U.S. Department of Justice
Office of Justice Programs

A Guide to Selecting Criminal Justice Microcomputers

Bureau of Justice Assistance

MONOGRAPH

U.S. Department of Justice
Office of Justice Programs

A Guide to Selecting Criminal Justice Microcomputers

Monograph

Bureau of Justice Assistance

July 1990

U.S. Department of Justice
Office of Justice Programs
Bureau of Justice Assistance

A Guide to Selecting Criminal Justice Microcomputers was prepared by SEARCH Group, Inc. under Cooperative Agreement No. 87-SA-CX-K086 by the Bureau of Justice Assistance, U.S. Department of Justice. The points of view stated in this document are those of SEARCH Group, Inc. and do not necessarily represent the official position or policies of the U.S. Department of Justice.

Bureau of Justice Assistance
633 Indiana Avenue, N.W., Washington, D.C. 20531
(202) 514-6278

U.S. Department of Justice
Office of Justice Programs
Bureau of Justice Assistance

Office of the Director

Washington, D.C. 20531

The Bureau of Justice Assistance (BJA) was established in 1984 as a component within the Office of Justice Programs, U.S. Department of Justice. As part of its charter and in direct response to justice practitioners' information management needs, BJA has provided national leadership in the development of justice information systems, programs and technical assistance for state and local agencies. Recognizing information technology as a vital component of the criminal justice infrastructure, BJA has encouraged practitioners to acquire and enhance automated information systems. The agency has also advocated the sharing of system information among justice agencies nationwide.

I am pleased to present *A Guide to Selecting Criminal Justice Microcomputers* as product of BJA's multi-faceted and continuing program of support for the development and acquisition of justice information systems. This publication should be of tremendous assistance to small criminal justice agencies seeking to work their way through the maze of computer-related information in their search for an appropriate system.

This guide presents a step-by-step process for planning, investigating and selecting the system software and equipment best suited for your agency's information needs. It describes the basic components of a microcomputer, suggests a course for preliminary planning, reviews the purpose and process of a needs assessment and helps you to define your user requirements. The text also leads the reader through the process of conducting a feasibility study as well as examining, comparing and then scoring alternative software packages and hardware components. For larger acquisitions which require vendor-provided configurations, *A Guide to Selecting Criminal Justice Microcomputers* describes the creation and distribution of Request for Proposals (RFP's). Finally, contracts and optional (perhaps unexpected) cost are discussed and installation requirements reviewed. A lengthy glossary of microcomputer terms and a resource directory round out this practical guidebook.

A Guide to Selecting Criminal Justice Microcomputers should be of practical assistance to criminal justice practitioners who are just beginning the oftentimes frustrating process of procuring an automated system that will meet their agencies' unique information needs.

Sincerely,

Gerald (Jerry) P. Regier
Acting Director

Acknowledgments

A Guide to Selecting Criminal Justice Microcomputers was completed under the auspices of SEARCH Group, Inc., the National Consortium for Justice Information and Statistics. Gary L. Bush is the Chairman and Gary R. Cooper is the Executive Director of this nonprofit consortium of the states which is dedicated to improving the criminal justice system through the management of information technology.

All work on *A Guide to Selecting Criminal Justice Microcomputers* was supervised by David J. Roberts, SEARCH Deputy Director, Programs. The *Guide* was written by Judith A. Ryder, Director, Corporate Communications; editorial and formatting assistance was provided by Twyla R. Thomas, Information Specialist and Jane L. Bassett, Publishing Assistant. The project was conducted under the direction of R. John Gregrich, Chief, Corrections Branch, Bureau of Justice Assistance, U.S. Department of Justice.

Table of Contents

Introduction

In rapidly increasing numbers, local criminal justice agencies are taking advantage of microcomputers' low costs and tremendous versatility. Departments are using microcomputers to speed simple administrative tasks — such as filing and retrieving information — and to assist in complex case investigations by analyzing relationships between disparate data on suspects, crime trends and statistics. Microcomputers can manage patrol cars, facilitate dispatching operations or maintain internal personnel and payroll records. At a more advanced level, computers can chart geographic information, track homicides, identify latent fingerprints and compile mug shots.

Obviously, microcomputers can be applied to a broad range of justice activities, quickly performing tedious and mundane tasks and efficiently managing a staggering volume of critical information. If properly selected and incorporated into the daily operations of an agency, a computer should enable that agency to increase productivity and free personnel for more creative and analytical work.

Buying a computer system, however, is much more than simply an exercise in equipment procurement. Selecting a computer is an important step in establishing your information management style, setting your priorities and managing your agency. The computer you select is a tool that should *support* your information management style or approach — not *determine* it.

You should not purchase any equipment until you can write down how you want to organize your agency over the next few years and determine how computers fit into that scheme. To improve agency productivity and efficiency, first examine any underlying problems and concentrate on organizational changes. Establish your priorities and consider the broad policy implications of those priorities. You should also consider future developments in organizational structure, technology, management, and the criminal justice arena; these will alter how your people work and how your agency serves its community.

The actual purchase of a microcomputer should come only after you have determined your needs, investigated what will best meet those needs, and justified the computer expenditure with the same analytic scrutiny you bring to any other agency purchase. Computer technology should enable you to better pursue your agency strategy: A computer is not an end in itself; it is a means of achieving strategic management objectives.

A Guide to Selecting Criminal Justice Microcomputers is designed to describe a basic, but comprehensive, process managers of small- to medium-sized criminal justice agencies should follow.

- Determine your information management needs.
- Acquire a system appropriate to those needs.
- Implement a system that enhances agency productivity.

It is our intention to provide you with enough information to become confident, knowledgeable consumers, able to identify your agency's computer needs and to translate those needs into a specific system configuration.

Although this *Guide* outlines the optimum or ideal process and is designed to lead you through what may be unfamiliar territory, the book should not restrict you in any way. Rather, we encourage you to read through it, review the major steps, and then tailor the procedures to suit your needs. For example, we suggest that the entire acquisition process be managed by a Project Team comprised of individuals from several units and having different areas of expertise. Pulling together a diverse group of people helps to ensure a comprehensive and credible review of the agency's needs, encourages cooperation and personal investment in the process, and builds acceptance for a new system. If you have only 10 people in your agency, however, it may not be feasible to set up a large task force or committee. With a proper understanding of the purpose of a Project Team, you might instead select three people with complementary backgrounds to manage the project.

THE ACQUISITION PROCESS. Acquiring a computer system is an exciting, if demanding, process. In this publication, we have attempted to outline a smooth and logical method for selecting the computer system best suited to your needs. How you choose to implement this process will depend on the amount and type of resources available to you (time, money and people) and the

extent of your automation needs. (Are you planning to automate an entire department, a unit within an agency or a single function within a unit?) Depending on the size of your agency and the extent of your automation needs, the acquisition process can be as limited or as expansive as you need it to be.

Too many people, in their haste to automate, have bought equipment before they really knew what functions they wanted the computer to perform. Such purchases can produce unnecessary or inappropriate equipment, as well as require expensive software modifications. Before you approach a computer vendor, it is important for you to "do your homework," which begins with a critical evaluation of your current system. Procuring the best system for your agency can be a complex project, so we have tried to simplify it by breaking it down into several manageable steps. You and others from your agency need to plan the acquisition, identify information needs, define user requirements, analyze the feasibility of the project, investigate computer components and negotiate a satisfactory contract with a vendor. Each of of these steps or subprocesses is important, but each can be easily modified to fit your resources and requirements.

CHAPTER ORGANIZATION. ***A Guide for Selecting Criminal Justice Microcomputers*** is divided into nine chapters that logically lead you through the procurement process. Each chapter represents a step in the process and builds on information relayed in the previous chapter. Although you are encouraged to tailor each step to your own agency's needs, it is important to understand the objective and rationale of each stage.

The first chapter, *Overview of Microcomputer Components*, is a general examination of hardware elements and software packages. If your agency has little or no experience with computers, it is critical that personnel involved in the procurement process familiarize themselves with basic computer terminology and components. The chapter discusses the workings of the central computer; describes input, output, storage and communication devices; and explains the different purposes and characteristics of system, utilities and application software. Additional informational resources (such as computer courses and seminars, magazines and books, and user groups) are also described.

This first section is followed by a chapter on *Preliminary Project Planning*, which stresses the importance of a formal planning structure. The creation of a Project Team (whether it consists of three or 13 people) and the selection of a Project Manager help

keep the acquisition process manageable and accountable. This chapter also explains the importance of reviewing your agency's goals and objectives to help determine which functions or areas could most benefit from automation.

Your agency's specific information needs — not the latest technological wonder — should determine which system is best for you. Thus, a first step in the process must be to define those needs as clearly and in as much detail as possible. What do you really want the microcomputer to do? How is your current system managing (or failing to manage) the defined tasks? Chapter Three, *Conducting a Needs Assessment*, outlines the process for defining your information problem or need and analyzing your existing system to determine exactly where automation can help.

Chapter Four, *Defining the User Requirements*, directly builds on the information gathered in the needs assessment. Having completed an extensive review of the current system, you are now ready to define two major objectives: the tasks the proposed system is required to perform and the specific capabilities the system must have. List the improvements in a new system both that users *require*, and those users *desire*.

The purpose of Chapter Five, *Conducting a Feasibility Study*, is to help your Project Team and agency management decide whether the acquisition process can and should move forward and to examine various system alternatives. The feasibility study should include an assessment of the costs and benefits associated with each alternative and a review of available financial, technical and administrative resources.

Investigating Hardware and Software is central to the acquisition process. It is essential that your Project Team explore the microcomputer world, talk to vendors and users, read reviews, and actually sit down at computer terminals and test the equipment and software. Chapter Six suggests a process for gathering information on available software and hardware, analyzing the systems that hold the most potential for your agency, comparing those options and scoring them against your required specifications.

The next chapter, *Requests for Proposals*, is aimed at those agencies that plan to purchase a specialized or particularly complex system, one that will require the assistance of a vendor to arrange the appropriate configuration at a reasonable cost. The chapter explains the process of developing a formal Request for Proposals (RFP) and reviews the major components to be included in the final document. This chapter also briefly explains how to evaluate the proposals and select a vendor.

Whether you decide to purchase a system off the shelf or require customization, at some point you will probably need a formal contract. Chapter Eight, *Contracts and Optional Costs*, points out some of the key provisions in typical computer system contracts (for both hardware and software) and alerts the reader to costs that often must be added to the basic price of the system. These provisions and costs can include shipment and delivery of the system, installation, maintenance and training.

In Chapter Nine, *Implementation*, you are urged to prepare a comprehensive implementation plan to ensure a smooth transition from your current system to a new one. The implementation plan should review and prepare for the system's physical requirements, include schedules for data conversion and training classes and make any necessary adjustments to job classifications.

The document concludes with an extensive *Glossary* of microcomputer terms and *Selected References*, which lists several books, directories and organizations that can assist you in your acquisition process.

1

Overview of Microcomputer Components

Knowing the parts and functions of a microcomputer is the first step toward choosing the system configuration appropriate for your agency's needs. Before you can begin to develop a plan for acquiring a microcomputer, you need to be familiar with microcomputer terminology and to gain a basic understanding of the parts of a computer system and how those parts work together. As the selection process continues, you will want to know fairly detailed information about particular microcomputers, including performance characteristics, costs and the types of compatible software. If you are not already familiar with microcomputers, this first section of ***A Guide to Selecting Criminal Justice Microcomputers*** will provide a brief overview of the major components of a microcomputer configuration, explaining how individual parts are put together to provide a range of capabilities.

This chapter draws heavily from SEARCH's ***The Criminal Justice Microcomputer Guide and Software Catalogue,*** which provides a very detailed explanation of microcomputer components and software packages designed specifically for criminal justice applications. If you have not already done so, we suggest you review that publication. You should also take advantage of microcomputer courses and seminars presented by local colleges and become familiar with the many computer magazines and handbooks that are available for study.

Basically, a computer is a system made up of a number of components or subsystems. Every computer has a central computer, which includes the Central Processing Unit (CPU) with its memory that represents the "brain" of the computer. The computer needs to communicate with its human operator and thus must have a way of receiving information from and transmitting information to the operator: Input/Output functions and circuits allow the computer to accomplish this task. Typically, the operator sends data to the computer by typing on a keyboard, and the computer responds on the screen in English symbols and graphics, as well as on a printer. Oftentimes, more memory is needed than is stored in the CPU, so the unit may require additional disk

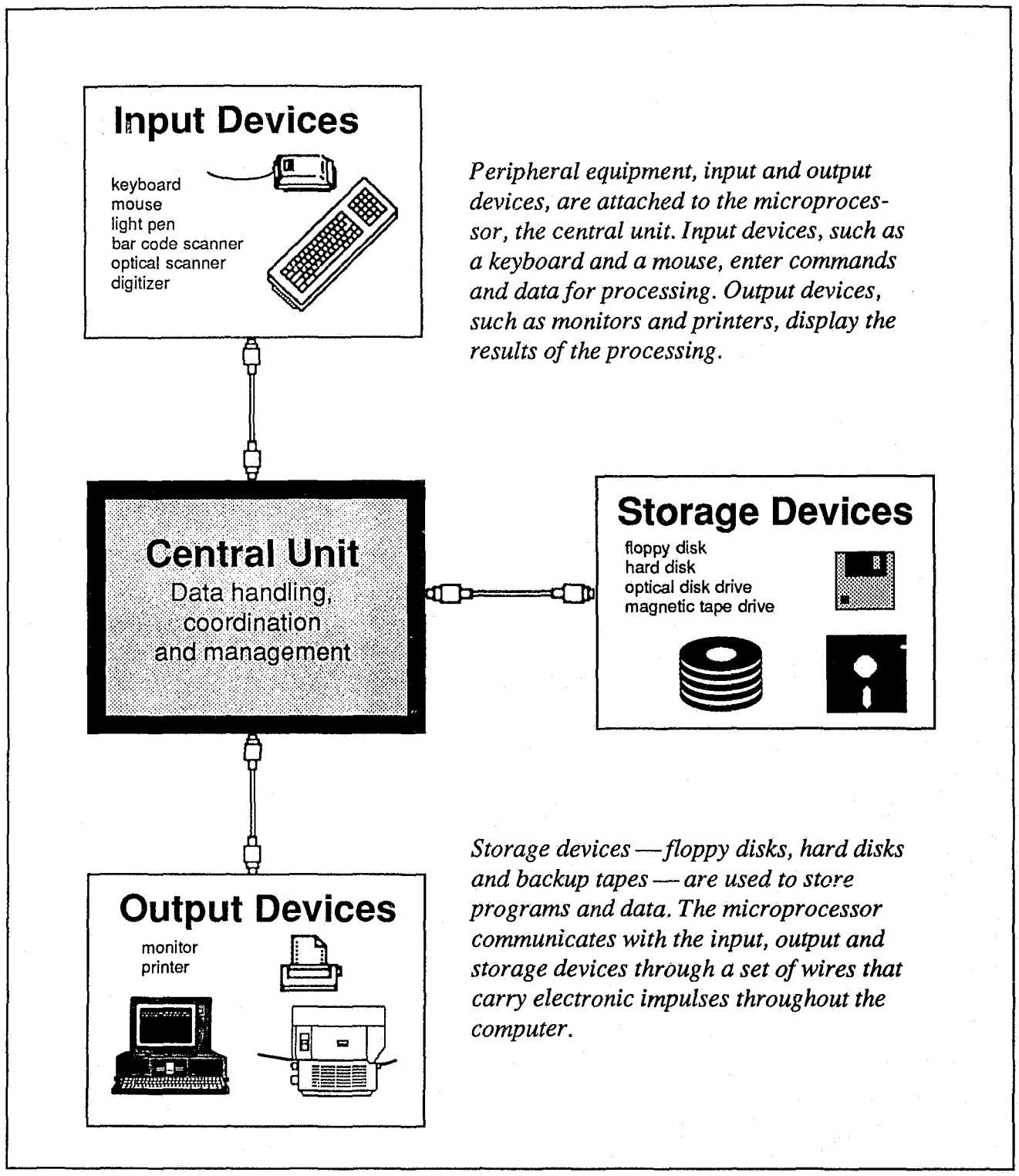

Figure 1-1: Physical Configuration of a Computer and Peripheral Units

drives (which provide mass storage). Thus, today's typical microcomputer system will contain the central computer, a keyboard, a display monitor, storage devices and a printer — the subsystems illustrated in **Figure 1-1**.

All microcomputer components can be categorized by hardware (the physical parts of the computer) and software (the instructions that tell the computer what to do).

Hardware

The Central Computer

Generally speaking, hardware consists of the physical, electrical and mechanical devices of a computer — the parts you can touch. Computer systems can be arranged in almost limitless configurations, but every microcomputer configuration requires a few essential components which, together, comprise the central computer.

- The CPU, also known as the microprocessor chip
- Memory
- Input/Output (I/O) functions and circuits

Briefly, the CPU reads the user's instructions and relays the information to the other parts of the machine, instructing it what to do. The I/O connection links the central computer to all the peripherals and accessories. There are two kinds of memory: One type holds the information you are working on at any particular time, and the other stores the information permanently. Each of these components of the central computer will be reviewed briefly before we discuss peripheral devices such as keyboards, video displays, storage and printers.

The Central Processing Unit (CPU)

The CPU is a microprocessing silicon chip upon which a tiny electrical circuit has been built. Its intricate wiring allows it to interact with the user and the software. In conjunction with the microcomputer's memory, the CPU manages and coordinates all operations, issuing orders throughout the system. As the "brain" of the computer, the CPU is the electronic device that interprets the software instructions and the data entered into the computer, places the information in internal memory, manipulates the data

and controls the sequence of operations contained in the software instructions.

The type of microprocessor determines how much memory is available for use, which operating systems it will run, how fast the microcomputer will perform and how compatible it will be with other systems. Therefore, it is important to know something about a microcomputer's capacity for storing data and processing speed.

Computers "think" in machine language, based on a binary code, which means that the language has only two digits: one and zero. Each digit, a one or a zero, is called a *bit*. (This acronym is derived from the words binary and digit.) Microcomputers store information in a *byte* (the number of bits needed to encode one character of data). Generally, a byte is composed of eight bits. A microprocessor is designed to "think" and "remember" in bytes of a minimum specific width. The microprocessor moves the bytes along a *bus* (an electronic pathway), which connects the microprocessor to the system's memory and other devices.

There are three basic classes of microprocessors: 8-bit, 16-bit and 32-bit. The classes are distinguished by the number of bits they can process at one time and the speed at which they can be processed. The width in bits is an indication of how many characters of information can be accessed in memory at any given time. For example, an 8-bit microprocessor can access only one byte (or one character) in a cycle and requires four cycles to access four characters. A 32-bit microprocessor, however, can access four characters in one cycle (8 + 8 + 8 + 8 = 32 bits). Thus, the amount of work done by the 32-bit microprocessor is four times greater than that of an 8-bit microprocessor at any given moment.

Processor speed involves three factors. The first is cycle time, which is measured in megahertz (MHz, millions of cycles per second). Cycle time is the amount of time it takes the microprocessor to get data from the memory and return it. The more cycles per second, the faster the microprocessor can process data. The second consideration is the amount of data that can fit into the Arithmetic Logic Unit (ALU), where all the computing takes place. And the third factor is the processor's architecture, which limits the size, and therefore the number, of addresses (location of data stored in the main memory) it can handle. The amount of addressable memory available should be of concern if you want to run applications that require a great deal of data to be in memory at any one time.

Common Microprocessor Speeds

Microcomputer	Microprocessor	Speed
Apple IIc	8-Bit	1.02 Mhz
Apple II GS	8-Bit	2.8 Mhz
IBM PC	8-Bit	4.7 Mhz
IBM AT	16-Bit	6.0 Mhz
Apple Macintosh SE	16-Bit	7.8 Mhz
Tandy 1000 TX	16-Bit	8.0 Mhz
IBM PS/2 Model 80	32-Bit	16.0 Mhz
Compaq 386	32-Bit	16.0 Mhz
Tandy 4000	32-Bit	16.0 Mhz
Apple Macintosh II	32-Bit	15.7 Mhz
IBM PS/2 Model 80 Enhanced	32-Bit	20.0 Mhz
Tandon 386	32-Bit	20.0 Mhz

Table 1-1: Common Microprocessor Speeds

The wisdom of acquiring a system with more processor speed depends on what you are doing. Speed is usually more important with multiuser systems than with stand-alone systems and for applications that require the system to sort or statistically analyze large quantities of data. For a general idea of relative CPU processing speeds, see **Table 1-1.**

Memory

• *Random Access Memory*

Random Access Memory (RAM) is the "thinking" memory of the microcomputer; it is where the microprocessor temporarily stores programs and data. The system can use RAM like a scratchpad — storing, moving and erasing data and programs at different times as needed. RAM is also erasable and "forgets" everything upon command or at any time the power is turned off (or unexpectedly goes off, in the case of a power failure).

RAM is the internal working area of the computer — much like your desktop where you place documents and writing materials. When you place a floppy disk containing a word processing program into your microcomputer, a copy of the program is read into the RAM. As you begin composing your report, the data are temporarily stored in the RAM as well. Through random access of the microcomputer memory, the microprocessor can call upon any piece of data or any word processing instruction with equal

speed. When you are finished with the writing session, you must save your typing to a storage device, because RAM provides only temporary storage; when you quit the program, everything in the RAM is erased.

When purchasing a microcomputer, you will want to know the minimum and maximum amount of memory available in any particular model. The maximum amount of main memory indicates (1) the maximum number of memory locations that can be addressed by the microprocessor and (2) the memory expandability of the system.

• *Read-Only Memory*
Read-Only Memory (ROM), unlike RAM, is not erasable and is permanently inscribed in the ROM chip. It is designed by the manufacturer to permanently store programs, such as the computer's startup program, operating system and compiler or translator programs, which manufacturers do not want users to modify. ROM is designed specifically for a particular system and is installed by the manufacturer.

All memory is measured in bytes, expressed in thousands (K) or millions (M) of bytes.

Input and Output (I/O) Interfaces

The third major component of the central computer is the section that links the CPU to the memory and both of them to all of the computer's peripherals, such as the keyboard, the screen, the storage filing units and the printer. The I/O interface serves as the computer's central switchboard or connecting link. For example, we send words and numbers into the computer by typing on a keyboard. The information goes to the I/O circuit, which sends it to the CPU and to the screen. If you want to print what you have typed, the message is sent to the I/O, which sends it to the CPU and then to the printer.

Peripherals

Peripheral equipment includes input and output, storage and communication devices. Input devices, such as a keyboard and a mouse, enter commands and data for processing. Output devices, such as monitors and printers, display the results of the processing. Storage devices — floppy disks, hard disks and backup tapes — are used to store programs and data. The microprocessor communicates with the input, output and storage devices through a set of wires (called a "bus") that carry

electronic impulses throughout the computer. Buses can transport one signal at a time in what is referred to as a "serial connection," or they can carry several signals simultaneously in a "parallel connection." Microcomputers can communicate with other computer systems through communications devices, such as modems and networks.

Input Devices

Very simply, input devices are used to enter data into the microcomputer. Common input devices include keyboards, a mouse, light pens, bar code scanners, optical scanners and digitizers.

• *Keyboards*

The most common keyboard is called QWERTY, named for the order in which the letters are arranged in the top row.

The keyboard is the most widely used input device. Most microcomputer keyboards use the key arrangement traditionally found on typewriters, but also feature tilting adjustments, audible key-click, numeric keypads and special "function keys." Many include type-ahead buffering or a "roll-over" feature, which allows you to type as fast as you can — even faster than the keys bounce back — without losing any of the typed characters.

• *Pointing devices*

Pointing devices (such as a mouse or a light pen) are another type of input device. These devices are designed to move the cursor — the indicator that shows where you are on the screen. A computer mouse may be a mechanical piece that is able to roll on your desk; an optical device that uses a beam of light on a special pad; or a hybrid, with a combination of optical and mechanical parts. Unlike a mouse, which moves on a desk or a pad, light pens are moved against the microcomputer's display screen.

• *Scanners*

A third type of input device is the bar code scanner. Bar codes are the patterns of black bars of varied width and spacing that represent numeric or alphanumeric sequences. When a special light pen, called a "bar code reader" or "bar code scanner," is passed over the pattern of bars on an item, the microcomputer translates the pattern of bars into a description of the item that the microcomputer can read as easily as if it were typed into the keyboard. Bar code scanners tend to reduce the human error normal in any data entry operation.

Other, more highly specialized, input devices include optical scanners and digitizing tablets.

Output Devices

Monitors and printers are typical output devices that transform digital data in the microcomputer into a form that is readable by humans or by another machine. For example, for a user to see information on a printed page, the output device — the printer — converts the digital data to characters.

• *Monitors*

The most common output device is the monitor or video display screen on a personal computer (also known as the display, the screen, the Cathode Ray Tube (CRT) and the Video Display Terminal (VDT).

Similar to a television screen, the monitor has a much higher resolution (there are more dots per square inch than on a TV screen), making images sharper and more accurate than a television can produce. All pictures or words displayed on a screen are made up of tiny dots called "pixels." The resolution is determined by how small the pixels are and how many are in a square inch. Resolution is measured in numbers such as "180 x 120," which means that the screen is composed of 180 horizontal columns of pixels and 120 vertical rows. The larger the number, the more pixels you have and the clearer the picture. For example, a picture with a resolution of "320 x 240" is clearer than a picture resolution of "180 x 120." As a general rule, buy the highest resolution color/graphic display you can afford. In addition to resolution, you should consider the screen's color, size, cursor and general capabilities.

• *Printers*

Printers are the devices that produce an output on paper or other permanent media (hard copy). They are capable of producing text characters and graphic output. Printers can be very expensive, and it is important that, as when purchasing any computer component, you first determine your needs. Does your printing need to be of highest quality, or is speed more important? You need to evaluate printers in terms of speed, quality of output and cost.

Printers come in two major varieties: impact and nonimpact. Impact printers are by far the more common type of printer and include dot matrix and wheel printers.

Dot matrix printers have a printhead composed of multiple pins in a rectangular pattern. These pins form characters by pressing a combination of the pins against an inked ribbon in the form of the desired character. The ribbon then transfers the character in

the form of tightly grouped dots on the paper. Dot matrix printers are popular because they are fast, inexpensive, and capable of printing both text and graphics. Some even produce colored output. Although some dot matrix printers can produce very high-quality, or near letter-quality output, the majority produce hard copy that is clearly composed of little dots.

Wheel printers include daisy wheel and thimble printers, both of which function in the same fashion. These printers have a plastic or metal print element, composed of individual arms; a fully formed character is raised on the end of each arm. This character is then pressed against the ribbon by a single metal hammer. The wheel or thimble is rotated into position for each of the characters as needed. Both printers produce letter-quality output, but have limited graphic capability. Wheel printers normally print fewer than 100 characters per second and generally produce only single-colored output.

Nonimpact printers include laser and inkjet printers. Laser printers are generally thought to be the top of the line for producing quality hard-copy output. They operate much like a photocopier, producing copy that approaches typeset quality. Laser printers are fast (six to eight pages per minute on the average) and extremely quiet. Disadvantages of laser printers include the fact that the consumable supplies (toner cartridges and drums) are expensive and the printers cannot handle large volumes of paper. Some laser printers now allow you to produce colored output.

Inkjet printers use a reservoir of magnetic ink, which is drawn out electrostatically and sprayed against the paper. They are usually very slow, but produce high-quality color and graphic output.

Storage Devices

Because information stored in RAM is erased when the power is turned off, and because you cannot change the information in ROM, microcomputer designers have developed various storage devices that allow you to keep programs and data indefinitely and still be able to change them when needed. Storage devices provide a medium for keeping programs and data where they can be accessed quickly without tying up active memory. Today, microcomputers usually come with some minimum amount of mass storage, generally at least one disk drive.

The type and size storage device you will use is one of the most important decisions you will make. No matter how much memory you estimate you will need, it is likely that it will not be enough. People generally find more uses for the computer as they become increasingly comfortable with it and as the computer proves its usefulness. Also, new, improved programs are constantly being introduced. These, generally, because of their advanced capabilities, require more memory.

Storage devices can be categorized into disk storage devices or tape storage devices. Disk storage devices include floppy disks, hard disks and laser optical disk drives. Tape storage devices include magnetic tape drives.

• *Floppy Disks*

The most common storage device associated with microcomputers is the floppy disk, also known as a flexible disk or floppy diskette. Floppy disks are available in three standard sizes: 8 inches, 5-1/4 inches or 3-1/2 inches in diameter. Information can be stored in "single density" or crowded onto the disk in "double density." Floppy disks are inexpensive, removable storage media which are reusable, allowing you to read and write to them many times before they wear out. They are, however, delicate and can be easily ruined if bent, written on, or soiled. Although they are adequate for backup and temporary storage of data, they are inappropriate for permanently archiving data. See **Table 1-2** for common disk storage formats.

Disk Storage Formats

Size	Storage	Characters	Words	Pages
IBM 5 1/4	360 KB	360,000	72,000	250
	1.2 MB	1,200,000	240,000	800
IBM 3 1/2	720 KB	720,000	144,000	480
	1.44 MB	1,440,000	288,000	960
Apple 3 1/2	400 KB	400,000	80,000	266
	800 KB	800,000	160,000	533

Table 1-2: Common Disk Storage Formats

• *Hard Disks*

Hard disks (also known as hard drives, fixed disks, rigid disks and Winchester disks) are very desirable because of their speed and convenience. They function like floppy disks, but hold much more data and are generally not removable. Hard disks are a fixed, sealed unit with a firm recording surface (as opposed to floppy disks, which have a flexible recording surface). The firmness allows more revolutions per minute, enabling the disk to access and transfer data much quicker than a floppy disk. A hard disk allows the computer to automatically load its program information into memory, saving time and allowing the computer to start up almost immediately. A hard disk also enables a user to store many programs and data, giving the user immediate access to the information whenever it is needed.

• *Laser Optical Storage Devices*

Optical disk drives, also known as CD-ROMs (Cartridge Disk Read-Only Memory) and laser disk drives, store data by representing bits in reflected patterns of light on the media. A laser beam makes a permanent microscopic hole in a computer chip; the information cannot be erased. These storage devices offer a huge capacity at a low cost. Optical disks are faster than floppy disks, but slower than hard disks. Optical disks, like floppy disks, can be removed from the computer.

• *Magnetic Tape Drives*

Magnetic tape storage is used to back up disk files for large systems. Magnetic tape can protect your programs and data against accidental loss as a result of power failure, system errors, fatal disk errors, theft or other catastrophes, as well as provide a medium for archiving data. Tape drives make the chore of backing up data from the hard disk much easier than when using floppy disks, but tape drives do not allow fast access to those files for processing.

Communication Devices

Communication devices are the hardware components that allow a microcomputer to communicate with other computer systems and with peripheral devices (such as disk drives, printers, input devices, etc., attached to the machine). These include I/O ports, modems, facsimile add-in cards and networks.

• *Input/Output Ports*

Ports are necessary for external interfacing and are the means for connecting the computer to such I/O devices as modems, external disk drives and digitizers. If you have a sufficient number of ports, then you will not have to keep switching connections. I/O ports can provide either serial or parallel transmission; some computers provide both types.

The terms "parallel" and "serial" refer to the means through which the data are transmitted from one component to another. In a parallel transmission, all of the bits of data are simultaneously transmitted as a group on several parallel wires (one wire per bit plus additional wires for control signals). Only one byte can be transmitted and received at a time. Parallel transmission allows large amounts of data to be transmitted quickly without complicated conversion or error detection techniques; it is usually limited to short distances.

Serial transmissions disassemble the byte into bits, quickly transmit the bits one at a time in a stream (using very few wires), and reassemble the bytes at the receiving end. Serial transmissions can be amplified as necessary when the signal weakens. Computers in two separate geographic locations commonly employ serial communications between each other. Until recently, serial ports were not usually included as standard equipment in microcomputers.

• *Modems*

A modem is a device to connect the computer to a telephone and requires special communication software to operate. The name comes from the tasks it performs — to MODulate and DEModulate. A modem converts digital signals from a microcomputer into audio signals. These audio signals can be transmitted over a telephone line and then reconverted in another, receiving modem attached to a terminal or another computer.

Generally, a modem is external to the microcomputer and looks like a flat metal box with cables. The box is typically attached by one cable to a serial port on a computer. Another cable links the box to a modular telephone jack, and a third cable provides the power to the modem. Some microcomputers have built-in, or internal, modems.

Modems are rated by their *baud* rate — the rate at which data can be transmitted. This rate is usually measured in bits per second. Because there are eight bits to a byte, a modem that transmits at 1200 baud is actually transmitting 150 characters per second (1200 ÷ 8 = 150) or approximately 1500 words per minute.

Most computers and peripheral devices allow the user to set the baud rate by commands or switches, and most modems are capable of multiple speeds. Low-speed modems generally are rated as 300-1200-2400 baud, which means the modem can transmit and receive at any of those three speeds. Medium-speed modems generally are rated as 4800-9600 baud, and high-speed modems as 9600-19,200 baud. The faster the baud rate, the more expensive the modem, but a fast modem may not be as expensive as daily one- or two-hour long distance telephone calls. If you are transmitting a large quantity of data on a regular basis, then you should buy the fastest modem you can afford.

• *Facsimile Cards*

One of the more interesting recent developments has been the facsimile add-in card for the microcomputer. This card gives the microcomputer the capability of sending and receiving handwritten text, pictures, or any other image that can be transmitted by a facsimile device. When the card is used with a high-quality scanner, you are able to have all the functions of a facsimile unit within your microcomputer.

• *Networks: Local Area Networks*

A network is a telecommunications and data communications system that allows a number of independent devices to communicate with each other. Local Area Networks (LANs) are systems for linking inexpensive microcomputers for the purpose of sharing programs, data, and peripheral resources (such as high-capacity hard disks and laser printers) in a limited geographic area, such as a building or office. By sharing software programs and a common database, as well as common resources, a LAN saves the user the tremendous expense of separate peripherals for each microcomputer. Moreover, networks can communicate with other computer systems through electronic mail, electronic bulletin boards and commercial networks.

A server is a microcomputer that supports the network by storing or printing files or providing communication capabilities. A server can be either a specialized microcomputer made by the manufacturer of the LAN hardware or a standard microcomputer running a specialized software application.

Microcomputer LANs generally follow one of two standard protocols or procedures for formatting and transmitting data: Token-Passing and Carrier-Sense Multiple Access/Collision Detection (CSMA/CD). Both LAN protocols have their strengths and weaknesses. In general, a network with a processing load composed of a number of short, intermittent data transmissions

will function with less delay in a token-passing network. Token-passing LANs, however, tend not to be as efficient at passing large amounts of data as the CSMA/CD protocol. CSMA/CD networks generally work best in environments where there are infrequent transfers of large files.

Software

Software is the term for the programs or instructions computers execute. It is composed of electronic signals or magnetically recorded elements which, unlike hardware, you cannot touch. There are a number of different types of software, divided into two standard categories: system software (which includes utility software) and application software. Traditional wisdom directs the potential microcomputer purchaser to select the desired software first, before choosing a system. The application is the reason for purchasing the system, and you must be certain that the computer you choose can handle its assigned task.

Today, this directive is slightly altered by the fact that microcomputers are frequently purchased for their facility in handling general-purpose software such as spreadsheets and database managers. Most microcomputers have such programs available. You should consider buying systems that will run the software of tomorrow, rather than just handling the single application needed today.

System Software

The system software controls the activity of the microprocessor circuitry, the functioning of the peripheral devices, and all software application programs. It includes operating system programs, which function as the "master control," and utility programs, which perform special tasks related to the operation of the system (such as storing the date and time). A good operating system does its job quietly and efficiently, in such a manner that most users never notice it is there.

Operating Systems

Operating systems are the programs that allow the computer to control its resources, execute programs and manage data. The operating system simultaneously executes highly complex tasks, such as tracking information, programs and files stored in memory;

logging user transactions; and compiling statistics on the use of the system's resources.

The operating system largely defines which applications will run on the computer. It is therefore critical that, when you are deciding upon a microcomputer, you consider the type of operating system it is capable of using so you can take advantage of the largest pool of available software.

Microcomputer operating systems can be classified into one of four categories, each of which has different benefits and limitations. The categories refer to the number of people using the system at any one time (single-user or multiuser) and the system's processing capability (single-tasking or multitasking). Operating systems become more complex and require more resources as they develop from single-user, single-tasking into multiuser, multitasking systems.

Sample Operating Systems

Single-user, Single-tasking
MS-DOS
Apple Finder

Single-user, Multitasking
OS/2
Apple MultiFinder

Multiuser, Single-tasking
Most LANs

Multiuser, Multitasking
UNIX

A ***single-user, single-tasking*** operating system is designed to allow a single user on a single system to perform a single task at a time. Single-tasking operating systems are relatively uncomplicated and are easier to write than multitasking operating systems. Single-user, single-tasking operating systems use very little memory and require simple hardware. Such systems are inexpensive, widely available and generally easy to learn. Single-user, single-tasking operating systems are the least complex of the four categories, and, although easy to learn, they generally cannot process large amounts of data as quickly as a more sophisticated system.

A ***single-user, multitasking*** operating system is a powerful system designed to allow a single person using a single system to perform multiple tasks simultaneously. If you are using word processing to compose a report, you can also place the report in a desktop publishing application and include graphics from a third application. This capability of running different programs almost simultaneously gives a single user a great deal of processing power. Generally, the 32-bit processors are the only family of processors capable of supporting such a complex operating system.

Multiuser capability means that the microcomputer can drive several terminals at once. Perhaps the most common example of ***multiuser, single-tasking*** operating systems are LANs. A LAN can connect multiple microcomputers, enabling them to share peripherals, software programs and (frequently) a common database. The LAN operating system controls, or overrides, the operating systems of the connected microcomputers. The micro-

computers still retain their single-user, single-tasking operating system but, when linked to the LAN, the effect is a multiuser, single-tasking system. In addition, the user still retains all of the advantages of a standard single-user, single-tasking operating system if the user chooses to disconnect from the LAN.

There are six popular operating systems, or microcomputer families:
1. Apple-DOS
2. Apple Macintosh
3. CP/M
4. MS/PC-DOS
5. TRS-DOS
6. UNIX

A ***multiuser, multitasking*** operating system for microcomputers allows several persons to share a single microprocessor while performing multiple tasks simultaneously. For example, such a system could simultaneously support one person using the word processor while another person performs inquiries against records in a database and a third person works with a jail management program. The availability of these capabilities places high-end microcomputers in direct competition with minicomputers. In general, multiuser, multitasking operating systems require a minimum of 20MB of hard disk storage and 1MB of RAM, because operating systems allocate separate areas of memory and storage for each user.

Utility Software

Utility software programs are the tools that support the operating system. Manufacturers of operating systems normally include utilities for such things as sorting, printing and editing files. These functions are not necessary to the operating system itself, but make using the computer easier. Utilities provide a standard means of performing generic file operations and a minimum set of tools for any programmer who will be working with the system. An operating system's "richness" or "fullness" is normally judged by the type and number of utilities that are included.

Application Software

Application software programs are the coded instructions that carry out *user* tasks, as opposed to system software, which assists the computer in carrying out *its* tasks. Application software usually will not function without the operating system already loaded and running, because the operating system must perform storage, retrieval and printing functions for the application programs.

Application software can be categorized in terms of its cost and availability (proprietary, public domain and "shareware") and in terms of market size and the generic or specialized nature of the software application (horizontal-market and vertical-market software).

• *Proprietary Software*

Proprietary— commercial—software is owned by an individual or corporation and is always protected by a copyright, making it unlawful for anyone to make copies of the program for others to use. Proprietary software usually includes comprehensive printed documentation, and publishers frequently provide telephone support for purchasers of the software package. The software is updated and improved on a regular basis; those who have already purchased the software are often able to obtain the upgrade at a cost much lower than the full price. Commercial software can be purchased in retail stores or through mail order.

• *Public Domain Software*

Public domain software refers to those packages developed by an author who has relinquished rights to the package. It generally is not protected by a copyright and, because it cannot be licensed, its use by the public is unrestricted. Public domain software is typically available without charge (or for a nominal fee) for use, copying, and general distribution. Documentation may be limited to a text file on the disk, or it may include several manuals. Usually, however, the software is provided without any guarantees, promises of support, or updates. Public domain software can be freely copied and obtained in a variety of ways including from user groups, computer bulletin boards, on-line information services and mail-order distributors.

• *Shareware*

Shareware—user-supported—software is an alternative method for distributing software that falls in between commercial and public domain software. Many of these software programs are professionally developed packages released by small companies with very limited advertising and marketing budgets. They are likely to be generic applications and often compete with products that are distributed commercially. The programs are copyrighted, but users are granted permission to make copies to share with others. The package comes with a request that the user send a specified contribution to the program's author if the user finds the package helpful. These programs usually include documentation in a disk text file and, after a user makes a contribution and registers with the author, the user may receive printed documentation as well. As is the case for public domain software, shareware is distributed through user groups, computer bulletin boards, on-line information services and mail order.

• *Horizontal-Market Software*

Horizontal-market software ranges across disciplines and market groups to serve the needs of a variety of users. For example, a database management package might be used by a police department, an accountant, a dentist, or anyone else interested in storing and retrieving information quickly. Horizontal-market software applications include accounting, word processing, spreadsheets, database management, communications, graphics and desktop publishing. You should be aware that most horizontal software packages cannot be modified; the manufacturer will supply you with only executable code, not the source code (the human-readable and modifiable form of a program). See **Figure 1-2** for samples of horizontal and vertical software.

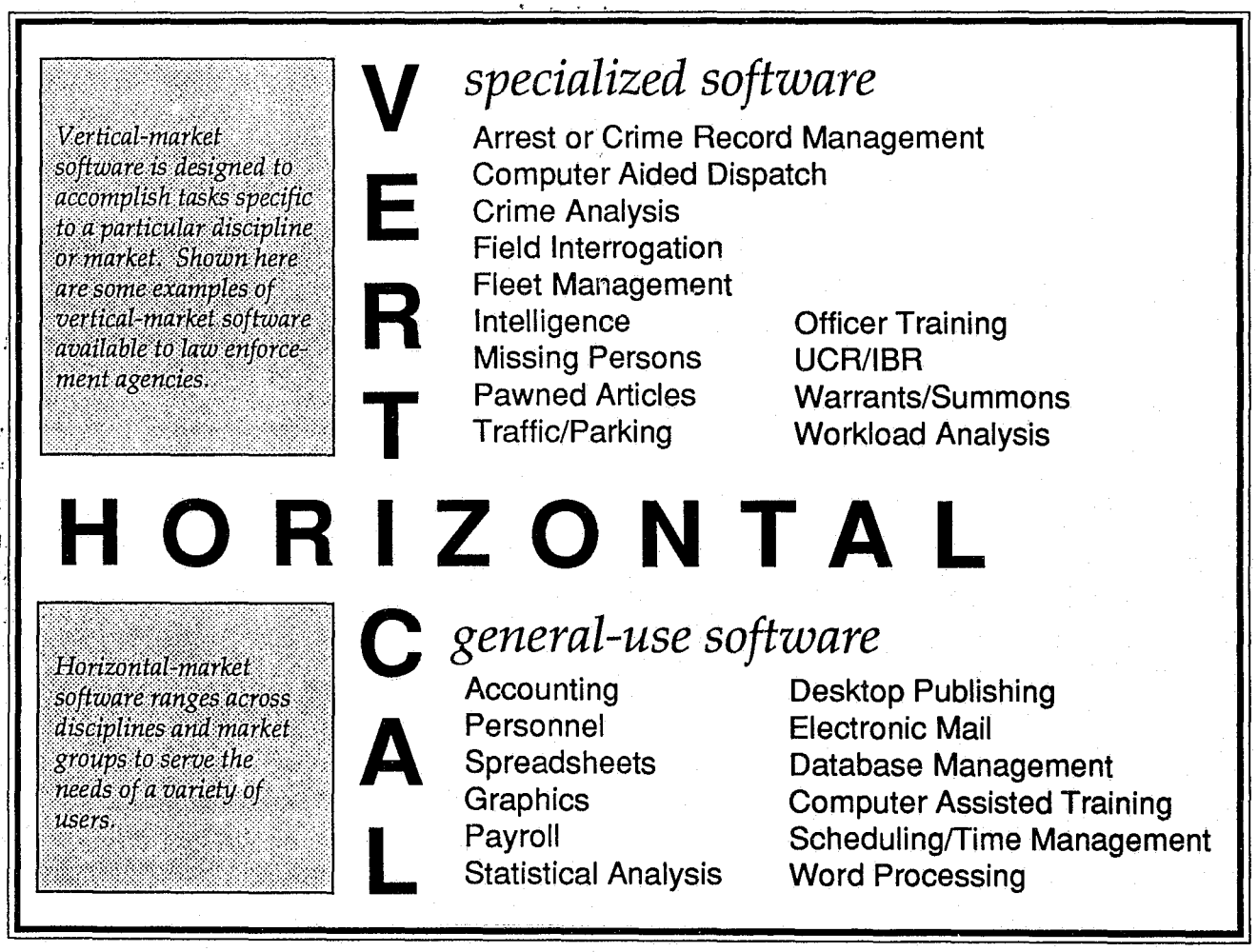

Figure 1-2: Horizontal- and Vertical-Market Software Examples

One of the original uses for computers was automating ***accounting*** systems. There are literally hundreds of applications available for every type of accounting — from accrual to cash-based, from private to public sector, and from not-for-profit corporations to sole proprietorships. Full-scale accounting packages tend to be expensive and perform a relatively limited set of functions. Because accounting software performs a particularly vital task, it requires special consideration in the areas of security and file sharing.

The advantage of a ***word processing*** program over a typewriter is that you do not have to completely retype a document when only small changes are required. A word processor allows the user to create, edit, revise and format text temporarily, holding the changes in the memory of the microcomputer until printed. Nearly unlimited drafts and revisions can be generated before the final product is printed. "Boilerplating" enables the user to create a prototype document as a source that can be modified rather than recreated each time a new document is needed. As state-of-the-art word processors become more sophisticated, additional capabilities such as spell-checking, thesaurus information, merging, headers and footers, indexing, automated tables of contents and a host of other features are available. Word processing is extremely helpful in generating daily reports, narratives or any standard format document.

A ***spreadsheet*** application is a management tool that can be used to organize and calculate statistical and financial data. It can be thought of as an electronic worksheet, with multiple columns and rows similar to those in an accounting ledger. Either alphanumeric or numeric data can be entered into the cells of the worksheet. The spreadsheet program on a microcomputer can automatically recalculate all the figures on the sheet when any number in the matrix is changed. This characteristic not only increases the quantity of work that can be done (reducing the amount of time necessary to make the calculations), but also improves the quality of the work, because projections are more refined. Complex mathematical and statistical functions can be performed on the numeric data, and it is advisable that the spreadsheet package selected include graphics capabilities. This way, you can easily create common business graphics such as pie, line and bar charts to visually display numeric data. A spreadsheet program is a valuable tool for manipulating empirical data to make "what-if" predictions; as you vary assumptions and calculations, you can measure the effect on the "bottom line."

Law enforcement agencies frequently use spreadsheets for crime analysis and workload distribution. Spreadsheets are also helpful in tabulating statistics, calculating overtime hours, preparing accounting or budget information, or performing nearly any activity that has some associated mathematical calculation.

Database management software systematically stores and retrieves large amounts of information in a structured fashion and has an organized method of inquiring about relationships in the data. Database management programs include, for example, simple file management-report generator databases, relational databases, and hierarchical databases. The type of program determines the type of problem the program can solve. Database management systems such as DataFlex™ and dBase III Plus™ allow you to create programs for your specific needs.

Database Types

1. File management-report generator
2. Relational
3. Hierarchical
4. Network
5. Free format
6. Multiuser
7. Computer languages

Database management systems are like filing cabinets: they store information according to the categories you create; but they sort and retrieve the information almost instantaneously. A relational database allows you to search and input across multiple files. The power of a relational database management program is not only that it stores and retrieves large volumes of data, but that it also allows you to examine relationships in the data. These relationships can provide you with new information. In crime analysis, for example, a comprehensive relational database of information about burglaries might enable a detective to detect a pattern and predict the next likely location.

Database management systems are useful if you have a frequent need for reports calling on information scattered throughout your entire system and if much of the system's data is accessed for different purposes by several different departments (thus reducing duplication).

Communications software applications let microcomputers share data and information with remote systems. Communications software affects the configuration of the entire microcomputer installation and involves both hardware and software. Most agencies now use some form of data communications system to allow them to communicate with systems such as the National Crime Information Center (NCIC), National Law Enforcement Telecommunications Systems (NLETS), or state motor vehicle systems. Microcomputers are capable of communicating with nearly any other computer available. The communications software can be one of two types. In a "terminal emulation" application, the software allows the microcomputer to imitate a different type of system and function as a workstation on the remote system. In a "file transfer system," data files are copied between the two separate systems.

Graphics application software allows data to be represented in a graphic format. A chart or graph is often the quickest and easiest means to clarify the analysis of statistical data. Graphic design software, such as computer-aided drafting or computerized "painting" programs, is also available for microcomputers and can be used to create designs of a crime or accident scene, rather than trying to draw them by hand. Because the graphics files are stored on the system, a case file can be stored with both the text and the graphic data.

Desktop publishing programs provide users with typesetting and design tools, allowing for a great deal of creativity in producing reports, newsletters, forms, overheads and instructional material. These programs can usually be integrated with other software, such as word processing, graphics and specific criminal justice applications; you can generally transfer data produced in another application into a desktop publishing system to create a final document. This convenience eliminates the necessity of preparing a document in word processing and then having it typeset by a commercial vendor. Previously, only the largest agencies could afford typesetting and sophisticated design services; with desktop publishing programs, agencies have in-house control of typesetting and design functions and can save both money and production time. Relatively inexpensive desktop publishing programs can greatly enhance the quality and range of communications generated within an agency.

• *Vertical-Market Software*

Vertical-market software is highly specialized; it is designed to accomplish tasks specific to a particular discipline or market. Criminal justice, medicine and law, for example, are considered vertical markets. Vendors with an understanding of the particular software requirements of these markets have created customized programs to fill certain criminal justice, medical and legal needs. A computer-aided dispatch application, for example, may have some features (such as word processing) used in a variety of other software packages, but the primary features have been developed to meet the unique needs of a law enforcement dispatch center. Such a product would not be appropriate for any other discipline.

Where To Go for Additional Information

Computer Courses

Many local colleges and universities, as well as manufacturers and computer stores, conduct introductory microcomputer courses. SEARCH provides a variety of introductory and advanced courses at its National Criminal Justice Computer Laboratory and Training Center in Sacramento, Calif.

A half-day or one-day course should familiarize you with the basic benefits and costs of using microcomputers, and a two-to-three-day course should provide more detailed instruction and hands-on demonstrations of various packages and systems. When considering a course or seminar, be aware of any biases: Courses put on by computer vendors will be less objective and may be little more than sales presentations for their particular systems.

Magazines and Other Publications

Almost any bookstore or newsstand now carries a variety of microcomputer magazines that can be quite useful. Not only are the articles highly informative, but the many advertisements can keep you up-to-date on the latest in microcomputing. Many business-related magazines and an increasing number of law enforcement journals also include articles on different aspects of microcomputers. These articles are usually less technical than the microcomputer magazines, because the articles are written with the nontechnical person in mind. Book review sections in various magazines can also steer you to some excellent current publications.

Computer User Groups

Check your local newspaper or computer store for notices announcing the time and location of computer user group meetings. These meetings are often informal and participants may include both hobbyists and business users. Meetings may be organized around particular types of microcomputers or may be very general in nature. Many groups have established libraries of public domain software and shareware for members to copy. You can also learn about the advantages and disadvantages of a particular program from others who have worked with the program. Whatever the focus, a user group meeting is an opportunity for users to share problems and solutions; trade software; and discuss their machines, software and suppliers.

Consultants

Any management and computer consulting firm will be able to assist you in learning about computer systems, but these resources are often quite expensive. As an alternative, you may want to consider your agency's auditors as a source of information. They are often microcomputer users themselves and have encountered other users whose names they may be willing to pass on to you.

2

Preliminary Project Planning

Before you undertake the acquisition process, it is important to recognize that, while improving productivity, the introduction of microcomputers into your agency will change traditional office routines and affect information management. Acquiring a system is not a simple project, and it has many lasting implications and ramifications. The process will require careful analysis and study before a final decision is made, and it will take a fair amount of time and effort — probably more than you anticipate. For all of these reasons, it should be apparent that careful planning and preparation prior to acquisition is critical to successful implementation. By carefully planning the acquisition, you can take advantage of many opportunities to use information technology to support your agency's programs effectively.

The planning methodology will depend on the size, resources, and style of your agency. The process can be as expansive and formal as you want or need, or it can be a limited and informal, but still effective, approach. The steps outlined here are meant to guide you through the process and you should tailor them to suit your needs.

Is There a Problem?

It may be useful (and enlightening) for you to take some time to generally review your agency's operations. Changes in an organization are usually gradual; in the hustle and bustle of the daily routine, these changes often go unnoticed. The cumulative effect on your productivity and effectiveness can be monumental. You need to step back and consider the following questions.

- Has your workload grown significantly in the past year?
- Have your responsibilities increased?
- Is the agency behind in paperwork because volume has exceeded the agency's capability to process it?
- Have errors and omissions increased in reports or routine correspondence?

- Are additional space requirements being projected?
- Is vital information missing or lacking when important decisions need to be made?
- Is the work environment such that there is not enough time to analyze problems — only time to react and apply quick fixes?

Affirmative answers to these questions may indicate a need to analyze your agency's operations more closely to determine if more efficient and effective procedures, tighter controls or additional personnel are needed — or whether a computer is part of the solution.

Planning Mechanisms

Acquiring a computer system is not a one-person task; you will need to establish some kind of formal planning structure. You might appoint a steering committee or task force, composed of a variety of people, to be officially responsible for planning and managing the acquisition process. Depending on the size of your agency, the task force may be only a few people or representatives from several departments. You may also wish to create a process for long-term planning. These two mechanisms will make the project manageable, provide guidance, centralize policy decisions and lend accountability to the acquisition process.

Assign a Project Manager

The Project Manager is responsible for directing the course of the project and coordinating the work of all participants. The manager must also ensure that the new system meets the actual needs of the agency. The Project Manager should be someone who can provide the leadership necessary to guide a very complex project to completion. The manager's support, direction and personal commitment are critical to the success of a new system. For a system to be effective, the manager must first recognize the need for a new or improved system and then be willing and able to follow through with the implementation process.

The Project Manager must also have direct access to, and the support of, top management personnel in order to keep them informed and involved in the process and to solicit their input in the overall strategy.

Create a Project Team

The introduction of a microcomputer into the office will affect virtually all staff. By involving key personnel in the planning process, the Project Manager can generate cooperation and an investment on the part of the users. The Project Team can be extremely helpful in collecting basic background information and preparing for the in-depth system analysis that follows. Members of the team should be involved in the process of identifying needs and problems as well as potential solutions.

A team that represents various office perspectives increases the likelihood that the process will be comprehensive and credible. (If possible, individuals should be from different units or departments within the agency, and from executive, programmatic, and technical levels.) Individuals do not need to have a technical background; they are providing information from the end user's point of view. It is beneficial, however, to include some personnel with computer expertise.

An initial step in creating the Project Team may be to draw up a list of key persons from each area (include each individual's field of expertise), who will be helpful in analyzing the current system. They should be familiar with the organizational tasks involved and be able to contribute experience and time to the analysis. You may wish to use all of these people throughout the entire process, or establish a core working group that can call upon the expertise of specific individuals at crucial points. For a sample form to use in compiling a key person list, see **Figure 2-1**.

NAME	POSITION	SUPERVISORY	EXPERIENCE	COMPUTER EXPERIENCE	EXPERTISE	COMMENTS
Max Buck	Finance Director	Yes	20 Years	Spreadsheet	Administrative	
Bruce Hall	Records Supervisor	Yes	4 Years	Database Management	Reports Organizational skills	
Sue White	Secretary	No	7 Years	Word Processing	Inter-office Communication	
Bill Long	Training Manager	Yes	2 years	Programming	Strong in micros, teaching	
Jane Francis	Personnel	No	5 Years	Accounting	Budgetary	

Figure 2-1: Key Person Sheet

To facilitate the work of the Project Team, create a charter that specifies the responsibilities of each team member, the commitment of resources, and a work schedule that outlines the team's projected activities.

Do You Need a Computer?

The purpose of introducing a computer into your agency should be to improve productivity while working to reach the agency's goals. Thus, if you are responsible for buying a new or enhanced system, you will first need to articulate the agency's mission statement, goals and objectives. You will then want to examine your agency's operations and the requirements for information to see how productivity can be improved. The reason for this exercise is to separate those functions that would benefit from automation from those that are operating efficiently and should perhaps be kept as manual operations. Remember, a computer cannot guarantee improvement if its capabilities are not well managed and applied to appropriate tasks.

Write a Mission Statement

The acquisition of a microcomputer should be planned with due regard to your agency's long- and short-term goals, not solely as a reaction to an existing problem. The microcomputer should, ideally, be an integral part of your organization's expansion plans over the course of several years.

Knowing exactly the purpose and goals of your agency will help you select the system best designed to meet those goals. If you know where your agency is headed in the next few years, it will be much easier to determine which piece of equipment is the best to help you get there. In addition, this information will keep you from being distracted by vendors who may want to sell you a different system, one that has a lot of "bells and whistles," but not the basic components you need to improve the efficiency and productivity of your agency.

• *Agency's Purpose*

You may want to discuss, in broad terms, the agency's purpose and the possibility that that purpose may change within the next five years. What direction is the agency headed during the next five years, and how will it meet its goals? How will shifting demographics, drugs, gangs and the AIDS epidemic affect your daily operations? What federal, state and local laws and regulations govern your agency, and what are the agency's obligations

and responsibilities? This information may be critical to the type of data you wish to maintain and analyze in a new computer system and critical to the creation of new agency guidelines. Discuss any pending changes in the social, political, economic and criminal justice environments that may affect how your agency functions and how the agency is prepared to respond to those changes.

It is useful to review the statutes that define your mission or purpose, as well as internal documents, such as the agency's strategic plan, its policies, program plans and budget. Ask executive managers to meet with the Project Team to discuss the agency's strategic plan and long-term objectives.

• *Key Programs*

As part of the Mission Statement, describe your agency's key programs and how these programs relate to the agency's mission. It is important to identify the key players and the role they play in each of these programs. The agency may have established a neighborhood crime watch program, for example. In addition to participants in the neighborhood and agency personnel responsible for coordinating the program, the city council, the local chamber of commerce, the media, and school personnel may have a vested interest in the program. You may find valuable information in statements of program goals and objectives prepared by program staff and in program budgets.

Measure Productivity

The primary benefit an agency gains from a microcomputer is increased productivity. Two common ways to improve productivity are (1) to reduce input resources without reducing output results and (2) to increase output results without increasing input resources. For example, in the first instance, a task that was previously accomplished manually can be accomplished faster with a microcomputer. Or, as in the second situation, there may be a new way to accomplish a task that, when done using a microcomputer, the product is of higher quality than was possible with manual tools.

In a small agency, productivity goals may be set by the manager, based on interviews with key personnel and the manager's judgment. A large agency's performance or production goals may come from organizational directives. Your agency may be seeking a system that will enhance productivity and reduce costs; system requirements may include a reduction in filing, photo-

copying and typing revisions, or a methodology to allow additional information to be recorded. Regardless of how they are set, productivity goals need to be established to compare the *actual* production of output to the agency's *expected* performance standards.

Of course, on close inspection, you may find that a task can be eliminated. You want technology to reorganize your operations to improve productivity. You do not want to spend a lot of money on a computer system that only codifies and automates old, inefficient habits.

• *Review the Current System*

To measure an agency's or department's productivity, you must examine the current system, identifying the flow of information and the inputs and outputs of major process areas. This initial overview of the system will supply general background information. In a needs assessment (discussed in the next section) you will conduct a more detailed analysis of the current system.

The Project Team, in conjunction with others on the key persons list, should interview personnel about the current system and observe how the process works. You may want to devise an interview guide so that everyone is asked the same questions (if applicable) and interviewers do not forget to ask for relevant information. You are looking for the following type of information.

- Where data enter the system, where and how the data are modified and the resulting outputs;
- A description of the output generated at each step of the process;
- A description of input preparation activities — the procedures that must be performed before data can be entered;
- A description of input procedures — the steps that must be followed to enter data;
- The annual volume of each output;
- Estimates as to the amount of time that could be saved if certain manual tasks were computerized;
- Recommendations for or against computerizing a specific output, area or process;
- Productivity factors affecting output such as timeliness, responsiveness, convenience and appearance; and
- Projected organizational changes, including workload changes.

If you are are thinking of automating your entire department or agency, your Project Team must consider all aspects of this ultimate system during the early planning and investigation stages. Although the "big picture" is reviewed in the initial stages, it is a much more manageable project if only portions of that ultimate system are acquired and implemented at any one time. If, for example, you are thinking about automating your accounting system and desire word processing and records management capabilities, you may want to start with the smallest and simplest of programs and, once you have those up and running smoothly, implement the next stage.

The results of your preliminary research should give you a general idea of which major functional or process areas (such as accounting, personnel, and records management) in the current system could benefit from automation. Your next step is to determine the amount of time, funding and expertise available to help you first assess your information and automation needs in detail, and then acquire and implement an automated system that will meet those needs.

Reducing Computer Anxiety

People are naturally resistant to change, and automating an agency can require major adjustments on the part of all staff members. There are, however, several ways a manager can reduce anxiety.

User Participation

To help ensure a successful acquisition, involve the end users — the people who will be working on the computer daily — in as many phases of the process as possible. People become committed to a project when they are asked for an opinion and encouraged to participate in the process.

Through interviews, find out users' specific needs and what bottlenecks they see. Ask their opinions about possible solutions and what components of their duties they believe would benefit from computerization. Listen closely — if you have not worked on the front line in a few years, you may be surprised by the answers you receive.

Communication

After soliciting input, it is critical that staff receive substantive feedback. Expectations have been raised; to maintain credibility, you need to explain why particular ideas and suggestions have or have not been implemented.

An acquisition project may extend over the course of a year: As the project develops, users should be regularly informed of its status. Monthly updates provide an opportunity for management to report on the project and to answer staff questions, thus demystifying the process.

People may feel their job security threatened by computers and may fear disruptions in agency routine. Make certain everyone understands why a computer is being purchased and how personnel will benefit. Assure them that automation will assume tedious, repetitive tasks, freeing individuals to do more creative and productive work.

3

Conducting a Needs Assessment

From your initial work in the preliminary project planning stage, you have determined which operational area or areas require more study. For example, after the Project Team reviews the entire system, the team may decide that the method currently used for handling records is deficient and, along with the dispatch and clerical functions, could benefit from automation. The Project Team will then take a much closer look at these functions.

You now must construct a *needs assessment*, which will help you identify your agency's information needs in one or more specific areas (such as records management, dispatch, and clerical tasks). This analysis will help determine if information technology will solve the problem identified and, if so, in what way. For example, the Project Team may find that offense and arrest reports are not being kept up-to-date and that a computer system can help solve this problem by automatically requiring a review of all arrest reports more than 60 days old. In a needs assessment, the Project Team conducts a detailed appraisal of the existing system (manual or automated) and determines what information it provides and what information needs are not being met. The information compiled in this stage helps you determine an effective response to the problem and forms the foundation for an enhanced or new system. Thus, it is important to anticipate future usage and needs in as much detail as possible.

Define the Problem

What do you want the new system to do that your current system does not do or does not do well? To answer this question, the Project Team will collaborate on writing a concise problem statement describing any general problems with the existing system. A problem may relate to the need to provide new services mandated by law or to provide existing services more efficiently. Other problems may coincide with the need to avoid budget increases or to reduce the costs of agency operations.

Perhaps your agency is having problems with misfiled, unnumbered, or lost reports. Or maybe your crime reporting forms are insufficient to meet the department's operational and management information needs. Is there an excessive amount of report reproduction? Are reports always backlogged? Are these bottlenecks directly related to other problems that should be investigated?

There may be several areas you think could benefit from automation and it is important to set priorities. Do you need a Computer-Aided Dispatch (CAD) system, or would your money be better spent on automating functions such as producing incidence reports by time of day and by beat, or by creating an alias file to be linked with a file of active warrants?

At this stage, you do not need to identify a solution or determine whether your agency needs a database management program or a more efficient method of reporting. You need only to describe the problem and how it is affecting your operations. Isolating the system's weaknesses is helpful later, when it is necessary to specify improvements the agency requires from a proposed system.

Again, depending on the size of your agency and the scope of the system you are investigating, the problem statement may be a paragraph, a page, or several pages.

Examine the Existing System

Having identified the perceived problems or difficulties with the agency's existing system, the Project Team eventually will be challenged with finding a solution to those problems. To devise solutions, however, the Project Team must first examine the existing system to thoroughly understand all its aspects. Analyzing your current operation may reveal ways to improve the system without automation, or it may indicate that automation is unnecessary or impractical. A precise picture of your agency's current methods, forms, work flows, and volumes is the basis for defining your needs and designing the new system. Understanding the current system also means understanding its costs. This knowledge is important for justifying proposed changes. As you go through the following steps, you need to keep notes that will help you prepare the final documentation.

Identify the Purpose of the Existing System

The first step in the process of understanding your current system is to learn the original purpose or purposes of the system. The

simple answer is usually "to provide information." What the Project Team needs to identify is the *purpose* of that information: Why is there a need for that system within your organization? Is the information necessary to comply with federal or state statutes, or is certain information being gathered solely because of "tradition"? If you determine that the purpose of the system is critical to the functions of the agency, then you will want to continue to investigate, seeking ways to streamline it for automation.

Interview Staff

The Project Team must begin by talking with those persons who work with the system daily. It is important now to call upon the key people identified in the preliminary planning stage for their input and expertise. These interviews need to be much more specific than those conducted previously and will serve to define the existing system's functions and to determine how information flows through the system. What information is provided and for whom? What kinds of input and output are used? What bottlenecks exist? Remember, before you can change anything, you must break the system down into separate tasks that can be analyzed and appraised.

Be careful not to assume that you are fully conversant with a system function if you are not involved with it on a day-to-day basis. You may have been very familiar with a system some years ago, but your information is probably dated, because systems change substantially over time.

Define the Information Flow

Using your key people from each area, examine the flow of information. Where does information enter the existing system, where and how is it modified as it travels through the system, and what comes out of the system? Try to define the information flow as a group of separate, interacting tasks or functions. Create a simple flowchart of the process by walking through the process. That is, have one or more of the key operational personnel *physically* walk you through the path of the information, following forms and documents as they move through the system. Identify every place the form or the flow of information stops to be manipulated in any way — from the point of entry into your agency, through every person and place the information travels, to the final destination of the information.

Define the Output

For each task, begin by identifying the output generated in each step of the process. Seeing each "result" helps you understand what your agency is trying to produce. When defining outputs, do not assume that every piece of information in your current system should become part of the computerized system. As systems mature and the information needs of administration change, information that was once useful may no longer be relevant.

Automating a manual process provides an excellent opportunity to rid your system of useless data and to streamline the data collection process. Gather samples of the physical documents generated, and review them with the people who receive them; note any unnecessary items. Once you have a good idea of the output requirements of the system, you will gain insight into the input requirements.

Define the Input

For each task, identify where and how information enters your system. How many people look at or modify the documentation created by a particular task? One? Five? Fifteen? Do they use the information as it exists, or do they add to it? One of the problems in analyzing a system is finding where and how information is modified or changed. If different people add to it, find the source of the new information. If they use the information, track where the information goes next. As you did in the output section, get samples of the forms and documents used to enter or modify the information. A comparison of the information on the input forms with the information on the output forms will clarify the ways in which the information is modified, manipulated, or changed.

Look for Bottlenecks

Eventually, you will begin to see several "streams" of information flowing through the system. Once you have a feel for where the information flows, look for any obvious bottlenecks or redundancies. For example, does a form normally spend a day or two in someone's in-box before being processed? Can this delay be avoided?

Carefully examine the purpose of any "blind" carbon copies in the system, and watch for duplicated forms or documents that have no real purpose other than tradition. Are recipients simply

filing or perhaps destroying their copies? Duplicate forms are an indication that the flow of information is not following a single path through the agency. Find out if there is a good, useful reason for this duplication. Remember, you do not want to automate a flawed manual system. At best, it will only do a bad job faster; at worst, the computer system will be rendered ineffective.

Documentation

Document the Outputs, Processes and Inputs

As you and your team determine the information flow, you should construct a matrix that identifies the outputs, processes, and inputs associated with a particular task. Arrange in rows and columns all the identifiable outputs; for each output, note the associated inputs and processes required to generate that output. For example, your outputs may include offense reports, arrest reports, and accident reports. A simplified example of a chart listing the inputs and processes for an agency's offense report can be seen in **Figure 3-1.**

Function Area: **Records**

Key Person: **Bruce Hall**

Output	Process	Inputs
Offense Report	Collect Data on Crime/Incident Assign Case Number Code Code and Index by • Name • Chronology • Type of Crime • Location	Dispatch Incident Report Complaint Receptionist Information Records Section Review

Figure 3-1: Sample Offense Report Chart

Create Flowcharts

Flowcharts are an easy and helpful way to condense your lists of data into a pictorial view of the information flow. For each output, create a simple flowchart to illustrate the sequence of operations involved in its production. Be sure that each process, no matter how small, is identified. Walk through the tasks, reviewing each chart to be certain all operations have been identified. When you are done, you will have a series of flowcharts identifying all information and paper movement activities in your agency; you will probably be surprised by the complexity of your operations!

Depending on the size of your agency and the amount of time and money available for this project, you may want to add notes to each of the charts to indicate the number of times a task is performed, the time required to complete the task, or the cost of preparation. The point here is to produce a chart with enough relevant information to assist you in understanding the current system as much as possible.

You will need to diagram each output. **Figure 3-2** is a simple example of how the information flow for an offense report might look. Information from the offense report may be used later in other reports, such as a master name list of arrestees, an officer activity report, and accident location reports.

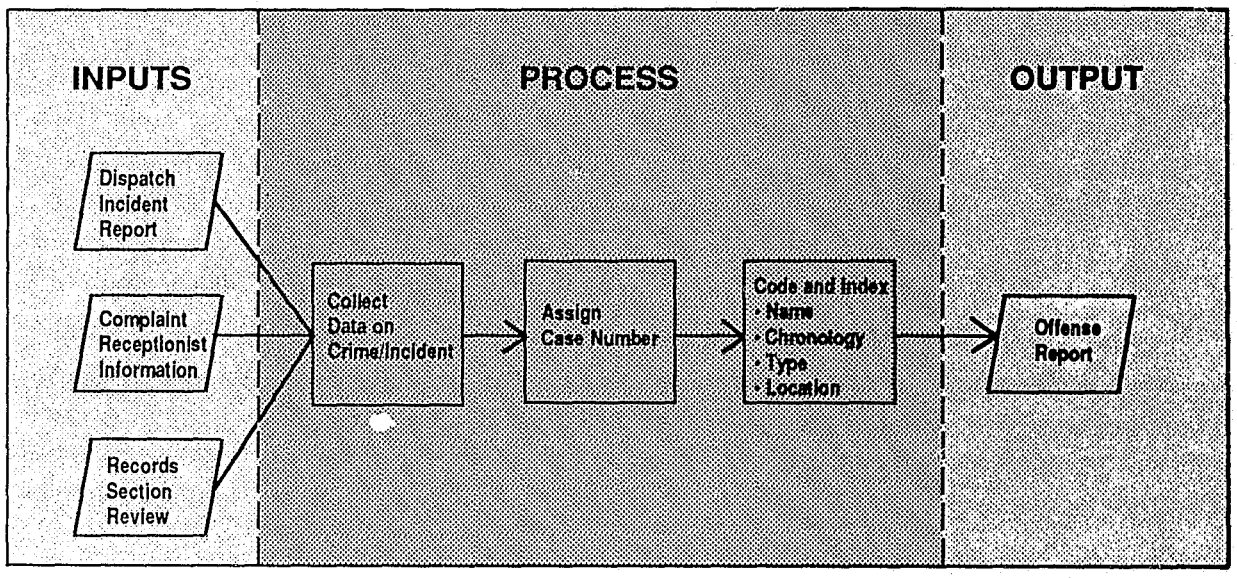

Figure 3-2: Offense Report Information Flow

Consolidate and Evaluate Information

As Project Manager, you will evaluate the information gathered up to this point to identify work blockages, unnecessary delays, redundant outputs, or other factors that decrease the efficiency of the current system. Examine each inefficiently produced output to determine what refinements and changes are needed to provide the optimal process of information management in your agency. When you have determined the optimal process, prepare a new flowchart of inputs, processes, and outputs to reflect the changes and record the increase in productivity you expect as a result of these changes. You may discover that immediate manual changes can rectify the inefficiency. Any changes that are made in the current system should be noted in the flowcharts so that, if you decide to continue with the acquisition process, your baseline is a true reflection of that system. These new flowcharts of your agency's outputs will be used to establish the user requirements of your enhanced or new system (addressed in the next chapter).

Draft the Needs Assessment Statement

All of the activities conducted and the information collected during the Needs Assessment should be well documented for future reference. From your notes, write an overall description of the existing system. This narrative, combined with the problem statement written at the beginning of the process and the flowcharts and matrix, will provide you and the Project Team with a clear picture of where the agency stands in terms of automation. These materials will help your agency determine whether to proceed with the acquisition process. If so, the Needs Assessment statement provides a foundation on which to build the new or enhanced system.

Know Your Facts

After working closely with key personnel in each area and having walked through the entire process, the Project Team should have factual answers (as opposed to "best guess" estimates) to the following types of questions concerning the current automated system.

- What is the current system's general configuration?
- What are the computing system's hardware resources (type, characteristics, location, age, etc.)?

- What are the computing system's software resources (type, function, age, usage, development method — commercial, in-house, public domain)?
- What are the computing system's personnel resources (title, function, positions filled, positions vacant, positions needed, budget source)?
- What are the computing system's data communications resources (type, characteristics, ports, lease/maintenance costs, etc.)?
- Who are the system users, and how much of the resources do they use?

4

Defining the User Requirements

The *User Requirements* statement will define two of your agency objectives.

- What the users require the proposed system to do
- What capabilities they wish the proposed system to have

The statement directly relates to the problem statement you wrote as you started the Needs Assessment: at this point, however, you are ready to make some preliminary decisions about what should be included in the proposed system. After examining the current system (including its weaknesses), the Project Team must list the improvements required in the new system. What does the system need to be able to do? How will it do it? Why are these functions justified? List those improvements that are both critical — those of key importance to your agency that *must* be computerized — and any enhancements that may be only desirable.

The User Requirements statement may be a lengthy document or it may only be a few pages, but it should be written and circulated so that all people involved in the system acquisition process have a common reference document clearly stating the project's original objectives. In the event that objectives seem to become blurred during the course of the project, this document can help keep participants on track. The user's problems and difficulties with the current system — and expectations for the proposed system — must be clearly understood by all personnel involved if the best system for the agency is to be purchased.

Writing the User Requirements

You have interviewed key personnel and users and have an understanding of the existing system and what the users need and want in a new system. Now you need to write that information down concisely in as simple of fashion as possible.

Prepare an Overview

The User Requirements statement should cite your objectives and provide an overview of the functions and capabilities you desire in a new system. In clear, descriptive, nontechnical language, begin by stating the purpose and primary objectives of the proposed system. List any problems that you want to overcome and improvements that you want to make in the existing manual or automated system. Describe how the proposed system addresses these problems. Consider how the proposed system will benefit your agency or the community it serves in terms of personnel savings, improved efficiency, ability to handle more work, or reduced costs. Remember, automating your data processing capabilities will enhance your officers' ability to work more effectively on the street, thereby providing better service to the public.

You will next need to identify the functional, performance, compatibility, and security requirements for the proposed system, including determining the file content and volume of activity, file size, and the types of reports required in the proposed system.

List General Objectives

Your objectives for the system will define the results that a proposed system must achieve for it to solve your problem or problems. The Project Team must set objectives that are measurable, achievable and complete; and the team should decide whether the final system should be primarily concerned with costs, agency operations, or both. If costs are a primary consideration, do you expect a new system to reduce your program costs, avoid additional costs, or increase revenue (such as being able to effectively collect ticket and license fees)? If improved agency operations and services are the compelling reasons for the system acquisition, consider any proposed system in terms of enhancing the comprehensiveness, timeliness and quality of service and products.

Define the Content of Files

From your work in defining inputs and outputs in the Needs Assessment, you should have a clear idea of the kinds of information your present system processes. This knowledge is necessary to determine if the files of the automated system will be able to accommodate the kinds of information you need to process.

What data will be entered into the computer? Will the new system be used for offense reports, warrants, name/vehicle searches or jail logs?

At this time, you should also determine the type and quality of information you intend to store in the new computer system and establish a reasonable purge schedule and criteria. Such a schedule will prevent the accumulation of unnecessary, useless or dated information in the system.

Define the Volume

In addition to the *kinds* of information to be processed, file definition includes the *amount* of information that you will store in the computer system. You will not know the exact amount of storage required by your system, as storage is a function of the software and operating system you will eventually acquire. At this time, however, you will need to estimate the average number of transactions, bookings, or actions that your system handles daily, monthly and annually. When are your peak times, and what time constraints exist? How quickly do you need an answer from a computer inquiry — what is the required response time? At what rate will the microcomputer be handling work? Realistically, you should realize that when a new system is introduced into an agency, initial utilization may be very low; it will take people time, perhaps 18 months, to grow from the learning stage to more advanced stages of using a microcomputer. After the first year, utilization of the system may increase by 50 percent.

Define the Size of Files

Whereas the volume of your transactions will not affect the cost of your system, the size of your files of information surely will. In estimating the size of your files, the simple objective is to find out how much space your filing cabinet full of paper reports and correspondence will occupy on the computer's storage devices. Storage space requirements are less dependent on applications software than on the amount of information you want to have at your fingertips at one time. What information do you want to access periodically, and what information do you want to have immediate access to on-line (for example, shift reports, master name list, incident reporting and tracking)?

The storage in an automated system is similar to a manual storage system in the sense that, just as a file folder in a drawer may contain several different reports or separate pieces of paper, a computer system will help you organize data into different files

(or folders). Within these files may be many different records. And, as is the case of a single printed page or report, the automated record is made up of characters, called bytes. The first step in estimating your storage requirements is to add up the number of characters required in each record of each file you will be using. Then, multiply that by the number of records you need. The result is the number of "bytes" of storage you need for that file. Note the following.

If an incident file has	1,000 records
and an average record length of	300 characters,
then it will require	300,000 bytes of storage

If you want to have several other files simultaneously on-line, then you will need to calculate the requirements for each and add them together.

Determine Additional Storage Needs

In calculating your data processing needs, you should also allow for growth. Any system will probably require modifications and expansion after installation, but it is best to plan for the future as much as possible. Growth can be estimated by examining your workload statistics for the past several years and projecting the minimum and maximum annual growth of your work for the next five years. Remember too, that in addition to the amount of storage you need for data, the operating system and the software programs you choose will require storage space. Generally speaking, purchase as much storage as you can afford.

Define Reports

It is necessary to define the scope and content, as well as the layout, of the reports you want the microcomputer to produce. This exercise may indicate the need to modify current reports, or you may decide that you want the proposed system to be able to modify reports over time and to generate ad hoc reports. It is important to recognize the system modification needs now, for this requirement may well disqualify many of the available package systems. Do not assume that packages can be easily changed to suit your specific requirements; often, an adjustment appearing effortless to the purchaser may be very difficult to incorporate.

Define the Security Required

You must also assess the security level required for the proposed system, evaluate the sensitivity of the information you are handling, and protect it to the same degree you would paper files. Are there any special risks associated with maintaining the integrity, confidentiality, and security of the data? What provisions are needed for disaster recovery? If the information is governed by federal or state security and privacy provisions, be certain you understand those provisions and that your security plans comply. Most multiuser operating systems have some type of security built into the software, such as password protection for programs and data. You might also consider purchasing removable disks that can be locked up separately from the system, or systems that require keys to be activated. When assessing security needs, consider the practicality of the plan, and guard against unnecessary precautions that only cause users to ignore or circumvent the security procedures.

Consider Compatibility

It is essential that the Project Team determine how proposed software and equipment must interact with existing or planned agency software or equipment. Do your agency's systems need to interact with software programs or equipment from an outside agency? What restrictions does that place on your acquisition process?

Determine Key Performance Factors

It is very likely that you will not be able to find a system that exactly meets your needs. In that case, you may be faced with the difficult task of discarding certain requirements or deciding to develop your own system. You must be able to set priorities and decide which of the many user requirements are of key importance: What functions are vital to your agency, and which can you give up without hindering the agency's effectiveness? Examples of ways to enhance productivity and perhaps reduce costs follow.

- Reduce filing time, photocopying and typing revision cycles
- Increase accuracy and timeliness of production
- Present information more clearly
- Record additional information

For each output that the Project Team identified during the Needs Assessment, list the mandatory and desired requirements. Mandatory requirements are those needed for a particular output to meet your productivity goals, those without which you would not consider acquiring a system. *Desired* requirements include functions that may be needed to meet estimated future workloads or to increase productivity beyond current goals. These decisions will be different for every agency, depending on the results of your Needs Assessment.

The User Requirements form the basis for the rest of the acquisition process. Thus, defining those requirements is a very important task, one that demands much time and effort. Although it may be impossible to define all of your requirements until you have actually used a package, include as many as you possibly can. (Documentation can be updated at a later date.) Remember, you are working to acquire a system that is appropriate not only for today, but also for the future: you want a system upon which you can build.

5

Conducting a Feasibility Study

The purpose of the *Feasibility Study* is to determine whether the system acquisition process should move ahead and, if so, whether the User Requirements statement can serve as the outline of a new or enhanced system. The Feasibility Study will help your agency analyze problems with the current system, examine various ways to respond to the problem, and evaluate those responses. The study also includes a review of the financial, technical, and administrative resources available for system acquisition, citing the estimates of the costs and benefits.

A feasibility study conducted for a microcomputer acquisition is much briefer than its counterpart for a mainframe or a minicomputer acquisition. Depending on the size of your agency and the parameters of the anticipated computer system, it may not be necessary to conduct a *formal* feasibility study, but you should know what it entails so that you can devise a checklist approach or otherwise modify the process to fit your needs.

Conducting a Study

Maintain Impartiality

Given that the purpose of the study is to determine whether moving forward with the acquisition process is a viable proposition, you will want someone other than a Project Team member (who is probably already committed to the process) to conduct the study. To provide objectivity, it is important that a disinterested party, preferably an outside consultant, execute the Feasibility Study. If the size of your agency and your finances preclude hiring someone from the outside, then try to find an aware, but independent colleague to manage this component of the acquisition process. The Feasibility Study manager you designate must have strong technical, analytical, and communications skills and be familiar with the workings of the agency and the results of the Needs Assessment and the User Requirements stages.

Review the Needs Assessment and the User Requirements Statements

The Feasibility Study repeats some of the elements of the Needs Assessment and the User Requirements, but it goes into much greater detail. And, to establish the validity of the User Requirements, the manager of the Feasibility Study will want to start from the beginning. Be prepared to have the work completed in the Needs Assessment and the User Requirements stages questioned in some depth. The Feasibility Study manager will try to answer, yet again, the following objectives.

- What problems or issues does the agency need to address?
- What are the agency's information needs, and do they relate to those described in the Needs Assessment statement?
- Can the agency's information needs be met by the elements outlined in the User Requirements statement?
- What changes to the User Requirements statement, if any, are necessary?

Interview Staff

The study will necessarily involve interviewing staff members to determine their report requirements (who needs what report and why); required response speed (when do they need it, and how quickly); and any fears or hopes they have in connection with the project. Depending on the agency's information needs, several microcomputers may need to be linked together in a local area network (LAN); thus, it is very important that the Feasibility Study identify the number of people who will need to use the same information at the same time.

Critical to the success of the Feasibility Study is the ability of the person conducting it to create an atmosphere of trust with the interviewees. Staff being interviewed must feel free to criticize the agency's existing systems, without fear of reprisal, and personnel must be encouraged to make suggestions. The person who works with the system on a daily basis can usually provide the most accurate and complete information; the manager of the Feasibility Study must be able to take advantage of that knowledge.

Consider Alternate Responses

After conducting extensive interviews, reviewing the work of the Project Team, and independently identifying the most critical user needs, the person heading the Feasibility Study should have

a clear understanding of the current system and its shortcomings. The manager must attempt to ascertain the reasons for those weaknesses. For example, has the volume of transactions escalated significantly since the existing system was designed, or are inaccuracies in data related to an inability to recruit sufficient numbers of qualified staff? Is it simpler to upgrade the current system rather than develop an entirely new method of processing? If the existing system cannot be improved adequately to meet the needs of users, what improvements are required from the proposed system?

The Feasibility Study manager must now evaluate possible responses to the existing system problem and justify recommendations for a preferred solution. The purpose of this appraisal is to discard obviously unsound proposals and to determine the direction for the future. Multiple options must be considered, including maintaining the current system, enhancing the system with automation, and developing or purchasing a new automated system. Options may also include developing or buying software. Each of these alternatives has its own set of technical, economic, operational, and political characteristics. Once all reasonable alternatives have been identified, the Feasibility Study manager will need to analyze each one in terms of the project's stated objectives, requirements, and associated costs.

Assess the Costs and Benefits

To provide the most viable options for your agency, the manager of the Feasibility Study must assess the costs and benefits of each alternative. This analysis will necessarily include a review of existing budgets and cost estimates for personnel, processing time required to implement an alternative, equipment, supplies, and recurring operating costs (such as equipment and software maintenance, staffing, security and backup facilities). Realistic time frames, the availability (or lack thereof) of in-house expertise, training needs and options must all be considered. The Feasibility Study manager will also need to determine what resources will be necessary to convert existing records from the current system to the proposed new system. The Feasibility Study must provide evidence that the proposed response to the problem will pay for itself over time.

The Feasibility Study manager must be as specific as possible when attempting to quantify the benefits of each alternative — this information will be useful when justifying an expenditure to upper management as well as to local and state control agencies. In addition to the benefits that can be easily quantified, list the

intangible advantages that will improve the timeliness and quality of products and services. For example, the manager might cite improved management control, increased data accuracy and reliability, better working conditions and expanded services. Identifying the benefits of each alternative should ensure that the alternative ultimately chosen has expected benefits that will outweigh the projected costs.

Prepare a Feasibility Study Report

The Feasibility Study manager must prepare a final report on the study for two main reasons.

- To satisfy internal reference and documentation needs; and
- To provide a comprehensive directive for the software and hardware investigation and acquisition that is to follow.

Reference documentation is important because it gives management the opportunity to question findings and to confirm why the project is headed in a particular direction. (Any working papers or notes created during the course of the Feasibility Study will be helpful in preparing the final report and should be saved for reference.) The Feasibility Study Report must clearly state the project's overall objectives, review the alternatives, justify a preferred option, and present an implementation schedule.

Document the Study

The final report should discuss the findings of the study and include a detailed description of the requirements and characteristics of the chosen alternative. This description should demonstrate how the chosen alternative meets the objectives and requirements originally set out in the User Requirements, and indicate how it will affect the existing method of operation, what general type of equipment will be necessary, and what funding sources are available. The scope of the report should be consistent with the complexity of any problems identified.

Develop a Plan and Schedule

Together, the Feasibility Study manager and the Project Team should develop a plan for managing the project and a schedule for implementing the process. Include this information in the Feasibility Study Report. The plan must include project assignments and estimates of the amount of time participants will devote to the project. The project schedule must establish dates for the start

Feasibility Study Objectives

1. To define the operations of the present system, its strengths and weaknesses.
2. To identify alternative solutions to the problems.
3. To identify the costs and benefits of the proposed system.
4. To justify a preferred solution.
5. To develop a plan and a schedule for implementing the new system.

and end of the project, for routine reviews of the project, and for the completion of major tasks. Also include procedures for timely reporting on the project's progress and for adjusting the schedule if and when that becomes necessary. (Project Team members will need to keep each other, as well as management, updated.)

The Feasibility Study Report will keep your acquisition project on track, preventing you from becoming overwhelmed by the many available, but not necessarily appropriate, system options.

6

Investigating Software and Hardware

Now that you have identified the user needs and requirements and (based on the results of the Feasibility Study) assuming that the acquisition process has been approved, you will next determine the actual specifications for the proposed system — what should you buy? To do this, you must thoroughly investigate available software applications and hardware configurations. This task may be the single-most important consideration: If you rush or disregard the investigation, then the system selected may not be able to provide the information and functions you require and may not be compatible with other operational systems in your agency. It is easy to get excited about a new technological innovation and then buy hardware and software systems that turn out to be unresponsive to your needs and incompatible with systems operating in related agencies. The installation of such unresponsive systems has degraded the effectiveness of agency operations by hindering decision-making capabilities and obstructing interagency and interjurisdictional information sharing.

The Project Team will have to investigate application and operating system software and computer hardware alternatives at virtually the same time, because they are dependent upon one another. We suggest, however, that you first examine software because it contains the problem-solving procedures: More success will come from using more productive software. Select software that is appropriate for both your present and future needs before selecting the hardware. After reviewing software applications and operating systems (which will automatically limit the type of hardware you need to consider), select the hardware that provides the power and flexibility necessary to execute the chosen software.

Investigating Software

We recommend that you start your software investigation with a three-step process.

- List all packages available for the kinds of applications you need.
- Analyze the systems that hold the most potential.
- Score those packages against your required specifications.

Assemble a List of Software Packages

Review the Literature

In compiling a list of appropriate software packages, you may want to start by reading reviews in computer magazines and newspapers. These publications also contain numerous advertisements that can keep you up-to-date on the latest in microcomputing. Keep abreast of new developments, but realize that oftentimes companies advertise new products that are not yet ready for the market. Called "vaporware" or "leading edge," such products may sound great but, by their very nature, be impractical at this time. You want current software and equipment, but you also want to be assured that all the kinks have been worked out before you spend any money. You do not want to be the guinea pig for a new product line. Do not overlook business-related magazines and law enforcement journals for nontechnical articles on different aspects of microcomputers. In addition, book review sections in these magazines will direct you to recently published works.

Contact User Groups and Other Agencies

Attend user group meetings to learn how other users have solved certain problems and to hear their opinions about various machines, software and suppliers. Try to find a professional group, preferably people working in criminal justice or a related field.

Also, contact your colleagues in other criminal justice agencies to see what kinds of hardware and software they are using. Ask for both the positive and negative aspects of each system, and take notes on their responses. By comparing the responses of all of your contacts (and your contacts' contacts), you may see a pattern develop. Keep following any leads you receive. It is very likely

that someone, somewhere, had a problem similar to the one your agency is experiencing and may have found a solution of interest to you.

Contact Vendors

Compile a list of software vendors headquartered in your area or with a nearby branch office, and contact them. For informational purposes, you may also want to contact mail-order software houses. Ask for demonstration disks and product literature, including copies of the documentation, if available. Ask for a list of users of the kinds of systems you are investigating.

Analyze Software Packages

Operating System Restrictions

The kind of system configuration you will acquire depends on a specific operating system, which is the first criterion by which you will narrow the field of available software packages. At this point, however, you need to decide only whether you need a single-user or multiuser, single-tasking or multitasking operating system. If, for example, you want to operate a multiuser system, then you will limit the software packages you investigate to those that can be supported by a multiuser operating system. The particular brand of multiuser operating system you select will depend on the particular multiuser software packages that you choose.

Establish Criteria

There are several methods for evaluating systems and selecting the most satisfying package. Common evaluation techniques include weighted score, cost/effectiveness ratio, and requirements costing. None of these techniques can make the decision for you, nor can they guarantee a solution. Each has its own advantages and disadvantages, but the point is that each of these various selection evaluation steps is a formal process, and comparisons are based on objective analysis.

No matter what evaluation technique you choose, when you are ready to evaluate software packages, you must establish general criteria for rating each package. A simple aid in this process is a chart that matches your objectives and functional requirements — as noted in the Needs Assessment and User Requirements

statements — with the software's functions or modules. In addition, because each agency is different, there may be special considerations that will affect your agency's criteria and the selection process. All agencies, however, must weigh the following.

Software Evaluation Criteria

Appropriateness and Performance Capability	Functions to standards; meets needs and expectations
Compatibility	Integrates with existing systems
Demonstrations	Can be evaluated based on information from hands-on demo
Documentation	Should be well written, well organized, well presented and complete
Ease of Use	Requires minimal instruction; speed is sufficient; excessive amount of RAM not required
Vendor Training and Support	Includes basic training plus technical support
Operating System Requirements	Choose software that runs on a standard operating system
Users' Comments	Get evaluations and opinions from firsthand users
Costs	Must be affordable for intended application

• *Appropriateness and Performance Capability*
Be certain that the package performs the functions that it was written to perform and is capable of handling all of the functions that users of that package require. What is the software's capacity to do what is expected, and how effectively does it perform under peak loads?

Compare — point-by-point — the requirements in your Needs Assessment with the capabilities of the software. During the comparison process, you may discover that a particular software package includes important functions that you did not consider or include in your list of desired capabilities. Update your requirements based on the capabilities offered by the different systems.

• *Compatibility*
Compatibility may be the most important criterion in selecting both software and equipment: It is essential to the creation of an integrated, coordinated and effective system. You will want to be able to integrate a new system with any packages or hardware that you already have in place. It is fairly easy to avoid compatibility problems (1) by selecting standard software packages that can share files with other packages and (2) by acquiring microcomputers that share the same operating system or support the same applications programs. Avoid obscure packages that do not support a common data interchange format.

• *Demonstrations*
Vendors are usually pleased to provide you with a complete demonstration of the software application. If they do not offer to provide a demonstration, insist on one. Hands-on experience can provide you with more information about the software and its performance than all the promotional literature the vendor can offer. Information such as ease of data entry and error message quality are subjective features and can be best evaluated by direct use. Sit down and work with the program as thoroughly as

possible. Take your time and thoroughly examine each operation that the software performs.

In addition to routine operations, ask to see a demonstration of the program's ability to recover from such things as users' mistakes and power failures. The software should be able either to correct simple errors automatically or to warn the user of their occurrence and prevent the program from continuing until the error is resolved.

If your needs analysis indicates that you should be examining multiuser software, try to evaluate the software with the same number of simultaneous active users as you anticipate at your agency.

• *Documentation*

Good documentation is easy to read and well organized; adequate reference material is available when a problem arises, and this information helps you to solve any problems with the system. The reference material should enable the user to take full advantage of all of the software's capabilities. Depending on the program and how thorough a review you want, your examination of the documentation can take an hour or more.

Traditionally, software documentation has been poorly written, primarily because it is often done as an afterthought. Although this situation is improving, you probably will not find many programs with documentation that meets all the criteria discussed here. Being aware of these criteria, however, should help you identify potential problems before you buy.

When browsing through a manual, note the table of contents, index, glossary, and general organization. Is the manual written in a conversational tone, is the language clear and concise, free of technical jargon? In determining how effective a tool the manual will be, consider the physical format, content, writing style, and the presence of a tutorial. The documentation should include a narrative description of the system, outlining its capabilities and suggesting how you will be able to use them.

Ideally, documentation should be organized into administrative, reference and user manuals. The administrative manual should illustrate the functions required to administer or maintain the software. Reference manuals should include a comprehensive list of available commands, modification procedures, file structures and other items of technical importance. A quick reference card summarizing frequently used commands and functions is

very helpful, as is a troubleshooting guide that tells you what to do when something goes wrong. The user manual contains the operator instructions and is therefore especially important. It should be written clearly enough to enable a novice to easily follow directions and quickly locate any necessary reference material.

Ask the vendor to let you load the program yourself; use the manual to work your way through various chapters. Does the manual include printed representations of the display screens? Do they match what you see on the actual screen? As you use the program and encounter problems, can you find the answers quickly in the manual?

Check to see if the documentation includes a customer registration card. Registration is important because it may be the only way that you are assured of learning about program updates and getting replacement disks if anything happens to your program. Also be sure that the software manuals you are reviewing are the same release version as the software being evaluated. Software companies are constantly updating their products. Sometimes these updates fix former bugs in the program, and sometimes the new version includes new features. If, after purchasing a package, you discover that your documentation is not the latest version, you should be able to get an updated copy from the vendor, but you may be charged for it.

• *Ease of Use*

The package should be capable of being used with minimal instruction and should include sufficient "help screens" or other reference devices. How easy is it to change report formats? How well will the program cope with your volumes? There are often tradeoffs between capability and ease of use: When selecting a package, you must find a balance between appropriate user applications and user expertise. You want to be sure that each software package can run on your agency's microcomputers, that it does not require an excessive amount of RAM, and that it operates with sufficient speed to be useful in your agency. Find out how difficult it is to install the program — will installation require an experienced programmer?

• *Vendor Training and Support*

Software vendors should offer some basic training in the use of the program they are providing. Short introductory sessions may include guidance on recovering from difficult situations and basic faultfinding. ("Check that you have switched on the power.")

Vendors usually also provide some level of technical support for software installation, enhancements and maintenance. Product support may include telephone support, a limited warranty, or more extensive support under a separate maintenance contract.

Telephone support is generally a toll-free phone number with a software technician available to answer your questions on the operation of, or problems with, the software. Vendors frequently offer an ongoing support contract or an extended software warranty for an annual fee. This type of contract usually provides for revisions or upgrades made during the term of the contract, along with either unlimited support calls or a base number of calls at a given fee. Find out how the vendor handles revisions and upgrades, whether new documentation is included and whether the software warranty extends to cover any new versions of the software.

• *Operating System Requirements*

At the same time you are evaluating software packages, you need to determine which operating systems will run which applications. The selection of the appropriate operating system is, obviously, critical to system performance. You want to be certain that the software you choose runs on a standard operating system.

Be sure to evaluate software on a variety of operating systems. Some packages, theoretically designed for both single- and multi-user operating systems, function very well on one but perform poorly on the other. If you require a multiuser operating system, make sure it will support the number of users and the type of environment (single-tasking or multitasking) you need. If a multiuser, single-tasking operating system is sufficient for your needs and you will be using a local area network, get assurance from the vendor that your proposed software will function with your specific LAN operating system. You may want to ask a technically qualified third party to talk to the vendor's support personnel to determine whether the operating system under which the vendor's software operates is capable of doing the job and whether the vendor's personnel can adequately support the system.

• *Users' Comments*

With the lists of users obtained from the vendor and your own contacts, ask people to evaluate the software in terms of their specifications, their expectations, and actual system performance. Try to get a sense of their satisfaction with the software. Ask

their opinions on the level of vendor support, the quality of training, and the average response time when software problems occur. Listen carefully to common complaints from different references. If everyone reports that the system does not generate reports well, perhaps there is a problem with the report module that you should discuss with the vendor.

• *Costs*

The expense of a software package is obviously a major consideration for any agency seeking to buy a system, but costs need to be balanced with the benefits provided. An inexpensive program that does not quite meet your needs is of little use and may actually end up costing you more money if it requires extensive programming changes. When considering the price of a program, also estimate the cost of producing a particular output if the task is automated, as compared to the present costs of producing that specific output.

Score Software Against Specifications

Create a Scoring System

By the time you have examined a number of packages and have talked to users, the field of potential software packages should have narrowed considerably. If you have a chart comparing the various packages on the richness of their capabilities and their ability to meet your requirements, you should have all the information necessary to create a scoring system to determine the package best suited to your needs.

You may wish to create a matrix comparing each program against the nine software criteria. You could then rank the software packages — for each criterion — in terms of quality, such as excellent, very good, good, fair, and poor. Another method is to assign points to each criterion and establish a rating scale of one-to-100. Consider deducting 10 points if a particular program does not respond to a criterion. The highest score will determine the winner.

If a couple of the packages are evenly matched, ask your agency people — those who will be using the system — if they have a favorite among the highest ranking two or three; then choose the most popular package.

Investigating Hardware

When selecting hardware, you should follow the same three-step process as you did for selecting software.

- List all hardware available for the kinds of software and operating system you need. (Identify your proposed system configuration).
- Analyze the systems that hold the most potential.
- Score those systems against your required specifications, including their compatibility with the chosen software.

Your evaluation criteria are also very similar to those for software and include (1) comparing the appropriateness and the capabilities of the hardware point-by-point with the requirements in your needs assessment, (2) assessing compatibility, (3) seeing a complete demonstration, (4) reviewing documentation, (5) ensuring adequate vendor training and support, (6) talking to users, and (7) evaluating costs. In addition, the Project Team must factor in a system's capability to expand, standard components and storage size. **Figure 6-1, Acquisition Checklist — Hardware** should assist you in comparing hardware components. Make several copies of the form; take it with you when you look at hardware.

Although the selection process and the criteria for evaluating hardware are similar to those for software (and we advise following the same steps outlined earlier under **Investigating Software**), there are additional specific hardware and environmental considerations that must be addressed to ensure the selection of the best system. Before considering system selection criteria, the Project Team must reaffirm the proposed system configuration.

Identify System Configuration

Based on its earlier work in the Needs Assessment and User Requirements stages, the Project Team should know generally how many people need to access the system, what information needs to be shared, and what the current inputs and outputs are. Compile a complete list of all hardware required for the project, and list specifications mandated by the system. Also consider installation requirements, including electrical, environmental and cabling needs. A careful review of all that you have learned will help you to narrow down your system choices. Generally, all systems fall into one of the following three broad categories.

System Name _____ **Vendor** _____

Manufacturer _____

☐ Component system
☐ Packaged system

☐ Portable
☐ Desktop

MICROPROCESSOR

8-bit
☐ Z80/8080/8085
☐ 6502

8/16-bit
☐ 8088

16-bit
☐ 8086
☐ 80286

8/32-bit
☐ 68000

Other (describe) _____

Memory

☐ 64K ☐ 128K ☐ 256K
☐ 512K ☐ 640K ☐ >640K

Expansion capabilities ☐ Y ☐ N

Floppy disk type

☐ 3½ " ☐ 5¼ " ☐ 8"

Price $ _____

DISPLAY

☐ Built-in CRT and keyboard
☐ Terminal needed

Does CRT display color? ☐ Y ☐ N

Graphics capability ☐ Y ☐ N

Screen size _____

CRT background

☐ Green ☐ Brown
☐ Black ☐ White

Other (describe) _____

CRT colors

☐ Green
☐ Amber
☐ Gray

Price $ _____

PRINTER

Type
☐ Dot matrix
☐ Character
☐ Sheet-fed
☐ Roll
☐ Fanfold

Output
☐ Letter quality
☐ Correspondence quality
☐ Draft quality
☐ Graphics
☐ Color graphics

Speed _____

Paper size _____

Other features _____

Price $ _____

KEYBOARD

Numeric keypad ☐ Y ☐ N

Number of keys _____

Other features _____

Price $ _____

MODEM

Does this microcomputer support a modem?
☐ Y ☐ N

Type
☐ Direct
☐ Acoustic

Speed _____

Other features _____

Price $ _____

STORAGE

☐ Floppy disks
☐ Winchester hard disk
☐ Bubble memory or RAM disk

Other (describe) _____

Price $ _____

Figure 6-1: Acquisition Checklist — Hardware

- A stand-alone, self-contained microcomputer supporting a single user.
- A microcomputer that processes data for several attached "dumb" terminals.
- Several connected microcomputers, each using its own microprocessor, i.e., a local area network.

With these system types in mind, it is important now to determine the input, processing, and output requirements for the proposed system.

Input Requirements

What type of keyboard, mouse, or scanner is needed? Will the software support optical scanners for input and, if so, what kind? What type of video display will the software and operating system support? Do you require graphics and/or color capability? Do the display screen and keyboard create a comfortable work environment? Ignoring seemingly insignificant details such as the display not tilting and swiveling or having too much glare can cause a great deal of discomfort to those using the system on a daily basis.

Processing Requirements

Can the microcomputer you are considering support the amount of processing that you anticipate? Obtain the specifications and all of the available documentation on the microcomputer from the vendor. Compare the vendor's literature to any available reviews of hardware systems that compare the performances of comparable microcomputers. Contact the list of references provided by the vendor, and discuss their hardware choices in the same manner that you discussed the software.

Output Requirements

Examine your output requirements. Do you require the high-speed output of a dot matrix printer or the print quality of a laser or letter-quality printer? Carefully compare the software and hardware documentation, noting any incompatibilities that may exist between your choice of printers and the system. Although the operating system may support a specific type of printer, note that the application software or hardware may not. It is important that your choice in printers be specifically supported by both the hardware and software.

Analyze Hardware Components

Establish Criteria

As you did when you evaluated software, you must establish general criteria for rating each hardware component. Consider creating a chart that matches your objectives and functional requirements — as noted in the Needs Assessment and User Requirements statements — with the hardware's functions. We suggest you consider the 10 following basic criteria, as well as any special considerations your agency may have.

• *Appropriateness and Performance Capabilities*
Above all, you want a system that matches your requirements. As you did for software applications, compare the capabilities of the hardware with the requirements you developed in the Needs Assessment and User Requirements statements. If an alternative does not meet your basic objectives and functional requirements, eliminate it from further consideration.

• *Compatibility*
Cooperative sharing of information is critical to criminal justice operations, and it is important that there is integration both *within* a computer and *between* computers. The capability to transfer files to and from databases and the capability of different microcomputers to share or accept the same physical parts are both forms of compatibility.

Compatibility.
If your agency is going to be using more than one microcomputer, it is important to consider compatibility. The microcomputers you purchase must be able to work together as a system. Often, incompatibility exists between different brands and even within the same brand of microcomputer produced at different times.

Be sure that the unit you are buying is compatible with units that have already been installed or that your agency plans to install. You will want to be able to share data disks among stand-alone systems, and you may want to link systems together in a network or link a microcomputer to a mainframe. To avoid compatibility and integration headaches, purchase microcomputers that share the same operating systems, that support the same application packages, and that can use the same expansion products. With fully compatible microcomputers, data transfer and training are simplified and bulk purchasing is possible.

• *Standard Hardware*
Selecting standard hardware is part of ensuring compatibility. Many different manufacturers develop hardware peripherals such as printers, modems, terminals and memory. When selecting a microcomputer, you want to be sure that you have many choices in peripherals and a great deal of flexibility. Your agency's best interests are served when you purchase a microcomputer

that will connect with the largest number of peripherals that meet your needs and budget. If your system will run peripherals from a variety of manufacturers, you know that you will be able to expand that system confidently.

The computer industry recognizes the value of having many choices and has developed standards to allow for greater connectivity between different microcomputers and peripheral devices. There are both official standards — created by professional organizations such as the Institute of Electrical and Electronic Engineers (I.E.E.E.) and the International Standards Organization (I.S.O.) — and informal, commercial standards. The commercial standards may have no legal force, but do enjoy general industry support because they are set by major manufacturers.

Of course, there are quality systems available that may not use one of the industry's standard hardware buses or interfaces. If you are interested in such a system, discuss your concerns with the vendor and request a list of third-party companies that manufacture components compatible with the system.

• *Demonstrations*
It is essential to get a live demonstration of a working system, configured identically to the system you are considering and operated by people with no vested interest in the vendor. Preferably, the demonstration should be conducted at a user site where the system has been operating for some time. Personnel from your agency who will be responsible for the system should participate in the demonstration. Discuss with your personnel any problems, concerns, or questions about the system.

• *Documentation*
Ask the vendor for complete system documentation for both the microcomputer system and all peripheral devices; verify that the documentation matches the system and peripherals you are evaluating. The documentation will help you assess the system's specifications, capabilities and performance. Review the documentation thoroughly, and ask for an explanation of anything you do not understand. Like the software documentation, hardware documentation should be clearly written and as free of technical jargon as is possible.

• *Expandability*
The amount of memory available and the machine's capability of handling multiuser or multitasking operations are important factors when estimating maximum usage. But no matter how you estimate usage, it seems as though more memory is always

needed. It is advisable, therefore, to purchase microcomputers that can be expanded through the use of expansion cards. Other options, such as spoolers and modems, should also be considered. Microcomputers will usually be integrated into networks, and you should keep this fact in mind when buying. Be certain that the selected network provides for expansion.

• *Size*

The amount of memory available and the machine's capability to handle multiuser or multitasking operations is critical to your choice of components. Of course, the amount of necessary storage and the file size will depend on the tasks the computer will be expected to complete, the number of people accessing the system at any one time, and the type of programs you will be operating. Do you need several terminals or several programs operating simultaneously? Multitasking capabilities are useful when the microcomputer is used for intensive processing efforts, such as calculation of statistics, or for running numerous programs at once. Multiuser capabilities are useful when the microcomputer replaces a minicomputer as the major processing tool of a department or small agency. In some cases, multiuser options should be considered as a less expensive alternative to linking large numbers of microcomputers together in a network.

• *Vendor Training and Support*

Microcomputers are relatively inexpensive, but they are also complex machines. Unfortunately, however, microcomputer vendors typically offer less advice on installation and maintenance than do vendors of other office equipment. The amount of backup a vendor is able (or can afford) to offer is limited. The Project Team's job, therefore, is to select a vendor who can assure high-quality and reliable service.

The version of the system you purchase is critical to the amount of support you can expect and the flexibility you will have if you want to expand the system in the future. Find out if the manufacturer is still advertising the hardware model the vendor is suggesting you purchase. You do not want to find yourself with an obsolete system, even if the price is right. Also realize that a manufacturer or vendor may stop supporting an existing product line after introducing a new system.

Service may be handled through the manufacturer, dealer, or a third-party service organization. In-house servicing is not advised unless agency personnel have special knowledge of microcomputer interface techniques and the appropriate tools. If your agency is considering purchasing a service contract, check to see

how much service is available and if a toll-free phone number is provided. Consider the vendor's reputation, the distance from your agency to the nearest office, the vendor's experience in handling the particular system you are considering, and the availability of backup in the event that units must be moved to the vendor's shop for repair.

• *Users' Comments*

In addition to acquiring a list of users from the vendor and asking those users for their comments, personnel from your agency need to talk to their counterparts at a user's site about any concerns or problems with the system. Find out how the vendor reacted to their requests for assistance. If you have follow-up questions after a demonstration, call key personnel at the site. Pay particular attention to comments about speed, storage capacity, and ease of use. Ask other agencies how the original sizing of their system measured up under actual working conditions. Find out what changes people would make to their system's configuration if they were to start over again. Most people will be happy to share their hindsight.

• *Costs*

The initial cost of the system may seem low, but you must factor in the addition of necessary software peripherals and networking, which can significantly raise your expenses. Also realize that computer procurement is not a one-time expense. When estimating system costs, include expenses associated with repairs, upgrades, expansions, and replacements.

For small systems, where network linkages are minimal and few peripherals are required, the total cost of a single unit is minimal compared to the salary of the individuals using the system. If a microcomputer enables personnel to be more efficient, considerable savings result, no matter how frequently the computer is used.

One method for limiting expense and providing access to sophisticated peripherals is the use of local area networks (LANs). Networks can link microcomputers together, permit sharing of data on large disk systems, and provide access to sophisticated, superior, high-cost peripherals such as laser printers. The file-sharing and communications advantages make LANs popular. Nonetheless, a network can be a complicating factor, and your agency's need for this arrangement must be justified.

Cost justifications must be flexible. Microcomputers can be continually upgraded; a static policy will result in inefficiency. The total system cost must, however, be controlled.

Scoring Hardware Components

To supplement inadequate documentation for a particular product, it is a good idea to check with a bookstore to see if there is a book available that explains how to use the software program. Also find out if there are any newsletters or other publications put out by user groups or software houses. These publications often include helpful tips and solutions to common problems.

If you have not already done so, you should consider creating a matrix or chart to evaluate the various components. Compare each component against the 10 criteria. Examine their capabilities and ability to meet your requirements. Your evaluation might be based on a weighted score, cost/effectiveness ratio, or some other scoring technique. One method is to rank components in terms of quality, or you may want to assign points to each criterion and establish a rating scale of one-to-100. Deduct points if a particular program does not respond to a criterion. The highest score will determine the winner.

Whatever method you decide to use, be certain it is a formal process and that comparisons are based on objective analysis.

By making some basic assumptions about your processing needs in relation to the desired physical configuration, you can direct the vendor to those systems that best meet your requirements. Each type of microcomputer system requires a different operating system and is therefore able to support different application software. As you have already discovered, microcomputer applications and systems differ greatly in benefits, cost, physical size, and structure. You need to carefully weigh the benefits and capabilities of each against your own system requirements and budget. Only after you have made these assessments should you seek out a vendor.

7

Requests for Proposals

In the context of a computer acquisition, a Request for Proposals (RFP) formally invites vendors to propose a solution to the agency's particular computer needs. Many microcomputer acquisitions will not require RFPs because the equipment and software applications can usually be purchased off the vendor's showroom floor or "off-the-shelf." If, however, your agency is interested in a very specialized or complex system configuration (i.e., a system with several workstations and a local area network that will affect many people and departments), then you may wish to develop a detailed RFP and ask several vendors to bid on the job. This chapter will briefly outline the major components of an RFP and discuss generally how to evaluate the proposals and select a vendor.

Brief Vendors on Your System Requirements

Before sending out a formal RFP to vendors, contact their representatives and invite them to attend an orientation meeting on the proposed system. Try to reach vendors from different sectors of the market, including manufacturers or sole distributors, software houses with an interest in microcomputers, specialist microcomputer suppliers, and vendors of general electrical and computing equipment.

Describe the functions and capabilities of your current system to this group, and explain what you would like your new system to do. Describe the areas within your agency that will need to be converted, discuss any general problems you anticipate, and provide an estimate of your current and future volumes. After this general discussion, ask those who wish to receive the formal bid invitation to submit a letter of intent. This way, you will not spend the time and money to prepare and mail a specifications packet (the RFP) to vendors who are not genuinely interested in bidding on the system. The manufacturers' letters of intent should be kept on file for reference.

Prepare a Request for Proposals

Much of the material that you need for the bid invitation can be taken directly from your Needs Assessment and User Requirements statements. The RFP should include the following sections.

- General agency information
- Administration and management of the RFP
- System specifications
- Data flowcharts
- Information requested of each manufacturer

General agency information and administrative issues should be brief so as not to detract from the other, more critical parts of the RFP.

General Agency Information

To help the vendors understand your operational needs, briefly describe your agency, stating its primary objectives and activities. List the tasks and activities (if any) that are currently computerized, and explain how they affect your agency's programs and organizational units. You might also describe any upcoming organizational changes or plans your agency has for future automation. This section allows you to discuss any general information about your agency that you believe will be useful to the vendors' understanding of your needs.

Administrative Matters

In this section, you will provide the vendors with an outline of the entire bidding process — explaining how the RFP document is arranged, what information will be provided, what information is expected from the vendors, what requirements must be met to be eligible for consideration, and what the vendors' responsibilities are before and after the installation. Define the scope of the proposed project, describing the area or areas affected by the new system. As clearly as possible, explain the criteria for analyzing and comparing vendors' proposals. Finally, you must include a schedule of key action dates, including the deadlines for submitting a draft proposal and a final proposal, date of a bidders conference, date of the contract award, and the target date for installation of the system.

System Specifications

In this section, you are asking the vendors to configure hardware that will meet your main requirements, which, of course, are based on your Needs Assessment and User Requirements statements. Drawing on the work that you have already completed, outline your key requirements, including portability and flexibility requirements. Begin with a fairly detailed description of your current system (equipment and software), so that bidders can review your present situation and recommend the best equipment for the workload. Include information on file characteristics, volume, frequency, and input/output requirements. Explain any unusual problems or idiosyncrasies inherent in your system. This information should be followed by a similar description of the *proposed* system.

Describe the proposed system's requirements, including the following:

- *Planned inputs*: Name and content of input data (such as documents and forms); origin of data within the system; and volume of inputs, including high and low points.
- *Procedures for handling data*: Types of transactions; transmission of local and distant data; computations required; new data generation within the system; and control points to test data accuracy.
- *Files to be maintained*: On-line and off-line storage; name and contents of files to be maintained; procedures for updating files; and size of files to be maintained.
- *Output needs:* Name and content of output data (such as reports and summaries); timely distribution of output data; and volume of outputs, including high and low points.
- *Other requirements:* Necessary policy changes; special internal controls; cost ceilings; compatibility of common data processing languages; data security measures; audit facilities; and any other special considerations.

Remember that clear, comprehensive descriptions will enable the vendors to respond with system configurations tailored to your particular needs; ambiguity or deficiencies in your information will cause the vendors to respond only with standard approaches that could be applied to any and all potential clients.

Data Flowcharts

Data flowcharts give the vendor a visual outline of the proposed system and its subsystems. Provide diagrams of each functional

area to be automated, plus an overview of the proposed system. The information necessary to create these charts was gathered and presented during the Needs Assessment stage and updated in the User Requirements stage. If the project has been properly documented all along, and if your data flowcharts are consistent with the information provided in the previous section, the material can be used here without any changes.

Information Requested of Each Vendor

In the final section of the bid invitation, identify what vendors should include in any bid. Specifying this information helps to ensure that you will receive comparable information from all of the vendors, which will make your final evaluation much easier. You will want a complete statement of the system specifications, price ranges, terms and timeframes. The vendors must specify the capabilities and technical features of both the basic equipment and its components and the peripheral equipment. What are the site preparation and installation requirements for each vendor's product (such as the amount of physical space and electrical power required)?

It is very important that you ask the vendor for information about the type and the extent of support and maintenance service the company will provide. What kind of training classes are available, and will they be conducted on-site? In the event of a breakdown, can repairs be made within hours? Will you be able to use an alternate component? Are compilers, programming aids and program libraries available? Are there any special products or services that the vendor can offer to ease the transition of moving to a new computer system?

Finally, ask potential suppliers for their cost estimates for the equipment and programs, as well as for whatever is needed to support their system. Require delivery, insurance, and equipment and software maintenance costs. Vendors must provide a financial statement and state any terms of license or contractual limitations.

Conduct a Bidders Conference

After you have sent out the RFP to those manufacturers interested in bidding on your project, it is useful to hold a Bidders Conference. The RFP will have delineated the many system requirements, but the various vendors undoubtedly will have additional questions that require clarification from you and members of the Project Team. Some may ask to modify the requirements to take

advantage of special features of their product. The conference can be very beneficial for both you and the vendors.

Evaluating Proposals

Once you receive all of the bidders' proposals, you will have to match the proposals against the selection criteria. Those that come closest should be invited to give a presentation and demonstrate their system, validating that the system matches the vendor's claims. The evaluation process then ranks the finalists, deciding which one is best suited to meet your needs. During this stage, you will have to assess price, availability, and technical support. The "winner" should have the best combination of cost, reputation, reliability, service record, training, delivery time, lease-finance terms, and conversion schedule.

It is easy to become distracted during the evaluation process and to "forget" your basic requirements — what it is that you really want from a new system. Ask yourself some basic questions about each proposed system:

1. Will the system solve my agency's central problem?

2. Is it reliable?

3. Is it easy to use?

4. Will it expand as the agency's responsibilities develop?

5. Is the cost justified by the benefits?

Compliance with the RFP

One of the first determinations you will make is how well each bid complies with and meets the requirements of the RFP. Consider clarity, accuracy, and completeness. Does the proposal respond to all the points presented in the RFP? Are the vendor's time and cost estimates feasible? Does the proposal reflect a good understanding of your requirements? Failures in any of these areas may cause problems for you in the future.

Establish Criteria

Just as you did when you investigated available software and hardware, you must establish meaningful criteria for evaluating each proposal from the different vendors. These criteria may include high performance levels, the availability of particular software and special application packages, the ease of conversion, and the response of other users currently using the equipment. You may find you can immediately eliminate some vendors because of a major failing — for example, the lack of an application package that is critical to your operating needs.

Establish a Scoring System

Examine the criteria that you listed earlier and rank them in order of importance. Assign a weight to each criteria, differentiating between mandatory and desired (or major and minor) features. It is unlikely that all user requirements can be met, so focus on satisfying the primary requirements. Rate each proposal according to the criteria. In **Figure 7-1,** three sample RFP's were scored

Software Criteria	Vendor A	Vendor B	Vendor C
Mandatory			
Low-cost throughput performance	Y	Y	Y
Expandability of equipment	Y	Y	Y
Compatible with existing systems	N	Y	Y
Low monthly rental	Y	Y	N
Dependable and efficient software	Y	Y	Y
Equipment backup in local area	Y	N	Y
Availability of operating personnel	Y	N	Y
Availability of technical support	Y	Y	Y
Desired			
Availability of equipment when needed	Y	N	Y
Compliance with installation requirements	Y	Y	N
Adequate equipment maintenance	Y	N	Y
Compliance with terms of bid invitation	Y	Y	Y

TO SCORE

Total points = 100

(8 mandatory criteria x 10 points) + (4 desired criteria x 5 points) = 100 points

Subtract 8 points for each no (N) answer of a mandatory criterion
Subtract 5 points for each no (N) answer of a desired criterion

In this example, Software A scored 90 points; Software B scored 70 points; and Software C scored 85 points.

Figure 7-1: Simplified System Evaluation Exercise

according to an agency's established criteria. The system proposed by Vendor A scored the most points for the total performance score and would thus be the agency's first choice. **Figure 7-1** represents a very simplified composite scoring exercise.

You would most likely want to evaluate each criteria separately first. For example, when determining hard disk drive options, you would score the capacity and cost offered by each vendor.

The acquisition of a microcomputer is a major decision. Advance planning and review are critical, and each of the steps leading up to the creation of the actual RFP is essential. If you have done all of your homework and can clearly articulate what you want a new system to be able to do, the responses you receive to the RFP will reflect that effort. Ultimately, your decision concerning the appropriate vendor will be that much easier. A careful comparison of your criteria to the plans proposed by vendors will help to ensure that satisfactory decisions are made in acquiring all of your computer equipment.

8

Contracts and Optional Costs

Whether you plan to purchase your system from a computer store or send out an RFP and have a vendor customize a system to meet your requirements, you will need to negotiate a formal contract. Obviously, a complex, customized system will require a more extensive contract with the vendor, but for even the smallest acquisition, you should understand the process and be aware of purchasing costs in addition to the basic price of the system. Many of these costs are negotiable, and some vendors will have more options to offer than others. This chapter will highlight some of the key points and provisions in a contract and will review typical optional costs, but you should seek professional advice in these areas.

Negotiating a Contract

Every contract is negotiable to some extent and, if you are prepared to accept some tradeoffs, you should be able to successfully negotiate amenities and terms with the vendor. Your objective in negotiating a contract with the selected vendor is to maximize contractual protections and minimize system expenses. To be enforceable, all points agreed upon must be reflected in the written contract. Even if you have a longstanding, friendly relationship with the vendor, do not assume anything. The vendor's specifications and claims, amount of support, and the type of input and output devices provided should all be spelled out in the contract.

Generally speaking, your contract should be goal-oriented, stipulating the results to be achieved with the system. Make certain it defines the system performance in terms of the expected functions. Clearly identify each component, and list the performance criteria that the processor must meet under regular operating conditions. When contracting for software, realize that you are usually purchasing the right to use the program; you are not assuming the title for the package.

The Uniform Commercial Code

The Uniform Commercial Code (UCC) is a set of commercial laws regulating business and trade in most of the United States. Although the entire commercial code has not been adopted by every state, in most situations, the courts decide issues consistently. The commercial code provides certain warranties as to the quality of the product sold (and also applies to many lease transactions). Essentially the UCC implies that the vendor has title to the goods when sold, the goods are merchantable, and the goods are fit for the purposes for which they are intended. Be aware that the UCC applies to goods, not services. Software publishers and sellers may argue that software is a service, but you and the vendor may agree to make the UCC applicable to the software. Without such a provision, the code may not cover software purchases and maintenance. Watch for language that may negate your rights under the UCC.

The following pages discuss typical contract provisions that cover the computer system, the purchase once the system is delivered, special software concerns and maintenance.

The Computer System

You should be certain that provisions for the following items are included in your contract, rather than relying solely on the reputation and integrity of the vendor. Remember, sales people may change and new management may take over a company, but the contract remains in force, so be certain it covers all your needs.

Security

Until the vendor receives full payment for the system, the company will retain a security interest in the system. This provision minimizes the vendor's risk of losing both the system and the payment if you fail to pay.

Shipment/Delivery

The vendor should deliver the system to your site, but a frequent problem is the failure of the vendor to deliver on schedule. Language in the contract may even permit the vendor to set back the shipping date. Set a realistic shipping date, realizing that a delayed delivery may cause you undue hardship. Your contract should specify the remedies provided for failure to meet the

agreed-upon delivery schedule. A clause stipulating a penalty or contract termination should provide the vendor with an incentive to deliver on time.

Risk of Loss or Damage

Who bears the risk of damage to the equipment once the system has left the vendor's shop and is in transit? Insurance will usually cover this risk, but the contract needs to state who will be responsible for purchasing the insurance and what contingencies the insurance will cover.

Definitions and Descriptions

Including technical definitions associated with the computer system makes clear to all parties just exactly what the contract covers. The buyer and seller are protected if you also include in the contract product descriptions of the computer system, features, capabilities, and configurations. The vendor's proposal may contain such descriptions and should be appended to the contract. This care can help to prevent future disputes over interpretations.

Purchase Agreements

The following provisions cover the computer system purchase after delivery to your site.

Installation

The contract usually states that the vendor must install the system according to the vendor's specifications and the purchaser must provide a suitable environment. Of course, you must find out what is required to prepare the computer site.

The contract should include installation details, such as the requirements, the procedures, and the definition of a completed installation. It should also address the installation of the operating system, any communication software, and all the application software. The next chapter discusses installation procedures in greater detail.

Acceptance

Once the vendor delivers and installs the system at your site, the vendor's technical staff should run diagnostics to ensure the operation of all components. In some cases, you may also need to contract with the vendor to do any required data conversion or data entry. Request a detailed description of the procedures necessary to install and test the system. Be certain that the technician tests each component and verifies in writing that the system is functional, including all communication devices, printers, terminals, disk drives, and peripherals. If you have a remote site, ask the vendor to document a successful connection and transmission of data to and from the site.

Make certain that you are allowed time for acceptance testing. Microcomputer vendors often make minor modifications to the operating system or the software without giving notice. Use all the programs and all facilities of the equipment: Check that they do what they are supposed to do. A 30-day in-use performance and acceptance test is not an uncommon request. Such a test often requires that the system maintain a performance ratio of 98 percent up-time during the first 30 days or the agency will not accept or pay for the system until it meets the requirement. Do not accept delivery of the system until you are assured in writing that all system components are installed, tested, and functional.

Limitation of Liability/Remedies

Vendors try to limit their liability for nonperformance as much as possible and include language that states that, as long as the hardware can be repaired, the vendor will make the repair as it chooses. If the hardware cannot be repaired, the vendor is liable for only a fixed amount of money, which is either stated in advance or determined by a set formula, not to exceed the cost of the system. This area is difficult to negotiate, but try to expand your remedies to include replacement of the entire system.

Finances

Another difficult item to negotiate is the time when payments begin. Payments may begin before or upon delivery and installation. You may wish to make partial payments as the vendor reaches certain milestones and full payment only after you have tested and accepted the system. (See the *Acceptance* paragraph above.) Withholding payment gives you some leverage to improve the vendor's performance.

The contract is usually signed before delivery and, thus, the prices defined in the contract should be protected. Read the contract carefully for language that allows the vendor to increase the price before shipment. You may wish to include a clause that specifically prevents a price increase over a limited period of time before shipment.

Training and Support

Adequate training on a system is imperative, but it can represent a substantial cost. Make sure that you have written confirmation of the exact amount of training and support to be included in the purchase price of the system. Contact other users about the quantity and quality of the training they received from your vendor. Verify that you are able to keep all training materials and have the right to reproduce the materials to train your staff.

Once you have expended the training included in the system purchase, you may need to purchase additional training. Most large vendors have a highly competent, full-time training staff and an ongoing schedule of classes. Find out if the vendor offers training for people who will be using the system on a regular basis and a separate program for personnel responsible for administering and maintaining the system. It may be cost effective to have the vendor train a few of your personnel who, in turn, will train others in your agency.

Acquiring software and hardware with on-line tutorials can also help to reduce training costs. It is important at the outset to designate someone in your agency to stay informed about the system. That person may need to attend additional training classes or be allocated time to master the operation of the system and to stay current with new developments.

Ongoing telephone support is very useful after the official training classes are over and staff are in the office working on the system. As questions arise or problems occur, it is important to be able to contact the vendor for assistance. Be sure you know what restrictions apply.

Documentation

Be certain the system's purchase price includes the documentation. Check to see that the contract states the quantity and type of manuals and any other materials that are included.

Maintenance

Most microcomputer systems have only a 90-day warranty, after which time you are responsible for paying for repairs and service. This can be a considerable expense so a maintenance agreement, which acts as an insurance policy against repairs and service, is useful. The following provisions concern hardware, but similar provisions can be used for a software support contract.

Term

Term refers to how long the contract is in effect. The initial term may be one year with automatic renewal. This provision gives you long-term, indefinite coverage and also allows you to terminate at any time.

Determine the starting date in relation to other contract provisions: You probably do not want the maintenance contract to overlap with the warranty or acceptance period. This provision is especially important when you have separate contracts for service or purchase. If you purchase the maintenance agreement at the same time as the system, you can frequently negotiate a discount.

Reliability Guarantee

When you do have system problems, it is obviously important to keep system down-time to a minimum. The vendor should specify the following.

- *Minimum hours of useable time per day*: The amount of time of computer operation before a shutdown.
- *Mean Time Between Failures (MTBF)*: The length of time the system will run without breaking down.
- *Maximum Time to Repair*: Response time needed to repair the system.

A four-hour response time is an industry standard, and your agency should not be charged a premium for a response time of more than four hours during normal business hours. If, however, you want 24-hour service or a response time of less than four hours, expect to pay more. You may want to insert a penalty clause (usually a monetary charge against the vendor) if the agreement is not met. Also, consider inserting a clause in the contract that mandates that the vendor lend or replace equipment if the needed parts or repair are not available in a reasonable time, as stipulated in the maintenance agreement.

Excluded Services

A maintenance contract does not cover all necessary service situations such as fire, water damage, power failure, and tampering by unauthorized personnel. Some of these contingencies can be covered by an insurance policy. Some vendors may perform excluded services for an additional charge. In the contract, include a list of excluded services to help you determine the total service cost.

Charges

Some contracts will charge a flat percentage of the total value of the equipment. A high maintenance fee may be realistic for mechanical devices such as a printer, but not so for the actual microcomputer, which (once installed) is unlikely to give you much trouble. Try to have your maintenance agreement "unbundled" so you pay a separate, lower fee for the computer. Also, it is best to have all maintenance performed by one firm to avoid one company blaming another for any equipment failure.

When purchasing a configuration with several stand-alone microcomputers, you can protect against high maintenance costs by purchasing an additional machine as a "floating spare." The spare is used in place of any system that fails. You can then take advantage of the vendor's normally lower maintenance fees for the failed equipment by taking it to the vendor's repair shop rather than paying for on-site service.

Payment may be a one-time annual fee or a fixed monthly charge during the maintenance period. Upon renewal, the contract may allow the vendor to raise charges.

Software

As stated earlier, software purchases are really a license to *use* the software for an indefinite period. Software license provisions may be separate or included as part of the main purchase agreement.

Licenses

The vendor is trying to prevent unauthorized use or distribution of the software so the purpose of the license is, for the most part, restrictive. You are not the only licensee, and you may not transfer the software to another person. The license may also mandate the specific machine on which the software can be used.

If you have several machines, the software suppliers may require you to make multiple license agreements. The license may also restrict the number of program backup disks that you can have in the event the original master disk is lost or destroyed.

The vendor that you are contracting with may not be the actual software developer so you need some assurance that the vendor has a license to distribute or sublicense the software.

UCC Applicability

As mentioned earlier in this chapter, the UCC generally applies to goods and not services. For your own protection, you and the vendor need to agree to consider the software as goods for purposes of your contract.

Upgrades and Enhancements

Software designers often find ways to improve their product. "Modifications" may correct program errors or additions may enhance and upgrade the product. Typically, the vendor has only to "advise" you of any changes. Include language in the contract that requires the vendor to send you design changes as they become available. Ask about the costs of upgrades and enhancements before signing any contract or purchase agreement. If you do not have an agreement for free or discounted enhancements, the cost of each upgrade could possibly equal or exceed the initial purchase price. Specify who will pay for and install the changes.

Source Code Access

Access to the software's source code is critical if want to modify the program, assuming that the license permits it. The vendor does not want to lose control over the software by allowing everyone to have the code. Sometimes, however, you can make special arrangements to have the vendor customize the software for you.

Signing the Contract

For a variety of reasons, the vendor's standard contract may not fully protect all of your interests. The burden is on you to be certain that you understand and agree with all of the contract's provisions. A deficiency in the contract may not be discovered until a dispute arises, which is too late. The contract is most likely enforceable. It is your obligation to know exactly what is , and what is not, contained in the contract. Your signature implies that you have read the contract, understand it, and agree to abide by its terms and conditions. Make sure you have some legal advice; do not sign until you are satisfied that the contract is acceptable.

9

Implementation

Typically, throughout the acquisition process, the majority of effort is focused on obtaining information that will assist you in determining the best system for your agency. Too often, preparation and planning for integrating the microcomputer into your agency is conducted as an afterthought. To assure a smooth transition from your current system to a new system, consider:

- Training plans
- Physical requirements
- Implementation plan
- Data conversion
- Job classifications

Early preparation in each of these areas should facilitate the new system's success.

Training Plans

Prior to installing the system, you should have in place a concise training plan. When creating this plan, encourage staff to make suggestions as to the amount and type of training they believe they need to do their job.

Determine who will receive training, what type of training will be offered, and when and where the training will be held. Who are the primary users of the system, and what is their level of expertise? Some vendors offer advanced courses, which would be appropriate for those personnel in your agency who have some prior understanding of and experience with microcomputers. Compile a list of training classes available from the vendor and in the community that would be useful for your staff.

With the help of the Project Team and the users, set milestones or review dates to evaluate training progress, and establish an implementation schedule with goals to be accomplished. The schedule should provide realistic time requirements to complete the training.

Physical Requirements

An area commonly overlooked during the hardware evaluation process is the relationship between the agency's physical environment and the proposed hardware. Agencies often neglect to examine the work area's physical characteristics and suddenly find themselves in need of additional space, power, cabling, and cooling facilities. A detailed site plan can be of tremendous assistance when choosing the appropriate hardware. Determine the physical requirements of the chosen computer system, and prepare the area well in advance of its arrival.

Certainly, before you select the microcomputer, obtain a copy of the floor plans of the area where you plan to install the equipment; mark to scale the proposed locations of the computer system, workstations, and proposed routing of cables. With your site plan, walk through the area and look for potential problems, such as insufficient power outlets, cooling, and water sprinklers and overloaded electrical circuits. Try to determine if the noise level from the new system's cooling fans or printers will disturb the normal function of the workspace.

Consideration should be given to the following requirements.

- *Power sources*: You will need enough power outlets to accommodate the computer and the peripherals that accompany it. Ideally, outlets should be dedicated for the computer's use.
- *Cleanliness*: Dust, dirt, and smoke are detrimental to computer operation. Choose an area where such hazards are reduced, such as a no-smoking area or an area where few personnel travel. Implementing controls that restrict eating and drinking in the computer area will help to maintain a clean environment.
- *Security*: Most microcomputers and peripherals are small enough that they can be easily stolen. Security measures should be established for hardware and software protection.
- *Storage*: Be certain you have adequate space for a computer's documentation, disks, printer, paper, and other supporting equipment.
- *Furniture*: Incorrect chair height, video display screen position, or workspace height can cause users to experience back or neck ailments. When selecting furniture, consider printer location with respect to the data entry area and space for collating printing materials.

- *Lighting*: Video display screens should be positioned to reduce glare from sunlight and indirect lighting (such as overhead fluorescent lights). The glare from the screen can cause eyestrain among users.

Implementation Plan

An implementation plan should be created to inform staff of their responsibilities and to explain how the implementation will progress. You should plan to continue to run your former system, along with the new system, for a stated period of time until you are confident that the new system is functioning as it should. Be as specific and as accurate as possible. Include the users in the development of this plan to encourage cooperation and to be certain that requirements and needs are being met.

Data Conversion

Data conversion from a current system to a new microcomputer can be very time consuming and tedious. Determine who will be doing the conversion, and plan accordingly. Oftentimes, this cost is included in the vendor's price. You need to establish controls, such as daily audits or reviews, to ensure that data are transferred accurately. Because it is impossible to transfer the data all at once, establish a realistic schedule that will reduce staff anxiety. The schedule should also indicate the phased in-use of the microcomputer.

Job Assignment

Introducing a new computer system into the agency may require you to redefine job classifications and shift job assignments. No matter how smooth a transition, a new system will affect established office routines and schedules and will change the way people do their jobs. Even the job itself may change. To reduce resistance and anxiety, make sure that all staff, not just those who will be working directly with the computer, understand why the agency has acquired the equipment and how personnel will benefit.

GLOSSARY OF MICROCOMPUTER TERMS

This glossary based on the glossary printed in
The Criminal Justice Microcomputer Guide and Software Catalogue
(Sacramento, CA: SEARCH Group, Inc., June 1988).

GLOSSARY

access: 1. The manner in which files or data sets are referred to by the system. Examples are serial, random or indexed access methods. 2. The ability to read and write files from disk storage.

access time: The time interval, usually in milliseconds, between the instant a request for data is made to the controller of a direct access storage device, such as a hard disk, and the instant the transfer of the data begins.

address: An identification (a label, number or name) corresponding to a particular location in storage or any other data destination or source.

address space: The complete range of addresses available to a central processing unit (CPU) or user.

algorithm: An established, finite set of well-defined rules or processes guaranteed to solve a specific problem.

allocation: The process of reserving storage space for programs or data.

alphanumeric: A set of characters that includes both uppercase and lowercase letters, digits and some special (-, /, *, $, (,), +, =, etc.) characters.

ALU: Arithmetic-Logic Unit. The section of a central processing unit (CPU) where arithmetic and logical operations are performed.

analog data: A physical representation of data that bears an exact relationship to the original data; for example, the electrical voltages of signals on a telephone wire are analog data representations of the original spoken voice.

analyst: A person skilled in the identification, definition and development of techniques for the systematic solution of problems.

ANSI: American National Standards Institute. The United States agency that coordinates the development of voluntary data processing standards for industry.

application: A specific task or set of tasks to be performed by a computer program or system.

application software: The prewritten computer programs used to perform some task or job. Contrast with "system software."

architecture: The physical structure, design and interconnection of the internal components of a hardware/software system.

archive: To back up or store data. Archiving preserves copies of files; if the original files are damaged or destroyed by some unusual circumstance — power failure, accidental erasure, system failure — important data are not lost.

argument: A variable to which either a logical or alphanumeric value may be assigned.

array: An ordered arrangement of items or numbers such as a matrix, vector or table.

artificial intelligence: The capability of a device to perform functions that are normally associated with human intelligence, such as reasoning, learning and self-improvement.

ASCII: American Standard Code for Information Interchange. The standard code used to represent letters, numbers and special functions as a series of zeros and ones. This seven-bit standard code was adopted to ease the interchange of data among various types of data processing and data communications equipment.

assembler: A computer program that takes a symbolic nonmachine language instruction prepared by a user and converts it to a binary, machine-executable form.

assembly language: A primitive, although precise, low-level language, specific to each processor. Instead of binary instructions, assembly language is made up of brief symbolic statements, which directly affect the internal components of the processor. Assembly language is noted for its obscure style and lack of required structure.

asynchronous: A method of operation in which processes begin in response to events external to themselves. Contrast with "synchronous."

asynchronous transmission: A technique of data transmission between two computer systems that operate completely independent of each other and do not share any timing information. Each system sends data in packets which have uniquely marked start and stop bits. Contrast with "synchronous transmission."

attenuation: The tendency of a signal to decrease in strength as it passes through a medium or control system.

back up: To copy files onto a second storage device so that they may be retrieved if the data on the original source are accidentally destroyed.

backup: A stored copy of a file, to be used in case of a malfunction that causes a loss of the data in the original file.

background: In multiprocessing, the environment in which low-priority jobs are executed. Two applications are run simultaneously on the same computer. The higher priority application is run in the foreground and the less important one in the background.

band: A range of frequencies between two specified limits.

bandwidth: The measure of the range of frequencies available for signalling. The difference, in hertz, between the highest and lowest frequencies in a band.

bar code scanner: A scanner designed to read bar-coded information by scanning the bar code with a beam of light (usually either laser or infrared light).

baseband: Communications facilities with a narrow bandwith (which is less than voice grade), and, combined with a carrier signal produces a signal capable of transmission. Baseband signals are transmitted at their original frequency, i.e., unmodulated.

BASIC: Beginner's All-Purpose Symbolic Instruction Code. A high-level, second-generation, algebraic programming language designed for ease of use and ease of learning. BASIC has a small number of commands and simple syntax. BASIC has been implemented on most computers.

batch processing: A data processing technique wherein a program with all the required and associated data is processed during a single machine run without user interaction or intervention. Contrast to "interactive" and "conversational processing."

baud: A unit of signaling speed generally equal to one bit per second. Thus, 8 baud equals 1 character per second.

benchmark: A point of reference from which comparisons between systems can be made, such as a program to evaluate disk performance.

binary code: A coding system in which the encoding of any data is done through the use of bits.

BIOS: Basic Input/Output System. The part of the operating system that handles the input and output to devices such as monitors, disks and printers.

bit: A BInary digiT, either 0 or 1, indicating one of two states: off (0) or on (1). The smallest unit of information and basic unit in digital data communications.

bomb: Failure of a system or program.

boot: 1. To load a computer's operating system into the random access memory of the computer. 2. To start up. Cold boot refers to reloading a computer's operating system by turning the electricity to the computer off and then on. A warm boot is a complete reload of a "warm" computer's operating system through the use of the system reset keys on the keyboard.

bootstrap: Also known as bootstrap loader. A small program that starts a "cold" computer. Generally, the bootstrap program tells the computer where to start so that it can clear the machine memory, load the operating system, and set up the machine to begin working.

BPI: Bits Per Inch or Bytes Per Inch. A measurement of recording density; the number of bits or bytes that can be recorded in one inch of magnetic tape.

bridge: A combination of hardware and software that allows either like or unlike networks to communicate with each other. Contrast to "gateway" and "router."

broadband: Communications facilities with a bandwidth capable of handling frequencies greater than those associated with normal voice range transmission facilities, e.g., cable TV signals.

broadcast: A message sent simultaneously to all the workstations of a multiuser system. Contrast with "multicast."

buffer: A software routine that controls a data flow or storage area in random access memory (RAM), used to temporarily store data being transferred at different speeds from one process or device to another (usually during input/output operations).

bug: A defect or malfunction in hardware, software or firmware.

bus: 1. Commonly used as the hardware interface in a computer through which additional central processing unit (CPU) components, options and peripherals are attached. Buses are normally manufactured to an industrywide interconnection standard or are proprietary. 2. A channel or path through which electrical or digital signals flow.

bus topology: A network topology which uses a single master cable for all data transfers, this common data path is the bus (backbone) of the network. All nodes are either attached by a short length of cable to the bus or interconnected by the bus itself. See "daisy chain topology."

byte: A group of adjacent binary digits operated on as a unit by the computer; normally, the eight bits that represent a character or number in binary code.

C: A general-purpose, third-generation, high-level programming language that features a highly efficient code, brief expressions, low-level hardware control and a rich set of features.

cache: A method for reducing the time it takes to access data stored in a large, slow device by retaining the most-often accessed data in a smaller, faster device. Normally, cache is used in reference to a combination of random access memory (RAM) and software for decreasing disk access times. Because memory access is much faster than disk access, accessing data from cache rather than disk increases performance.

canned programs: Prewritten programs prepared by the publisher and made available to the user in a machine-readable form, usually on disk or tape.

chip: Slang for a semiconductor-based integrated circuit; may apply to anything from microprocessors to memory arrays.

CMOS: Complementary Metallic Oxide Semiconductor. A method of making MOS chips which use less power and work faster than MOS chips. Contrast to "MOS."

coaxial cable: Cable composed of one wire, called a conductor, surrounded by a stranded shield that acts as a ground. The conductor and the ground are separated by a thick insulating material, and the entire cable is protected by an insulating jacket. Coaxial cable is usually used in broadband or high-speed applications.

COBOL: COmmon Business-Oriented Language. An "English-like" second-generation, high-level programming language designed by the U.S. Navy in the 1950s.

command: A user-entered instruction that specifies a computer operation to be performed.

communication device: Hardware components that allow a microcomputer to communicate with other computer systems and peripheral devices, such as ports, modems and facsimile devices.

communication software: Software that enables microcomputers to share data and information with remote systems.

compatible: Able to be used together, a software application may be compatible with the IBM PC, which means it will run on a PC. Some computers are PC-compatible, which means they work like a PC.

compile: To prepare a machine language program from a program written in a higher level program language, such as COBOL, FORTRAN, C or PASCAL.

compiler: A computer program that converts an entire high-level language program into a machine language program from a source program. The compiler can replace single program statements with a series of machine language instructions or with a subroutine.

concurrent processing: Performing two or more data processing tasks within a specified time interval, but not at the same exact time. Contrast with "simultaneous processing."

configuration: 1. In hardware, the equipment to be used and the way it is to be interconnected. 2. In software, a procedure performed to prepare a software program for operation or to define a system's resources to the software program.

console: The primary (or master) display screen and keyboard from which the user controls the computer system.

constant: A value that does not change during the execution of a program.

controller: A device required by the central processing unit (CPU) to operate and control peripheral components such as disk drives.

convention: A standard or accepted procedure in programming or systems design.

conversational processing: An application where an input or inquiry by a user causes an immediate response from the system. Conversational systems give the appearance of conducting a dialog with a user. Contrast to "batch processing" and "interactive."

CPU: Central Processing Unit. The component of a computer that includes the circuitry controlling the interpretation and execution of all instructions. The electronic components that cause processing to occur.

crash: A slang term indicating that the hardware and/or software has stopped functioning properly.

CRC: Cyclic Redundancy Check. An error detection method whereby a value is calculated based upon a specific block of data and stored. The CRC value is then compared to the value that is recalculated when the data are reread or transmitted. If the CRC numbers match, the data are free of errors.

CRT: Cathode Ray Tube. The picture tube on a terminal or workstation used to display text and graphic images.

CSMA/CA: Carrier Sense Multiple Access/Collision Avoidance. A multiaccess transmission protocol that uses carrier detection and collision avoidance to ensure reliable data transfer. A typical CSMA/CA system is Apple Computer's AppleTalk network.

CSMA/CD: Carrier Sense Multiple Access/Collision Detection. A multiaccess transmission protocol that uses carrier detection and collision detection to ensure reliable data transfer. Typical CSMA/CD systems are Ethernet and Starlan.

cursor: A block, an underline, a blinking character, or a reverse video image, which indicates on a video terminal, where the next character will appear on the screen.

cylinder: A vertical column of tracks on multiple magnetic disks treated as a unit by the hard disk.

daisy chain topology: A type of bus network topology in which all nodes are directly interconnected by the bus. See "bus topology."

data: The basic units of facts, concepts, or instructions represented by numeric, alphabetic, graphic, or special characters that are used for communication, interpretation, or processing by human or automatic means.

database: Information stored in a computer for subsequent retrieval. A large set or ordered collection of interrelated data (with limited duplication) designed to serve one or more applications and stored in such a fashion that the data are independent of the programs that use them. Thus, a common, controlled approach can be used for adding, modifying and retrieving data.

data processing: The process of converting data into information and the manipulation, storage and retrieval of that information.

data-transfer rate: The speed at which a computer reads or writes data to a storage medium.

DBMS: Data Base Management System. The collection of software programs used to create files, maintain data, and generate reports stored in a database.

deadlock: An unresolved contention for the use of a resource. Normally associated with two users attempting to open a file simultaneously. Also known as a "deadly embrace."

debug: To detect, correct and eliminate mistakes and logical flaws in computer systems and programs.

dedicated: Hardware, software or procedures designed for a specific use.

default: A value or option assumed by the computer or program when no other value or option is specified.

default drive: The drive currently in use by a workstation. In disk-operating systems (DOS), the drive prompt (A>, C>, etc.) that identifies the default drive letter.

delete: To remove or eliminate.

demand paging: In virtual storage systems, the transfer of a page from external page storage to real storage when the page is needed for execution.

device driver: A data structure that allows an operating system and its applications to access low-level hardware or nonoperating system services.

digital data: Data represented in a discrete, discontinuous form, as contrasted with analog data, represented in a continuous form.

digitizer: A device that converts visual images into digital data and inputs those images into the computer.

DIP: Dual In-Line Package. The most common form for an integrated circuit (IC), which has two rows of pins for connectors.

direct access: Also known as random access. The technique of obtaining and retrieving data from a storage device in such a manner that the time required for such access is independent of the location of the data most recently accessed. In effect, the device "goes directly" to the data. An example is a hard disk drive subsystem. Contrast with "sequential access."

directory: 1. A logical portion of disk space named by a user when created. A directory is used to hold a list of logically related files. A directory may be part of another directory (a subdirectory), or it may contain several other directories and files. A directory is analogous to a drawer in a file cabinet in which the user stores files. 2. The list of files displayed when a listing command is entered at any directory level of the file structure.

disable: To deny access, remove, inhibit or turn off a normal capability.

disk: A flat, circular, rotating magnetic media upon which magnetic pulse-coded data can be written and read. See also "floppy disk," "hard disk," "microfloppy," "minifloppy," and "Winchester disk."

disk cache buffer: An area in the memory of a system into which large portions of disk files are read prior to an actual request to read the entire section. This area increases the chance that the next disk read request can be pulled from memory rather than from the disk. Reading from memory is much faster than reading from disk, so performance is greatly improved.

disk caching: A process of reading large portions of a file or disk into a special buffer, thereby decreasing the number of "disk reads" required for the rest of the data and improving performance.

DMA: Direct Memory Access. A technique in which data can be transferred between peripheral devices and random access memory (RAM) without intervention by the central processing unit (CPU).

DOS: Disk Operating System. A generic term for the operating system for IBM and IBM-compatible personal computers. DOS is based on MS-DOS, the Microsoft operating system.

download: The transfer of information from a remote computer to a local host computer.

downtime: That period of time in which a system is inoperable; normally used to denote time in which the system is down due to a fault.

driver: Software that provides the computer with a series of instructions for the reformatting of data to enable transfers with a particular peripheral device.

dump: Copying all or part of the contents of a storage unit, usually from random access memory (RAM), into an auxiliary storage device (usually a disk) or onto a printer.

duplex: Deals with the capability of a communications channel to handle two signals headed in opposite directions at the same time. Full duplex modems can transmit and receive data at the same time. Half duplex modems alternately transmit and receive data.

EAROM: Electronically Alterable ROM. Read-only memory that can be selectively altered without destroying all the stored data.

edit: 1. To check the correctness of data. 2. To modify or change a program or data. 3. To prepare data for further manipulation, such as format changes or code conversions, by a subsequent process.

EEROM: Electronically Erasable ROM. Read-only memory that can be electronically erased, as opposed to EAROM.

electronic mail: The transmission and storage of messages through computers and telecommunications.

emulate: To imitate one system with another so that the host system accepts the same data, executes the same programs, and maintains the same functionality as the imitated system.

emulator: A type of program or device that allows programs written for one system to operate on another. Emulators make a system "look like" the emulated system to the programs.

enable: To turn on, allow access to, or place in a state that will allow operation of a computer or peripheral.

environment: In a computing context, normally refers to the mode of operation (e.g., conversational, stand-alone environment).

EOF: End-of-file. The termination or end point of a file or quantity of data. End-of-file marks are used to designate this point on magnetic files.

EPROM: Electronically Programmable ROM. A special programmable read-only memory (PROM) that can be erased under ultraviolet light and reprogrammed repeatedly. EPROMs are normally manufactured unprogrammed.

erase: To remove data (without replacing) from storage.

EROM: Erasable ROM. See "EPROM."

error: Any deviation of a computed or measured quantity from the theoretically correct or true value.

error-correcting code: A coding system in which transmission errors are automatically detected and corrected.

error-detecting code: A coding system where errors in transmission are automatically detected as forbidden combinations of bits and are reported but not corrected.

error file: A file generated during processing to capture error messages or as a "catchall" for bad data.

error message: A printed or displayed statement signifying that the computer has detected an error condition.

ESDI: Enhanced Small Device Interface. A peripheral device interface standard that allows for high data-transfer rates.

executable: A program statement that gives an instruction (as opposed to labels or comments) for some operation to be performed.

executable code: Programming code that is immediately executable by the computer. See "machine code."

execute: To run a program or carry out an instruction.

expression: A source code combination of one or more operations.

facsimile: A device that will allow the telephonic transmission of hard-copy graphic images to another facsimile device. The images may be handwritten text, typed pages, pictures, or any black-and-white image that can be digitized by the facsimile device.

fault: A physical state or condition that may cause the failure of a functional system.

fault tolerance: The capability of those systems designed to continue to function satisfactorily in the presence of faults. Some systems achieve this goal with multiple processors and/or by duplicating data on different storage devices; if one storage device fails, all data are available from the other device.

fiber optic cable: A glass or fiber cable which transmits large amounts of data by light pulses.

field: In a record, a group of related characters treated as a unit.

file: A collection of records. Related information or data sets stored and manipulated as a single unit. Files can be stored on disks, in memory, on tape, or they can be sent to a printer.

finite: To have limits or an end; for example, a terminating number or sequence.

firmware: Software that resides in read-only memory. Refers to the execution of programs from read-only memory (ROM) or programmable forms of ROM, so that the programs seem to be as much a function of hardware as of software.

fixed-length record: A record that always has the same number of characters. Contrast with "variable-length record."

fixed-point arithmetic: A type of arithmetic where the computer does not keep track of the radix point, and all operands and results must be between certain fixed values (e.g., integers between -65536 and +65536).

floating-point arithmetic: A method of calculation that automatically accounts for the location of the radix point.

floating-point routine: A set of subroutines that perform simulated floating-point math on a computer system that does not have built-in floating-point hardware.

floppy disk: Also known as floppy diskette and flexible diskette. A removable magnetic storage medium composed of a thin mylar plastic, coated with a magnetic oxide material and enclosed in a semirigid protective plastic jacket. Floppy disks come in three sizes; standard 8 inches, minifloppy 5-1/4 inches, and microfloppy 3-1/2 inches.

flowchart: A diagram that uses symbols and interconnecting lines to show the logic and sequence of a specific set of operations. Flowcharts are used to break a complex problem into smaller, more manageable pieces.

font: A complete assortment of type of one style and size (e.g., Courier, 10 point).

footer: In word processing, refers to the text that appears in the bottom margin of each page of a document.

foreground processing: In a multiprogramming environment, refers to the automatic execution of programs that have been designed to have pre-emptive use of the system resources.

foreground program: In a multiprocessing environment, a program that has a high priority and therefore takes precedence over other concurrently operating programs.

format: 1. The logical and/or physical arrangement of tracks and sectors on a floppy diskette or hard disk drive. To be usable, a disk must be formatted so that the tracks and sectors are laid out in a manner compatible with the operating system in use. 2. To prepare a disk or diskette to accept data.

FORTRAN: FORmula TRANnslation. A second-generation, high-level programming language designed for mathematical and scientific use. It is generally considered obsolete, although it is still in limited use.

fourth-generation language: A programming language, usually based on a database management system, which uses a nonprocedural (menus) interface for programming rather than a procedural (source code) interface.

frequency: The number of recurrences of a periodic phenomenon in a unit of time. Electrical frequency is specified in hertz.

front-end processor: A dedicated communications computer connected to a host at the entry point for communications to the host computer. It may perform communications line assignment, data conversion, message-handling, or other data communications functions.

gateway: A device that interconnects two dissimilar networks or a network and a host system. Contrast to "bridge" and "router."

GB: Gigabyte. One billion bytes or one thousand megabytes.

generator: A program that constructs other programs to perform a specific task, such as a report generator.

graphics: Pictorial display of information. Illustrations or diagrams produced by a graphics application software program. Includes drawings created by using boxes, arcs, and circles as well as those done freehand.

handshake: The initial exchange between two data communication systems prior to data transmission. The first device sends a predetermined signal, then waits for a predetermined response from the other device. A handshake method (such as XON/XOFF) is part of a complete transmission protocol.

hard copy: A paper printout. The term "hard" is used because the copy output can actually can be held, as opposed to appearing on a screen display.

hard disk: A high-capacity, high-speed, rigid, rotating magnetic media random access storage device. Hard disks may contain multiple read/write heads and multiple disks and allow users to read, write and erase data and files.

hardware: The physical equipment of a computer system. The electrical and mechanical components. Contrast with "software."

hard-wired: Physically connected to a computer, usually by electrical wiring.

head: 1. A device that reads, writes, or erases data on a storage medium. 2. A special data item pointing to the beginning of a list or set.

header: 1. In communications, refers to the first part of a message containing all the necessary information for directing or decoding the rest of the message. 2. In word processing, refers to the text that appears in the top margin of each page of a document.

hertz: A measure of frequency or bandwidth equal to one cycle per second, abbreviated as "Hz."

hex: Hexadecimal. A number system with a radix of 16, in which the digits greater than 9 are represented by the letters A through F of the alphabet.

hierarchical: Organized according to levels.

high-level language: A programming language oriented toward the problem to be solved or the procedures to be used. In a high-level language, the instructions more closely resemble English.

high order: Pertaining to the digit or digits of a number that have the greatest weight or significance. In computer memory, the most significant portion of the word.

horizontal market: Pertains to software packages, such as word processing, spreadsheets, database management, communications, graphics, and accounting, which are generic in nature and can be used by most businesses with little or no modification. Contrast to "vertical market."

housekeeping: Computer operations or programsthat do not contribute directly to the desired results; in general, initialization, setup, and cleanup operations.

IC: Integrated Circuit. A microscopically sized electronic circuit containing multiple electronic components (which replace thousands of transistors) bonded on or within a single nonconductive ceramic or plastic carrier, usually a Dual In-line Package (DIP).

icon: A simple graphic image used to represent a physical device, program or activity.

impact printer: A printing device that prints by striking a raised element or pins against paper, using ink or a ribbon. Dot matrix printers and wheel printers are two types of impact printers.

implementation: 1. The process of installing a computer system, including choosing the equipment and software, installation, training, and establishment of procedures and policies. 2. The act of installing a program.

initialize: To preset a variable to proper starting values prior to execution.

input: Verb or noun pertaining to entering data, information or instructions from an external device into a system.

inquiry: A request for data from storage.

instruction: In a programming language, a meaningful command or formula that defines one operation with its parameters, if any.

intelligent terminal: A terminal that has a number of computer processing capabilities built into, or attached to, the terminal unit.

interactive: An application where an input or inquiry by a user causes an immediate action by the system in response to the input. Contrast to "batch processing" and "conversational processing."

interface: 1. A common boundary between two systems or devices. 2. A specific hardware or software connection or a connection between systems or devices. 3. The combination of hardware and software used to present the operating system and programs to the user. Example: the keyboard, mouse, and icon-driven display on an Apple Macintosh. 4. To make two devices or components capable of communicating.

internetwork: Two or more networks connected together by means of special bridge hardware and software in order to pass data and share resources as if they were a single network.

interpreter: A program that translates and executes each source code expression before translating and executing the next, without first translating the entire source program into executable code. Interpreters are slow, but provide greater interaction than a compiler.

interrupt: A signal that causes the computer to interrupt the normal flow of a program to perform some other function and then return to the program after the operation has been completed.

I/O: Input/Output. The process of moving data or information. For example, the process of transmitting data from disk to a printer for printing or out of disk storage for display on a workstation screen.

IRQ: Interrupt request. One of a group of signals normally associated with the request of an interrupt.

ISAM: Indexed Sequential Access Method. A method of high-speed data access in which one or more index files are used to point to data in a data file. The data file is organized based on a key field (a field or combination of fields that creates a unique identifier) and is usually not sorted. The index file contains a sorted list of copies of the keys from the data file and a pointer (a number) representing the position of the record from the beginning of the data file. This procedure allows the use of a simplistic sequential data file and yet gives high-speed reads by allowing a program to quickly find the exact record requested.

JCL: Job Control Language. The language used in batch control cards. JCL cards are used at the head and tail of a batch job to identify the owner of job, required system resources, etc.

job: A collection of tasks that compose a unit of work for a computer.

justification: The process of aligning, adjusting, or shifting digits or text to the right or left to fit a prescribed pattern (e.g., aligning a column of numbers by decimal point).

justify: To align the characters of a field to one extreme or the other. In right justification, the rightmost character of the item is written into the last or rightmost position of the field. To left justify, the first character is written in the leftmost position in a field.

KB: Kilobyte. A unit of measure for memory or disk storage capacity. Two to the tenth power or 1024 in decimal.

kernel: The set of programs that make up the most basic or fundamental functional elements of an operating system.

key: 1. The field or combination of fields that uniquely identify a record. 2. The field that designates the position of a record in sorted sequence. 3. A button on a machine such as VDT keyboards.

keyboard: A standard device composed of marked levers operated manually for the entry of characters onto a recording medium.

key data entry device: The equipment used to edit and prepare data for storage, such as key-to-disk or key-to-tape units.

key-to-disk: A data entry system where data are typed directly into a disk storage device from a keyboard, usually in a high data-volume environment.

key-to-tape: A data entry system where data are typed directly onto a tape storage device from a keyboard, usually in a high data-volume environment.

label: A name or identifier used in a program associated with a particular instruction, statement, message, value, field, record or file.

LAN: Local Area Network. A collection of computers connected and able to communicate and share peripheral devices (such as hard disks and printers) and possibly access remote hosts or other networks.

light pen: A hand-held pointing device designed to be pressed against the CRT, which senses the beam of light generated by the CRT and translates the time it takes for the beam to strike the sensor in the light pen into a coordinate position on the screen.

line printer: Prints one line at a time.

link: To connect one location with another for the transfer of data. To give access to a directory or device.

linker: A utility program that assembles the separately coded pieces of a program into one executable piece.

list: 1. To print every applicable item of input data. 2. To print out all the statements of a program. 3. A collection of ordered items. 4. The arrangement of data using index and pointers to allow for nonsequential retrieval.

listing: Any report produced on a printer.

load: 1. To move a program or data into memory prior to execution or modification. 2. A ratio of the amount of processing capacity of a system to the actual levels of processing utilization. 3. To initially install software onto a system.

loader: A utility program that reads programs into memory before execution.

log on: The process of establishing a connection with and confirming authorization to use a conversationally programmed computer.

logical file: One or more logical records.

logical instruction: An instruction that executes a symbolic logic operation, such as AND, NOT, OR or NOR.

logical operator: Any of the Boolean operators such as AND, NAND, OR, NOR, EXCLUSIVE OR and NOT. (Boolean math deals with logical operators in a binary format.)

logical record: The structure of a record as designed by the programmer.

logical value: A value that is either "true" or "false," depending on the results of a logical operation.

looping: Repetitively executing the same instruction or routine.

low-level language: A machine-dependent programming language, which is translated into instructions by an assembler; as contrasted to a high-level language, which is machine-independent.

LSI: Large Scale Integration. The method of constructing electronic circuits in which thousands of circuits can be stored on a single silicon chip.

machine code: Binary code directly executable by the processor. Normally created by an assembler or compiler. Commonly known as "executable code."

machine-independent: The ability of a program to run on a number of different computers made by various manufacturers or on the assorted models made by a single manufacturer.

machine language: The binary code that the language computer understands. Machine language programs require no modification to run on a computer. Also known as "executable code."

macro: A single symbolic statement that, when translated, results in a series of executable statements.

macro assembler: An assembler that allows the user to create and define new computer instructions, composed of existing instructions.

macro instruction: A source code instruction that is the equivalent of a specific number of machine language statements.

magnetic tape: Plastic tape with a magnetic surface layer on which data can be stored. Data are magnetically stored in an eight-bit or six-bit coding format. Magnetic tape is normally one-half-inch in width, with 9 recording tracks and a recording density of 1600 bits per inch (bpi) or 6,250 bpi.

magnetic tape cassette: A cartridge tape containing magnetic tape (normally one-eighth-inch wide tape) in such a fashion that the tape is inseparable from the container. It is similar to a stereo cassette tape cartridge.

mainframe: A large-scale, multiuser computing system and associated peripheral equipment. A mainframe typically has a 32-bit bus or greater, a performance greater than 12 MIPS, and a serving capability of more than 256 users. Mainframe computers typically require a dedicated support staff of multiple personnel for operation and programming.

main memory: Obsolete term, see RAM.

mass storage: Long-term storage of very high capacity, normally rigid disk, kept online to the central processing unit (CPU).

master file: A relatively large file of information, updated on a periodic basis and used as a primary source of reference.

matrix: A group of numbers, characters or symbols arranged in a rectangular, gridlike pattern. Individual elements can be referenced by their row and column locations.

matrix printer: A printer with a printhead composed of multiple pins that strike an inked ribbon to form characters on paper. Also known as dot matrix printer.

MB: Megabyte. A unit of measure for memory or disk storage capacity. Two to the 20th power or 1,048,576 in decimal.

memory: One of the essential components of a computer's central processing unit. Information and programs are actively processed in memory, which includes both read-only memory (ROM) and random access memory (RAM).

menu: A list of the options available at a particular place in a computer program. Menus allow users with little computer knowledge to utilize powerful systems.

merge: To combine the items of two or more sets that are each in the same given order into one set in the same order.

microcomputer: A small-scale, single-user or multiuser computing system and associated peripheral equipment. A microcomputer typically has a 32-bit bus or less, performance less than 5 MIPS, and serves fewer than 32 users. Microcomputers typically do not require a dedicated support staff for operation or programming.

microfloppy: A 3 1/2-inch floppy disk.

microprocessor: An integrated circuit containing the arithmetic, logic and control units of an entire central processing unit on a single chip.

microsecond: One millionth of a second.

millisecond: One thousandth of a second.

minicomputer: A medium-scale, multiuser computing system and associated peripheral equipment. A minicomputer typically has a 32-bit bus or greater, performance greater than 5 MIPS (but less than 10 MIPS), and is capable of serving more than 32 users (but fewer than 256 users). Minicomputers typically require a dedicated support staff of one or two personnel for operation and programming.

minifloppy: A 5 1/4-inch floppy disk.

MIPS: Millions of Instructions Per Second. A method of measuring processing power.

MIS: Management Information System. Data processing system designed to furnish management with information to aid in the performance of management functions and decisionmaking.

modem: MODulator/DEModulator. A hardware device that converts serial digital data from a computer to analog format to transmit or receive data by telephone lines. There are both internal and external modems.

module: One logically separate piece of a program that is identifiable when combined with other modules and independently performs a unique function. Modules may be written and tested separately, then combined together to form one complete program. A module is usually composed of multiple subroutines.

monitor: The video display unit on a personal computer. See "VDT," "CRT."

monitor program: A computer program that observes, regulates, controls, or verifies the operations of a data processing system.

MOS: Metallic oxide semiconductor. A transistor or capacitor in which a semiconductor material forms one part of the component and a metallic oxide forms the other.

motherboard: The circuitry and mechanical assemblies used to connect printed circuit cards, boards or modules. Main logic board.

mouse: A hand-controlled pointing device designed to move on a flat, horizontal surface. The mouse creates a digital signal indicating its position; the computer uses the signal to position the cursor in an analogous position on a display screen.

MTBF: Mean Time Between Failures. The average time a component or system is supposed to work without a failure.

MTTF: Mean Time To Failure. The average length of time a component or system works without a failure.

MTTR: Mean Time To Repair. The average time required to find and fix a fault in a component or system.

multicast: A message sent to a specific group of workstations on a system. Contrast with "broadcast."

multiplexer: A device that allows simultaneous or interleaved transmission of multiple signals over a single transmission medium.

multiprocessing: The parallel execution of multiple series of instructions by multiple central processing units (CPUs) under a unified control.

multiprogramming: The apparent execution of two or more programs at the same time on a single computer. Each program has individual memory space and peripherals, but shares the central processor.

multitasking: The ability of an operating system to allow separate processes to run at the same time.

multiuser: The ability of an operating system to allow many users (at separate workstations) to share a system's processing power and perhaps also to share data and peripherals.

narrowband: Used to describe transmission media that can transmit only low volumes of data. Contrast with "wideband."

native language: A computer language specific to one model of processor, for example, an assembly language.

NETBIOS: Network Basic Input/Output System. An interface designed by IBM to facilitate communication between its network architecture and the higher level application programs.

network: A system of interconnected computers that sends and receives data and messages via cable or some other such communication medium.

network communication: Data transmission between network stations. Requests for services and data are passed from one network station to another through a communications medium.

node: Any computer, terminal, workstation or communications controller in a computer network; a point at which terminals have access to a network.

non-impact printer: A printer that prints without making an impression on the page. Typical examples are laser, inkjet and thermal printers.

null: The empty set, literally nothing, as contrasted with a zero, which implies a lack of numeric value, or a blank, which implies a lack of alphanumeric value. A null is the "black hole" of data.

object code: Also known as an object program. Output from an assembler or compiler which is itself executable machine code or is suitable for processing to produce machine code. See "source code."

Octal: A numeric system with a radix of eight.

OEM: Original equipment manufacturer. A manufacturer who purchases components or equipment from other manufacturers and integrates them to produce a complete system for resale.

office automation: The application of computer and communications technology to office environments.

offline: Pertaining to equipment or persons not under direct control of, or in direct communications with, the central processor. Sometimes used to indicate batch processing.

online: Pertaining to equipment or persons under the direct control of, or in direct communications with, the central processing unit (CPU).

operating system: The resident master control program of a computer system. It controls the overall operations of the system and execution of all other programs.

optical scanner: A scanner designed to read text and/or graphic information from a printed page.

output: 1. Pertaining to a device, process or program (including monitors and printers) that allows data, information or instructions to exit a system. 2. Data on which the computer has completed processing.

overlay: A technique where multiple segments of a program are swapped in and out of the same auxiliary storage space at different times as needed by the execution of the program. Overlays allow the execution of a program much larger than physical memory.

page: A segment of a program or data of fixed length (usually 2K or 4K words), which has a constant virtual address, but can exist in any region of the computer's memory. See "virtual memory."

page frame: A computer location designed to hold a page of commands or data.

paging: A technique for moving programs and data from virtual memory to real memory.

paging rate: The number of page-ins and page-outs, per unit measurement of time, in a virtual memory system.

parallel processing: Pertains to the execution of multiple processes simultaneously or concurrently on multiple processors or channels. Contrast with "serial processing."

parallel transmission: A means of data transfer in which each bit of data has its own wire. All the bits in one byte are transmitted simultaneously rather than serially. Contrast with "serial transmission."

parameter: 1. A variable that can assume the properties of a constant. 2. A modifier to a command that changes the command's default action.

parity bit: Used in an error-checking technique known as parity checking. An extra bit added to a character or block of characters to ensure that either an odd or even number of bits are transmitted. Allows an error to be detected by checking the parity of the word.

Pascal: A third-generation, high-level programming language noted for its logical constructs and embedded modularity. Pascal is popular in schools and universities due to its syntax, which facilitates the teaching of structured programming.

patch: A small section of code inserted into a program to fix a problem or modify the program.

peripheral equipment (or device): In a computer system, any equipment distinct from the central processing unit (CPU) that may provide the system with outside communication, such as a modem, cathode ray tube (CRT), printer, or additional facilities, such as an external hard disk or tape drive.

physical record: A record whose characteristics depend on the manner or form in which it is stored, retrieved or moved. A physical record may consist of partial, entire or multiple logical records.

pixel: The individual dots on a computer screen; a picture cell; the smallest addressable dot on a CRT (cathode ray tube) or VDT (video display terminal). Letter, numbers and symbols consist of pixels arranged in a matrix.

plotter: A hard-copy output device, which uses movable colored pens to graph data.

pointer: An address or memory location that indicates the address or location of a data element.

polling: A serial interrogation of devices for purposes such as to avoid contention, to determine operational status, or to determine readiness to send or receive data.

port: 1. In hardware, connectors that allow two devices to be cabled together to allow communications. 2. In software, a memory address used to transfer information. 3. To rewrite a program in order to enable execution on a type of computer other than the computer on which the program was originally written.

post: To enter information into a record.

print spooler/despooler: A print spooler is a program that controls the way in which print files are stored to disk or memory. The despooler is a separate process that controls how print files are taken off the disk or out of memory and sent to the correct printer.

process: A systematic sequence of operations or thread of execution initiated in response to a specific request or instruction.

process manager: A software module that allows processes to run in an orderly fashion. Normally, process managers are a multitasking kernel that control multiple processes, passing control back and forth so that each process executes according to some established priority.

program: The smallest set of computer instructions that can be executed as a stand-alone unit; ordered to perform a particular task.

programming language: A language used to express computer languages. See "BASIC," "C," "COBOL," "FORTRAN" and "Pascal."

programmer: A person who designs, writes, tests and debugs computer programs.

PROM: Programmable Read-Only Memory. A memory chip that can be programmed by electrical pulses. Once programmed, the PROM is read-only. PROMs are purchased blank and programmed on a device called a PROM Burner or PROM Programmer.

prompt: A character or message from the software, which appears on the display screen and requires a user response.

protocol: A set of specified procedures or conventions used routinely for determining how and when to format and send data between devices.

query: An operation that invokes a response from a computer system.

query language: A language built into a database management system wherein the user can ask questions of the database in an "English-like" language.

queue: A temporary buffer in which the computer keeps a list of data in the order in which it was received (as in a waiting line).

radix: The base number in a numerical system; for example, the decimal system has a radix of 10.

radix point: The dot or symbol in a numerical system that separates the integral part of the number from the fractional part.

RAM: Random Access Memory. A memory into which data can be placed (written) and from which data can be retrieved (read). RAM is the amount of open memory a computer has available for running programs. RAM is temporary storage; turning off the electricity causes all data to be erased.

random access: An access method in which specific logical records are obtained from or placed into a file in a nonsequential manner.

read: The process that takes place when one device receives data from another.

read-only: A type of data protection that allows data to be read, but not erased or modified.

realtime processing: Pertaining to the processing of data from outside a computer quickly enough to affect the process creating the input data. Usually applied to systems that control some physical process such as a nuclear reactor. Contrast with "batch processing" and "conversational processing."

record: A collection of related information that is treated as one unit within a file.

recursive: Defining an item in terms of itself. A repetitive process that is dependent on the results of the previous repetition.

recursive procedure: A procedure that either calls itself or calls another procedure, which in turn calls itself (the first one) back.

re-entrant: The attribute of a program or routine that allows the same copy of the program to be used concurrently by two or more tasks (e.g., multiuser word processors).

relocatable program: The attribute of a program or part of a program that allows it to adjust its internal address references to allow execution after being moved from one location into another.

relocate: To move a program in and adjust its internal address references to allow execution after the movement.

remote: Refers to a connection that allows data to be sent and received across physical distances (normally using telephone lines or some other means).

report: Generally, any hard-copy data output; involves the grouping of information to assist the reader.

report generator: A program that converts stored data into a printed report.

reserved word: A word that holds special meaning to a language or operating system; a reserved word cannot be used in an application program.

reset: To return the components of a system to a known default state.

restore: To copy archived data back onto a system.

RGB: Red-Green-Blue, the primary additive colors. An RGB color monitor uses three separate electron guns, one for each color, to light up the screen pixels (color dots) for additional colors. Normally a high-resolution device.

ring topology: A network topology in which all nodes are cabled together in a ring. Every piece of data transmitted on the network flows through each node sequentially and is retransmitted by every node on the network.

RJE: Remote Job Entry. The introduction of a job through an input device that has access to a computer through a remote link.

rollback: A system that will restart the running program after a failure by taking snapshots of the data and programs at periodic intervals. At a restart, the system rolls back to start at the most recently stored snapshot.

ROM: Read-Only Memory. A type of memory used in computer hardware that is permanently programmed with one set of instructions. Data can be read from and used but cannot be written on or altered. ROM is not erased when the electricity is turned off.

router: The software that keeps track of, and directs packets to, other similar networks connected locally. Contrast to "bridge" and "gateway."

routine: A set of instructions for some specific purpose that has general or frequent use. Normally applied to segments of program code.

RS232C: Also RS232. A data communications standard for the interconnection of serial devices.

run: The execution of a program by a computer.

scanner: An optical input device that recognizes a specific group of visual symbols and translates the visual signals into digital signals.

scratch: An item, such as a tape or disk, or an area, such as a directory, file or volume, used as a temporary workspace.

screen: See "monitor."

scrolling: Moving the images of a display screen up or down.

SCSI: Small Computer Systems Interface. Standard interface for connecting peripherals to the microprocessor. Also known as "scuzzy interface."

sector: The smallest addressable portion of a disk surface. Sectors are used as location references in disk storage.

seek time: The amount of time, usually expressed in milliseconds, needed to move the read/write heads of a disk or tape drive to a specific position.

sequential access: An access mode in which files are searched serially from beginning to end to find a record and new records are written onto the end of the file. The records must be processed one after the other, based on the order in which the records were entered into the file. Contrast with "direct access."

sequential processing: The computer technique of performing actions one at a time, in sequence.

serial: Pertains to the handling of data or processes in a sequential fashion.

serial processing: Pertains to the sequential execution of multiple single processes or instructions (in the order in which the processes or jobs are stored or requested). Contrast with "parallel processing."

serial transmission: A means of data transmission in which all the bits are sent a bit at a time over a single wire. All bits are transmitted serially rather than simultaneously. Contrast with "parallel transmission."

server: A centralized piece of software that performs some action upon request from a network workstation (usually either naming, printing, storing, transmitting or mailing a file). The term "server" is sometimes used to refer to the computer that houses the server software.

share: To give access to a file, directory or a device to more than one process or user.

simultaneous processing: The execution of two or more processes or jobs at the same instant, usually on multiple processors. Contrast with "concurrent processing."

software: A generic term for all computer programs. A set of stored instructions and procedures that can be recalled as needed for the computer to execute. Contrast with "hardware."

sort: To divide records into groups according to specified logical patterns.

source code: A computer program in a human-readable (symbolic) form, which can be translated into a machine-executable format. Also known as source and source program. See "object code."

special character: A graphic character that is not a number, a letter or a space.

spooling: 1. The apparent simultaneous operation of multiple input/output devices because the system is using multiple buffers for queuing input and output. 2. The storing of data to disk or tape prior to processing.

spreadsheet: A mathematical formula processor, like a word processor for numbers. A spreadsheet program allows you to enter a series of mathematical formulas, also known as mathematical models, which the spreadsheet software will constantly calculate.

stand-alone: A self-sufficient computer system with its own peripherals, not connected to any other system.

star topology: A network topology in which all the nodes are connected by a separate and independent wire to a central hub, which is usually the network server.

storage device: A device or portion of a device capable of receiving data, retaining it for an indefinite period of time, and supplying it on command.

string: A sequenced group of characters or bits that is treated as a single item.

structured programming: A technique for organizing and coding programs that reduces complexity, improves clarity, and makes the programs easier to debug and modify. Typically, a structured program is a hierarchy of modules and each has a single entry point and a single exit point; control is passed downward through the structure without unconditional branches to higher levels of the structure.

subdirectory: Any directory that is below another directory in the hierarchical directory structure.

subroutine: Procedure that performs some task without returning a value. A series or group of related statements that performs some specific action and can be used at one or more points in a computer program; for example, a subroutine that transforms lowercase characters to uppercase. Subroutines can be either inserted where needed or stored in one place and linked to multiple calling routines.

subsystem: A unified set of processes or set of components that perform a specific set of tasks or services and are connected to and controlled by another system.

swapping: 1. In a time-sharing system, the process of either reading a program into memory from disk at the start of a time slice or writing it out to disk prior to execution of another user's time slice. 2. In virtual storage, the process wherein a page is brought from auxiliary storage and swapped for an active page.

switched line: Normally a phone line connected to the switched telephone network.

synchronous: A method of operation in which processing begins in response to internal events or to a clock-generated signal. Contrast with "asynchronous."

synchronous transmission: A technique of data transmission in which data are transmitted at a fixed rate, and both the sending and receiving devices use the same clocking signals to synchronize transmission rates. Contrast with "asynchronous transmission."

syntax: The rules governing the structure and grammar of a language.

syntax error: The breaking of a rule governing the structure or grammar of a programming language or operating system.

system: All of the equipment, personnel, material, procedures, documentation, and information that form a self-sufficient unit capable of attaining specified objectives.

system software: Those programs that directly affect or control hardware functions of a computer, such as an operating system or caching program. Contrast with "application software."

systems analyst: A person who performs analytical functions in defining computer-related problems. Systems analysts normally provide system definitions to the programmers who write the actual code.

table lookup: A technique that uses a known value as a pointer to an unknown value in a table.

telecommunications: A means of communication in which computers use telephone lines to transmit and receive information.

terminal: A device, usually equipped with a keyboard and some kind of display, capable of sending and receiving information over a communication channel.

throughput: A measure of the amount of work performed by a computer system over a given period of time and under a given job load.

time sharing: The use of a central processor for two or more purposes during the same overall period of time by sharing, in fractions of a second, the amount of time available with each job.

topology: The architecture of a network; the way circuits are connected to link the network nodes together.

track: A specified path parallel to the reference edge on a disk or tape where the data media move past a given read/write head position.

transaction: In batch or remote batch entry, a job or a step. In a system with time sharing, an exchange between a terminal and another device that accomplishes a particular action or result; for example, the entry of a customer's deposit and the updating of the customer's balance.

transaction file: A file containing temporary information, which is processed against a master file. Also known as a detail file.

translator: A program that performs translations from one language into another, such as a compiler.

transparent: Pertaining to information or activities that are not recognizable by a program, device or user.

troubleshoot: To detect, locate and eliminate errors in software or faults in hardware.

truncate: To drop off the least significant digits of a number series (to cut off a number at a certain position to the right of the radix).

TTL: Transistor-to-Transistor Logic. Normally refers to the transistor-level logic applied to digital circuiting.

turnkey system: A computer system that contains all the hardware, software, training and installation required to perform a given application.

upload: The transfer of data from a local host system to a remote system. See "download."

user: A person who uses a computer.

utility software: A computer program created for general support of the processes of a computer or a program created to perform everyday tasks (e.g., a diagnostic or sort program).

validation: The checking of data for correctness or compliance with applicable standards, rules and conventions.

variable-length record: A record having a length independent of the length of other records with which it is logically or physically associated. Contrast with "fixed-length record."

VDT: Video Display Terminal. A terminal with a cathode ray tube for video display and limited processing or editing capabilities, which is attached to a host computer (as opposed to a monitor attached to a microcomputer). See "terminal," "CRT," and "monitor."

vendor: A company that sells computers, peripherals, and/or operating systems and other software.

verify: To determine whether the transcription of data or other operation has been accomplished accurately. To check the results of keypunching.

vertical market: A highly specialized section of the software marketplace which requires a specialized understanding of the needs of the industry. Usually applied to a type of business or activity that has specialized software and hardware needs, but is limited in the total number of possible customers. Contrast to "horizontal market."

virtual memory: A system of managing RAM and disk space so that a computer appears to have more memory than is physically installed. Virtual memory utilizes a technique of swapping segments of programs and data, called pages, to and from a disk storage device. Virtual memory transparently allows the user to use a small area of fast random access memory (or "real memory") in conjunction with a large amount of slower disk storage (or "virtual memory"), while appearing to have a large, contiguous amount of memory in which to work.

virtual storage: A technique wherein two or more storage devices are combined to form what appears to the system as one large storage device. The size of virtual storage is limited by the maximum amount of storage addressable by the central processing unit (CPU) and the physical amount of storage available.

VLSI: Very Large Scale Integration. A semiconductor device with a very high density (up to 10,000 circuits) of electronic circuitry contained on a single chip.

voder: Vocal coder. A speech synthesizer.

volume: A physical unit of a storage medium from which data can be written and read.

wait state: The period of time during which the processor waits for accompanying circuitry, like memory, to complete other activities, such as memory refresh cycles.

WAN: Wide Area Network. A group of local area networks joined together and sharing the same overall protocols to enable sharing of information or peripherals.

wideband: A channel width greater in bandwidth than a voice grade (300Hz to 3,000Hz) channel. Normally associated with circuitry or cabling that can transmit video, audio and digital signals.

Winchester disk: A model name that has become generic for a hard disk drive. This fixed, rigid disk is sealed within the computer or within a separate peripheral, providing greater storage capacity and faster access than a floppy disk drive.

window: A section of the video display used for a special purpose.

windowing: A means of dividing the computer screen into several areas so that a variety of information can be displayed simultaneously.

word: A string of bits that is the same size as the smallest individually addressable element of memory and capable of being manipulated by the central processing unit (CPU) as one item. In microcomputers, usually 8 bits or 16 bits.

word processor: A program that allows the operator to create, edit, format and store text as needed in preparing reports, letters, etc.

workstation: A microcomputer or terminal attached to a multiuser system.

WORM drive: Write Once Read Many drive. Typically an optical disk drive where information is digitized and stored on removable media by a laser process.

write: To record data on a hard disk, floppy diskette or memory.

zero suppression: The elimination of zeros that have no significance to a numeral (i.e., 0000155 yields 155).

SELECTED REFERENCES

REFERENCES

Automated Court Management Information Systems Directory. U.S. Department of Justice, Bureau of Justice Assistance, 1988.
This *Directory* of more than 1,800 court management systems enables practitioners to identify and contact agencies with similar application and information-processing needs.

Catalog of Personal Computer Software Developed within the Federal Courts. The Federal Judicial Center, Innovations and Systems Development Division, February 1989.
The catalog was compiled to facilitate sharing of court-developed computer software among the federal court community. It also fosters the exchange of ideas among software developers, courts that are considering software development, and those that have completed the task.

The Criminal Justice Microcomputer Guide and Software Catalogue. U.S. Department of Justice, Bureau of Justice Statistics, 1988.
The *Guide* defines the basic components of microcomputers and describes a process for selecting a system. The document also lists more than 150 operational criminal justice software applications.

Datapro Management of Microcomputer Systems. Delran, N. J: Datapro Research, 1988.
A two-volume directory, this collection of articles provides subscribers with the latest information on more than 50 categories of microcomputer-related topics. In addition to hardware and software, the directory has sections on system selection and acquisition, planning for a computer, management and administration, and future systems and trends.

Evaluating Donor Systems — A Software Transfer Technique. Sacramento, CA: SEARCH Group, Inc., 1980.
This guide assists practitioners in evaluating software systems for possible transfer; it includes a methodology for identifying alternative systems, selection criteria, and descriptions used to narrow the search for proven, operating systems available for transfer.

How to Manage an Information Technology Project. Sacramento, CA: Office of Information Technology, 1988.
One in a series of guideline publications developed by the OIT, this 77-page booklet was designed to assist California State agencies with critical aspects of the management of information technology.

Prosecution Management Support Systems Directory. U.S. Department of Justice, Bureau of Justice Assistance, 1988.
Intended to assist prosecutors and their data processing support personnel in developing, enhancing or acquiring a PMSS. It provides detailed information on 140 automated Prosecution Management Support Systems operating across the country.

For further information contact:

National Criminal Justice Reference Service
P.O. Box 6000, Rockville, MD 20850
(301) 251-5500 or (800) 851-3420

Bureau of Justice Assistance
U.S. Department of Justice
633 Indiana Avenue, N.W.
Washington, D.C. 20531
(202) 272-6838

SEARCH Group, Inc.
The National Consortium for Justice
Information and Statistics
7311 Greenhaven Drive, Suite 145
Sacramento, CA 95831
(916) 392-2550

The Bureau of Justice Assistance

BJA *Announces its New Clearinghouse*

To fulfill its mission under the Anti-Drug Abuse Act of 1988, the Bureau of Justice Assistance (BJA) provides funds and technical assistance to State and local governments to control crime and drug abuse and to improve the criminal justice system.

In support of these activities, BJA has created the **Bureau of Justice Assistance Clearinghouse**, a component of the National Criminal Justice Reference Service (NCJRS). The Clearinghouse informs State and local criminal justice practitioners about BJA products and programs.

The following BJA publications, now available from the Clearinghouse, provide a wealth of useful information on some of the most critical issues affecting criminal justice and will be valuable additions to your professional library.

An Invitation to Project DARE: Drug Abuse Resistance Education, Program Brief. NCJ 114802. Free.

Building Integrity and Reducing Drug Corruption in Police Departments, Monograph. NCJ 120652. Free.

Drug Recognition Program, Monograph. NCJ 117432. Free.

Electronic Monitoring in Intensive Probation and Parole Programs, Monograph. NCJ 116319. Free.

Estimating the Costs of Drug Testing in Pretrial Services Programs, Monograph. NCJ 118317. Free.

FY 1988 Report on Drug Control, Executive Summary. NCJ 118277. Free.

FY 1988 Report on Drug Control, Full Report. NCJ 117435. $6.50. Call for ordering information.

Prosecution Management Support System, Program Brief. NCJ 117093. Free.

Reducing Crime by Reducing Drug Abuse: A Manual for Police Chiefs and Sheriffs, Manual. NCJ 113110. Free.

Treatment Alternatives to Street Crime (TASC): Participant's Manual, Training Manual. NCJ 116322. Free.

Treatment Alternatives to Street Crime (TASC), Program Brief. NCJ 116321. Free.

Treatment Alternatives to Street Crime (TASC) Resource Catalog, Resource Guide. NCJ 119847. Free.

Urinalysis as Part of a Treatment Alternatives to Street Crime (TASC) Program, Monograph. NCJ 115416. Free.

Wisconsin Drug Abuse Treatment Unit, Monograph. NCJ 120655. Free.

Write or call right away—limited quantities of these valuable resource documents are available.

Bureau of Justice Assistance Clearinghouse • Call toll-free 800–688–4 BJA(252)
or write: BJA Clearinghouse/NCJRS, Box 6000, Rockville, MD 20850

U.S. Department of Justice
Office of Justice Programs
Bureau of Justice Assistance

Washington, D.C. 20531