DOSEMU and FreeDOS: past, present and future

Bart Oldeman <bart@dosemu.org>
Maintained DOSEMU (2001-2013)

FreeDOS slides based on those from FreeDOS project coordinator
Jim Hall <jhall@freedos.org>
FDISK Options

Current fixed disk drive: 1

Choose one of the following:

1. Create DOS partition or Logical DOS Drive
2. Set active partition
3. Delete partition or Logical DOS Drive
4. Display partition information

Enter choice: [1]

Press Esc to exit FDISK
DOSEMU

Originally: using the Linux vm86() syscall to run DOS and DOS programs in a lightweight virtual machine

- Sept 3, 1992: Matthias Lautner (Linux was only a little over 1 year old.)
  Version 0.1: “DOS EMULATOR for LINUX 0.97 pl2
  This is a pre alpha version. This means there are some bugs and lots of things aren't implemented such as screen attributes, some video interrupts, some other bios calls, any port io, ...”
- 1993: Robert Sanders (0.47-0.49)
- 1993-1997: James MacLean (0.49pl2-0.64.3)
  - DOS Protected Mode Interface (DPMI) support using modify_ldt() syscall added early 1994.
- 1997-2001: Hans Lermen (0.64.4-1.1.1)
- 2001-2013: Bart Oldeman
- DOSEMU2: 2013- Stas Sergeev
Please test against a recent version before reporting bugs and problems.
Get the latest code at http://stsp.github.io/dosemu2
Submit Bugs via https://github.com/stsp/dosemu2/issues
Ask for help in mail list: linux-msdos@vger.kernel.org

FreeDOS kernel - SVN (build 2042 OEM:0xfd) [compiled Feb 2 2018]
Kernel compatibility 7.10 - GNUC - FAT32 support

(C) Copyright 1995-2012 Pasquale J. Villani and The FreeDOS Project.
All Rights Reserved. This is free software and comes with ABSOLUTELY NO
WARRANTY; you can redistribute it and/or modify it under the terms of the
GNU General Public License as published by the Free Software Foundation;
either version 2, or (at your option) any later version.
C: HD1, Pri[1], CHS= 0-1-1, start= 0 MB, size= 2000 MB
D: HD2, Pri[1], CHS= 0-1-1, start= 0 MB, size= 2000 MB
dosemu XMS 3.0 & UMB support enabled
dosemu EMS driver rev 0.8 installed.
Kernel: allocated 35 Diskbuffers = 18620 Bytes in HMA

FreeCom version 0.84-pre3 - GNUC - XMS_Swap [Feb 2 2018 11:27:01]
C: \>
This is the dumb terminal mode; S-Lang (terminal), SDL, and X outputs are available.
Running Win3.1 Netscape with network (T.P. Reitzel)
DOSEMU’s other uses

Asked around on mailing list and github and received the following replies:

- Running very old specialized cross compilers, all setup with wrappers so from the user’s point of view they are just Linux programs.
- A yearly ephemeris for astrologers with a QuickBasic IDE.
- Running old DOS accounting packages.
- Running DOS CAD programs for the design of electronics and astronomical instruments.
- Backing up Psion-3 personal organiser, using 1991 Psion link software via a standard USB-Serial (RS232 9600 baud) hardware module.
- Running MIDI sequencers.
- POS Terminals with some commercial MUMPS interpreter.
- Stable and fast environment for running PC/GEOS.
- Talks to ISA hardware controlling an antenna receiving satellite images.
DOSEMU vs other options

Where does DOSEMU fit in among other solutions such as DOSBox and QEMU?

Heterogeneous “competition”:

- **DOSBox**: Aimed at games and games only, much more portable, using CPU emulation. Doesn’t have a command line “dumb” mode, terminal mode, support for printers, etc. Note: DOSBox-X is a maintained (2013+) fork.

- **Bochs, QEMU, VirtualBox, etc.**: “heavyweight” emulators and virtual machines, work well but harder to access host file system; DOSEMU can easily access the Linux host file system through the “undocumented” DOS network redirector interface.
DOSEMU vs hardware and Linux kernel

- Originally: a somewhat dangerous program
  - Ran as root or suid-root
  - Direct VGA hardware access
  - Direct (FAT) partition disk access
    (it was common to dual boot with DOS/Windows on FAT)

- Linux kernel forced some changes:
  - Disabled mmap of /proc/self/exe (used for EMS)
    - map a file or use shm_open() and co.
  - Disabled mmap of page 0 (needed for vm86())
    - Use CPU emulation or KVM instead of vm86()
    - modify_ldt() can still be used (offsetting the base addresses for DPMI)
  - KMS: direct VGA gone for most people
    - VGA emulation was quite mature by then

- Hardware forced some changes:
  - x86_64 does not do vm86() in long mode:
    - Use CPU emulation or KVM instead of vm86()
DOSEMU2 changes

- Fully rewritten video stack, much improved sound stack, mostly rewritten signal handling and DPMI context switching, better EMS support.
- Many other bug fixes.
- Supports KVM (see next slide)
KVM support in Dosemu2

- Use of KVM within Dosemu2 to allow running DOS applications at near-native hardware speed on CPUs that no longer support the vm86() syscall.
- But... those old machines were slow too, right?
- Still nice for doing compile jobs inside DOSEMU, e.g. COMMAND.COM: 9.5 secs with KVM, 57 secs with JIT CPU emulation, 4 mins with simulated CPU emulation.
- My main contribution to Dosemu2, based on this article: https://lwn.net/Articles/658511/
  - The DOS applications still run in V86 mode inside KVM; a tiny protected mode monitor in KVM traps into DOSEMU on demand.
  - DPMI is possible too but no speed gain versus modify_ldt(), though KVM provides better isolation.
  - Memory mapping was tricky, not properly finished.
Public Domain DOS

Newsgroups: comp.os.msdos.apps
Subject: PD-DOS project *announcement*
Date: 29 Jun 94 00:24:11 -0600

ANNOUNCEMENT OF PD-DOS PROJECT:

A few months ago, I posted articles relating to starting a public domain version of DOS. The general support for this at the time was strong, and many people agreed with the statement, “start writing!” So, I have...

Announcing the first effort to produce a PD-DOS. I have written up a “manifest” describing the goals of such a project and an outline of the work, as well as a “task list” that shows exactly what needs to be written. I’ll post those here, and let discussion follow.
Welcome to the FreeDOS Homepage

Cool! Get the FreeDOS Beta 1 "Orlando" Release!

What's New with FreeDOS?

2 May: Hey! Minor revision to Orlando release News Item 006 Download!
25 Apr: New FreeDOS available! News Item 003
25 Apr: Call for volunteers - FreeDOS compiler project News Item 005
25 Apr: FreeDOS compatibility list available News Item 004 Go there!
16 Apr: FreeDOS Beta 1 "Orlando" release due soon. News Item 002
16 Apr: File f dismalpha5.zip now includes Installer v1.2. Download!
15 Apr: Can someone help us fix command.com? See Tech Note 002
15 Apr: Can FreeDOS use high capacity HDD's? Not yet. See Tech Note 001

What is FreeDOS?
1994 to 2006

Free-DOS Alpha 1 (16 September 1994)
Free-DOS Alpha 2 (December 1994)
Free-DOS Alpha 3 (January 1995)
Free-DOS Alpha 4 (June 1995)
FreeDOS Alpha 5 (10 August 1996)
FreeDOS Alpha 6 (November 1997)

FreeDOS Beta 1 “Orlando” (25 March 1998)
FreeDOS Beta 2 “Marvin” (28 October 1998)
FreeDOS Beta 3 “Ventura” (21 April 1999)
FreeDOS Beta 4 “Lemur” (9 April 2000)
FreeDOS Beta 5 “Lara” (10 August 2000)
FreeDOS Beta 6 “Midnite” (18 March 2001)
FreeDOS Beta 7 “Spears” (7 September 2001)
FreeDOS Beta 8 “Methusalem” (7 April 2002)

FreeDOS Beta 9 RC1 (July 2003)
FreeDOS Beta 9 RC2 (23 August 2003)
FreeDOS Beta 9 RC3 (27 September 2003)
FreeDOS Beta 9 RC4 (5 February 2004)
FreeDOS Beta 9 RC5 (20 March 2004)

FreeDOS Beta 9 (28 September 2004)
FreeDOS Beta 9 SR1 (30 November 2004)
FreeDOS Beta 9 SR2 (30 November 2005)

FreeDOS 1.0 (3 September 2006)
FreeDOS aims to be a complete, free, 100% MS-DOS compatible operating system.

**FreeDOS 1.0** Yes, the FreeDOS Project has reached the "1.0" milestone. [Download FreeDOS](#) or buy it on CDROM. This is a very important day for FreeDOS. A lot of you have put in so much work over the years, helping to make everything perfect. Even if you didn’t contribute code, you helped out the FreeDOS Project by submitting comments and bug reports. Thank you!

**What is FreeDOS**

FreeDOS is a free DOS-compatible operating system for IBM-PC compatible systems. FreeDOS is made of up many different, separate programs that act as "packages" to the overall FreeDOS Project.

These days, there are three main uses of FreeDOS:

1. To run classic DOS games (like Doom, MAME, etc.)
2. To run business software that only supports DOS
3. To support an embedded DOS system, such as a computerized cash register or till
Welcome to FreeDOS

FreeDOS is a complete, free, DOS-compatible operating system that you can use to play classic DOS games, run legacy business software, or develop embedded systems. Any program that works on MS-DOS should also run on FreeDOS.

It doesn’t cost anything to download and use FreeDOS. You can also share FreeDOS for others to enjoy! And you can view and edit our source code, because all FreeDOS programs are distributed under the GNU General Public License or a similar open source software license.
FreeDOS 2.0 (Jim Hall's proposal)

What is “DOS”? 

- 16-bit
- single-tasking
- single-user
- command-line
- run on old hardware

Compatibility is key!

What is a “modern DOS”? 

tools
utilities

What isn’t needed anymore? (“Compat”)

APPEND
ASSIGN
GRAPHICS
JOIN
SUBST
FreeDOS compiler issue

- Issue: GCC could not produce 16-bit x86 code
- Until 2002, proprietary compilers were used to compile FreeDOS
- As DOSEMU can’t function without a DOS, DOSEMU was pushed out of distributions’ “main” repositories.
- 2002: Open Watcom comes around, OSI approved but “Oops... it looks like OSI smoked something especially bad this time, I'm afraid. This license looks like someone took his time to collect every single problematic clause.” (Adam Borowski, debian-devel)
- 2007: Rask Ingemann Lambertson contributes ia16 codegen to gcc.
- 2017: Andrew Jenner refines into GCC 6.2 based patchset (tiny model only)
- 2017-2018: FreeDOS kernel (mostly written in C) and command.com (FreeCOM) ported to ia16-elf-gcc.
Differently: Stas Sergeev: run FD kernel C-code in PM
More information

http://www.dosemu.org/
http://stsp.github.io/dosemu2/
http://www.freedos.org/
http://freedos-project.blogspot.com/
http://twitter.com/FreeDOS_Project