## DOS History:

| Date | Comments |
|---|---|
| 1973 | Gary Kildall writes a simple operating system in his PL/M language. He calls it CP/M (Control Program/Monitor or Control Program for Microcomputer). |
| 1979 | Apple Computer releases DOS 3.2.<br><br>Apple Computer releases DOS 3.2.1 |
| 1980 | Tim Patterson begins writing an operating system for use with Seattle Computer Products' 8086-based computer. Seattle Computer Products decides to make their own disk operating system (DOS), due to delays by Digital Research in releasing a CP/M-86 operating system.<br><br>QDOS 0.10 (Quick and Dirty Operating System) is shipped by Seattle Computer Products. Even though it had been created in only two man-months, the DOS worked surprisingly well. A week later, the **EDLIN** (**DOS's version of vi!**) line editor was created. EDLIN was supposed to last only six months, before being replaced.<br><br>Tim Patterson shows Microsoft his 86-DOS, written for the 8086 chip.<br><br>Microsoft's Paul Allen contacts Seattle Computer Products' Tim Patterson, asking for the rights to sell SCP's DOS to an unnamed client (IBM). Microsoft pays less than US$100,000 for the right.<br><br>Seattle Computer Products renames QDOS to 86-DOS, releasing it as version 0.3. Microsoft then bought non-exclusive rights to market 86-DOS. |
| 1981 | MS-DOS runs for the first time on IBM's prototype microcomputer. |

## MS-DOS History:

| Version | Date | Comments |
|---|---|---|
| 1.0 | 1981 | The original version of MS-DOS. This was a renamed version of QDOS which had been purchased by an upstart company called Microsoft. |
| 1.25 | 1982 | This added support for double-sided disks. Previously the disk had to be turned over to use the other side. |
| 2.0 | 1983 | This added support for IBM's 10 MB hard disk, directories and double-density 5.25" floppy disks with capacities of 360 KB.<br><br>IBM introduces PC-DOS 2.1 with the IBM PCjr. |

| | | |
|---|---|---|
| 2.11 | 1983 | Support for foreign and extended characters was added.<br><br>Microsoft releases MS-DOS 2.1 for the IBM PCjr. |
| 3.0 | 1984 | Support for high-density (1.2 MB) floppy disks and 32 MB hard disks was added. |
| 3.1 | 1984 | Network support was added. |
| 3.3 | 1987 | This release was written to take advantage of IBM's PS/2 computer range. It added support for high density 3.5" floppy disks, more than one partition on hard disks (allowing use of disks bigger than 32 MB) and code pages. |
| 4.0 | 1988 | This version provided XMS support, support for partitions on hard disks up to 2 GB and a graphical shell. It also contained a large number of bugs and many programs refused to run on it.<br><br>Digital Research transforms CP/M into DR DOS. |
| 4.01 | 1989 | The bugs in version 4.0 were fixed.<br><br>Microsoft introduces Russian MS-DOS 4.01 for the Soviet market.<br><br>Digital Research releases DR DOS 5.0. |
| 5.0 | 1991 | This was a major upgrade. It allowed parts of DOS to load itself in the high memory area and certain device drivers and TSRs to run in the unused parts of the upper memory area between 640K and 1024K. This version also added support for IBM's new 2.88 MB floppy disks. An improved BASIC interpreter and text editor were included, as was a disk cache, an undelete utility and a hard-disk partition-table backup program . After the problems with MS-DOS 4, it also provided a utility to make programs think they were running on a different version of MS-DOS.<br><br>Digital Research Inc. releases DR DOS 6.0 |
| 5.0a | 1992/3 | This was a minor bug fix which dealt with possibly catastrophic problems with UNDELETE and CHKDSK. |
| 6.0 | 1993 | This was a catch-up with Novell's DR-DOS 6. It added a disk-compression utility called DoubleSpace, a basic anti-virus program and a disk defragmenter. It also finally included a MOVE command, an improved backup program, MSBACKUP and multiple boot configurations. Memory management was also improved by the addition of MEMMAKER. A number of older utilities, such as JOIN and RECOVER were removed. The DOS Shell was released separately as Microsoft felt that |

| | | |
|---|---|---|
| | | there were too many disks. |
| 6.2 | 1993 | Extra security was built into DoubleSpace following complaints of data loss. A new disk checker, SCANDISK, was also introduced, as well as improvements to DISKCOPY and SmartDrive. |
| 6.21 | 1993 | Following legal action by Stac Electronics, Microsoft released this version which had DoubleSpace removed. It came with a voucher for an alternative disk compression program. |
| 6.22 | 1994 | Microsoft licenced a disk-compression package called DoubleDisk from VertiSoft Systems and renamed it DriveSpace, which was included in this version. |
| 7.0 | 1995 | This version is part of the original version of Windows 95. It provides support for long filenames when Windows is running, but removes a large number of utilities, some of which are on the Windows 95 CD in the \other\oldmsdos directory. |
| 7.1 | 1997 | This version is part of OEM Service Release 2 and later of Windows 95. The main change is support for FAT 32 hard disks, a more efficient and robust way of storing data on large drives. |

**DOS Internals:**

- A single user OS, no multitasking supported.
- Kernel and command interpreter in memory.
- A program loads and part of the command interpreter is overwritten to free up memory (memory was expensive).
- The program terminates and the resident part of the command interpreter reloads the remaining (overwritten) part of the command interpreter.

**DOS System Files:**

- IO.SYS
- Hidden, system, read-only file.
- Loaded by the bootstrap program.
- Allows RAM to manage I/O.
- Loads MSDOS.SYS.

**MSDOS.SYS**

- Hidden, system, read-only file.
- IO.SYS and MSDOS.SYS create (combine to form) kernel.
- Manages file creation.
- Manages directories.
- Looks for CONFIG.SYS.

**CONFIG.SYS (optional)**

- System file.
- User created.
- Contains system configuration info.
- Loads device drivers.
- Loads HIMEM.SYS (if available).
- Loads EMM386.EXE (if available).
- Looks for COMMAND.COM.

**COMMAND.COM**

- System file.
- Command interpreter (shell).
- Performs DOS commands.
- Generates DOS error messages.
- Looks for AUTOEXEC.BAT.

**DOS Interrupts:**

- DOS programs not only had direct access to the hardware (**a no no**), in some cases, they had to directly access the hardware to execute (**a real no no**).
- Examples of interrupts that any program could envoke:
- Interrupt 32 – Program terminate (also implemented as a DOS function call).
    - This interrupt does not automatically close open files and any files left open may be incorrect.
- Interrupt 34 – Terminate address.
    - Indicates what address control should be passed to on program termination (typically back to DOS or a "shelled" from program).
    - Not really an interrupt. The interrupt table simply stores the address that control should be passed to.
- Interrupt 35 – Break address.
    - Points to interrupt routine to handle control-breaks.
    - Control returns to DOS, which can resume operation of the program or exit to DOS.
- Interrupt 36 – Critical-Error address.
    - Points to interrupt routine to handle critical-errors detected by DOS (typically disk errors).
    - Returns control to DOS, which can ignore the error (resume the program), try the operation again (possible resume the program), or terminate the program.
- Interrupt 39 – Terminate-but-stay-resident (also implemented as a DOS function call).
    - Similar to interrupt 32, except it does not erase the program from memory. A portion of the program remains in memory (for example, we can discard the initialization code) and becomes an extension of DOS (DOS's record of the first usable part of memory is adjusted to beyond the TSR).
    - Frequently used to establish a new interrupt handling routine for some device or to replace an existing interrupt handling routine.

- Can also be used to load data into a "common" area that all proceeding programs can access.
- Interrupt 39 only works with low memory, hence .EXE files cannot use interrupt 39.
- It was not uncommon for TSRs to require reloading of the TSR program upon being called (since only a portion of the TSR may be in memory). If multiple TSRs were in memory and if a TSR in lower memory required extra memory to reload itself, the system frequently crashed. I.e. the order in which multiple TSRs were loaded was not arbitrary.

**DOS Function calls:**

- In order to make DOS behave/look more like unix - where all hardware accesses are controlled by the OS, IBM added a set of functions that could be called to acquire low-level access to the machine.
- Examples of functions that any program could call:
  - Function 72 – Allocate a block of memory.
  - Function 75 – Load/Execute program
    - Allows an overlay to be loaded and jumped to (no PSP created).
    - Allows data to be loaded.
    - When a program is loaded and executed any active file handles are available to the subprogram.
    - The load is performed by the command interpreter. Therefore, it may require reloading prior to the subprogram load.
    - Function 75 clobbers most of the registers. Therefore, these should be preserved before and restored after function 75. The stack cannot be used for this. Therefore, one way to preserve the data is to paste it into the program code as well as the appropriate address to jump to (to get the register data) at the point of re-execution.
  - Function 73 – Free allocated memory.

**DOS CPU Scheduling:**

- There isn't one!
- Once a process is started in retains the CPU until it is finished or an interrupt has occured. If an interrupt does occur the process will be forced to wait until the interrupt finished before resuming.

**Imagine how fast a modern PC would be if it only ran DOS!**

**DOS Memory:**

- Conventional 0 to 640K :
  - Provides the base memory (to run programs, etc).
- Reserved 640K to 1M :
  - Reserved by IBM for expansion (video, NICs, etc).
- Expanded EMS - LIM 4.0 :
  - Uses 4 16K pages to create a 64K frame page which is swapped into/out of the reserved memory region. Could only be used for data storage.

- Extended (generic) 1M to 4G :
  - Could not be used by DOS in most circumstances (except RAM drives, ...).
- XMS (extended memory specification) 640K to 4G :
  - Provided 3 types of virtual protected memory:
    - UMA - 640K to 1M : Used to load device drivers, etc freeing up more of conventional memory.
    - HMA - 1024K to 1088K : Originally a (popular?) "bug" in the 286 design, now used to load part of the DOS kernel (**DOS=high**). Can only be used by 1 program at a time.
    - EMA - 1088K to 4G : Used by protected mode programs.

**DOS File System:**

- FAT

**DOS I/O System:**

- Like much of DOS the I/O system interacts directly with the hardware using:
  - Memory - By writing or reading from memory locations, a (memory-mapped) device can be controlled.
  - Ports - Ports are numbered (like memory locations) and can be used to control a device. Note that ports are not memory.
  - Interrupts - DOS can also use software interrupts to interact with devices.

**Conclusion:**

- DOS/PCs were never meant to be what they are today. In fact:
  - IBM never thought the personal computer would ever "catch on".
  - IBM never thought anyone would need more than 64K of RAM, nor hard drives, nor high density floppies, nor color graphics, ... on a PC.

**DOS lives! It has been revamped for hand-held computers and embedded computers. And don't forget, it's still buried in Windows.**