THE MINICOMPUTER SIMPLIFIED

An Executive's Guide to the Basics

Carol W. Brown

THE MINICOMPUTER SIMPLIFIED

An Executive's Guide to the Basics

Carol W. Brown

THE FREE PRESS
A Division of Macmillan Publishing Co., Inc.
NEW YORK
Collier Macmillan Publishers
LONDON

THE FREE PRESS
A Division of Macmillan Publishing Co., Inc.
866 Third Avenue, New York, N. Y. 10022

Collier Macmillan Canada, Ltd.

Library of Congress Catalog Card Number: 80-1031

Printed in the United States of America

printing number

2 3 4 5 6 7 8 9 10

Library of Congress Cataloging in Publication Data

Brown, Carol W.
The minicomputer simplified.

Bibliography: p.
Includes index.
1. Minicomputers. I. Title.
QA76.5.B776 001.64'04 80-1031
ISBN 0-02-905130-4

Preface

You must have said to yourself, "Is it time to consider getting a small computer for my business?" as you heard how popular they were getting and how prices are dropping. Maybe one or more of your friends have taken the plunge. Maybe a hungry salesperson has already found you and painted a picture that sounds too good to be true. Maybe a competitor has boasted of his tremendous satisfaction with his new system and how his business has improved. Maybe another friend suffered endlessly in getting his system going and told you a real horror story.

You want to look into the facts and make your own decision but have been put off and have felt ill at ease with the lingo and the novel concepts and the high-pressure salesperson and the strange-looking machines. You continue to postpone the investigation, hoping that at some point either computers will get easier or you will get smarter about them!

This book is intended to make you smarter about computers. It's been written by a person who's been through all of it, with lots of hands-on experience in the many aspects of matching minicomputer solutions with business needs. The information in this book will help you make educated decisions about the place for a minicomputer in your business.

I am grateful to a lot of people. Special thanks go to attorney Bob Blumberg for help with Chapter 14, on contracts. Thanks also to Gail Pyle for providing me with a friendly electric typewriter, the correction key of which I have worn to a nub. Bob Wallace at The Free Press provided encouragement when my confidence ebbed and was wonderfully patient and considerate. My husband, Cyrus, always believed I could do it and made it all possible in a myriad of ways.

Contents

CHAPTER ONE

Enough Computer Jargon to Get By

The purpose of this chapter is to give you enough computer vocabulary to get by. The terms I discuss will get you through a sales presentation without looking or feeling ignorant.

It is unfortunate for everyone involved that computers always seem to be discussed in a foreign language. It immediately puts the uninitiated at a disadvantage. Knowledgeable computer people use jargon as a shorthand way of communicating with each other. But even they have trouble keeping up with the acronyms and terms that are constantly being invented and bandied about.

Computer jargon comes in several varieties. First, we have acronyms, which are seemingly meaningless words made up of the first letters of meaningful words. Some acronyms are pronounced as words, such as ROM (pronounced "rahm"), which stands for "read-only memory," a term you can immediately forget. Other acronyms have their letters pronounced separately, as in CRT (pronounced "see-are-tee"), which stands for "cathode ray tube." (Cathode ray tube is a fancy name for something that you could consider a black-and-white TV set with a typewriter keyboard built into the bottom of it.)

Our friends at International Business Machines (you undoubtedly know that acronym) are inveterate users of acronyms. When the IBM salesperson comes to call, ask him to avoid using acronyms. The length of his visit may double, but you will save yourself a lot of confusion.

Another kind of computer jargon is the so-called weird word.

The one that most often strikes terror into the heart of the non-user is "byte" (pronounced "bite"). If you substitute the word "character" evey time you hear "byte" you will have no problem, as a byte is just that, a character. The word "byte" has four bytes in it. The name "Carol Brown" has eleven bytes, as the spaces between words of text are often counted as bytes, too.

Normally bytes are discussed when people are talking about how much storage space a computer system has. A computer that can store a million bytes is smaller than one that can store 5 million bytes. By the way, the nickname for one million bytes is "megabyte," which is abbreviated MB. When a salesperson says that a computer has 50MB, he means that the computer has the ability to store 50 million letters and numbers simultaneously.

Another kind of jargon is the one-letter word. The letter K is used a lot in computer talk. It means a thousand characters, the same kind of characters we talked about as being bytes. Thus, 64K means 64,000 characters. Usually, when salespeople talk about how many K a computer has, they are describing the size of its "central processing unit," also called CPU or "memory." The more K a computer has, the more versatile it is likely to be. Later we will discuss what happens inside the CPU.

A lot of other jargon consists of strange words used for perfectly normal things. Take, for example, the term "input-output device." This term is fancy language for a machine that can both *give* and *receive* information. Your mouth could be considered an input-output device, as it both accepts food and drink and produces speech for others to receive. Your ears and eyes, however, would normally be considered "input-only" devices. Other parts of the body would be considered "output-only" devices. The printer on a computer system is logically considered an output-only device, as it can only produce information and is unable to receive information from the user. A terminal with a keyboard attached to it can display information to an operator and accept information through its keys; thus, it qualifies as an input-output device. The acronym for input-output is I/O (pronounced "eye-oh"). Frequently you will hear the term "I/O devices" used to describe the entire group of input-only, output-only, and input-output devices that surround and serve the central processing unit by giving information to it from the user and returning information from it to the user.

Some words that are very familiar to us are used in specific ways in computer talk. These words represent concepts that are important to understand. Several of these terms are interrelated, and you'll need to feel comfortable when they are used by others.

Let's say that you had a list of customers with whom you had done business over the years. Computer people would refer to this list as the customer "file," meaning a collection of related information. In this case, each name shares with each other name the relationship of being your customer. Each customer name would occupy one customer "record." In other words, each customer would constitute a *record* in the customer *file*. A customer's record contains all the pertinent information about the customer, including perhaps name, address—street, city, state, zip code—and current balance. Each of these facts would be considered a "field." A field is a discrete fact, or piece of information, within a particular record. And each field is made up of bytes. Isn't it a little bit like "the frog on the bump on the branch on the log . . . "?

Files, records, fields, and bytes will come up as computer salespeople attempt to "size your files." In simpler words, they want to figure out how large a computer they can justify selling you. They will attempt to identify your major files, estimate the number of records in each, determine the likely fields and their size within each record, and eventually come up with the total number of bytes of storage required for the jobs you need done by the computer.

Let's use another example so that you can prove to yourself that you are comfortable with the terms "file," "record," and "field." An inventory *file* would normally contain information about all of the products a company carries. An inventory *record* within the inventory *file* would contain a lot of facts about one particular product. One of the *fields* within an inventory record might be the selling price of the product; another *field* might be its name.

Computer people often want to describe a field by its format or appearance, as well as by its contents. They may use the words "numeric" and "alphabetic" to do so. When they say a field is numeric, they mean that the field should contain only numerals, say, a dollar amount or a domestic zip code (many foreign zip codes use letters). Fields that have only letters in them are called alphabetic. Many fields have a hybrid format, containing both letters and numbers. Accordingly, they are called "alphanumeric" (or

"alphameric" for short). An example of an alphanumeric field is a street address, which typically includes a house number and a street name.

Why do computer people care whether a field has numbers, letters, or both? Well, there are some considerations involving the way numbers and letters are stored in a computer, but the best reason for caring is yet another acronym, GIGO (pronounced "guy-go"), which stands for "garbage in, garbage out." This is a frequently heard—and apt—phrase.

Once bad information gets into a computer system, it will almost certainly come out the other end, perhaps many times. It is in everyone's best interest to prevent bad data from getting in. And one of the ways to validate the information put into a computer is to test its format. As a down-to-earth example, if an operator is requested to enter a dollar amount but accidentally hits a letter instead of a number, the system should reject the entry and ask for re-entry. We will discuss this topic in much greater depth later on.

We have been discussing data here. "Data" is a term you will hear often in computer talk. You might hear, for example, "Programs and data are stored on the disk." Just what is "data" and how is it different from "programs"?

The data in a system consists of all the information about the customers, products, receivables, orders, payroll, and every other area the system processes. It is the *data* that is being ***processed*** for the user. The data in a system is the most valuable part of the system to the user. Visit a fire drill in a midtown Manhattan office building sometime and watch the people carrying disk packs containing all their data files out to the street! The physical machinery can be replaced easily; obtaining complete, up-to-date lists of your customers, products, and orders would take much more time and effort.

"Programs" are the instructions the machines follow to process a user's data. There are several varieties of program, which we shall discuss in greater detail subsequently. But keep in mind that without some sort of program in it, a computer is utterly and completely *stupid*. It is dumb to the point that it cannot add or subtract or print or display anything on a screen. It does not know the date or the time of day and can't do anything the least bit useful without a program!

This brings us to the terms "hardware" and "software," which no salesperson will ever fail to use. Hardware refers to all the

machinery that is involved with the computer. Hardware can be touched and seen and heard and watched. Some people even call it "iron." Software, on the other hand, is the general name for all of the programs that make the system operate. You cannot see software in action, as it is, in its simplest form, merely electricity coursing around inside printed circuit boards within the central processing unit. Because of its amorphousness, its intangibility, people call it software.

In general, there are two kinds of software, and it is helpful to be aware of the difference between them. "System software" comes with the machine, from the manufacturer, and is the general name for the set of programs that make the machine know *how to run itself.* Another name you will hear for this is the "operating system." Each manufacturer's operating system is different because the hardware being operated is itself different. Manufacturers like to boast about their operating systems and their ingenuity. True, the job of the operating system is to run the machine itself as efficiently and reliably as possible. Like a traffic cop, the operating system decides which user shall have control over which device. For example, if two users each ask for a printed report, it would not make sense for the computer to combine the reports into one! The operating system must grant to one user the use of the printer until his report is printed and place the other user in line to wait briefly. So the operating system has to be a fairly smart program, well designed and well written, as it becomes the boss of the entire system. Indeed, the quality of the operating system has a great influence over the way a given machine performs. Remember that the operating system is software but usually comes along with the hardware from the manufacturer.

The other kind of software is "applications software." This is the general name for all of the programs that perform the specific jobs to be done in your business. Examples of applications software are programs that perform order entry, accounts receivable, general ledger, and accounts payable functions.

If all businesses performed their functions the same way, the job of installing computer systems would be infinitely easier! Obviously, this is not the case. Businesses operate in unique and diverse ways. Therefore, the computer programs they need to assist them may be very different from one company to the next, even when the companies are direct competitors. Applications software is generally unique at least to the industry involved, if not to the

company. Remember, applications software is the group of programs that perform your business applications, no matter where the programs come from.

The concept of a language is not strange to most of us. Even if we haven't actually traveled abroad, we understand that if we go to a foreign country, we might either speak that country's language or go without communicating verbally, as the people there will not necessarily understand our language. The same is true with computers. Computers have a language of their own, which we do not understand easily. In general, the language a computer understands is called its "native language" or "machine language." Each machine has its own machine language. Were we to see machine language, it might be all ones and zeros or strings of meaningless letters and numbers. With some effort and quite a bit of knowledge, we could decode the machine language, but it is very difficult and wasteful of time and effort to try. The machine doesn't understand our language either, although over the years we have come closer and closer to that idyllic state. Programming languages have been developed so that people can give instructions to computers in people-language (or as close as possible to people-language). Once people give their instructions, a translation process takes place. A special program called a "compiler" tries to translate the people-language into machine language. As with any language, in order to be understood, one must follow rules of logic, grammar, and syntax. Computer languages have their own vocabularies and rules of grammar, logic, and syntax. If the writer has obeyed the rules, the compiler will be able to make a successful translation of the instructions into machine language. Interestingly enough, the compiler itself is also a program, whose job it is to turn the user's program instructions into something the machine can understand. The concept of a special language for a computer is not difficult. Think of a translator at the United Nations translating a speech for the delegate from the country called Computerland. Each salesperson you see will tell you what language his computer uses and boast of its wonders. We will continue the discussion of the various computer languages in Chapter 4.

With the advent of minicomputers, people have begun to give more consideration to the nature of "interactive systems." What is an interactive system?

In the early days, computers were so expensive that you couldn't have one in the middle of your office. The computer was

located in a central location so that all the users (who were helping pay its enormous bill) would visit it, bringing their own work to be done. Such systems operated in what we call the "batch mode." They would do your job for a while, then mine, then someone else's. While your job was running, mine was not. In fact, my data existed in the machine only when my job was actually running. When my job was over, I would take my data away, and you would put your data in the machine and run your batch of work. Each of us would have up-to-date totals right after running our jobs, but not again *until the next time* we ran a job; so we got our information from our batch system at intervals. Generally, system designers tried to set the size of that time interval to be consistent with the costs and benefits inherent in producing the information. For example, if you were producing quarterly financial statements, it was fine to accumulate information for three months and then visit the computer. But can you imagine running an airline reservation system that way? The businesses that really had to have information *quickly* (it's called "real-time" when it's almost instantaneous) began to spend money on developing interactive systems. These systems gave their users almost immediate responses to requests and transactions. If a ticket agent reserved a seat on a given flight, the very next agent to query that flight would be shown one less available seat. You can see how different this sort of processing is from batch processing. It's an important distinction to remember, as now real-time, or interactive, processing is available to minicomputer users and may be something you will want to consider carefully.

This next section will be fun to read since it is going to cover phrases you don't need to bother with! Human minds are sometimes perverse, however, and you will probably remember the meanings of these terms because you don't have to.

A "bit" usually gets mentioned along with the ubiquitous byte. Bit is a shortened form of "binary digit." Suffice it to say that there are eight bits in a byte; their coded makeup tells the computer what character the byte is. You should not have to worry about bits. But, after all, the computer has to know one byte from another somehow, doesn't it?

Another weird word that sounds like something from the Wild West is "baud," often used in the phrase "baud rate." Baud means "bits per second" and refers to the speed at which devices can exchange information over communications lines. The whole subject

can be dropped until your computer system grows to the point where you want several locations talking to each other.

A few other terms worth ignoring are "microsecond," "millisecond," and "nanosecond." They are all names for unbelievably small fractions of seconds within which computers can perform discrete operations. Salespeople will try to impress you with how fast the computer can go. You are much better off asking down-to-earth questions like "How many invoices can I print in an hour?" or "How long will I wait for a response to my inquiry to the system?" Frankly, you really shouldn't care whether the inside of the computer is full of hamsters on treadmills or mirrors or gnomes so long as it does *what you need done.*

A final word on the subject of jargon. You wouldn't be totally wrong if you had the suspicion that computer jargon is a conspiracy to make computers sound more difficult than they really are. Some of the world's biggest braggarts are people who have the least to claim for themselves. In the same way, some of the people who use the most computer jargon really know the least about computers. Don't be afraid to ask for plain English from salespeople, consultants, programmers, or anyone else who starts to spout computer talk. And add a little extra skepticism to your judgment about anyone who tries to wow you with a lot of jargon. They wouldn't do it if they really cared whether you understood what they were saying.

CHAPTER TWO

Just What Is a Minicomputer?

Within our short lifetimes, computers have been invented, refined, and dramatically improved.

Few of us were able to view the ENIAC, developed in 1946 at the University of Pennsylvania School of Engineering. That computer was used for calculating ballistic trajectories. The ENIAC filled several rooms with its eighteen thousand vacuum tubes. In addition to the electrical and reliability problems caused by the sheer number of tubes, the amount of heat generated, which had somehow to be dissipated, was enormous. The ENIAC became the UNIVAC, also enormous, inefficient, and unbelievably expensive.

In those early days, only big, well-funded research sites and big, well-financed companies (and of course the government) could afford computers. They were placed in specially air-conditioned rooms with special electricity and strengthened floors raised to allow hundreds of snakelike cables to run beneath. White-coated technicians hovered around the equipment much like worker bees attending their queen. Some technicians actually were located within the computer itself! Programmers agonized over tortuous programs written in machine language that were impossible to create and even worse to test and correct. Somehow the rest of the world got the idea that these machines possessed extrahuman intelligence, calling them "giant brains" and worse. Books like Orwell's *1984* further encouraged the fear of computers and their Big Brother potential.

Computers did get cheaper, smaller, and easier to operate in

the fifties and early sixties. Most large companies and some medium-sized ones obtained their own computers. This class of computer received the name "mainframe computer" or simply "mainframe," as they usually constituted the company's centralized data processing facility.

Most data was prepared away from the computer, in the form of punched cards. Operating departments of the company would submit large batches of paperwork to employees called "keypunch operators," who would "type" the relevant data onto punched cards. Those cards would then be processed by the computer in large batches. The resulting reports would be returned to the operating department sometime later.

The machines themselves performed one job at a time. The vacuum tubes were gone, replaced by transistors. The mainframes were still tended by technicians, now without white coats. The equipment still lived in heavily air-conditioned rooms with raised floors but was easier to program and operate. Memories were larger, and communication with the user improved. Programmers of this era were drawn to their vocations by a love of machinery and problem-solving and were not known for their normal or socialized behavior. In fact, at one point long hair, a beard, sandals, and an Indian belt around the forehead seemed to be the programmer's uniform!

The term "minicomputer" is extremely difficult to define in specific terms. The term used to mean small. In the early days of computers it meant home-built, such as a bright engineer might construct in his basement in his spare time, or special-purpose, such as used in specialized industrial applications. Indeed, the early minicomputers were developed for very specialized industrial and factory tasks, for example, controlling an oven or keeping a chemical solution at a constant acidity. Since the early computers themselves were so large, these special-purpose computers seemed small in comparison and thus received the "mini" designation.

The dividing line between a minicomputer and a mainframe computer has itself been getting fuzzier over the years as both have become substantially larger. Today's minicomputers are frequently larger than yesterday's mainframes! Suffice it to say that defining a computer as a minicomputer based upon its memory or disk storage capability has turned out to be a short-lived solution. Similarly, calling computers below a certain price minicomputers fails as a

workable definition because the prices of the components of computer systems have fallen dramatically since the early days. To illustrate this point, IBM has estimated that calculations costing $1.26 to perform in 1952 can now be done for seven-tenths of a penny!

For a time, the definition of minicomputer had an unspoken element of "not from IBM," in addition to the "smaller than a mainframe" aspect. This was because IBM really did not enter the interactive minicomputer marketplace until the late seventies. For a good ten years, the other manufacturers had the minicomputer marketplace to themselves. Now that IBM, too, has strong offerings in this field, the entire minicomputer movement has increased credibility and would appear to be strongly entrenched in the computer scene.

Perhaps the only definition that will stand the test of time is that a minicomputer is a computer system that does not require a raised floor! Today's minicomputer normally does not need the extensive and expensive air-conditioning and electrical arrangements of the mainframes of the past. Often they do not require their own computer room but are located in the departments they serve. They are designed to co-exist well with the people who use them and function at the same temperature and humidity as their users! They are substantially smaller than the early mainframes, as the vacuum tubes were replaced by transistors, which in turn gave way to semiconductors. Indeed, now "microcomputers" physically smaller than minicomputers are common; their entire processing logic and internal memory are contained in a single "chip" smaller than your fingernail. Entire computer systems no larger than a desk or a desk top are common.

The people who operate today's minicomputers are no longer professional keypunch operators but normally are ordinary office staff members, perhaps the very people who performed the same applications manually before the minicomputer system appeared. Most often data is not keypunched into cards but is entered directly into the system via a terminal, or CRT. Normally the work is not batched but is entered as it occurs or is received, at which time the system's files are immediately updated, as in real-time processing. The systems are commonly able to run more than one job at a time, and the jobs are designed to have interactive dialogue with the user. This means that the operator's entries are requested one at a

time in simple English terms and are tested for validity. Should the entry be invalid for some reason, the operator is informed of that fact, again in simple English terms, and asked to reenter the data. In this way, the operator can be led through rather complicated entry sequences and can correct errors on the spot.

In many ways, today's minicomputers should be considered office tools like copiers and typewriters. The term "small business computer" probably makes more sense in defining the function of today's minicomputer in the business environment.

Moving the computer out of the computer room and onto the user's desk has had some dramatic effects on the organization and management of many companies. When computers were terribly expensive and difficult to operate, there was no choice but to centralize the data processing function within the business in order to operate efficiently. One of the results of extensive centralization was the creation of computer "empires" within companies. The user departments whose data was being processed had to submit their work to the centralized facility and wait to get it back. Conflicts frequently arose in part because many users were not good at communicating their needs and desires to the computer facility. On the one hand, they were unfamiliar with data processing concepts and practices; on the other hand, the computer department was not brimming over with patience and spoke that dreadful computer jargon. The managers of computer facilities often judged their own success by the amounts of new and sophisticated equipment they could amass for their operations. Top management, wringing their hands over ever increasing expenditures that they could neither understand nor prevent, exacerbated the trouble by failing to set computer processing priorities. In some companies where the costs of the central computer facility were "rebilled," or charged against the user departments, warfare broke out. Small wonder, then, that minicomputers that could belong to single departments and be under local control have become very popular indeed.

Large companies that have decentralized their computing resources have been able to delegate to their managers clearly defined areas of responsibility and accountability. And they have been able to analyze profits and losses in these areas once they were not dependent upon a central computing facility.

We can see that the minicomputer has really taken the opposite role from that of the giant, decentralized mainframe.