

DOS/65 Update April 2025
Version 3.21 released

DOS/65 was written by Richard A. Leary and was licensed, maintained and updated for many years by Richard. DOS/65 provides a similar code structure to CP/M and is compatible at a disk filesystem level. The intent is to provide a standard “CP/M-like” operating environment on the 6502 that has a common filesystem and system calls. While this does not provide a binary level of compatibility with CP/M, it does provide a common programming structure.

To the best of my knowledge, the last version that Richard worked on is Version 3.0, of which there are two implementations: A RAM based, bootable version (using SYSGEN to create) and a ROM based version, which uses WDC Tools to assemble and link. The final version provides additional features and functions beyond Version 2.1, which are highlighted below:

- A CP/M compatible User from 0-15 is now supported.
- The SUBMIT batch capability is added to CCM
- The IOSTAT variable provides the User number and Drive number

There are some functional limitations of DOS/65, which are highlighted below:

- A maximum of 8 drives (drive letters A thru H) are supported.
- Each allocated drive is limited to 8MB total allocation size.
- A single file has a maximum size of 512KB.

Around 2021/2022, Richard changed the license agreement for the 3.0 ROM Version documentation as shown below:

“The DOS/65 V3 ROM software and all other DOS/65 software are free software; 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 3 of the license, or any later version.”

My view on this is that Richard stopped supporting/updating DOS/65 and changed the licensing to GNU GPL Version 3 or later. By referencing “all other DOS/65 software”, that should include earlier releases and utility programs, which were released with Version 2. This includes the Assembler, Basic Compiler and Runtime, Editor, etc. If anyone knows this is not the case, please contact me so I can update this. My last correspondence with Richard was back in 2021. Hans Otten corresponded with Richard in 2022 and has provided some details and links on his website for DOS/65 here:

<http://retro.hansotten.nl/6502-sbc/dos-65/>

There has also been a fair amount of discussion on 6502.org and some other sites which Richard participated in directly for older systems such as the OSI, which can be found here:

<http://osiweb.org/osiforum/viewtopic.php?f=4&t=235&start=40>

In 2023, I did a fairly large rewrite of DOS/65 Version 3 (RAM version). The goal was to provide a more modern version for newer 65C02 systems with faster I/O and larger block devices for storage. By using 65C02 opcodes and addressing modes, this resulted in a smaller and faster code base. This new code base also removed various routines that were specific to diskette-based media. Other changes were to eliminate the Track and Sector based disk access and change to a Record based access mechanism. I released this code as Version 3.20, which was a bootable RAM based version and some of the core features are:

- Separation of CCM, PEM and SIM into standalone source modules
- Updates to CCM and PEM to take advantage of CMOS opcodes and addressing modes
- Fixed problem areas related to Drive Letters and out of range inputs for CCM/PEM
- SIM is a completely new module and uses PEM Records to Logical Block Addressing
- Elimination of Checksum routines in PEM only used for diskette-based media
- Update to the Disk Control Block (DCB) for Record Based media
- Source code ported to WDC Tools for assembly and linking
- Version 2.1 utilities are still fully compatible (assembler, editor, Basic, etc.)
- Updates to some core utilities such as SD to accommodate the updated DCB.

Since the release of Version 3.20, I've done a large amount of testing and have concluded that the code base is extremely stable and can likely run indefinitely on a good hardware design. I've also continued working on some updates. By going thru CCM and PEM modules, which make up the core of DOS/65, I've completed another release with additional changes. These changes have resulted in more efficient memory use for 512-byte block storage devices. There were a number of routines in PEM that were called by CCM only. These were likely put into PEM as the CCM code size needed to be contained into a smaller size.

With the new version (3.21), these routines have been moved to CCM and the end result is that CCM and PEM are now fully isolated from each another. CCM only accesses PEM via standard calls, i.e., no JSRs to routines inside of the PEM allocated memory range. I also changed the Record based access in PEM so that Reserved Records are passed to SIM and handled there instead of within PEM. CCM has grown a bit and now occupies 2.5KB of space, which is 5 blocks from the disk drive. PEM is also contained in 2.5KB and (also) occupies 5 blocks on the disk drive. This new isolation of CCM and PEM allows CCM to be reloaded with a warm boot without having to reload PEM.

A change has been made to the erase command within CCM. By default, checking for the existence of the file to be erased is not done, so if you type in the wrong filename, the command just returns without any indication that nothing was done. The change now shows a "NOT FOUND" message, which also works for wildcard entries, except for a "*. *" filespec, where the standard "ALL FILES (Y/N)" message is displayed.

I also added a "SYSTEM" command, which displays memory usage for DOS/65, which includes Page Zero, System BIOS, TEA, CCM, PEM, SIM and Disk Allocation Map areas.

The new SIM version, which was a clean rewrite from scratch for Version 3.20, has been updated with some additional functions and some minor changes for Version 3.21. It is still contained within 1KB of space and provides additional routines to aid in a more complete system and requires a simple BIOS to operate. Another minor change is to use 4KB allocated block size for the drive definitions, which uses half the memory as before. This results in 8KB of RAM used for CCM, PEM, SIM and Disk Allocation Maps, which provides eight 8MB drives. CCM and PEM are 2.5KB each, SIM is 1.0KB and the full set of Disk Allocation Maps is 2.0KB. Reducing the number of supported drives and/or reducing the size of the drives can reduce the space required for drive allocation maps if needed.

With rare exception, any existing application for DOS/65 should run without changes. To date, I've assembled and tested almost every utility that was provided with DOS/65 Version 2.1 and 3.0. There are certain applications/utilities that are specific to the hardware platform, such as DEBUG, that need to be customized for the hardware being used. My C02 Pocket system already has a Monitor, so I've not altered the Debug program. Currently, I've no plans to do this, as I don't need it for my system. For anyone else wanting to use DOS/65 version (3.21), they might want to consider updating the DEBUG program to use with their system.

The hardware system I have used to develop and test the new updates on is a 3.3-volt prototype based on my C02 Pocket SBC with CF-Card/RTC adapter, which is described below:

- W65C02S Processor running at 8MHz
- AT28BV256 EEPROM addressed from \$F000 - \$FFFF (4KB)
- Alliance 128KB SRAM addressed from \$0000 - \$EFFF (60KB)
- I/O Window configured at \$FE00: 2- I/O selects, each at 32-bytes wide
- NXP SC28L92 DUART – both ports and timer configured
- Maxim DS1318 Realtime Clock (not used with DOS/65 currently)
- Hitachi 5GB or 6GB Microdrive PATA ZIF with 8/16-bit IDE interface
- BIOS supports DUART and Microdrive via interrupt-driven code
- BIOS Jump table at \$FF00 is called by SIM for all hardware access
- DS1233A is used as a Panic NMI trigger

DOS/65 Version 3.21 requires 5KB of memory for CCM, PEM and their required Data areas. The SIM module is mapped to 1KB of memory after PEM. All modules are mapped to page boundaries. There are 10 pages for CCM, 10 pages for PEM and 4 pages for SIM. This totals to 6KB of RAM and the default storage configuration provides eight drives, each at 8MB in size. There is an additional 2KB of RAM used for the disk allocation maps. This corresponds to 2048 total blocks which are 4096 bytes in size and 1024 directory entries per drive. The additional 2KB for allocation maps brings the DOS/65 size to 8KB. The C02 Pocket SBC prototype uses the first 2KB for hardware and buffer support and the upper 4KB for a limited Monitor, BIOS and I/O region. Loading from the reserved records area of the first logical drive, DOS/65 is loaded starting at \$D000 and the TEA starts at \$0800. This provides 50KB of free memory for user programs, with an additional 2.5KB by overwriting CCM, if required, for a maximum of 52.5KB.

It is possible to reduce the ROM size by shrinking or eliminating the Monitor code, which could increase RAM to 62KB. Note however that the limited Monitor has some useful functions, which can help with debugging and allows full examination and modification of memory. It also includes a Xmodem-CRC loader with S19 record support. Currently, the C02 BIOS reserves the upper 64 bytes of Page Zero and includes the 8-bytes for DOS/65. The Monitor reserves 32 bytes of Page Zero directly below the BIOS, but no Monitor functions are required by DOS/65. This provides a fairly large contiguous Page Zero space of 192 bytes for User applications and utilities.

To date, extensive testing has been done with Version 3.21 release, which includes extensive disk activity with utility applications including:

- Submit
- Assembler
- MakeCom
- Alloc
- SD (Super Directory)
- Basic Compiler and Runtime
- COPY and UCOPY
- COMPARE
- DUMP, MORE, PAGE
- XMODEM

The system has been completely stable. CCM and PEM now check for drive input range and show an "UNKNOWN DRIVE" error if out of the range for the system. All "phantom" accesses are eliminated as well. i.e., "t:" or a non-alpha character, which could cause disk problems if inadvertently used from CCM or a call to PEM.

The new SIM module provides Blocking/De-Blocking for 128-byte Records to the 512-byte LBA size of the storage device. This also includes delayed LBA writes using a "dirty block" flag to improve disk write performance. SIM also provides some additional routines to eliminate the need for functions contained in the C02 Monitor code.

Note, there is no SYSGEN for system creation. WDC Tools are used for assembling and linking CCM, PEM and SIM, the output is a Motorola S-Record (S19) file. The limited Monitor supports download via XMODEM-CRC. On download, it will automatically sense and process S19 records. I've written a disk utility (MDUTIL) that provides numerous functions for testing and examining an IDE device. One function is to write a section of memory to a contiguous set of blocks on the IDE device. The C02 Pocket SBC uses a disk partitioning scheme to provide multiple bootable partitions. A partition offset is added in SIM to define the starting LBA for all of its drives. The Utilities to setup and configure the storage device are still a work in progress. For now, the Monitor loads the image from the drive and jumps to SIM to boot DOS/65. SIM's warm-boot call reloads CCM from the reserved records section of the boot drive.

I'm hoping others can take advantage of this new version of DOS/65 and get it running on a larger range of 65C02 systems. The only functions required to implement this are a compatible BIOS to support the console and disk-based calls, which are outlined in the SIM source code file. The C02 BIOS and limited Monitor code is also provided in source form.

Regards, KM