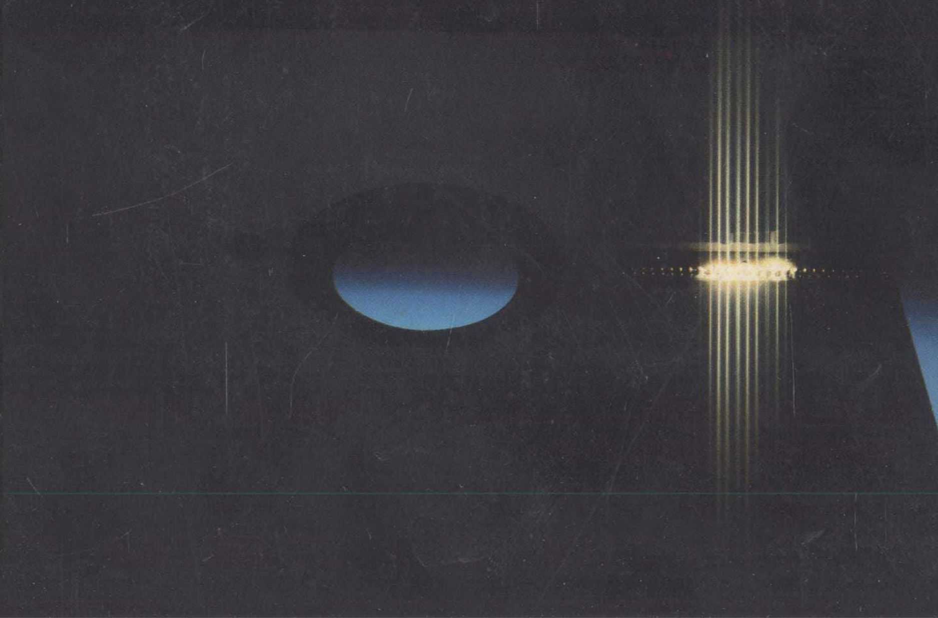
Microcomputer Disk Techniques

by Paul Swanson



Library of Congress Cataloging in Publication Data

Swanson, Paul
Microcomputer disk techniques.

Includes index.
1. Microcomputer—Programming. 2. Data disk drives. I. Title.
QA76.6.S916 001.64'42 82-1313
ISBN 0-07-062582-4 AACR2

Edited by Bruce Roberts.
Cover Design and Production Supervision by Ellen Klempner.
Cover Photo by Paul Avis.
Drawings by Eddie Germano.
Production Editing by Tom McMillan.
Production by Mike Lonsky.
Typesetting by Debi Fredericks, Donna Sweeney, and Valerie Horn.
Printed and Bound by Halliday Lithograph Corp.
Arcata Company
North Quincy, Massachusetts

6

5

Microcomputer Disk Techniques

by Paul Swanson

BYTE/McGraw-Hill, 70 Main Street, Peterborough, NH 03458

ACKNOWLEDGMENTS

I would like to thank my wife, Eleanor, for her help in proofing and in selecting illustrations for this book. I would also like to thank my mother-in-law, Petroneli Sanda, not only for her help in the proofing, but also for the glossary, as she selected each word in it.

INTRODUCTION

Over the past few years, the cost of purchasing a computer system has plummeted, and, as a direct result, many people have purchased computer systems for personal use at home and many others have bought systems for small business. But because the cost of the software, or programming, has not dropped that much, if at all, many of these new computer owners have decided to program the machines themselves. These people have had varying success, but most have discovered the difference between an amateur programmer and a professional one. The net result is thousands of very under-programmed computers, computers whose programs are not nearly sophisticated enough to make full use of their powers.

The inexperienced or amateur programmer needs to have years of instruction and experience to know enough to increase the sophistication of the programs. If the computer is there, the experience is possible. But adequate instruction is not that easy to get. The various manuals and other instructional materials that come or can be purchased with the computer do a very good job of teaching the basic concepts behind the language the computer uses, usually Beginner's All-Purpose Symbolic Instruction Code (BASIC). By trial and error the amateur programmer can usually become quite proficient at programming the basic model of the computer. The real confusion begins when the amateur adds a disk drive. Adding a single drive to a computer transforms it into a system with greatly increased power and flexibility, particularly in comparison to that of a tape cassette player. This increase is due to the availability of fast data storage and retrieval that the disk provides. The problem is that, although the basics of the language are described well in the manuals, the various techniques required to make effective use of the disk are not there.

By discussing disk drives with amateur programmers and observing what literature and programming aids are available to them, I have learned that the average owner of a personal computer knows how to handle the disk in a relatively primitive way. The fault is in the items that are available. This book will show some fairly sophisticated methods of using disks in microcomputer systems.

The first part of this book is basically a long introduction to disks. In it are descriptions of the disk itself and how the computer physically uses it and how the operating system in the computer is written to make using the disk simpler. The rest of the book is divided into four sections, each section building on the information presented in the preceding one.

The first section starts where most disk manuals stop. It explains the random-access file, one of two types of files available on almost every disk-based computer. The random-access file is the type of file responsible for the increased flexibility the disk affords. The other type of file, the sequential file, is also used in the book for some required functions, but sequential files are like tape files (which, technically, are one form of a sequential file) and have the same basic restrictions. The contents of the first section represent the approximate limit most personal computer owners achieve using the learning material that comes with the computer system.

Section II deals with three different types of key files. Each of the three is used in professionally written programs for different purposes. Each one has its strong and weak points, so all should be examined to see which of the three types to use. The key file is a more sophisticated use of the random-access file than is explained in Section I. It has an index, which is a look-up table. The data is stored and retrieved by name instead of by number.

Section III adds to the programs developed in Section II. It explains what parameter-driven subroutines can do and increases the flexibility of a key-file system developed in Section II.

Section IV deals with the program itself. It shows how to effectively structure the program and the data to increase a system's usability. This section also deals with such subjects as using program overlays to segment the functions of a program system.

This book is not meant for the professional programmer, who either is familiar with most of the information presented here or uses larger computer systems that have more automatic, built-in methods for handling the disk. Instead, it is meant for the amateur programmer who is serious about learning how to effectively use a disk drive on a microcomputer system.

Microcomputer Disk Techniques


EDDIE GERMANO

TABLE OF CONTENTS

PART ONE

CHAPTER 1

THE DISK AND DISK DRIVE

Deciding whether to add a disk drive to an existing computer system or to buy a new computer system with disks is confusing. Almost all of the microcomputers can have disk drives, but many of them give you choices that, unless you know a lot about computers, mean little. And although disks are made of very simple parts, there are many of those very simple parts. That is complicated.

A disk is often compared to a phonograph record, but this analogy is flawed. First, a record has two grooves on it—one on each side. A disk does not have grooves. It does, however, have something comparable—tracks that are the magnetic equivalent of grooves. The analogy fails here in that the disk has several concentric tracks instead of the single spiral pattern that is on a record. Also, while records cannot be erased, disks can, and information can be recorded over whatever was there, much the way a tape recorder can record over the former contents of a tape. Although there is really no analogy to the disk drive that explains it, you can explain the disk simply enough without an analogy.

Most disks are metal or plastic platters coated with a material that can be magnetized. Recording tapes are coated with a similar material. But unlike the tape recorder and player, the mechanism used to read and write the information on a disk can select a portion of the surface by a mechanical index. The logic circuits that control the disk drive, which is the mechanical part, can select how far the reading arm goes out across the surface of the disk by number. Each number on this index corresponds to an area of the disk that takes the shape of a circle. There are many of these circles on the disk, each one having its own index number. These circles are known as tracks.

A track may contain thousands of characters of information. In order to handle this information more easily, the logic that controls the disk can break down the information in each track into smaller units called sectors. Every track on a disk will have the same number of sectors as every other track on the same disk. Some disks are *hard sectored*, which means the tracks on the disks are mechanically divided into sectors, usually by a series of equally spaced holes on the disk. Others are *soft sectored*, which means that there is only one hole on the disk. This hole, and one selected hole on a hard-sectored disk, are called timing holes. A timing hole lets the logic handling the disk know where the tracks begin. Usually, the disk interface logic circuit handling the disk requires the computer to give it the number of the track and sector along with a code telling it whether it should read or write and the information to write. But remember, most of the physical properties of the disk are invisible to the computer program and to the computer programmer.

Several types of disks are on the market. They range in size from the small disk to the large hard disk. All of them use the same method of reading and writing.

The types of disks fall into two general categories. There are the *hard disks*, which are, as the name implies, rigid platters. On hard disks the mechanism used to read and write (called the read/write head) does not make contact with the platter. Instead, it literally flies over the surface on a bed of moving air continuously pumped across the disk platter. This read/write head flies at such an incredibly low altitude that a smoke particle will not fit between it and the surface of the disk platter. Some hard disk drives contain platters that are not removable but are permanently fixed inside the drive, while others have the disk platters enclosed in removable hard plastic cartridges. The most popular hard disk drives are a combination, having one platter fixed in the drive and another fixed in a removable cartridge. This type is quite logically called a *fixed/removable* disk drive.

The other general category of disks, much more common with microcomputers, is the flexible diskette or floppy disk. This disk is flexible, although your testing this quality will soon destroy the disk. The main difference between this type and the hard disks is that, on the flexible disk, the read/write head is in contact with the disk platter when it is in use. Although this allows the circuitry driving the mechanics to be a lot simpler, it also allows the disk platters to wear out. While hard disks will almost never wear out, floppy disks will.

There are two basic sizes of floppy disks available. The older of the two forms is 8 inches in diameter. The other, usually called the minifloppy, is 5¼ inches in diameter. Both are enclosed in an envelope that should never be removed. This envelope protects the surface of the disk against dirt and contact with materials that may mar the surface. It also has a material on the inside that constantly wipes the surface of the platter so that it will not accumulate dust particles. If dust were to accumulate, the read/write head, pressing on the dust particles, could scratch the magnetic material off the surface, making the platter unreadable. Even with this protection, the surfaces do wear out. Their life spans are usually fairly long, although this does seem to depend upon the logic circuits that read and write the information. If these circuits are not very sensitive, a disk that would work on a more sensitive system may not be readable. Unfortunately, many of the microcomputers on the market, because of efforts to keep costs down, do not have the best access circuitry and are not sensitive enough. Chapter 3 contains some information on the proper care of floppy disks that will help you make yours last a little longer.

There are also many differences in the individual drives that can further complicate the operation. The main difference is alignment. This is more of a problem on hard disks than on floppy disks because on hard disks data are stored in a more compressed form than on floppy disks. But floppy disks can also have alignment problems. Alignment refers to the index that allows the computer to find a given track. The index is the distance that the arm moves out onto the disk. If this index is off, the disk is said to be out of alignment. This is relative. Depending on how far out of alignment the disk drives are in relation to each other, a disk that is out of alignment might be able to read information that it stored itself, but may not be able to read information stored by another disk drive. There are standard alignment distances that are used so that most disks can read data stored by other disks and the alignment, if off, can be adjusted.

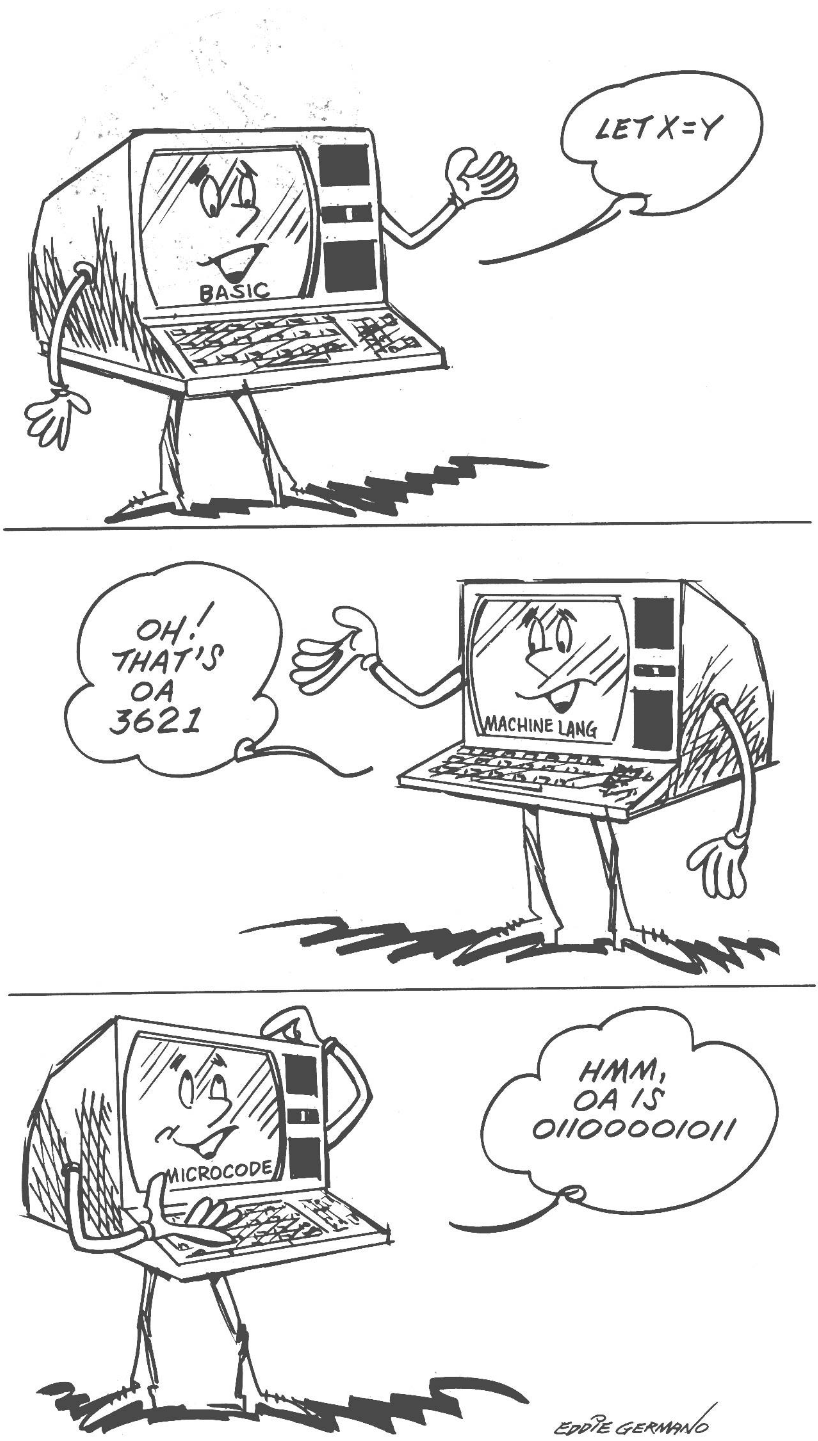
The idea of floppy disks wearing out brings me to backup, another important topic. Backup is an extra copy of software, normally kept on file in case the original program is damaged or lost. One rule on backups is that they should not be used just when the main disk becomes unusable. In that case a copy of the backup disk should be used. That way, the backup disk stays on file and is not destroyed. Backups of more important information should be made often. But because the information on a disk can be lost over time, even when the disk is not being used, a backup disk should be periodically refreshed by again copying it from the main disk. Refreshing also helps avoid reliability problems with disks, particularly microcomputer disks. Even systems that have good reliability records can have a disk that becomes, on occasion, unreadable. But always keeping two or three copies of all-important disks guards against this problem, too.

Another basic difference between the floppy and the hard disk is that some hard disks can store more data than any floppy disks. Here are some details. Both hard and floppy disks are divided into tracks and sectors. Although the sizes of the tracks are different on different types of disks, the sizes of the sectors are generally the same. All measurements are usually in bytes and all information used in most computers is stored in bytes. One byte is composed of 8 binary digits called bits (a bit is either a 1 or a 0, which are the binary digits) and stores a value between 0 and 255. On the disks, one sector is usually 256 bytes. The track is no longer important in that the computer will take care of the division into sectors. Hard disks can store 1,000,000 bytes of information (1 megabyte) and up while floppy disks usually store 1 megabyte or less.

The floppy disks, being the most commonly used disk on the microcomputers, require more discussion. There are several different types of floppy disks. The basic type is the single-density, single-sided floppy disk. The density refers to how compactly the data is stored. There are single-density and double-density drives. The double-density drives, as their name implies, store twice the amount of data as the single-density equivalent. Double-sided floppy disks simply use both sides instead of just one side. This requires two read/write heads in each drive. This also doubles the amount of information that can be stored on the floppy disk. The 8-inch floppy disk can have a capacity of about 250,000 bytes if it is single-sided and single-density. The capacity increases if the disk is either double-sided or double-density to about 500,000 bytes. If it is both double-density and double-sided, the capacity is about 1 megabyte. Minifloppy disks have similar options. The basic, single-sided, single-density minifloppy disk stores about 90,000 bytes. Dual-density and double-sided minifloppy disks are newer to the market and are not as widely available, but a minifloppy disk that is double-density or double-sided will store about 180,000 bytes, and one that is both double-density and double-sided will store 360,000 bytes.

The type, size, and number of disk drives required depend upon the intended application. Two minifloppy disks on a system with 32K (K stands for kilobytes—1 kilobyte is 1024 bytes) of memory create a very powerful personal computer system and will handle many small businesses. A single-drive system has its uses, but it also has many built-in handicaps. Backups, for example, are not as easy to make. To make a backup on a one-drive system usually requires removing and inserting the old and new disks several times. This also adds some extra wear to the disks and to the disk drives. Most users I know who started with a single drive system have purchased a second drive. A one-drive system is not a bad system on which to learn about disks, but if you purchase a one-drive system or add only one disk to a system that had no disks, expect to purchase another drive later. Planning it that way may be a good way to spread out some of the expense, in that the purchase price of the second drive is delayed until you learn how to handle the disk.

Knowing what a sector is helps in designing the programs, but the other physical information about the disk is only background. Most of the microcomputers use a dynamic allocation type of system, which is explained in the next chapter.


BASIC
LET X=Y
MACHINE LANG
OH! THAT'S OA 3621
MICROCODE
HMM, OA IS 01100001011
EDDIE GERMANO

CHAPTER 2

DISK OPERATING SYSTEMS

Understanding the physical properties of the disks is only a beginning. Another important thing to know about is the operating system, the program built into the computer that supervises the disk's activities. It is this part of the computer that understands all of the commands listed in the manual.

To understand the disk operating system, you may need an overview of all of the systems operating and all the layers of languages present in the computer. At the lowest level is an understanding of the internal workings of the microprocessor chip itself. In that one chip is a full central processing unit (CPU), the brain of the computer, and some memory of its own, not available to the computer system in general. The CPU in the microprocessor is programmed to understand the microprocessor's machine language. When a machine language command is executed by this chip, it is read into the chip's internal memory and there triggers a series of commands inside the chip. These are simple commands. One command may be to get the contents of a register (a register is a temporary storage place where 8 bits can be stored), or to place a byte (8 bits) into a register. This language is called microcode and to computers it is the lowest-level language. A machine language can be used in the memory that is outside of the chip, and this memory is referred to as the computer's memory. BASIC and the disk operating system are both written in this language. Usually, an assembler is used to program the BASIC interpreter and the disk operating system. An assembler is simply a way of using a series of letters (called *mnemonics*) to represent the machine language commands. This makes it easier to program the machine language by making the computer do some of the computations involved in the programming.

Because the computer is usually precoded to accept some predetermined machine language, the computer's user cannot use the microcode. The machine language, however, generally is available to the user. BASIC is an interpretive language, which means that the computer looks at each