Optimizing Memory for Legacy DOS Systems

|  |
| --- |
| **Denise de Vries** |
| *Swinburne University of Technology*  *Australia*  *dbdevries@swin.edu.au*  *0000-0001-9061-6471* |
|  |



Figure : Types of Memory

**Abstract – Running software from the early 19980s and 1990s, often problems are encountered due to the limitation of conventional (base) memory. Even though the system may have a sizable amount of memory, only conventional memory was used to load and run programs. Customizing the system to load other necessary executable files such as drivers into memory was required. In this paper, an overview of the memory architecture of IBM-compatible personal computers is given and approaches to memory management configuration are presented.**

**Keywords – DOS, memory management, IBM-compatible**

**Conference Topics – Resilience**

# Introduction

The Play it Again 2 project “Preserving Australian video game history of the 1990s” (LP180100104) is focusing on 51 games from that period across eleven different computer platforms. The DOS and MS Windows games number 28 games from 1992 to 2000, ie MS DOS versions 6 and later and MS Windows versions 95 to 2000. When executing these video games difficulties arise with respect to memory limits due to the 640K conventional memory barrier.

In 1988, Microsoft launched a specification to manage previously unmanaged regions of memory in an IBM-compatible personal computer [1]. While this memory management solved the problem of applications overwriting memory addresses used by other applications, it introduced the need for the computer owner to configure a computer’s use of available memory. There were various disk operating systems (DOS) released, such as PC DOS (IBM), MS DOS (Microsoft) and DR DOS (Digital Research), all of which offered memory management. All variants are referred to as DOS in this paper. This memory management architecture was also implemented in MS Windows versions to Windows 98.

Therefore, configuring original hardware and emulators to run software from this era, it is necessary to have an understanding of the requirements of the day. In this article, I present a background to the system architecture of the IBM PC-compatible of the early 1990s and examples of configurations to optimize system memory.

# Types of Memory

shows types of memory in the IBM PC-compatible systems; each of these is described as follows.

## Conventional (Base) Memory

The original PC/XT-type computer was designed to use 1M of memory workspace (RAM). This 1M of RAM was divided into several sections. DOS could read and write to the entire megabyte, but could manage the loading of programs only in the portion of RAM space called **conventional memory**, which at the time the first PC was introduced was 512K. The other 512K was reserved for use by the system itself. IBM decided that only 384K was needed for these reserved uses, and then began marketing PCs with 640K of user memory.

Thus, 640K became the standard for memory that could be used by DOS for running programs and resulted in what was known as the 640K memory barrier. The remaining memory after 640K was reserved for use by the graphics boards, other adapters, and the motherboard ROM BIOS. [2]

## Upper Memory Area

The term Upper Memory Area (UMA) describes the reserved 384K at the top of the first megabyte of system memory. While this area was termed *reserved memory*, it was possible to use unused regions of this memory to load device drivers and memory-resident programs to free up the conventional memory they would otherwise require.[2, 3]

## High Memory Area

The High Memory Area (HMA) is the first 64K of extended memory. The HMA’s size was fixed, no matter how much extended memory was available.[3] HMA was used to load device drivers and memory-resident programs, to free up conventional memory. [2]

## Extended Memory

Extended memory is all memory past the first megabyte, which could only be accessed while the processor was in protected mode. The extended memory specification (XMS) was developed to specify how programs would use extended memory. Extended memory required an extended-memory manager, such as HIMEM.SYS.[2, 3]

## Expanded Memory

Some DOS programs used a type of memory called Expanded Memory Specification or EMS memory. Expanded memory was installed on an expanded memory board and came with an expanded memory manager. As EMS was designed for 8-bit systems, The memory manager EMM386 was used instead to convert extended to expanded memory for backwards compatibility. Only programs written specifically to make calls to expanded memory require it. Therefore, some DOS programs use expanded memory while others do not.[2, 4]

# Out of memory? But there is a lot!

An often encountered message is “not enough conventional memory”. As explained in the previous section programs must be loaded into the conventional memory of the system (640K). So no matter what the total memory is of the system, if there is not enough space in conventional memory, the program cannot run.

Most DOS and many early Windows systems load numerous device drivers and TSR (terminate-and-stay resident) programs during the boot cycle. These programs are, by default, loaded into conventional memory, taking up valuable space. Memory management techniques are needed to load these device drivers and TSRs into the upper memory, allowing more conventional memory to be made available to programs.

Customizing the CONFIG.SYS and AUTOEXEC.BAT files to manage the placement of device drivers and TSRs into upper memory blocks (UMBs) in the upper memory area (UMA) on booting, in maximizes the conventional memory available for applications.

The CONFIG.SYS file is a text file containing commands that configure the computer’s hardware components (memory, keyboard, mouse, etc). When DOS starts, it carries out the commands in the CONFIG.SYS file first.

The AUTOEXEC.BAT file is a batch program that DOS runs immediately after carrying out the commands in the CONFIG.SYS file. The AUTOEXEC.BAT file contains the commands to be executed when the system is started. DOS carries out the commands in both the CONFIG.SYS and AUTOEXEC.BAT files **each** time the computer is started.[4, 5]

## CONFIG.SYS

Each hardware component of your computer is called a device. The keyboard, mouse, display, printer, disk drives, and memory boards are all devices. Each device has its own characteristics that can be customized. DOS has built-in device drivers for the keyboard, display, hard drives and diskette drives, and communication ports. Other devices, such as memory boards, a mouse, or CD-ROM have device drivers that are that are not built into DOS. Such a device driver is called an **installable** device driver: these are installed by adding a command to the CONFIG.SYS file. [5, 6]

While most CONFIG.SYS commands can appear in your CONFIG.SYS file in any order. The order of the device and devicehigh commands is important because some device drivers enable devices that are needed by other drivers. The HIMEM.SYS extended-memory driver must be loaded before any drivers that use extended memory.

The order in which device drivers should appear in the CONFIG.SYS file is as follows:

1. DEVICE=C:\DOS\HIMEM.SYS
2. DEVICE=C:\DOS\EMM386.EXE NOEMS
3. DOS=HIGH,UMB
4. Any other device drivers.[5, 6]

The DEVICE commands load the HIMEM.SYS and EMM386.EXE device drivers. The HIMEM.SYS driver manages extended memory. The EMM386.EXE driver, provides access to the upper memory area and simulates expanded memory. The DOS=HIGH,UMB command runs DOS in the high memory area and specifies that programs should have access to the upper memory area.[5, 6]

If programs require expanded memory (EMS), start EMM386 with the NOEMS switch. This can give you an additional 64K of UMBs. The NOEMS switch instructs EMM386 not to create an EMS page frame in the upper memory area. If EMM386 is started with the NOEMS switch, programs will be unable to use expanded memory.

In Code 1 is an example of CONFIG.SYS commands to load the device drivers for memory management, followed by loading DOS onto upper memory. On line 4 is the command to load the driver for a CD-ROM drive. This is specific for the installed drive. The driver might have another name and be located in another directory..

The parameter L/:2 sets the UMB block where the driver should be loaded. The /D: MSCD000 is the device name, not the driver. It is important to note that this name must match what is in AUTOEXEC.BAT. If these names do not match, the CD-ROM drive will not load.

Other commonly used commands in CONFIG.SYS are:

* BUFFERS to specify how much memory is reserve for transferring information to and from disks.  
  e.g. BUFFERS=20



* COUNTRY to set the language conventions for the system.  
  e.g. COUNTRY=031,850,C:\DOS\COUNTRY.SYS for The Netherlands.
* FILES to specify how many files can be open at a time.  
  e.g. FILES=30
* LASTDRIVE to set the number of valid drive letters.  
  e.g. LASTDRIVE=J

## AUTOEXEC.BAT

Figure 2: An Example MEM Report

Code 2: Example AUTOEXEC.BAT

@ECHO OFF

LH:2 C:\DOS\MSCDEX /D:MSCD000

LH /L:2 C:\MOUSE\MOUSE

SET SOUND=C:\SB16

SET BLASTER=A220 I5 D1 H5 P330 T6

SET MIDI=SYNTH:1 MAP:G

PROMPT $p$g

PATH C:\DOS;C:\SB16

SET TEMP=C:\TEMP

SET TMP=C:\TEMP

AUTOEXEC.BAT executes each command in the exact order in which they have been placed. These commands specify where device programs are loaded, and DOS environmental settings are set.

In Code 2 the example AUTOEXEC.BAT file shows loading the CD DOS extension to access a CD drive matching the device specified in CONFIG.SYS. LH loads the driver to high memory.

The mouse driver may have a different name and be in a different directory from the example.

The directory is set the location of the sound card files. The sound card is installed at Address 220 with IRQ (Interrupt Request) 5, Low DMA (Direct Memory Access) on DMA-channel 1, High DMA on DMA-channel 5, MIDI (Musical Instrument Digital Interface) address 330 and that the soundcard is Type 6 (Sound Blaster 16 compatible).

SET MIDI-sets how MIDI files are played. MAP:G ensures that both basic MIDI and extended MIDI play.

The DOS prompt is set to the current drive and path followed by “>”.

DEVICE=C:\DOS\HIMEM.SYS

DEVICE=C:\DOS\EMM386.EXE NOEMS

DOS=HIGH,UMB

DEVICEHIGH /L:2=C:\CDROM\CDROM.SYS /D:MSCD000

The search path for executable files is set.

Code 1: Example CONFIG.SYS

Directories names for temporary files generated by programs are set. Both TEMP and TMP were used as variable names for DOS programs.

## Analyzing Memory

In DOS version 4, the command MEM was provided to analyze memory. With DOS version 6 the MEM command came with additional switches to produce more detailed reports on memory usage.[3, 5]

By running MEM a report similar to Figure 2 is produced showing the different types of memory and what has been used and still available. Most importantly can be seen what the largest executable program size is.[3, 5]

To further analyze memory usage, the /C switch lists all the drivers and programs that have been loaded with their installed sizes.

Altering the CONFIG.SYS and AUTOEXEC.BAT files , rebooting the computer and then running the MEM command enables more control over memory usage by fine tuning the computer’s configuration.

# Summary

In the late 1980s, a memory management architecture was implemented in IBM-compatible personal computers. The memory comprised four defined types: conventional, upper memory, high memory blocks, extended memory plus in some systems expanded memory.

The 640K limit of conventional memory became the standard memory size for running programs in DOS which became known as the “640K memory barrier”.

In order to use other areas of memory available, it is necessary to customize the CONFIG.SYS and AUTOEXEC.BAT files to load the operating system, device drivers and TSRs into these regions.

Examples of possible customizations of commands have been presented to explain the syntax and recommended order in which these commands should be executed on booting the system.

Analyzing the result of configuration amendments can be made using the MEM command which displays a summary of the memory configuration. It shows how much of each kind of memory there is, how much is currently in use, and how much is currently free.

Whether original hardware or emulated systems are used to run legacy software from the late 1980s to the 1990s, knowledge of the memory architecture and how to optimize it are required for a successful experience.

# References

1. Gibson, S., *Placing the IBM/Microsoft XMS Spec into Perspective.* Info World, 1988. **10**(33).

2. Mueller, S., *Upgrading and repairing PCs*. 4 ed. 1994, Indianapolis USA: Que.

3. Prosise, J., *PC magazine guide to DOS 6 memory management with utilities*. 1993, Emeryville, USA: Ziff-Davis Press.

4. Microsoft Press, *Microsoft MS-DOS programmer's reference* 1993, Redmond, WA USA: Microsoft Press.

5. Microsoft Corporation, *Concise User’s Guide Microsoft MS-DOS 6*. 1993, USA: Microsoft Press.

6. IBM, *PC DOS 7*. 1995: International Business Machines Corporation.