RUNNING LARGE MODELS ON A PC PLATFORM

Dudley J. Benton
P-Squared Technologies, Inc.
10938 Hardin Valley Road
Knoxville, Tennessee 37932

presented at the
7th Tennessee Water Resources Symposium
February 24-26, 1997
Nashville, Tennessee

INTRODUCTION

P-Squared Technologies specializes in environmental fluid flow modeling--especially groundwater and surface water systems. These applications are often large models which require large programs. The 640KB[1] memory limitation of standard DOS[2] is far too small for modeling complex environmental phenomena. Utilizing more than 64MB[3] of physical memory or more than 256MB of virtual memory on a PC[4] is extremely difficult and requires software and hardware capability.

PHYSICAL MEMORY

The most significant increase in PC hardware capability came with the introduction of the 80386[5]. The 80386 and subsequent Intel processors are capable of operating in *real* and *protected*[6] modes, or 16-bit and 32-bit[7] operation. In protected mode, these processors are capable of addressing 4GB[8] of memory. Memory above 1MB on these machines is called *extended*[9]. This processor capability is misleading; however, as no currently available PC can contain 4GB of physical memory (often called RAM[10]). Furthermore, the top 2GB is reserved for the operating system and is not available for user programs. Thus, no more than 2GB will ever be available on a 32-bit machine--regardless of the claims of software vendors promising access to 4GB. Practically, most PCs, even Pentium[11]-based machines, are limited to 128MB of RAM and very few single-processor Intel systems are commercially available which can support more than 512MB of RAM.

Even if the memory chips can be purchased, and even if they can be inserted into a machine, this does not assure that it will actually recognize any more than 64MB of RAM. Even if the BIOS[12] recognizes the memory above 64MB during boot-up, this does not assure that any available software can utilize this additional memory. Some systems (e.g., Gateway P5-90s) when augmented from 64MB to 128MB actually report less than the original 64MB.

The DOS and Windows[13] extended memory manager[14], HIMEM.SYS[15], can not report any more than 64MB of extended memory. This is by virtue of its protocol, and is not correctable. A different protocol is necessary in order to report more than 64MB. Some

---

[1] KiloByte or 1024 bytes

[2] Disk Operating System - a trademark of Microsoft

[3] MegaByte or 1024×1024 bytes

[4] Personal Computer - a trademark of IBM

[5] microprocessor and trademark of Intel

[6] The etymology of the terms *real mode* and *protected mode* arise from the respective memory addressing. In *real mode*, memory addresses correspond directly to *real* locations in physical memory. In *protected mode*, memory addresses are translated within a program context and may not have a literal correspondence to physical memory. In *protected mode*, some areas of memory are *protected* from alteration and/or access.

[7] 16-bit and 32-bit refer to the size of the integers used to address memory. A 16-bit integer can range from 0 to 2^{16}-1 (or 65536). Thus, a processor operating in *real* mode, using 16-bit integers to address memory, must use *segmented* addresses, or a combination of two 16-bit integers. Intel processors overlap these two to form a 20-bit address. As 2^{20} equals 1048576, or 1024×1024, this is the source of the 1MB memory limitation of DOS. A 32-bit integer can range from 0 to 2^{32}-1 (or 4294967295); thus, accessing megabytes or memory requires 32-bit operation.

[8] GigaByte or 1024×1024×1024 bytes

[9] physical memory extended beyond the 1MB 16-bit addressing capability

[10] Physical memory *chips* are often referred to as *RAM* or Random Access Memory. This is to distinguish it from Sequential Access Memory (or *SAM*) devices such as a tape or disk drive.

[11] microprocessor and trademark of Intel

[12] Basic Input/Output System, or the lowest level software which supports the operating system. The rolling numbers displayed during the power-up procedure indicate the physical memory recognized by the BIOS.

[13] Windows is a trademark of Microsoft

[14] An extended memory manager is memory-resident (i.e., ever-present) software which controls access to the memory above 1MB.

[15] HIMEM.SYS is an extended memory manager developed by Microsoft.

machines (e.g., Zeos Pentiums), have a switch which enables Windows-NT to access up to 256MB of RAM by altering the HIMEM.SYS protocol. Enabling this switch causes the BIOS to report up to 256MB. However, this causes non-NT programs which utilize this protocol to report only 16MB of RAM, making this a non-solution to the 64MB RAM ceiling problem.

VIRTUAL MEMORY

Virtual memory, or disk-swapping, is provided by software. Windows-V3, Windows-95, and Windows-NT all provide virtual memory access (VMA). This is done by creating a swap file and placing *hooks* in the protected-mode memory access which will generate an *interrupt*[16] when these phantom memory locations are accessed. The VMA software can then swap in and out active and inactive memory. This enables the computer to function as if it had more memory than is physically present. As this additional processing takes time, and disk drives are much slower than RAM, programs utilizing virtual memory may run much slower than those which can function entirely within the limits of physical memory.

The biggest limitation of VMA is seen with Windows-V3, which will create, but refuses to utilize a swap file larger than four times the physical memory. As HIMEM.SYS is limited to 64MB, this results in a maximum Windows-V3 swap file of 256MB. It would seem only logical that, were a user willing to commit the disk space and runtime, Windows should have no upper limit on the swap file size; however, this is not the case. There is a marginally-documented Windows SYSTEM.INI parameter, *PageOverCommit*, which has a default value of 4 and is said to accept any value between 1 and 20. This would seem to indicate that the 256MB ceiling is linked to this parameter. The *PageOverCommit* parameter can indeed be set in SYSTEM.INI to values between 1 and 20; however, Windows-V3 refuses values above 4; thus, the 256MB limitation remains. Windows-95 and Windows-NT will create any size swap file, limited only by available disk space. Rather than dynamically growing as needed, a Windows-V3 swap file larger than 20MB must be made *permanent*, which results in a dedicated allocation of disk space.

[16] An *interrupt* is a notification of the operating system and can be hardware or software generated. Depending on the nature of the interrupt, the processor may suspend the current activity, process the interrupt, and then resume the previous task; or it may save the interrupt in a queue and process it later.

As different compilers (which may be necessitated by the use of several models, pre-processors, and post-processors written in different languages or which depend on compiler-specific features) utilize different virtual memory managers, separate hard disk space must be dedicated to each. When running several models which require in excess of 256MB of virtual memory and may require/produce 512MB of input/output files, available disk space becomes a serious problem. Repeatedly switching from one virtual memory manager and/or operating system to another is not only time-consuming and tedious, it increases the likelihood of hardware failure.

Pharlap sells a DOS extender, TNT, which allows non-Windows programs to run in protected or 32-bit mode. Pharlap also sells a virtual memory manager, VMMDRV, which adds disk-swapping. This software, while expensive and cumbersome, does increase capability, as its virtual memory is limited only by available disk space, unlike the virtual memory provided by Windows-V3. VMMDRV, however, will not provide virtual memory inside any Windows environment. It must be run in DOS, outside Windows. Under DOS-6.22, the Pharlap DOS extender will only access the first 64MB of physical memory. This eliminates any advantage in purchasing more memory. However, running in what might be called DOS-95, or the residual DOS left after a shut-down of Windows-95 and warm boot, physical memory above 64MB can be accessed, provided the hardware and BIOS are compatible.

COMPILERS AND DOS EXTENDERS

Two languages are essential: FORTRAN and C. Large models require that the compilers be able to generate 32-bit code. Running 32-bit code on a PC requires either a DOS extender or a Windows-NT executable (Windows-95 will run Windows-NT programs). The project in which these operating systems and compilers were evaluated required running pre-processors and post-processors written in C and a solver written in FORTRAN. Getting these programs to run on a single platform and in a single configuration is very important.

WATCOM, Lahey, Salford, and Microsoft PowerStation FORTRAN compilers were tested. All but the PowerStation produce extended DOS executables. WATCOM uses the Rational Systems DOS extender. Lahey uses the Pharlap DOS extender. Salford uses the DBOS DOS extender. PowerStation executables will only run under Windows-95 or Windows-NT and will not run in DOS outside of Windows.

Due to an internal limitation of the Microsoft PowerStation FORTRAN Version 1, programs having more than 256MB of static arrays will not run under Windows-NT, but will run under Windows-95. Microsoft indicates that PowerStation programs larger than 256MB should not run on any system. The Microsoft *solution* to this discrepancy, which came out with Version 4 of the PowerStation (there were no versions 2 and 3), was not to repair the problem so that larger programs would run under Windows-NT, but to assure that larger programs would no longer

run under Windows-95. Thus, if more than 256MB is required, the old version of PowerStation along with Windows-95 is required.

The Rational Systems and Pharlap DOS extenders supplied with the WATCOM and Lahey compilers, respectively, are *crippled* in that they are limited to a few megabytes. An expanded version of the Rational Systems DOS extender for use with WATCOM FORTRAN which would provide access to the hundreds of megabytes required for the current project is unreasonably expensive. A network license or a distribution license (required if the executables are to be run by another party) for the Rational Systems DOS extender are quite expensive.

Lahey FORTRAN suffers from a similar problem as WATCOM in that large fees are required for Pharlap support--a situation not made clear when purchasing the compiler. Perhaps the most irritating factor associated with the Pharlap DOS extender is that the versions which support the FORTRAN compilers will not support the C compilers, or vice versa; thus making it necessary to purchase two costly DOS extenders.

C compilers also present an array of frustrations. Microsoft C Version 7.00 is an excellent example of conglomerated problems. The documentation states in many places that compilation can target the 80386 processor family. This capability, however, is not part of the software. The compiler will produce some form of 80386-targeted code if a Windows executable is generated. The problem with this, is that a Windows-based executable requires a Windows-based, rather than console-based interface, as well as many more changes. Building a Windows-based interface for an already-functioning computational model can be an enormous effort. There is a simulated console interface available with several C compilers which can be built into a Windows executable; however, this is limited to small programs. The further limitation of this compiler is that it does not utilize the power of the 80386 and later processors for even simple calculations involving integers. This console interface limitation and processor 32-bit function utilization is not a problem with Microsoft PowerStation FORTRAN.

WATCOM C suffers from the same limitations as WATCOM FORTRAN, arising from a dependence on the Rational Systems DOS extender.

Symantec C (formerly called Zortech) is by far the most impressive 32-bit compiler for the PC platform. The switch from 16-bit to 32-bit compilation is effortless. Symantec C supports several 32-bit options, including DOS, Pharlap[17], and Windows-NT. The Windows-NT option will only run inside Windows-NT, Windows-95, or WIN32s; however, virtual memory is not accessible under WIN32s when compiled with this option.[18] In the case of Windows-NT and Windows-95, a *console interface* is provided by internally. No such compatibility is available with WIN32s.[19] Thus the only solution that Symantec C provides for more than 256MB is to produce a Windows-NT executable with the native console interface provided by Windows-95 and Windows-NT. Symantec C does not provide any solution for large models running in a DOS or Windows-V3 environment.

High-C is an expensive and cumbersome compiler which seems to work well with the Pharlap DOS extender; but offers no advantage over the Symantec C compiler; and, was found to be inferior to Symantec C in each of the categories discussed above.

SUMMARY

In summary, many more difficulties and frustrations are associated with running large models on a PC platform than one might expect, based on the claims of compiler vendors. Repeated references to gigabytes of memory are misleading, at best. The availability of memory slots capable of seating many megabytes of RAM are also misleading. For FORTRAN code, only one configuration has been found which can support very large models and at least 256MB of RAM: early versions of Microsoft PowerStation running under Windows-95. For C code, the Symantec compiler with the Windows-NT target operating system, and running Windows-NT or Windows-95 provides access to more than 256MB of physical and virtual memory. Therefore, the combination of Microsoft PowerStation and Symantec C compilers targeting Windows-95 is the only combination found which enables all of the programs to run on a single platform.

---

[17] Note that the recent documentation for Symantec C and Pharlap DOS extender do not indicate that these are compatible; however, this option functions quite well through Version 6.11.

[18] There is the further, marginally-documented limitation of WIN32s, that no single array can be larger than 16MB (the practical limit is more like 15MB).

[19] While a program compiled with this set of options will indeed run, no output will be displayed during operation, and it must be launched from FileManager, not from the DOS box.