PERIPHERAL DEVICES

IVAN FLORES, Ph.D.

IVAN FLORES

Computer Consultant
and Professor of Statistics
Baruch College
City University of New York

PERIPHERAL DEVICES

Prentice-Hall, Inc., Englewood Cliffs, N.J.

The author and publisher wish to acknowledge the contribution of the International Business Machines Corporation. The following illustrations were either adapted or reproduced from material supplied by I.B.M.: Figures 5.1.1, 5.1.3-5.1.10, 6.5.1-6.5.3, 6.5.4, 6.5.6, 8.1.2-8.1.9, 8.2.2, 8.2.3, 8.3.1, 10.1.1, 10.1.2, 10.2.1-10.2.3, 10.3.1, 10.3.2, 10.4.1, 12.1.1-12.1-6, 12.2.1, 12.2.3-12.2.7, 13.1.1, 13.1.2, 13.2.1, 13.2.2, 13.3.1, 13.3.2, 14.1.1, 14.1.2, 14.1.6, 14.3.1, 14.3.2, 15.1.1, 15.2.1-15.2.3, 15.2.5, 15.3.1; Tables 5.1.1, 5.3.1, 5.4.1, 5.4.2

Library of Congress Catalog Card Number: 73-9253

0-13-657205-7

Current printing (last digit):

10 9 8 7 6 5 4 3 2 1

Printed in the United States of America

Prentice-Hall International, Inc., *London*
Prentice-Hall of Australia, Pty. Ltd., *Sydney*
Prentice-Hall of Canada, Ltd., *Toronto*
Prentice-Hall of India Private Limited, *New Delhi*
Prentice-Hall of Japan, Inc., *Tokyo*

Preface

At this writing, there is no book entirely devoted to PERIPHERAL DEVICES. When one looks at a large computer installation, a little thought will assure one that from the point of view of quantity, cost and importance to the proper functioning of the installation, the PERIPHERAL DEVICES occupy a paramount place. There is as much money invested in them as in the MAIN FRAME, if not more. How is it this area has been so neglected?

It is my aim to remedy this situation. I think a book on the subject should be self-sufficient. I do expect that my readers have had exposure to computer science and have worked in the field, so that the terms have a familiar ring to them. However, I will devote a good deal of time to developing a common background so that the rest of the material will fall into place.

The book may be used also for a course on PERIPHERAL DEVICES intended for the senior or the graduate student.

This is not light reading. The information presented is very technical. It ranges from the logic level in the hardware to the operational software level. I present a current and coordinated explanation of how DEVICES work as well as how they are integrated into the overall system.

The exposition is aimed at third generation equipment. Since a large majority of third generation equipment is manufactured by IBM, there is a good reason for using IBM equipment in all the examples that arise. Individual manufacturers produce equipment which is plug-to-plug compatible with IBM System 360. Hence, whatever is covered for IBM equipment remains fairly true for the independent peripheral manufacturers.

Chapter 1 is important for establishing a common fund of knowledge. It begins with an explanation of data and the relation of the data format to the user's problem. The reader must get an appreciation of a record and the field to see how these units of information compare and contrast with the units used within the COMPUTER and the PERIPHERAL DEVICES.

The book is intimately concerned with hardware details, at least at the functional level. It behooves the reader to become acquainted with the functional building blocks, their pictorial representation, and what they do. He should understand also how these blocks are used to make up the hardware of the COMPUTER, but this is reviewed for him in the first chapter. Along with the structure of the COMPUTER, we examine how data flows through it. The difference between the DEVICE and the medium from which the DEVICE gets information or on which it places information is emphasized.

Chapter 2 discusses DEVICES and media. It analyzes the components that make up an IO DEVICE in general terms. Then it examines the geometry of various media. A number of specific kinds of media are discussed. All medium and large third generation computers use the CHANNEL principle. It's a very complicated arrangement and not easy to grasp initially.

Chapter 3 provides an in-depth discussion of how the CHANNEL works. It also examines the DEVICE CONTROLLER and the MULTIPLEXER. The latter is a topic that has been neglected almost entirely in literature.

The CHANNEL INTERFACE is uniform throughout System 360 and System 370. This provides for total hardware modularity and enables the independent DEVICE manufacturer to hook his DEVICE into IBM equipment without a moment's worry that it won't work properly since he has observed the INTERFACE restrictions. That is why it is so important for us to know how the INTERFACE works.

Chapter 4 discusses the software required to handle IO activities. Since the software is most often involved by means of the interrupt, we examine in detail how the hardware interrupt system works. It is then possible to examine the two important software components, the **IO Supervisor** and the **Access Method**. These are covered only in terms of function, so that the reader can understand their need in managing the DEVICE. We discuss how the program talks to the **Access Method**, how the **Access Method** talks to the **IO Supervisor**, and how the **IO Supervisor** talks to the CHANNEL. Details of the functions of these software components are provided in the author's *OS/MVT* (published by Allyn & Bacon, Boston, Massachusetts, 1972).

The most important means of bringing information into the COMPUTER is the punchcard. There is much information on the punchcard and the CARD PUNCH. These are examined briefly, along with off-line operations, in Chapter 5. Cards are read and punched at the request of the computer, by means of the CARD READER PUNCH. It does not interface directly with the CHANNEL but rather with the INTEGRATED CONTROL UNIT. The operation and features of the DEVICE and the CONTROLLER are now examined.

The LINE PRINTER is the main source of immediately usable output from the COMPUTER. It is examined in detail in Chapter 6, along with those features of the ICU that are necessary for its operation.

The remaining important DEVICES all involve magnetic recording. It is not strange that an entire chapter is used for the principles of magnetic recording; this requires that we review some of the principles of physics. Thereafter, the several forms of magnetic recording are discussed with regard to their physical properties.

With this in hand, it is possible in Chapter 8 to describe the MAGNETIC TAPE UNIT and its CONTROLLER.

Other chapters describe other DEVICES with equal thoroughness as shown in the Table of Contents.

Many people have helped me to make this book a work of which I am really proud:

- The students at Baruch College, CUNY struggled through purple notes crammed full of errors.
- Max Ferder at Baruch College taught several classes using those same awful notes.
- Marilyn Pronko struggled through countless days of dictation, transcription and corrections to produce the manuscript.
- Ken Cashman, the production editor for P-H, was always helpful, understanding and polite, even when I lost my temper.
- Bert Schneider and his staff at Bertrick produced this fine art from my crude sketches.
- Chet and Arlene Abend produced their usual fine cover design.
- Sweet Abby Cook got eyestrain, headaches and distemper from proofreading the final page proofs.

IVAN FLORES

Contents

CHAPTER 1

Data and the Computer

1.0 INTRODUCTION

Chapter 1 introduces the difficult topic of IO. It is difficult both for the student and the writer because

- little has been written about IO;
- hardware knowledge is essential but the programmer is afraid of hardware details;
- programming types don't immediately see the benefit of this "detailed" knowledge.

The System

The COMPUTER SYSTEM consists of three parts:

- the architecture—the functional aspects of the COMPUTER that are readily available through some programming language;
- the operating system—the system software that keeps the system functioning;
- the hardware—the electronic and mechanical components on which the others depend for life.

Just as the institutions in society depend on the people that organized them to perform their function, the COMPUTER SYSTEM depends on the hardware. Can you make sense of the institution without knowing a little something about the people behind it?

Of course, it's up to *me* to give *you* just the right amount of hardware background—enough to understand well how the system works—not too much so that you are burdened with details. Regardless, you will still think that this exposition is too detailed with hardware. It's not!

Architecture

I eskew this word! How to program to get the computer to do the job you want it to is discussed at the assembly language level elsewhere and hardly touched here.*

Operating System

To make proper use of the COMPUTER, you should have a strong background in the programs that are performing the complicated services you request with simple requests such as READ or GET or OPEN. This should be acquired gradually. The more you know about the hardware, the easier the software will be. I think the best way is to attack on two fronts. Alternate your study—learn some hardware, then some software, etc.

All you need to know about IO software is in DSM.† An advanced exposition is in OS/MVT.‡ A brief introduction appears in Section 1.8 of the book.

Hardware

Hardware for third generation computers has indeed become complicated. So has the software. I don't believe that today's effective programmer can get by without understanding this typical flow of data.

- A card is read by the CARD READER.
- Its image is sent to the INTEGRATED CONTROL UNIT.
- Upon request it goes to the CHANNEL.
- The CHANNEL interrupts the COMPUTER when the image is available.
- The **IO Supervisor** accepts the image.
- The **Access Method** gets it next.
- He finally gives it to the program.

By the time you get through the book, all this will become clear.

Naturally, at least a superficial knowledge of hardware atoms and their

* Ivan Flores, *Computer Programming, System 360.* Englewood Cliffs, N.J.: Prentice-Hall Inc., 1971, 376 pp.
Ivan Flores, *The BAL Machine.* Boston, Mass.: Allyn and Bacon, Inc., 1972, 256 pp.
Ivan Flores, *Assemblers and BAL.* Englewood Cliffs, N.J.: Prentice-Hall, Inc., 1971, 420 pp.

† Ivan Flores, *Data Structure and Management.* Englewood Cliffs, N.J.: Prentice-Hall, Inc., 1970.

‡ Ivan Flores, *OS/MVT,* Boston, Mass.: Allyn and Bacon, Inc., 1972.

function is essential—it is the job of *this* chapter to present the preliminaries.
If you really get stuck, you can consult my hardware book.*

Conventions

In discussing COMPUTERS in a system environment, we have seen how many things concern us:

- the hardware itself;
- problem solving programs;
- the programming systems and parts of them;
- the language elements with which the programmer speaks to the computer;
- signals floating about in the hardware for internal communications.

The printed book is a way for the author to talk to the reader—for me to talk to you. This communication can be more effective if ambiguity is removed. One way of removing ambiguity is to use different typefaces to distinguish different kinds of objects. That is the device I use in this book—different kinds of typefaces to refer to different classes of things.

The conventions I employ are listed below. The reader should become familiar with them to facilitate his reading of this volume. Some of the words below have not yet been defined or even explained and so will be strange; but bear with me.

1. Whenever definitions are made, the name of the object being defined appears in **boldface type**. Generally these definitions are interspersed in the text. The use of boldface enables the reader to find the term more quickly when needed for reference.
2. The names of all kinds of hardware units appear in SMALL CAPITALS. This applies to everything from subsystems through functional units to logical units.
3. When software is discussed by name, this name appears in outline type. Thus, the name for the system supervisor is SYSTEM. All capitals are used for the largest systems; initial capitals are used for important routines, such as AMETHOD.
4. The programmer deals with several kinds of programming languages. These range from high level down to assembly language and, in some cases, machine language. I deal here mainly with assembly language and mnemonics. Whenever these are used, they appear in uppercase Gothic type. For example, when the programmer specifies addition, he uses the mnemonic, ADD. Mnmenonics I have invented for subcommands used mixed uppercase and lowercase Gothic. Thus, DRead is a *data read* subcommand.

* Ivan Flores, *Computer Logic*, Englewood Cliffs, N.J.: Prentice-Hall, Inc., 1960.

5. When parts of the hardware communicate with one another by control signals, the names of these signals appear in lowercase Gothic type. Thus, start is a signal to initiate some computer processes.

Many diagrams appear in the text. These include arrows of different kinds to indicate particular activities.

- The double arrow indicates the movement of information toward the arrowhead.

from ⟹ to

- The single arrow indicates that control is relinquished from the tail of the arrow and given to the program or subsystem at the head of the arrow.

from ⟶ to

- The dashed arrow indicates that the block at the tail of the arrow gives directions and controls the activity at the head of the arrow.

from ⇢ to

- A dotted line indicates a pointer to whatever is at the head of the arrow.

from ⇢ to

1.1 WHAT IS DATA?

Accounting

The business organization keeps information about groups of individuals or other kinds of entities. For instance, a firm has customers to which it extends credit. The amount outstanding is kept track of so that the customer can be billed periodically. In a similar way, for accounts payable, the firm keeps track of the amount it owes to its vendors and those who supply it with merchandise and parts. A third example is the inventory: an up-to-date knowledge of the availability of each item carried in stock, is a requirement for a modern firm.

For the three examples above, **individuals** or **entities** are defined differently: we have, respectively, customers, vendors and items in stock. We call the collection of entities a **universe**; those that make up the universe are called **individuals**.

Each individual has characteristics of which we keep track—their **attributes**. The individual may have many other characteristics of which we need not keep track. These are ignored. It is the present state of each attribute of

each individual that is of importance. These should be recorded and kept track of so that management can have an up-to-date knowledge of the individuals that comprise the universe. The more thoroughly this is done, the more effective it will be as a tool to management.

Before the Computer

You can imagine this scene at the turn of the century: a room crowded with tables attended by green-visored bookkeepers sitting on stools; each of them is bending over his own set of books, entering into them the transactions from the previous day. All the information is kept current totally by human effort.

But firms have grown so big and transactions so numerous that it is almost impossible for such a team of bookkeepers to be kept down to a reasonable size and yet keep the books sufficiently timely.

The entries in the books reflect the present state of the universe and the current attributes of the individuals involved.

Now

Today information is handled by the high-speed COMPUTER. Its task is also to keep available the latest information about the entities of interest. At frequent intervals the computer produces an output giving the current status of the universe. Many reams of paper may be produced at this time. Additionally, some systems provide for inquiry at CONSOLES provided at various sites throughout the company. Such a system produces replies at the same CONSOLE within reasonable time intervals. Such a facility is undoubtedly useful to management who can then retrieve only that information that is momentarily pertinent.

Quanta

A **quantum** is an amount, here, of information. The COMPUTER is viewed as a system that works under its own accord but must also interact with human operators and users. Hence there are three kinds of quanta to be considered.

- The user deals with information in terms most easily handled by the human, **user quanta**;
- The COMPUTER can most conveniently manipulate information in terms of how its MEMORY is divided, **computer quanta**;
- Then there are the PERIPHERAL DEVICES and their media where information is stored in bulk before and after it is degested by the COMPUTER, **IO quanta**.

We shall discuss appropriate quanta in several places which follow. The next section is devoted to the user and his view of the data. Section 1.3 examines the structure of the COMPUTER and prepares us to examine the quanta in which the COMPUTER deals with data in Section 1.4. PERIPHERAL DEVICES have their own needs and their examination is postponed to the next chapter.

1.2 USER QUANTA

Definition

Precise definitions of basic terms are difficult to come by in all categories: this is expected in the computer field where things are constantly changing. I shall use three criteria to determine the fitness of a definition:

1. it should adhere to common use if possible;
2. it should include all those things to which the definition is supposed to apply;
3. it should exclude all those things to which it is not supposed to apply.

Although many of the terms I used are found throughout the computer field, occasionally there are differences in terms that I originate that differ to some extent from common usage, from person to person, among various installations, and even from one problem to the next. An area where differences are most notably absent is the naming of information quanta. It is apparent that the user needs different quanta from that used by the hardware person. The user structures his problem and its solution in terms of the individual in the real world which he is describing. The information is converted into another form for temporary storage on a PERIPHERAL DEVICE. It is converted once again to another form when it is stored and worked on in the COMPUTER. Thus, there are three different quanta with which we are concerned.

We examine first those most important to the user.

The Universe

I like to think of the terms *file*, *record*, and *field* pictorially. I shall use the terms *universe*, *individual*, and *attribute*, respectively, as the practical correspondents in the real world to which the former terms apply. Consider Fig. 1.2.1; it shows a collection of individuals, a universe. I shall not define individual; here they are just people! There is some property about the people that makes them members of *my* universe. For instance, they may be employees of *my* company in a payroll application; they may be companies that do business with *me* for accounts receivable; etc.

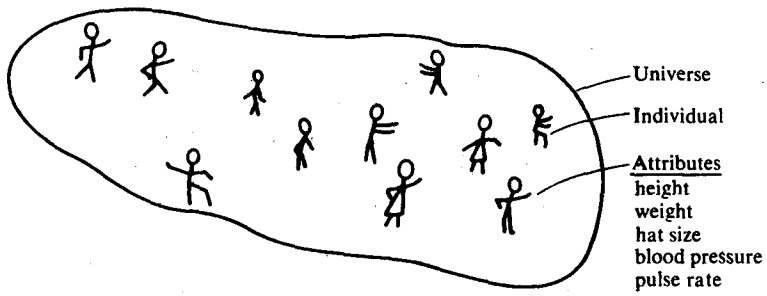

Figure 1.2.1. The universe consists of individuals each with attributes. The latter have values.

Each individual has many qualities. Only some of these are of interest to us. These kinds of qualities are called **attributes**. We name the attribute—this is the **attribute name**. Table 1.2.1 lists a few physical attributes people might have: height, weight, etc. Given an attribute, we can *measure* and *record* the **attribute value** for a particular individual. Thus we might construct a table for each individual such as Table 1.2.1. The *attribute name* of interest to us does not vary from one individual to another. The *value* of the *attribute* certainly does vary.

Table 1.2.1. Individual Attributes

Attribute Name	Attribute Value
Name	Harry Kaplan
Weight	183 lb
Height	5′ 10″.
Pulse	75 beats per minute

Now we can show how the concrete object and its name are related to the data about the object and the *name* for the data as shown in Table 1.2.2.

Table 1.2.2. Names for Objects and Data

Name for Object	Name for Data About Object
Attribute	Field
Name	Name
Value	Value
Individual	Record
Universe	File