Why We Love FreeDOS

Stories from developers and users about why we love FreeDOS

Why We Love FreeDOS

Stories from developers and users about why we love FreeDOS

Jim Hall, ed.

Thanks to everyone who responded to interviews for this book.

I also want to thank everyone who is—or has been part of—the FreeDOS Project since 1994. FreeDOS would not have succeeded were it not for Jerome, Ralf, Tom, Gregory, Eric, Tk Chia, Steffan, Bart, Pat, Aitor, Bernd, Wilhelm, Mercury, Michael, Paul, Liam, Bret, Rugxulo, and everyone else. Thank you!

Table of Contents

What is your earliest memory of using DOS?

I first used DOS on a home computer, on the IBM Personal Computer 5150. At the time, I was amazed by DOS. I had previously used only the Apple series of personal computers, which had a minimal interface that only let you enter BASIC programs and run them.

By comparison, the DOS command line was much more powerful. DOS offered tools and commands that you could use to process data.

We later replaced the original IBM PC with newer models, and continued to upgrade the DOS operating system. I explored the new features in every release.

For me, DOS 5 was a turning point. Until then, the DOS command line was functional but unimpressive. But DOS 5 included a host of new tools and features, including a rewritten BASIC environment called QBASIC.

Ralf Quint

Just before Christmas in 1981, my former boss got from the newly opened Computerland store in Bonn an IBM PC on loan, to test the suitability to port our civil engineering software, which up to this point ran on various HP calculators including the HP-67/97/41 and HP Series 80 desktop computers. I was able to take the system home with me over the Christmas and New Year holidays to give my boss a report what I thought about the capabilities.

This was an IBM PC with 256 KB of memory, both MDA and CGA video adapters and matching monitors, dual 320 KB 5.25-inch floppy disks and a third party Tecmar 10 MB hard drive, with a patched PC DOS 1.1 to support the additional space of that humongous 10 MB hard drive.

Needless to say, trying out one of the first IBM PCs in Germany at the turn from 1981 to 1982 was rather underwhelming. A couple of months earlier, a couple of hours after the SYSTEMS computer fair in Munich show officially closed, we had played with a brand spanking new Sirius 1, known as a Victor 9000 here in the United States, running CP/M-86. But that machine had been loaded with the full 896 KB of memory, boasted a graphics resolution of 800×400 pixels in monochrome only, but that was *huge* back in those days. And of course, it featured dual 1.2 MB 5.25-inch floppy drives.

Porting software to PC DOS and MS-DOS was put on the slow burner for about a year until we got a pre-series HP-

150, a not-very IBM "compatible" PC with a 9-inch touch screen and using the same 3.5-inch floppy drives as the HP Series 80 computers we were already using, connecting externally via HP-IB, and running with HP's adaptation of MS-DOS 2.0.

We never got very far on that one either, as a short time later, we got one of the first HP-9816 workstations, running a Motorola 68000 CPU and either HP's Rocky Mountain BASIC or UCSD Pascal.

Gregory Pietsch

I started using MS-DOS in 1983. I was still a teenager and my father purchased a few PC clones from a company called Corona. I trained myself to run programs on it.

When I got to college, I had an 80286 computer running MS-DOS. I kept upgrading it until I couldn't do it anymore.

Burkhard Meißner

I got my first impressions of PC DOS when I was visiting a friend at my old College, St. John's, Oxford, in 1984. There was a semi-secret room with an IBM PC, which selected students had access to and could do word processing with WordStar, provided they brought their own diskettes.

I instantly decided to get such a machine for myself, to learn how to program it, and to eventually teach it writing and analyzing ancient texts, especially in Greek. This was what I did during much of the 1980s, besides writing my doctoral dissertation on ancient Greek historiography.

In April 1986, immediately before buying my own personal computer, I was granted access to a networked PC with a hard disk at the University of Heidelberg. Very few people used the machine, so I had plenty of time and opportunity to acquaint myself with its features. In the accompanying documentation, there was a self-training text book which I read thoroughly.

The last lesson I completed on that day was "Formatting a hard disk." When I returned home in the evening of that rainy day, I finally saw the news about the Chernobyl reactor crash two days before.

Diego Zuppardi Bernini

My first contact was in the late 1980s, at my mother's job, but only as a launcher tool. We used floppy disks at the time.

In the early 1990s, I started learning and using MS-DOS 3.3. I found it easy to use and far better than BASIC from other personal computers.

At high school, they had some MSX computers, so I used MSX-DOS too. I quickly moved to MS-DOS 5 for real work and for better hardware support.

Franco Bugnano

I started using DOS as a little kid on an Olivetti M24 machine in the mid to late 1980s. At the time, I was mostly playing games like Microsoft Decathlon, Alley Cat, Digger, Hopper, Tapper, and Moon Patrol.

Other than that, I was also learning to program using GW-BASIC, and I was intrigued by the graphics and sounds that a PC could produce.

Anthony J. Williams (Rugxulo)

Our first IBM PC was in 1994: an 80486 SX (no floating-point unit) 25 MHz with 4 MB of memory and 170 MB hard drive running MS-DOS 6.00 and Windows 3.1. It came with a floppy drive, CD-ROM drive, SoundBlaster16, mouse, and King's Quest 6 on CD-ROM. I thought King's Quest 6 was brilliant.

The very first time I saw a DOS batch file was when downloading the shareware version of Ed T. Toton III's UFO card game, a clone of the Uno card game. It took thirty minutes to download over our 2400 baud modem! I hoped it was worth the delay, and it was. I had lots of fun with that game, especially the voice sound effects.

Arad Zinalpoor

I got interested in virtual machines and my first choice was DOSBox. After that I moved on to VMWare, which I used to experiment with all DOS operating systems and installing Windows on them.

Javier Gutiérrez Chamorro

Even if I played a bit with an Amstrad 1512 running MS-DOS 3.20 at my father's office, it was not until Christmas in 1990 that I was gifted an Amstrad PC-2086 that came with MS-DOS 3.30. It was an affordable machine with lots of special particularities, specially its Paradise VGA graphics.

Then I moved to DR-DOS 3.40 and PC DOS 5.00. After that, I "specialized" in niche DOS, including PC DOS 2000, PTS-DOS, and then OpenDOS featuring the best of DR-DOS in an "open source" fashion.

Berki Yenigün

In the student dorm in 1994, there was a computer with DOS and Windows 3.11. We all tried to get some time on it to play Descent. When some people were playing, others were watching and giving "advice." I must say it was my first contact with a real PC after a Commodore 64 and it was so different. No direct BASIC interpreter, a hard disk, so many things to configure, directories, and a weird `C:\>` prompt.

Without Internet access at the time, and without a manual, it took some time to get the hang of using DOS, but at least if you typed `HELP` there was a decent help system.

Mercury Thirteen

Way back in the 1990s, my uncle, a professor at Carnegie Mellon University, found a portion of his building in the midst of a systems upgrade. My love of all things digital apparently made a lasting impression upon him, as he later told me I instantly came to mind when he saw all those old systems getting loaded up for recycling.

He grabbed one off the stack and gifted me with the first computer of my childhood: an 8088 IBM PC XT with no hard drive, that love-it-or-hate-it clicky keyboard, and the iconic green on black phosphor monitor. As any young nerd would be, I was beside myself with excitement!

I proceeded to learn the beginnings of programming over the next few years by entering—and sometimes even translating from other dialects—BASIC programs out of every book and magazine I could get my young hands on. Beneath the BASICA interpreter, though, the computer ran PC DOS 3.20, the first operating system I cut my teeth on.

Although I found it archaic and clumsy at first, it quickly grew to hold a certain charm for me. I remember feeling like some kind of wizard when I figured out the exact syntax to make DOS jump whatever hoops I required of it. I worked to improve my comprehension of the operating system, and by the time I moved to my next computer—an 80486 hand-me-down—I was comfortable enough at the command line to feel right at home with my new computer's operating system, MS-DOS 6.22.

Even though it also had Windows 3.11 bundled with it, I didn't much care for Windows at first. In my young mind, it felt clunky and limited, like a shallow attempt to achieve something it couldn't fully attain, and I often found myself avoiding Windows in favor of the old faithful command line.

Steven Neal

My earliest clear memory of using DOS was working with my father and EA technical support to create a boot disk for Ultima VII: The Black Gate. I was already 15 years old but our family had used a VIC-20 and an Amiga 500 earlier in life. We transitioned directly to Windows 98, skipping MS-DOS and early Windows PCs. My father had some experience with DOS from work, but we both needed the technical support people to help us with this.

We never did get that version of the game working, but I have been playing with these things on and off ever since.

How did you first get interested in FreeDOS?

We started FreeDOS in 1994. I was an undergraduate physics student, and I used DOS to run all my applications. I had experimented with an early Linux distribution and dual-booted my computer with DOS and Linux. I was impressed at how developers from around the world came together to create this Unix-like system.

I read articles in computer trade magazines about how Microsoft was going all-in with Windows. If you remember Windows 3.1, you know that Windows was not very good. I never used Windows 1.0 or 2.0, but Windows 3.0 was slow and buggy. A poorly behaved application could freeze the entire system. Windows 3.1 didn't seem much better to me; it was also slow and prone to freezing. I used Windows 3.11 briefly, and though it was more stable than the versions before it, I was stuck on how Windows didn't offer very many features. I thought I could do my work more quickly using the DOS command line and DOS applications.

So you can imagine that I wasn't very keen on Microsoft going all-in with Windows. Yet the trade magazines said that with the next version of Windows, DOS would be "dead."

I didn't want DOS to die, and asked on several Usenet discussion groups if anyone was working on a free version of DOS using the "Linux" model. The response was generally, "No, but that's a good idea, and you should do it."

I had written my own command line tools that replaced or improved on the DOS command line, so I started there. I also wrote a few other programs that replicated the basic DOS features. I provided the source code for everything and wrote a brief "manifesto" of a new project that would create a free version of DOS, and shared it on Usenet in June 1994.

I found hosting on the SunSITE server at the University of North Carolina-Chapel Hill. SunSITE later became Ibiblio, which is how many people know it today.

The project quickly became popular. Over the next few days, developeres reached out to share their own tools that replaced or improved on the original DOS, and a volunteer started writing a new command shell that eventually became FreeCOM. We released everything under an open source license in our new "freedos" folder at SunSITE.

A friend connected me with Pat Villani, who had already written a DOS-compatible kernel called DOS/NT, and he contributed that to our FreeDOS Project as DOS-C, later renamed the FreeDOS Kernel.

Ralf Quint

Sometime in 1994, I saw an announcement from Jim about a new project to create an open source MS-DOS clone in a Usenet group.

I had pretty much worked with DOS from the inside out at my current job for about eight years at that point. And Microsoft had announced the end-of-life for MS-DOS. And I had gotten my feet wet with open source software trying to run early versions of Linux. So I was curious how well FreeDOS would work compared to MS-DOS 6.22, which was our workhorse back in those days.

I don't recall what might have been the first thing I tried to run on FreeDOS, but probably some versions of Turbo Pascal and Turbo C were among my initial tries.

Diego Zuppardi Bernini

In the mid 1990s, I was always looking for a free and open source operating system, and I found FreeDOS. I was very happy because I was upset about the DOS end of life. I know "DOS" was still hidden in Windows 95, but I felt that it was not the same.

The first thing I did was test my Clipper's apps. Then I tried to run Windows 3.1, but only out of curiosity.

Anthony J. Williams (Rugxulo)

There was *way* too much shareware back then. I got very frustrated, so a lot of my searching stemmed from that. We used to dial a lot of bulletin board systems until 1997 or so.

I don't remember exactly what I first did with FreeDOS, probably some trivial assembly utilities. But I do remember making floppies of the FreeDOS Beta 8 distribution.

I had first used DJGPP and Allegro back in 1997 with MS-DOS, but I didn't know what I was doing. My experiences with FreeDOS came a few years after that. I didn't even start using DJGPP again seriously until 2007.

I did buy a pre-standard, DOS-based book *Learn C++ Today* (1995) in 1998 but only halfway read it and didn't fully understand it. My attempt to learn Forth from GNU Forth 0.5.0's tutorial in 2005 failed, but I did finally learn Pascal in 2010.

Gregory Pietsch

I stumbled upon FreeDOS about 1998. At the time, I thought, "free DOS clone, it must be a fun project." The first thing I did was take a look at the code on the website, taking a mental note about things that could be improved upon or things that were missing.

Berki Yenigün

I don't remember how I first heard about FreeDOS, but I was interested in free software for a long time. When I got my first PC, I quickly installed a Linux distribution. After the second or third try and some effort to get used to it, Linux became my daily driver.

I really liked the “copy it, share it with your friends and improve it” philosophy instead of the “don’t make unauthorized copies” injunction. I liked it so much that after my switch, I emailed Richard Stallman, thanking him for starting the Free Software Movement. He took the time to kindly answer a random stranger, explaining the difference between the open source and Free Software approaches, the latter focusing on user freedom.

Regarding FreeDOS, what got my interest was freedom from vendor lock-in when I wanted to go back to the good old days. Why bother trying to get a copy of MS-DOS when you can a get much better DOS more easily? The fact that users could have control, and not be dependent of a company’s wishes, was very interesting. Also, this is a great way to keep all the software from the DOS era functioning.

The first thing I did was to grab a copy of Descent and try to finish the game again!

Burkhard Meißner

It must have been in 1998 or 1999; by then I had switched to machines with MS-DOS and Linux installed in parallel, to be booted alternatively.

At some stage I also started using some DOS programs under DOSemu on Linux. And since FreeDOS was mentioned as especially suitable to be used in connection with DOSemu, I used FreeDOS, quickly realizing that its functionality surpasses that which MS-DOS provided at any time. Eventually I dropped MS-DOS altogether in favor of FreeDOS. I especially like long file name support.

Javier Gutiérrez Chamorro

At that time I was very active in different communities, and FreeDOS sounded interesting. I tried it, and at first glance was impressed that it simply worked!

Open source was not so popular at the time, with the exception of Linux, so having a DOS compatible environment which was free to use was great. I remember playing with the Micro-C Compiler, the tool recommended at that time for creating FreeDOS programs. It was years behind the Borland C Compiler, but was a good start.

Franco Bugnano

I don’t remember exactly when I first heard about FreeDOS, but it was in the early 2000s.

At the time I, was mostly interested in Linux and open source software, so I must have read somewhere that there is an open source version of DOS. Or was it because DOSemu shipped with the version of Linux I was using at the time, maybe Red Hat Linux 7, and that included a version of FreeDOS? I don't remember.

Other than running it for the novelty factor a couple of times, I didn’t do much with FreeDOS at the time.

I rediscovered FreeDOS in 2020, when I was researching about DOS memory management, and also discovered the PCem emulator. I was curious about how FreeDOS manages memory compared to MS-DOS, and I was pleasantly surprised about how memory efficient its mouse and CD-ROM drivers are.

I also noticed the lack of GEMMIS support in JEMM386, and linked to some documentation in the JEMM386 GitHub issues.

Other than memory management, I tested FreeDOS compatibility with some games, and I was impressed how good it is. FreeDOS ran almost every game I tried, except for Ultima 7.

Mercury Thirteen

I had heard of FreeDOS from colleagues in the early 2000s, but only got into it in earnest as part of my job as Tech Administrator at a small machine shop. The owner was always on the lookout for inexpensive machines he could buy and fix up, either for use in our shop or to sell on to others.

In his travels, he located a small lathe which was in good condition visually, but needed some repairs before it could actually be useful. While our electrician and machinist worked on fixing up their respective parts of the machine, I got the job of finding a suitable replacement for the non-working control system it used. The control systems on these machine tools were usually proprietary units designed by key players such as Siemens or GE, without which a machine tool amounts to nothing more than a very heavy doorstop.

Because of their key importance, these control systems can cost tens of thousands of dollars, depending on their complexity. Thankfully, this was only a small machine, and we planned to replace the broken control with a like model for probably no more than a few thousand dollars, if we were lucky enough to find one for sale.

Imagine my surprise when I opened the cabinet to find a low powered DOS PC running the show instead! In an imaginative twist, this lathe's manufacturer opted to use a regular old desktop computer running a custom application

using DOS4GW to supply the logic and user interface, and simply designed some custom PCI cards which interfaced to the machine as opposed to spending thousands to bring their machine to life using a traditional control system.

The owner was nearly beside himself when I told him how much money he would *not* have to spend, since FreeDOS is more than capable of running the machine's proprietary software. All I needed to do was replace the computer's defective hard drive. We sold the machine, and some years later crossed paths with its buyer once again. He was happy to say that—to that day—he still frequently turned out parts on his FreeDOS powered lathe.

The first thing I did or tried on FreeDOS? I ran pretty much every program I had. I had to see how well it lived up to its impressive compatibility claims! I threw everything from games to text editors to compilers at it, and it handled each with ease.

Steven Neal

I found FreeDOS a few years ago because I have children of my own now and wanted to share this world with them so they can understand what computers are actually doing on the inside. Most of what people interact with is touch

screens, designed to hide all the inner workings. You don't learn much doing it that way.

Using an older DOS based machine was a simple way for me to give them age appropriate computer programs like games without Internet connectivity and at low cost. Most modern software, including operating systems, require Internet connections for licensing and verification.

I still have access to plenty of old software for my kids to choose from but lacked an operating system. I was able to get an older machine for free from someone's basement and quickly found FreeDOS, which appealed to me. Using MS-DOS without a license is still pirating software and Microsoft won't sell you a license anymore and there are no updates. Using FreeDOS was the obvious choice.

Arad Zinalpoor

After realizing it's much more stable for today's DOS usage, I immediately fell in love with FreeDOS. I found out about FreeDOS after reading about it on a computer science encyclopedia. The first thing that I remember I did was trying out its games and networking programs.

How do you run FreeDOS today?

I don't actually run FreeDOS on real hardware these days. I just don't have the space to set up an extra computer, especially a "classic" PC with the accompanying CRT display.

Instead, I run FreeDOS in a virtual machine on my Linux computer. I like VirtualBox because it is fast, although it lacks PC speaker emulation. QEMU provides better hardware emulation, but seems to run a little slower.

Booting FreeDOS in an emulated environment has its benefits.With a virtual machine, I can easily set up multiple instances of FreeDOS: one for my regular work, and another for testing new releases. If I break my testing environment, my regular work is unaffected. And I can easily set up another test instance.

Today, many people run FreeDOS in a virtual machine. But there is a dedicated core of fans who like to run FreeDOS on legacy hardware.

Berki Yenigün

In Turkey, and probably in other developing countries, many computers are sold with FreeDOS to avoid paying the Windows license fee. When I bought a laptop a few years ago, I chose one that came with FreeDOS. I installed a Linux distribution and kept something like 1 GB on the hard disk for FreeDOS, so it is now a dual boot setup.

Most people simply wipe the disk and install something else. My way of giving back is to help people learn more about FreeDOS by translating, especially the documentation, so that others may learn about FreeDOS and want to keep it around.

Other than that, I use virtual machines for testing purposes, with VirtualBox and sometimes with QEMU for the PC speaker support. With VirtualBox, the setup is almost the default; just typing `FreeDOS` as the name in VirtualBox automatically sets the operating system family as DOS. I just give the system plenty of memory (128 MB) and disk space (4 GB).

The main difference with the default suggestions is the disk type, which is VHD instead of the default VDI, so that I can exchange files between the host and the guest systems.[1]

1 See http://wiki.freedos.org/wiki/index.php/VirtualBox_-_Chapter_6 for more information about transferring files between a virtual machine and the host system.

Anthony J. Williams (Rugxulo)

I still use MetaDOS (customized version of FreeDOS) under VirtualBox occasionally, but mostly I just boot up a USB flash drive with a custom FreeDOS install on real hardware with my old 2010 Dell laptop. I've gotten back into programming as a hobby in recent months, so I always use that. It has a huge 128 GB FAT32 drive with only a few GB used. Mostly I boot up with a very large RAM drive and copy a few files there to manipulate and play with.

Burkhard Meißner

Today, I have all my machines set up as Linux machines, using FreeDOS in virtual machines or emulators. The emulator used most is DOSemu2 which is especially good at using DOS extenders and meeting huge memory requirements.

I also use Oracle's VirtualBox from time to time. DOSemu2 is set up to provide direct access to the Linux file system; I do not have any DOS file systems left. VirtualBox has the advantage of easy access to do networking.

Arad Zinalpoor

I usually run FreeDOS on virtual machines, where it runs incredibly fast with 2 GB of memory. FreeDOS performs network activities very well in bridge mode.

Diego Zuppardi Bernini

I mainly use FreeDOS on a virtual machine; the typical set up is 32 MB memory, 2GB hard disk, with network and sound card enabled. I love restoring old computers and I use FreeDOS and OpenGEM on most of them.

E. C. Masloch (ecm)

I started out running DOS on a 2001-vintage machine, with an i686 class Intel Pentium 3 processor, that I got in about 2006. At the time, it was running Windows ME and its bundled MS-DOS 8.00. I still have that machine in my room and occasionally boot it, though now usually running MS-DOS 7.10 from Windows 98 SE.

I have also used an older FreeDOS kernel on this machine. I used the FreeDOS editor from FreeDOS 1.0 as my main editor on there, renamed to `FDED`.

As an example of what I still use it for, I recently built a new revision of my debugger[2] with a special build option to allow checking whether the Trace Interrupt `01h` has its own `FL` word's Trace Flag set or clear.[3]

Since 2007, I ran the modified MS-DOS bundled in the 32-bit Windows XP NTVDM on an AMD processor. In 2013, I switched to Debian Linux, and picked up DOSemu and then DOSemu2, both running a FreeDOS system.

Around 2014, I started using my partner's server across SSH, running with Debian Linux on AMD64 processors, as a platform for DOS development. It also runs DOSemu2, which I build on the server itself from the DOSemu2 Git repository. It runs without access to either real V86M or KVM, so that V86M code needs to be simulated while DPMI code may run natively. Occasionally, I use QEMU as well, either packaged by Debian or compiled by me.

Most of the time now, I connect to the server via SSH in an Android app called ConnectBot from my mobile phone running Hacker's Keyboard. On the server, I use a session of a Linux program called `screen`, which allows me to switch between different terminals and keeps my session alive even

2 See https://hg.pushbx.org/ecm/ldebug/rev/45067f20604b

3 See https://stackoverflow.com/questions/72116029/does-single-step-interrupt01h-interrupt-clear-tf-flag-after-every-instruction#comment127422313_72116029

if the phone gets disconnected and has to reconnect to the server. Most of the usage and development of DOS I do today is with this mobile phone and server setup.

In DOSemu2, I use an assortment of programs, mostly from FreeDOS. Some of those are from a package that was intended to bundle FreeDOS for the old DOSemu. However, I have updated the kernel multiple times in recent months to ones I built myself on the server using TK Chia's IA16 GCC compiler.

Gregory Pietsch

To be honest, I don't run FreeDOS all that much. The last time I tried, it was as a virtual machine. But I am vested in the success of the operating system, and I am nostalgic about DOS.

DOS was my go-to operating system for twenty years or thereabouts. Microsoft has not given away the source code to MS-DOS, or made it free and thus improvable by others, so FreeDOS is the only real way I or anyone else can take this clunky operating system and improve it.

Franco Bugnano

As much as I would have liked to run FreeDOS on my Olivetti M24, running FreeDOS on such old hardware is not ideal, and there is a video on YouTube that shows the hurdles of running FreeDOS on an IBM 5150. For such old machines, I think that MS-DOS version 3.3 is unbeatable in terms of memory consumption, disk space required, and features offered.

I'm running FreeDOS emulated both on PCem and 86Box. I'm emulating an 80486 DX2 66 MHz processor, 32 MB of memory, and a CL-GD5428 VLB SVGA card, and a 2 GB hard drive. I had a similar Olivetti M4 PC at the time, except less memory.

FreeDOS runs well on that machine, except for some unresponsive user interfaces both in the installer and in FDIMPLES, which isn't a big deal.

Javier Gutiérrez Chamorro

I still have an old Pentium II computer with 32 MB of memory running DOS. I used to upgrade from one distribution to another from time to time. I do not use this computer very regularly, but I think it has MS-DOS 7.1 currently. My daily usage is with VirtualBox virtual

machines—if we don't take into account DOSBox, which is a nice project too.

My setup is targeted more on nostalgia than usability. I use QEMM 97, DESQview, and also a custom personal distribution I have created by mixing the latest versions included in DR-DOS since version 7.01 to the controversial DR-DOS versions 8.0 and 8.1.

Also I have a FreeDOS 1.3 image where I like to play with open source tools, including the OpenWatcom V2 fork.

Mercury Thirteen

Currently, I'm running FreeDOS in a variety of places. I have one VirtualBox virtual machine in use on my desktop to play games or use actual DOS software, along with another virtual machine that gets cloned as an isolated development area for each new DOS-based project I dream up. Right now I have no less than half a dozen virtual DOS machines on my system!

In the realm of real hardware, I have my "Devbook," a Compaq Mini 102 netbook computer running FreeDOS to test the performance and usability of any software I need on an actual system.

Ralf Quint

When I get the time and the space for it, I run FreeDOS both on "real iron" and on VirtualBox. On hardware, I have two laptops (one Dell and one HP, both Intel Core Duos), a Compaq Presario mini tower with a 500 MHz PIII CPU, and a Gateway desktop with a fan-less 300 MHz PII CPU which currently has a dead PATA hard drive. I also tried to install FreeDOS on various incarnations of PCem, but it's too bad that this emulator has since been abandoned.

Steven Neal

The main setup I used was a circa 2003 computer that had originally been set up with Windows XP but later in life became a Linux based server in someone's home. That person no longer needed it, so they gave it to me. It has a 2.2 GHz Pentium 4 CPU, 2 GB of memory, 300 GB hard drive, CD-ROM, floppy drive, and (awkwardly) and Accelerated Graphics Port (AGP) 2.0 slot on the motherboard from back when that interface was a serious competitor. The power supply died on it a while back and I need to either replace it or find a new system.

What's cool about FreeDOS 1.3? What features do you use the most?

I love that people find different things they like in FreeDOS. One of our goals in the FreeDOS Project was to create a modern version of DOS. FreeDOS 1.3 was released in February 2022, with a long list of features and included applications.

Compatibility critical in FreeDOS. We try to carefully balance the features and behaviors in FreeDOS with the original DOS. At the same time, we aim to include new features found on more modern systems. FreeDOS provides all the commands that replicate classic DOS, but we also include new device drivers, compilers and programing tools, editors, archive programs, networking, games, and "desktop" applications.

I often use the IA-16 port of the GNU C Compiler and the OpenWatcom C Compiler to write new programs. My favorite editor is FED. I like playing several of the DOS games, including Simple Senet and Block Drop.

Anthony J. Williams (Rugxulo)

I use the FreeCOM shell for batch files, and HimemX for extended memory (XMS) more than anything else. I still use TDE as my text editor, which I've been using for almost twenty years! Other than that, I mostly play with various assemblers like NASM, FASM, A86, Wolfware WASM, and A72; compilers like DJGPP, FPC, OpenWatcom, and P5; and interpreters like SED, AWK, and REXX.

Franco Bugnano

First of all FreeDOS is cool from an ideological point, being fully open source. Then, FAT32 and LFN (Long File Name) support out of the box is really cool.

All the included software that is easily installable with FDIMPLES is FreeDOS's killer feature, in my opinion. Personally, I love the Doszip Commander file manager. It is so convenient and lightweight, it should be part of the base install.

Other than that, I use the OpenWatcom C compiler for developing programs for DOS. I love that it is packaged with FreeDOS and ready to use.

Diego Zuppardi Bernini

There are a lot of cool tools in the FreeDOS 1.3 release, such as networking, 8086 FAT32 support, an updated command line interpreter, an updated kernel, tons of applications and utilities, and more.

I use FDIMPLES (package manager), developer tools like IA-16 GCC and OpenWatcom, Internet text browsers, and emulators for ZXSpectrum (my favorite 8-bit computer).

Javier Gutiérrez Chamorro

One think I love, and that was dreamed when only commercial DOS was available, is the possibility of having different kernels, each compiled with a different compiler and optimized to a different CPU. It is so nice to see how new hardware and capabilities have been implemented, with huge memory in JEMM386, FAT32 support, and fast disk caches and virtual disks.

E. C. Masloch (ecm)

I am very happy that there is a DOS distribution that promotes and mostly contains Free and open-source software. I believe that all software ideally would be Free software. In turn, I mostly use FreeDOS to develop DOS applications or other system components such as the loaders, kernel, or shell.

Burkhard Meißner

Very important is the compatibility with MS-DOS. Also very important is long file name support. And the sheer number, quality, and functionality of utilities far exceeds that of MS-DOS.

Berki Yenigün

What I really like about FreeDOS 1.3 compared to previous versions is the built-in networking support. Everything is detected automatically in VirtualBox. Just type `fdnpkg checkupdates` and admire the result. Last but not least, many bugs from FreeDOS 1.2 were fixed.

Gregory Pietsch

The cool thing about FreeDOS is that there are many die-hard DOS fans out there that have contributed to the success of FreeDOS over the years. In regard to the features, I have attempted to get the best out of the operating system with each program or library I write.

Mercury Thirteen

The updates! It's nice to have the latest kernel, FreeCOM, and other packages right out of the box without having to set up networking and update everything manually.

Why is FreeDOS important to you? Why do you run FreeDOS today?

I think FreeDOS fills an important niche in computing. FreeDOS started as a simple replacement of classic DOS, and it has maintained that core identity. If you need to run 16-bit DOS applications, you can do that very easily using FreeDOS. While you can also use other DOS-like environments such as DOSBox, only FreeDOS provides a full DOS command line experience.

DOS played an important part in desktop computing. We had countless DOS applications and games that filled every use case throughout the 1980s and into the 1990s. Businesses relied on DOS, especially the now-standard "office" applications including word processing, spreadsheets, presentations, and email.

And with FreeDOS, you can continue to run those great applications. DOS lives on. I'm glad that FreeDOS is part of that history.

Berki Yenigün

To play old games, and sometimes to play with software that I get from the BBS servers that are still online. I play new games, too. Right now, BlockDrop is eating too much of my time. It's better than many shiny smartphones games, and it has no ads whatsoever!

Burkhard Meißner

There are two application programs which I still use quite regularly, and they are more or less bound to the operating system: WordPerfect 5.1 (word processor) and a text analysis program for Greek and Latin texts which I developed in the 1990s. The latter is written in assembly language and MacroSPITBOL, for which there was a powerful compiler in those days which allowed for external assembly routines to be linked in at runtime.

For some reason, SPITBOL (Speedy Implementation of SNOBOL) for Linux lacks this functionality, and since I need to use my program (called View&Find) as well as the word processor quite often, I will certainly continue using FreeDOS for the foreseeable future.

Actually, I think word processing using DOS is more efficient for the author than under newer operating systems

with their huge graphics overhead and all sorts of distractions and gimmicks that may serve the casual user, but take away efficiency from the regular writer. For me, FreeDOS helps concentrate on the essential, and it is also great fun.

Anthony J. Williams (Rugxulo)

FreeDOS is simple to use and comfortable and fits the bill for simple programming with files, strings, and arithmetic. I will also occasionally play a game, but that's rare these days.

Arad Zinalpoor

FreeDOS is like a reboot of a great game. MS-DOS was just getting old, but FreeDOS pushed its way into the present day, containing the same usefulness as the DOS we all used to have.

Diego Zuppardi Bernini

FreeDOS and other open source operating systems and tools are fundamental to give opportunities for all people to use, learn, and develop skills in the digital world. This provides a fundamental platform to be more equal and fair, especially for people who don't otherwise have access to software licenses and appropriate training.

This also gives the opportunity to learn and experiment even for those who aren't in the IT field, or those who have few or no computing skills.

No less important, open source operating systems are more secure, because you can audit, improve or modify them as you need.

FreeDOS also demonstrates that if people work together, wonderful things could happen. FreeDOS developers make a great gift to all of us, giving the opportunity to work, learn, or do whatever we want or need with the operating system.

Mercury Thirteen

I've moved on from other more DOS-centric positions in my career. At this point in my life, I mainly use FreeDOS for development of new and exciting software!

Franco Bugnano

I find it fun to tinker with operating systems different than Linux from time to time.

Javier Gutiérrez Chamorro

When developing, I like the freedom DOS provides. You have full control and full access to the hardware. You run on real hardware, not in a virtual environment controlled by the OS. With FreeDOS you have also the freedom of it being free.

E. C. Masloch (ecm)

I started out on DOS and got involved in Assembly and C programming, thanks to interesting documentation such as what Henrik Haftmann wrote about DOSLFN in an older German file and in a newer English one. It is inertia that keeps me coming back to FreeDOS, and occasionally other DOS systems in this day and age. I like to think of myself as an amateur developer, one who does this because they love what they do.

Steven Neal

There are several reasons, not least of which is because it is the modern and actively supported version of DOS. It is also easy on hardware, so I don't need to buy expensive parts to run it.

FreeDOS does not constantly run some "Updater" program in the background, nor send me push notifications about whatever it deems important. When I boot up FreeDOS, it just works and lets me get straight to my programs.

Modern systems have their conveniences, but running FreeDOS gives me the same satisfaction that using a stick shift transmission does. It does this by providing a sense of control and connection; you always feel that if there is a problem, it is solvable, and finding the solution will be enriching.

FreeDOS is also partly a nostalgia thing, especially as I missed out on the DOS era of gaming, having an Amiga 500. It is just plain fun to play DOS games like Dark Forces, Betrayal at Krondor, and Ultima Underworld on a legacy computer. I miss my manual transmission car and motorcycle, but I can keep using FreeDOS and I will never run out of garage or basement space for it, as happened to the vehicles.

Ralf Quint

Personally, for mainly nostalgic reasons. At some point over twenty years ago, I was part of a project in Los Angeles that refurbished old, discarded office PCs; we installed FreeDOS send them to various organizations in Middle America. Unfortunately, after some time, the organizers decided instead of using a clean and free FreeDOS on those machines, to use "liberated" Windows 95 or Windows 98. And once they ran into legal trouble, they decided to use some graphical Linux distribution instead.

But I still see a use for FreeDOS in some niche applications, like small single-board computers that require a file system but are not powerful enough to run a graphical Linux. I hope to play with such systems again.

What programs do you like to run on FreeDOS? If you could only pick one, what program would you take with you to a desert island?

My favorite DOS program "genre" is the spreadsheet. My undergraduate degree is physics, and I relied on spreadsheets to analyze my lab data. I experimented with a few DOS spreadsheets, and fell in love with As Easy As by TRIUS Inc.

I still use As Easy As today, from time to time. My fingers still remember the keystrokes and keyboard commands to activate the different features. And I find As Easy As remains feature competitive with modern spreadsheets. The spreadsheet hasn't changed that much since the 1990s.

My favorite DOS game is TIE Fighter by LucasArts. I often play it today, even though I know all the missions by heart. The game is just fun to play. And I appreciate the simplicity of flying a TIE fighter sortie against waves of rebels, pirates, and defecting Imperial forces.

Anthony J. Williams (Rugxulo)

My favorite tool is probably still SED, but I like a lot of others, too. AWK is very cool, so is REXX. Turbo Pascal (FPC) is great. DJGPP is probably a must-have, but it has so many pieces. But I still enjoy my one-floppy EZ-DJGPP v2 for C89 programming.

My favorite games are probably PSR Invaders (it's relaxing) and BioMenace (an old favorite). Retro City Rampage DX (2015) is also very cool!

E. C. Masloch (ecm)

Honestly I mostly run my own programs, chiefly lDebug (that's "L Debug" with a small "L"). And the debugger is mostly used to debug its own "debuggable" build, or one instance of the debugger running within another instance.

For the desert island, it'd be a toss-up between lDebug or a recent release of NASM (plus a DPMI host to run it), both of which would be useful to build upon. In any case, I wouldn't want to miss my very own free x86 instruction reference[4] based on the English-language reference included in older NASM versions.

4 See https://pushbx.org/ecm/doc/insref.htm

Diego Zuppardi Bernini

I like running a lot of programs for testing or daily use such as compilers, spreadsheets, database and hardware information systems. I don't really have a favorite one, but these days if I can have only one program, I would definitely choose a C compiler.

Arad Zinalpoor

One of my favorite games is Freedoom. It's free and replaces DOOM. If I had to take just one program to a desert island, it would be FreeDOS EDIT.

Franco Bugnano

My favorite program for FreeDOS is Doszip Commander. My favorite game would be Commander Keen episode 4, and that would be the DOS program that I would take on a desert island.

Burkhard Meißner

Given its special functionality, the program or system which I had written myself would accompany me on an island trip.

In the 1990s, Packard Humanities Institute and Thesaurus Linguae Graecae issued ancient texts on CD-ROM, encoded in a scheme which they call “beta code.” This code uses Latin letters for Greek scripts, special codes for special symbols and codes to switch between different coding styles; there is also a comprehensive encoding for meta-textual information (provenance, date, and references).

Since the coding was published and can be reproduced easily, I was able to transfer other textual material from other text data bases to this coding, eventually to compile a text data base that comprises much of ancient literature, inscriptions and some medieval and modern texts. Using these text data, I have produced quite a few studies of word frequency distributions on ancient texts to answer questions of authorship, genre and style. For these analyses I used the View&Find system I mentioned earlier, and software written on the spot, mostly written in SNOBOL4, SPITBOL, C, and Assembly language. I published a couple of papers about these analyses:

B.Meißner, Sum enim unus ex curiosis, Computerstudien zum Stil der Historia Augusta, Rivista di cultura classica e medioevale 34 (1992) 47-79

B.Meißner, Computergestützte Untersuchungen zur stilischen Einheitlichkeit der Historia Augusta, in: G.Bonamente u. K.Rosen (edd.), Historiae Augustae Colloquium Bonnense, Bari (1997) 175-215

B.Meißner, Polybios als Militärhistoriker; in: V.Grieb, C.Koehn (edd.), Polybios und seine Historien, Stuttgart (2013) 127-157

Berki Yenigün

That's a difficult one, but I'd say why choose just one app when you can get everything for the same price, at no cost? I particularly like PGME; it's a good app launcher that allows complete customization. In its default configuration in FreeDOS 1.3, it gives direct access to many things, including the help system and FDIMPLES, the package manager. From there, it's only a matter of curiosity to find what you like from all that's offered.

Javier Gutiérrez Chamorro

Borland C++ 3.1 is the program I spend the most time with. But I would also like to mention WordPerfect 6.1 and 6.2, the epitome in DOS evolution.

Steven Neal

One of my favorite things to do is try and get something new to run that I haven't tried before. I often start something, and once it works, I move on to the next thing. So there is a lot of change in what I like to run on FreeDOS.

For example, at one time my favorite program was MuPDF because I was just so excited to get PDFs working in DOS. Of course, it is a bit misleading to say that is my favorite program!

That said, if I could take only one DOS program with me to a desert island, it would have to be The Ultimate DOOM. I have been playing this recently, and it is just plain fun. I find I can replay it again and again, always finding a new challenge to pursue. The core game loop is so well crafted that I just keep coming back to it, making it the perfect choice for me, if I could only have one program.

Mercury Thirteen

Some of my favorites are NASM, OpenWatcom's C compiler, and mTCP to maintain connectivity between all my machines, both real and virtual.

I'd take the NASM assembler to a desert island. Then I could eventually make anything else I needed!

Ralf Quint

If it's one program only, then I'd pick Borland Pascal 7.0. The whole suite. Then I can develop everything else I need myself.

I never have been a gamer, so I don't care about those on DOS. Favorite applications are a bit tough these many years later, but I would be okay with WordPerfect 5.1 and Lotus 1-2-3. Or Lotus Symphony. Or Framework II. Or Microsoft Works.

What's the coolest program you've written for DOS?

DOS has a lot of great programs, including games, applications, tools, compilers, and simulators. If you can imagine a program that does something, you can probably make it happen on DOS.

My favorite contribution is a multilanguage library called Cats. Previously, programs supported the native language of the programmer, which was usually English. This worked well for others who understood English, but frustrating if you only spoke another language like German, Italian, French, or Spanish.

Cats was a compatible implementation of the `catgets` *library from Unix systems. Programs that used the Cats library could immediately support multiple spoken languages. I think that's pretty cool for a project with users around the world.*

Tom Ehlert updated Cats to be more efficient, removing the `catgets` *components that we didn't need, and released the new version as Kitten.*

Mercury Thirteen

The program of which I have the fondest memories would have to be a cheat code utility I wrote as a teenager, which I called CXD, the COM and EXE Debugger. However, it was less of a debugger, and more like the "code search" tool one might find in the old GameShark devices. It allowed you to search a compiled game executable for specific values or strings, then give you a list of matches through which you could step incrementally, automatically replacing the matched value with a specified one and running the game to see if the swap resulted in the desired effect.

I remember using it to hack Mike Wiering's game "Mario & Luigi" to start with 99 lives. Unfortunately, in those days, I had a very poor backup strategy (none) and the source code of many of my old creations, including CXD, became corrupted or completely lost after a hard drive crash.

The only applications I've released publicly so far are ListPCI and ListVESA, both of which answered specific needs I had while writing other software still in the works.

But back in my BASIC days, I wrote a slew of tiny programs such as BitPaint, a program to design sprites for use in games and other software, DiskWipe, a program to thoroughly wipe a drive (similar to BleachBit) before passing a PC on to another person, a pair of utilities to split files to chunks of a specified size and rejoin them once again to a single file, a Pong clone, a mathematics educational game aimed at young children featuring custom fonts and

rudimentary animation and sound, a utility to list all the vectors in the IVT, and a handful of simple geometric screensavers.

Ralf Quint

This might depend on your definition of "cool." For almost a decade, I was part of a software company that developed CAD software running on DOS, which ran circles around both AutoCAD as well as workstation based CAD systems like Medusa, CADDS, and CATIA. At that company, I also wrote a ton of Novell Netware based tools for internal use, like compiler stations, network chat programs, network print servers.

Once I came here to the United States, the "coolest" program was probably that of an environmental monitoring station in the Ballona Wetlands. That was on DOS, but it later got replaced with Windows 95B to take advantage of the larger disk space of FAT32. These days, I think it is running on Raspberry Pi and Linux.

Burkhard Meißner

The most useful will probably be View&Find. However, if coolness is meant more in the stylistic than pragmatic sense, then probably "DateChi" and "MergeChi" will have the better of it: These were two programs which I wrote for a scientific word processor (ChiWriter) that was fashionable in the 1980s and 1990s, in order to provide this word processor with a mail merge capability and the capacity to produce automatically dated documents. You would eventually end up producing Greek or Russian invoices, automatically dated and with addresses appropriately inserted from a data base. The programs were very useful for my own software marketing and employed by other ChiWriter users, too.

Gregory Pietsch

I have written many cool programs over the years. Could I mention this line editor that I think is awesome?

First of all, there's FreeDOS Edlin. I wrote it because I noticed a vacuum in the FreeDOS distribution and decided to fill it, hoping to get fame and fortune along the way. I have also written a version of ar, the Unix archiver, as well as a few other clones of Unix programs.

I also contributed a few libraries such as a version of libm and a multi-precision integer math library. I wrote those because I wanted to write a C compiler and it turned out to be too much of an undertaking to write everything that would go with the compiler: the compiler itself, the binary utilities, the development utilities, and the libraries. There was a libc clone already, but not a libm clone. Anyway, I wrote a few of the things I would need for this before my brain exploded, and thought somebody could reuse them, so I sent them in.

Franco Bugnano

In 2021, I wrote a Bomberman game clone in C for the 8086 processor with CGA graphics and PC speaker sound. I used the OpenWatcom C compiler that is included with FreeDOS, but I edited the sources with Vim on Linux.

Unfortunately I cannot release it, as I'm using assets ripped from Bomberman on NES and music ripped from the 8088 MPH demo.

Anthony J. Williams (Rugxulo)

I'm not a professional programmer, so my contributions are usually small:

I wrote a REXX script to update a gzip file's external date and time with its internal timestamp.

I patched PAQ8O8 to use `CPUID` to determine NOASM, MMX, or SSE2 at runtime.

I wrote a 7-bit ASCII executable stub in Assembly so that VMS `CONFIGURE.COM` wouldn't crash on DOS.

I wrote a Befunge-93 interpreter in several languages.

I wrote a bunch of code to convert PSR Invaders from MASM syntax to NASM.

I rewrote Makefiles to rebuild NASM 0.98.39 for 8086 using OpenWatcom and TurboC++, since previously it was only for 186.

I wrote some batch-oriented command line binary patching tools: `BINHACK` in C and `BYTEFIX` in Assembly.

E. C. Masloch (ecm)

That'd be lDebug.[5] I didn't write most of it, but I did greatly extend the base of FreeDOS Debug that I started with in 2009, plus the occasional thing picked from more recent FreeDOS Debug revisions since. I started to track my progress in a Mercurial repository beginning in 2010 (still called NDebug then). In 2021, I started to make regular releases and also noted down most of the high-level changes I had added in the manual's News sections.

Other than the debugger, I am also somewhat proud of two more projects:

First, my TSR handling as developed for RxANSI and re-used in the TSR example project,[6] in lClock (with a small "L"), SEEKEXT, and partially for my updates to FDAPM and FreeDOS SHARE. This contains what I call the "optimal installation" and the "advanced deinstallation method," and utilizes the Alternate Multiplex Interrupt Specification (AMIS) as well as IBM Interrupt Sharing Protocol (IISP) headers.

Second, the ldosboot collection (small "L" again) of FAT12/FAT16/FAT32 boot sector loaders and the next stage loader, called iniload. These especially are an exercise in size-optimized programming. This project is used by my work on the RxDOS kernel but particularly for lDebug as a bootable debugger. The iniload stage allows the program to

5 See https://pushbx.org/ecm/web/#projects-ldebug
6 See https://pushbx.org/ecm/web/#projects-subsection-tsr

be loaded in many different load protocols, as an IBMBIO.COM kernel (PC DOS 6 or 7 style), an IO.SYS kernel (MS-DOS 6 or the different MS-DOS 7 style), a KERNEL.SYS kernel (FreeDOS style), a Multiboot specification or Multiboot2 specification kernel, an lDOS or RxDOS.3 kernel, or as a DOS application or DOS device driver.

ldosboot[7] also integrates with my project called inicomp, which supports many different compression formats to compress the program, other than the outer iniload stage and inicomp's depacker. Both of those projects fully support compressed or uncompressed files larger than 64 KB, something that (as for compressed file size) the FreeDOS kernel's UPX compression support only learned recently through a contribution I made.

Steven Neal

As the machine I was using was mostly intended for my son to use, I wrote a batch program that would keep him away from administration functions and minimize the command line navigation he would have to do on his own.

I used the Phil's Computer Lab Starter Pack for FreeDOS as a base and then wrote my own version that presented the

7 See https://pushbx.org/ecm/web/#projects-ldosboot

user with a choice of games on startup, or if the correct options were selected you could go to the command line. This way, I could still use the computer without having to modify the AUTOEXEC.BAT every time I wanted access to the command line. The menu was written so that my son could choose a game, and when closing the game, he would return to the menu so that the experience was seamless for him.

This meant that he could use the computer without adult supervision, which he liked a lot. And it allowed me to leave all of my software on there so I could still play more mature games by navigating to them using the command line—but they would be out of sight and out of mind for him.

Javier Gutiérrez Chamorro

I can remember UPTIME and ZEROFILL, both of which are included in FreeDOS. Also BEEP.

The reasons for each program are different, but all of them start because they are useful to you. I started UPTIME in order to try the improvements on the FAST compiler after Bruce Axtens released FAST and SOFA as open source. I tried FAST decades ago before learning Assembly and it amazed me. So as a challenge, I decided to write UPTIME in FAST.

Unfortunately, the latest FAST version was buggy, and it required to apply several workarounds. So for UPTIME version 2, I decided to rewrite it in assembly with JWASM.

What's different about writing programs for DOS than for other operating systems like Windows or Linux?

Writing programs for a 16-bit DOS comes with a different set of assumptions. You need to make tradeoffs between memory and performance, which requires more creativity in constructing programs.

If you write programs for other, more modern operating systems like Linux, it's easy to load a bunch of libraries to do all sorts of nice things for you. Programmers might load one library to build a user interface, another library to draw game graphics, and another library to provide a graphics filter effect like shaders. On a modern computer with 32GB or more memory, there's a lot of room to load a lot of libraries.

But on DOS, programs have limited memory. The CPU speed on older systems may be slower. DOS programs need to account for these and other system limitations. Writing programs that run well on DOS requires balancing features and carefully tuning performance.

Franco Bugnano

It's a totally different experience. Writing programs for DOS is much like writing embedded firmware:

- You want performant graphics? Write directly to the video memory.
- You want non-blocking keyboard input? Write a custom keyboard interrupt handler.
- You want non-blocking music? Write the sound frequency to the timer, and write a timer interrupt handler to change to the next note.

None of these operations are possible on modern operating systems. Not in user space, at least.

And then there's memory management. If you use a DOS extender like DOS4GW or DOS32A, then memory management is no different than on a modern operating system. But if you target the 8086, then you have to deal with near and far pointers, all due to Intel's segmented memory model.

So even if you're using a high level language like C, you still have to deal with lots of low-level hardware details when programming on DOS.

Burkhard Meißner

With DOS, you have to do a lot more on your own. My View&Find system even had its own graphics system to write Greek or Coptic to the screen, and this had to be programmed differently for EGA, VGA, CGA and Hercules systems. At some stage I even had to directly program the graphics card's hardware registers for the Hercules system. The upside of it was, given the appropriate hardware, it is extremely fast.

Mercury Thirteen

To me, DOS is a palette of tools and a blank canvas upon which I can code whatever I can imagine. There's no "nanny" process protection to hold your hand along the way, no underlying system software with which you must "play nice." Just raw capability.

It welcomes you in and gives you the ability to dream up anything you want because, unlike so many other systems, the operating system doesn't stand in your way. It's actually on your side.

I think aspiring developers appreciate this simplicity because, although high-level libraries are available should they choose to go that route, DOS doesn't force them to

jump through hoops and learn arbitrary access protocols or function classes before writing a simple "Hello world!" program on their computer. DOS simply delivers the entire processing power of your system's CPU directly to whatever task you set before it, and lets you steer that power in any direction you see fit. And that's exactly how it should be.

Gregory Pietsch

DOS is a single-tasking operating system that's barely more than a memory. It only exists so that you can run other programs on top of it. Graphics tend to bog down the CPU, so the best programs you can run on DOS are simple filters such as Edlin. If you have to do windowing on a 25×80 screen or access the serial ports of the computer, the best way was to poke values into absolute addresses or peek at those values.

For Windows and Linux, you can't really do that. Native Windows code relies on a complicated event-driven model of input and output. Linux, like Unix, has its own quirks.

To write programs for DOS, remember the "KISS" principle: "Keep It Simple, Stupid." Dynamically allocate your memory instead of using a giant pre-allocated block. Make it so portable that it runs on other systems.

Ralf Quint

With DOS, you are closer to the metal. Both the DOS API and BIOS functions are rather limited, in a good way. And you have direct access to pretty much all the hardware, for better or worse.

Also, as far as memory goes, things are pretty limited, but that rather makes you think more about what you are doing instead of brainlessly just use libraries, objects, and methods like there's no tomorrow. When you accept its limitations, DOS teaches you to be much more conscious about your coding habits.

Javier Gutiérrez Chamorro

One additional advantage of DOS is that it still has the 8-bit philosophy. It allows you to create small but useful programs that can be developed in a few hours or days, in contrast with modern systems in which the effort can easily surpass that by a factor of five.

Anthony J. Williams (Rugxulo)

DOS doesn't have bogus antivirus heuristics that flag half your Windows program files as "generic" or "bad." And DOS doesn't take gigabytes of memory just to boot up. DOS doesn't need hundreds of megabytes to host a C compiler.

DOS doesn't have forced reboots to update. DOS doesn't have tons of loadable libraries to constantly reinstall. DOS doesn't require a bloated user interface. DOS doesn't make you upgrade your hardware every few years.

E. C. Masloch (ecm)

In DOS, particularly when writing in Assembly language, you need to do a lot of things "by hand." That can be a good thing, because it is cool and educational to get to know the ins and outs of low-level programming concepts. However, it can also be a bad thing, and is the reason that my non-DOS programs use other languages such as C, Python, Perl, and Bash scripting, which have a lot more features built-in or easily available, or simply put require a lot less verbosity.

Of course, another difficulty (and advantage) with DOS programming is that you get total control over the part of the machine that's running your DOS program. This can be

unforgiving in that many errors may crash and lock up the entire machine.

Fortunately, as we are beyond the days of Windows 95, 98, and ME, we can simply terminate the offending virtual machine and restart our applications, to try hopefully to pin down the exact error conditions and fix them.

How did you get interested in translating FreeDOS programs?

Translation is an important part of any open source software project. Without translation, the audience will be very limited, useful only to those who understand the same language as the developer. I am grateful to all our contributors who translate the FreeDOS programs, help pages, documentation, and all parts that make up a package. This makes the software available to everyone, all around the world.

Berki Yenigün

I contribute to various free software projects. Since I liked FreeDOS, I contributed a few translations years ago. But then the project accepting them got abandoned, so I went on a hiatus. But when I heard work for FreeDOS 1.3 was

ongoing, I took that as an opportunity to contribute again because I wanted to give something back.

As decent documentation for DOS is hard to find today, I thought translating the Help system would be useful to other newbies, so I did that. It took time, but now it's done in French. I'm currently working on the Turkish version.

I must say none of that would have happened without the help of Jerome Shidel who set up a Git repository for translations. Jerome has been very supportive, reviewing and accepting feature and pull requests, answering questions, fixing bugs. The same for Wilhelm Spiegl, who maintains the Help files and who put up a colossal work updating over three hundred Help files and translating them in German. He fixed my line lengths, pointed out typos, and incorporated the French translation.

Translating program messages and documentation is a specialty area in technical communication. What do you find most challenging about translation?

Understanding precisely the context, the meaning to convey in order to minimize the loss of information and testing the translations. Not being a developer, compiling is not easy. Thankfully, many FreeDOS programs use files in an `NLS` (National Language Support) directory, so changing them is enough to test the translations and fix typos and mistakes.

What skills do you need to be a good translator?

You just need patience, curiosity, and experience. And contributing to Free software projects is a great way to gain that experience.

Unlike programming you don't have to be good at mathematics, you don't need to learn a programming language. If you speak English and another language, come join us! You'll make a difference and you'll learn or remember a lot about the inner workings of computers and their history, as I did.

These days, you just insert a USB flash drive or a LiveCD into your computer and everything gets recognized automatically—which is great, but that also hides a lot from the user. Editing the FDAUTO.BAT and FDCONFIG.SYS files, managing the memory, the disk, and all of that teaches quite a bit about what's really happening under the hood.

Colophon

Produced using LibreOffice

Main body font is Crimson Pro

Title and heading font is Liberation Sans Narrow

8 See https://creativecommons.org/licenses/by-sa/4.0/