HRN
Papers file.
Include copies with Chinese packet.
2

Exploration Software Development Using the IBM-PC

Terence L. Smith
Landmark Graphics Corporation
Houston, Texas

ABSTRACT

The IBM PC is an important tool for developing application software packages for the exploration and production industry. It offers a wider range of languages, linkers, debuggers, and advanced hardware tools than any other microcomputer system. In addition, the IBM PC family offers a variety of communications methodologies for transferring data to mainframe and large computer systems. The connectivity between PCs, workstations, and mainframe computers includes both lower-speed asynchronous communication and higher-speed networking solutions.

This paper describes how off-the-shelf IBM-PC capabilities were used to develop an advanced turnkey interpretation workstation for the exploration and production segment of the petroleum industry. Topics covered include the requirements of having dedicated computer cycles for the software engineer, code development tasks, and code maintenance.

A key factor influencing the applicability of the IBM PC as a software development tone is its general availability and low cost.

Introduction

When Landmark began building its stand-alone interpretation workstation, in late 1982, we were immediately confronted with the need to produce hardware and software simultaneously. This meant we needed to identify and use software development tools while the initial hardware prototypes were being built. At a minimum, we needed development computer systems that would allow us to edit source code and to do test compiles. We knew it would be desirable to link and test as much as possible on the development system, since available time on fully functional interpretation workstation would continually be scarce. Our solution was to provide each programmer on staff with an IBM PC. Since that time, Landmark has acquired a mixture of IBM PCs, XTs, ATs, and IBM compatible PC's such as Compaq portable computers. Hereinafter, any of these computers are referred to as "PCs".

Architecture

The baseline architecture for hardware was defined concurrently with an analysis of the software tools available. Landmark settled on FORTRAN as the standard language for application development, because it is the de facto language of the petroleum industry. When libraries of code are provided to customers, for example, they are generally required to be FORTRAN-callable; or when vendors supply drivers for new devices (graphics, array processors) they are generally FORTRAN-callable. While there are some limitations to microcomputer FORTRANs, it was the first high-level language to meet key requirements. Our basic need was to have access to the entire megabyte address range available under MS-DOS on the Intel-8086 and 80286, which forms the basis for Landmark's workstation architecture.

Software Development Tools

The PC was an excellent choice for development of scientific software for several reasons. It offers a rich set of programming tools, including a variety of source code editors; good implementation of all the currently popular programming languages, such as FORTRAN, C, PASCAL, and BASIC; good debugging tools; and an expandable capacity. In addition, the PC provides access to most of the latest hardware technology, such as high-capacity disks, networking support, and some graphics. The PC itself has available a floating point chip that offers substantial speed improvement on floating point operations.

Dedicated compute cycles

The most heavily used resource in a software development environment is computing cycles. The PCs provide dedicated compute cycles to each programmer. This enables editing of source code, program compiles, and linking and testing modules that go into many of Landmark's basic libraries. Since we can run the same operating system and compilers on the PCs as we run on the workstation, programs that execute successfully on a PC can be transferred directly to the workstation. When each programmer has a own PC for the bulk work of editing, compiling, and linking, there is no competition with other programmers for those cycles.

Networking

A second component of Landmark's highly productive development environment is our local area network. As soon as we integrated graphics with a prototype workstation, we installed an Ethernet local area network (LAN). Each programmer's PC is connected to the interpretation station via the LAN. This enables us to do very rapid prototyping of new applications. The programmers can link their own version of the application, ship it to the prototype over the network, test it, record bugs, and go back to a PC to edit source code, compile, and link. This use of the LAN has virtually eliminated competition for graphics resources.

The PC was an excellent choice, once again, to support our networking activities. A basic reason is the wide variety of networking products to choose from for both hardware and software. Multiple vendors of boards enable us to keep from being dependent on a single source. We are also able to write our own networking software, and a substantial amount of this work was done on PCs using C and assembler language. Only testing of boards and final integration must be done on the workstation.

Code development

In our networked environment, access to workstations is necessary only for testing of code, never for compiling, linking, or other conventional software maintenance activities. Expensive resources like tape drives, array processors, and high-performance graphics are attached to the full interpretation workstations. Compute-cycle intensive activities, such as coding, take place on PCs. The result is greater utilization of all hardware resources. At the same time, this environment provides the maximum amount of computer time to the development staff. Several Landmark programmers have PCs at home to allow continued testing and development in off hours. Using these concepts, we have not been seriously constrained by our hardware resources.

A most significant fact is that because programmers have his or her own PC, the feedback loop is shortened on many phases of the software development, hence "flow" is established with the work. This high-frequency feedback loop is instrumental in achieving high individual productivity.

Code maintenance

Code maintenance is handled by having a file server on the net. Each programmer has a copy of working code and libraries on their PC. Production modules that need to be updated are "checked out" by means of a standard set of procedures and copied to the programmer's PC via the network. However, changes to be submitted for final integration are not copied back via the network, but are submitted in a controlled way. Changes are compiled afresh from the submitted source code, new libraries are built, and test versions of executables are linked. The source changes are not certified until the regenerated executables are tested by the submitting programmer. The logistics of this are all quite simple, because testing can be done on any LAN node.

The file server containing the master copy of the software is backed up daily. Additional backups are made at key points in the development cycle, such as major releases. However, programmers are responsible for backing up the contents of their own PC. This shares the burden of making backups. Daily copies of the latest verified code are saved. In addition each programmer backs up the portion of latest changes that are worth keeping.

Printers

An additional advantage of the networked development environment is that it is always possible to increase capacity by adding more nodes. For each new programmer, we simply add a new node. We can also add relatively expensive devices to the network by placing them on an existing node where everyone can access them. Examples are color plotters, digitizers, host computers, microVAXs, or special-purpose printers. While there is shared access for quick turnaround, performance does not degrade.

We have installed printers on the network on a print server. This allows all of the programmers to have shared access to the printer. Near-future plans call for installation of a variety of plotters on various servers in order to share access.

Development environment cost

One of the key benefits in selecting the PC is the price. It costs approximately $7,000 for a network-configured PC with development software. This is extremely inexpensive compared to allocated mainframe costs. The low cost of PCs also make it possible for many members of the development staff to have machines at home.

Summary

The IBM PC is an important component of Landmark's software development strategy. It provides each programmer with state-of-the-art tools and dedicated access to computing cycles. At the same time, file servers provide centrally managed storage for production code; print servers provde shared access to printing; and other expensive resources can be added to the network to be shared as needed. Nodes dedicated to programming activities and rapid prototyping of new applications will continue to be an important component of exploration and production software development.