



How to compile MS-DOS 1.25

The source code of Seattle Computer Products's release of MS-DOS 1.25 was published on GitHub by Microsoft and may be freely published and distributed by anyone. This tutorial will help you to compile the MS-DOS 1.25 source code.

1. How to compile?

MS-DOS 1.25 was completely written in 8086 assembly so it needs to be assembled using an assembler (MASM and SCP 8086 Assembler). The object file(s) produced by MASM will then be linked to produce an executable which will then be converted to a raw binary executable (.COM). The .HEX files produced by SCP 8086 Assembler will be converted directly to raw .COM files using the HEX2BIN utility.

2. How to make DOS run?

Well, this is a bit challenging as the source code for the DOS BIOS (IO.SYS) was designed for SCP's own hardware and is not PC-compatible. Theoretically the compiled binaries should work on SCP's hardware but using the source code release by Microsoft alone will not produce a valid bootable disk as no source code for the boot sector is given. I do have access to the source code of the boot sector but as a result of Microsoft not releasing it publicly, I will not share it. This basically means with only the source code published by Microsoft, it is impossible to produce a working copy of MS-DOS 1.25.

3. What is required

The following is required for compiling MS-DOS 1.25:

1. Microsoft Macro Assembler (MASM)
2. Microsoft 8086 Object Linker (LINK)
3. EXE to Binary Conversion Utility (EXE2BIN)
4. Seattle Computer Products 8086 Assembler (ASM)
5. HEX to Binary Conversion Utility (HEX2BIN)
6. A copy of the MS-DOS 1.25 source code

4. Details

The files STDDOS.ASM and MSDOS.ASM can be assembled to produce the DOS kernel and they are MASM compatible. All switches are set in the file STDDOS.ASM. MSVER generates binaries for MS-DOS with Microsoft copyright strings, IBM produces IBM PC-DOS binaries with IBM copyright strings. There is no need to change HIMEM and DSKTEST unless you want to experiment more with it.

COMMAND.ASM can be assembled using MASM to produce the command interpreter for MS-DOS. Just like the DOS kernel, you may use switches to produce both MS-DOS and IBM PC-DOS binaries. The HIMEM switch is also present.

ASM.ASM, HEX2BIN.ASM, IO.ASM and TRANS.ASM were written by SCP and must be assembled using their own assembler. IO.ASM have switches to configure the floppy disk controller and the Cromemco 4FDC is currently supported by The SIMH Altair 8800 Z80 simulator.

5. Compile it

Now, we are ready to compile MS-DOS 1.25 for the first time! If you are experienced in assembly, used those assemblers and tools listed above, then you should be able to get everything done by yourself. For those unable to assemble it on their own, my build disk will save your time.

Here is how to compile MS-DOS 1.25 with the build disk:

1. Download the build disk [here](#)
2. Create a new virtual machine using your favorite emulator or hypervisor

3. Add a floppy drive, load the build disk
4. Start the virtual machine (it should boot from the build disk)
5. Check the filenames using the DIR command and it should return:
6. Type in MK <FILENAME> to assemble that particular component or MK ALL to assemble everything.
7. Once you have everything assembled, type in DIR *.COM and DIR *.SYS to see the executables produced, it should be similar to:
8. You have done it! You can now copy those executables to another disk or extract them or run them!

6. Experiments

Now we can experiment more with the source code and make modifications! Go ahead and try assembling it with the IBM switch on! Try customizing the strings and add easter eggs...

7. Summary

Now we have MS-DOS 1.25 compiled/assembled and here are the binaries just in case.

Top