these instructions allow 8-bit arithmetic to be performed.


1.1  LOAD A REGISTER WITH A VALUE

Any of the single registers may be loaded directly with a value using the two-
byte instruction

      LD  r,n

where r is any of the registers A,B,C,D,E,H or L and n is an unsigned number
in the range 0 to 255 or a signed number in the range -128 to +127.  For
example, LD B,99 will load the B register with the value 99.


1.2  ADD AND SUBTRACT A VALUE

Quantities can be added to, and subtracted from, the accumulator by means of
the

      ADD  A,n         and         SUB  n

instructions, respectively.  For example, the sequence of instructions

1 100%  2 200%  3 Full Page  4 Facing Pages: 1            Doc 1 Pg 1
```

WordPerfect 5.1 running in vDosPlus: (Left): A document opened, and (Right): The print preview screen

No issues with the standard version of vDosPlus, but apparently WPDOS as I mentioned earlier supports multiple ways of exchanging data with the Windows clipboard, open files on a network share, has a load of additional WordPerfect macros, and up to 142 columns by 60 lines display. Other graphical modes and other fonts and codepages are also supported.



```
File: C:\WP\WP51\DOCS\ANDIZ80                 WP5.1     Revised: 18-07-02 08:02p


1.7  ADDRESSING MODES - IMMEDIATE AND EXTENDED

The three instructions  LD r,n,  ADD A,n and SUB n are using the immediate
addressing mode - so called because the value of the operand is included in
the instruction (in the second byte for these particular instructions).

Another addressing mode - extended addressing - allows the address of the
operand to be specified.  For example,

        LD  A,(127AH)

specifies that the contents of byte 127AH are to be loaded into the
accumulator.  Conversely, the contents of the accumulator can be stored in a
byte using the instruction

        LD  (nn),A

where nn is the address of the byte in which the contents of the accumulator
are to be stored.

Look: 1 Next Doc; 2 Previous Doc: 0
```

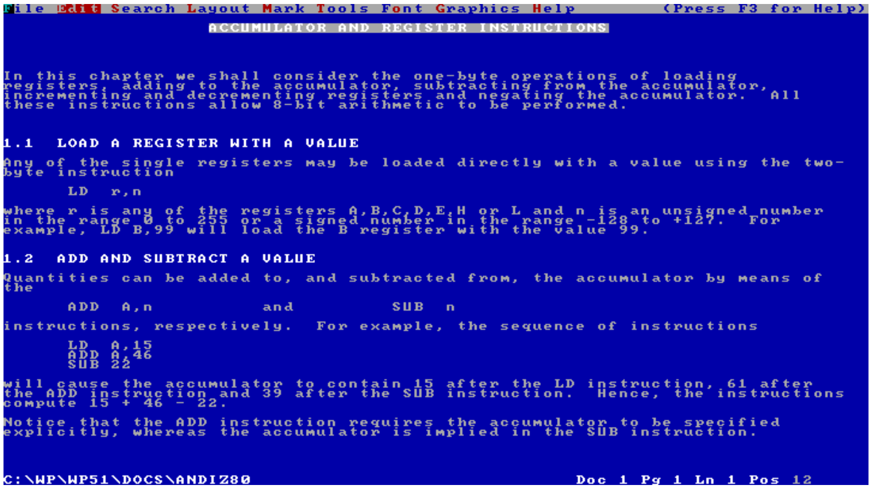
```
                         CHAPTER 1

            ACCUMULATOR AND REGISTER INSTRUCTIONS


In this chapter we shall consider the one-byte operations of loading
registers. adding to the accumulator. subtracting from the accumulator.
incrementing and decrementing registers and negating the accumulator.  All
these instructions allow 8-bit arithmetic to be performed.


1.1  LOAD A REGISTER WITH A VALUE

Any of the single registers may be loaded directly with a value using the two-
byte instruction

      LD  r.n

where r is any of the registers A.B.C.D.E.H or L and n is an unsigned number
in the range 0 to 255 or a signed number in the range -128 to +127.  For
example. LD B.99 will load the B register with the value 99.


1.2  ADD AND SUBTRACT A VALUE

Quantities can be added to. and subtracted from. the accumulator by means of
the

      ADD  A.n         and         SUB  n

instructions. respectively.  For example. the sequence of instructions
1 100%  2 200%  3 Full Page  4 Facing Pages: 1              Doc 1 Pg 1
```

WordPerfect 5.1 running in VirtualBox: (Left): A document opened, and (Right): The print preview screen

VirtualBox also runs WordPerfect 5.1 well, though its performance is slower than that of vDosPlus and DOSBox when displaying the print preview.

Now to try out 132-column mode with VirtualBox.... Incidentally, WordPerfect can be set to run in a different text mode via the main menu's File->Setup->Display option. Be careful here, as it doesn't work under DOSBox or vDosPlus, and will corrupt the screen so you cannot even navigate back to a normal mode unless you know what you're doing! If this happens to you, hit Esc, Esc, Shift-F1, 2, 3, 2 to set the text mode back to normal 80x25.

Since VirtualBOX is running a version of FreeDOS 1.2, the DOS MODE utility is available and it works just fine:



```
File Edit Search Layout Mark Tools Font Graphics Help            (Press F3 for Help)
                    ACCUMULATOR AND REGISTER INSTRUCTIONS


In this chapter we shall consider the one-byte operations of loading
registers, adding to the accumulator, subtracting from the accumulator,
incrementing and decrementing registers and negating the accumulator.  All
these instructions allow 8-bit arithmetic to be performed.

1.1  LOAD A REGISTER WITH A VALUE
Any of the single registers may be loaded directly with a value using the two-
byte instruction
        LD  r,n
where r is any of the registers A,B,C,D,E,H or L and n is an unsigned number
in the range 0 to 255 or a signed number in the range -128 to +127.  For
example, LD B,99 will load the B register with the value 99.

1.2  ADD AND SUBTRACT A VALUE
Quantities can be added to, and subtracted from, the accumulator by means of
the
        ADD  A,n            and           SUB  n
instructions, respectively.  For example, the sequence of instructions
        LD  A,15
        ADD A,46
        SUB 22
will cause the accumulator to contain 15 after the LD instruction, 61 after
the ADD instruction and 39 after the SUB instruction.  Hence, the instructions
compute 15 + 46 - 22.
Notice that the ADD instruction requires the accumulator to be specified
explicitly, whereas the accumulator is implied in the SUB instruction.


C:\WP\WP51\DOCS\ANDIZ80                                  Doc 1 Pg 1 Ln 1 Pos 12
```

WordPerfect 5.1 running in VirtualBox 7.0 with 132 x 50 text mode

In Part 3, I continue with spreadsheets, finance programs, and databases.

Home | Contact Us