

Part II
DOS Programming

“writing BASIC for the Altair was exhausting...Paul and I didn't sleep much and lost track of night and day. When I did fall asleep, it was usually at my desk or on the floor. Some days I didn't eat...But after five weeks...world's first microcomputer software company was born.”

—**Bill Gates**

Courtesy: The Road Ahead (ISBN 0-14-024351-8)

14

"If you love to sleep, you will be poor."

DOS Secrets

To program well, you have to know more about your hardware and DOS internals. This book is neither a hardware book nor a beginners' book. So I would slightly touch the hardware and DOS internals in this chapter. In many Institutions hardware & software are being taught as different subjects. And people don't know how both are related. For system programming you must know the relationship between the two. This chapter will help you to understand why a programmer should know hardware & DOS internals for DOS programming.

14.1 Prelude

DOS (Disk Operating System) is the widely used operating system. It is a single-user operating system. DOS is designed to provide an easy way to use disks for storage. It is very efficient in controlling, accessing and managing the data from disk drives. The basic operations performed by DOS are regulate space allocation, keep track of files, save and retrieve files and manage other control functions associated with disk storage. Thus using DOS an interface is made between the user and the computer. This DOS is same for all the systems. For loading this DOS to the memory BIOS, bootstrap program, diagnostic testing programs are very essential and we will discuss it in the coming sections.

14.1.1 BIOS

It is a program that provides link between the hardware and the operating system. It is a firmware (Firmware is a program or data stored in ROM. These are not altered by software, and are not lost when the power is turned off). Since it is stored in ROM, it is usually called as ROM BIOS. It contains many low level routines. It is responsible for basic hardware operations such as interactions with disk drives and keyboards. It also has drivers and other software that manages the peripheral devices.

The basic operations performed by BIOS are

- Keyboard routine
- Video routines
- Printer routines

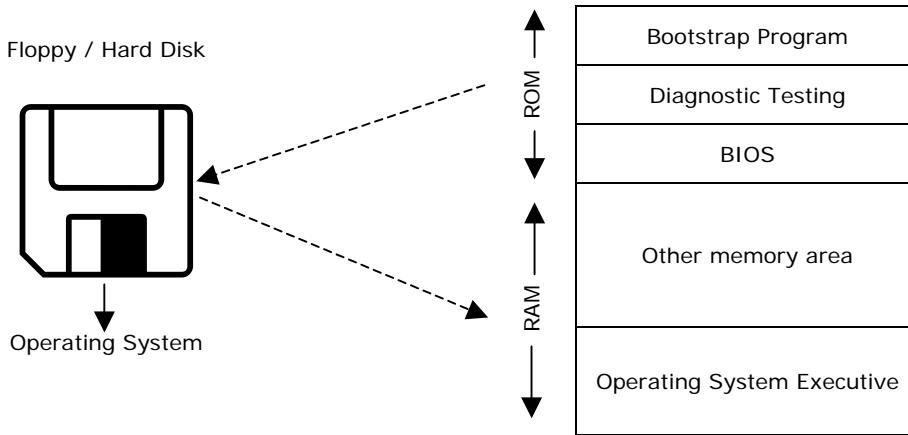
This BIOS program differs from system to system. For getting good results we can use BIOS functions along with the DOS functions.

14.1.2 Bootstrap Program

Bootstrap program is responsible for loading the operating system from the disk to the memory. When the computer is switched ON the process of bootstrapping takes place, which initializes the computer for use, by automatically clearing memory and loading the first few instructions that call other instructions in the disk (Nowadays the remaining part of the operating system resides in the hard disk itself).

The basic operations performed by bootstrap program are

- It runs the diagnostics testing programs to check the status of RAM.
- It makes a call to the disk for loading the operating system into the memory.
- After loading the operating system, it transfers control to the operating system.



14.1.3 Boot Sector

The boot sector on a disk is always the first sector on the first track on the first head. BIOS starts up and does the POST, when computer is powered ON. It initializes all its data and then looks for a valid boot sector. First it looks at the Floppy disk (A:), then at the Hard disk (C:). After this process, the operating system is loaded into the memory, which is explained in the figure. If it doesn't find it then interrupt 18h is called (on original IBM PCs this started the ROM BASIC). A valid boot sector (to the BIOS) is one that has 0AA55h at offset 510 in the boot sector.

When the BIOS finds the boot sector, it reads that sector (512 bytes) off of the disk and into memory at 0:7C00h. Then it jumps to 0:7C00h and the boot sector code gets control. BIOS data area (40h:0) and the BIOS interrupts (10h - 1Ah) are initialized. At this point, memory is mostly unused, but not necessarily cleared to 0.

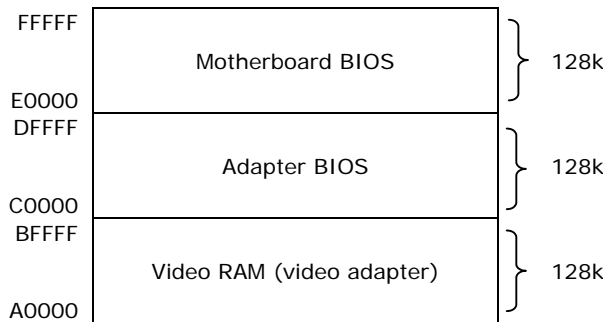
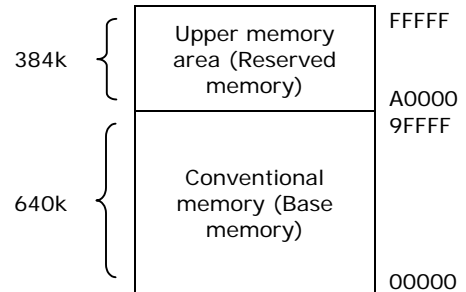
14.2 Memory Layout

For better programming in DOS we must also know the memory layout of DOS. In the system there is 1MB of addressable memory area, in that 1024K(1MB) of addressable memory first 640K is called *conventional memory area*, it addresses from 00000 to FFFFF and the remaining 384K is called *reserved memory* or *upper memory area*, it addresses from A0000 to FFFFF.

The conventional memory (which is also called *base memory*) is reserved for the use by the system and the upper memory area is reserved for the use by the graphics boards, other adapters and motherboard ROM BIOS.

14.2.1 Upper Memory Area (UMA)

The 384K of upper memory is further divided into three equal parts of 128K each. The first 128K above the conventional memory area is reserved for the use by the video adapter and it is also called video RAM. The next 128K is reserved for use by the adapter BIOS and the last 128K is for Motherboard BIOS.



In the video RAM area the information related to text and graphics display on screen is stored. The address range of this video adapter is A0000-BFFFF. If we use monochrome graphics adapter (MGA) then the information about the display is stored between B0000 and B8000. If we use CGA then it occupies the address range B8000-

BFFFF. Graphics mode video RAM occupies A0000-AFFFF.

In the 128K area of adapter BIOS, the first 32K is used by VGA compatible video adapters and the remaining area is used by network adapters and some other adapters.

In the 128K of the motherboard BIOS, the first 64K is called free UMA block space and most of the systems use only the last 64K. In this area POST (Power On Self Test—which is a set of routines that test motherboard, memory, disk drives, adapter, keyboards, other devices and components in the system), bootstrap loader (which is set of routines to start the operating system) and CMOS (Component Metal Oxide Semiconductor—which is used to configure the system by pressing some key while booting) reside.

14.3 Segment Address

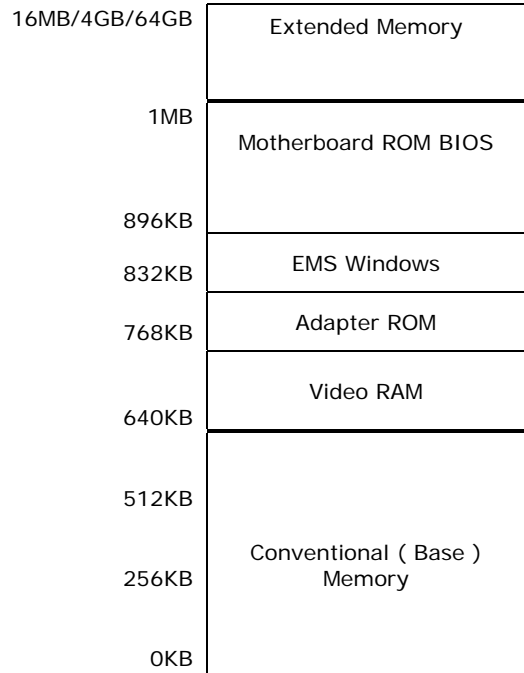
In the system every instruction is addressed by 20-bit linear address from 00000-FFFFF. This is called real address or physical address of the system. The total memory area in the system is divided into different segments. These segments use only 16-bit address for storing and retrieving data in each segment. The real addressing has 20-bits and so to represent this 20-bit physical address we are using 16-bit segment address and offset address.

For example, if the segment address is B000 and the offset address is 8888 then the corresponding physical or linear address will be

B000	
8888	Thus we have've got
B8888	a 20-bit real address

In this method overlapping is possible. For example, we can get the same physical address in various segments and offset combinations.

B080	B880	B008	B808
8008	0088	8808	0808
B8888	B8888	B8888	B8888



14.4 Extended Memory

Any memory above 1MB is called extended memory. The size of the extended memory changes from system to system. For example, the size of extended memory for 286, 386DX and Pentium II are 16M, 4G, and 64G respectively.

The diagram with the conventional, upper and extended memory is given above.

14.5 Limitations of DOS

It is a single user operating system and it does not support multitasking and it is not designed for networking. It does not support GUI (graphical User Interface), which is popular in Windows. Virtual memory area is not present in DOS. Now DOS is given a graphical user interface and limited multitasking capability by combining with Windows. This DOS/Windows combination was first introduced in 1995 with Windows 95.