THE
PICK
OPERATING
SYSTEM

JOSEPH ST. JOHN BATE
AND MIKE WYATT

The PICK Operating System

**Joseph St. John Bate
and
Mike Wyatt**

Also by Joseph St. John Bate

The Automated Office
Joseph St. John Bate and Ross Burgess
0 00 383008 X

Office Automation using the IBM Personal Computer Systems
Ross Burgess and Joseph St. John Bate
0 00 383104 3

The PICK Operating System

Other books of interest

The Automated Office
Joseph St. John Bate and Ross Burgess
0 00 383008 X

Office Automation using the IBM Personal Computer Systems
Ross Burgess and Joseph St. John Bate
0 00 383104 3

MS/PC-DOS Prompt
Randall McMullan
0 00 383288 0

dBase III – A Practical Guide
M. de Pace
0 00 383076 4

Working with dBase II
M. de Pace
0 00 383251 1

The dBase Programming Language
M. de Pace
0 00 383267 8

WordStar Prompt
Randall McMullan
0 246 12446 6

WordStar in Action
Randall McMullan
0 00 383107 8

SuperCalc Prompt
Randall McMullan
0 00 383004 7

Preface

We wrote this book to dispel the belief that there is a mystique in computing, especially in the commercial and business environment.

In our daily work as information technology consultants, we found that our clients expressed the firmly held belief that there was only one way to computerise their system – and that was the hard way! The number of times we have demonstrated how the DP department can be developed to increase its productivity and give the user what they require makes us realise it is time the world got to know PICK. It is time to uncover the best kept secret in the computing industry.

Computing is not just about hardware, software and computer jargon. Hardware and equipment are important and necessary elements in the computing equation but they should never be considered the prime elements. That role should be left to information. It is information – the way in which questions are answered by accessing data – that is important. The computer should be the information centre of any organisation, not a cash-draining black box. Computing, therefore, should be a method of improving the flow and the effectiveness of data by turning it into information.

We consider that the PICK operating system, above all other operating systems, enables the lay computer user within the commercial organisation to make use of the data available within the system to make the company more effective.

In writing this book we hope that we have opened up the opportunity for many more commercial and business organisations to increase their effectiveness – that is if they use the PICK operating system.

Joseph St. John Bate and Mike Wyatt

Contents

Chapter One
The PICK Operating System – An Overview

Introduction

The PICK operating system is first and foremost a database management system and secondarily an operating environment. It is this that differentiates it from virtually all other operating systems. As an operating system upon which to base a distributed data processing strategy for both large and small organisations, it meets the requirements of data driven systems and provides the tools necessary to allow the engineering of data into information.

Dick Pick, the designer, first implemented a query-type database system on an IBM System/360 at TRW. Once the potential had been realised, he worked with Microdata in California to implement the system on a micro-programmable microcomputer – the M1600 – and the 'REALITY' was launched. By implementing most of the system in microcode, a balance of functionalism and performance was achieved.

Dick Pick left Microdata to found Pick and Associates and then went on to form Pick Systems. In addition to the Microdata range, several other Pick implementations have appeared and the system is now available on IBM, Honeywell, DEC, Prime, and HP systems, as well as Intel, Zilog and Motorola chip-based micros.

Operating system architecture

The key to understanding the PICK system is its database structure. No single element of the file structure is constrained in number or real length: the database has any number of variable size files, each containing variable length records comprising variable length fields. This is implemented in a non-hierarchical delimiter-oriented structure. The PICK system file structure is managed by a comprehensive series of data dictionaries – implemented in a hierarchical structure – which provides a true relational database capability. Each file or set of files in the database has an associated data dictionary which logically defines the structure of the data file. This data dictionary may easily be changed, and complex data inter-relationships may be defined without affecting the actual data stored in the file.

At the highest level is the **system file** which contains user information (for example, passwords, access and security codes), along with pointers to the user's data files in virtual memory. The first file associates with each user in his **master dictionary** which contains his personal vocabulary. It includes the following:

- **Commands** (or 'verbs' to use PICK parlance) such as EDIT, LIST, COPY
- **File-names** that the verbs operate upon, such as Inventory
- **Verb associated words** (connectives), such as with, heading
- **Procedures** which can be used interactively to prompt, display, update or pass control.

Every file is structured in the same logical manner, even though the physical organisation is different. This means that ACCESS – the system query language – can provide a report on any data in the system, provided there is authorisation.

Each data record or item has a unique key for that file and can be up to 32 K bytes long. All fields – attributes, values and subvalues – are variable length and type. The only system generated data pertinent to an item is a code for its length, along with the system delimiters used to signify field intervals. Data files are structured through a hashing algorithm random access technique, using the item name (or item-id) as a key. This allows efficient retrieval of any file record, no matter how large the file grows. Other than the primary file space, which is user-defined at file creation, the data in a file may be located in any available disc space, even if this means the data spans multiple disc drives.

These 'low-level' details of the file structure are managed entirely by the virtual memory and database firmware routines. They are totally transparent to the user and the application developer. The user only 'sees' the database at the logical level. Apart from the occasional requirement to change user access or update lock codes, the database is truly self-maintaining and will operate indefinitely without user involvement.

A computer to do the business

The main task of a computer in a management context is to change data into information and help people to manage information. To measure the difficulty of developing systems to achieve this state of affairs, consider the number of times data is confused with information; how many times a business computer ends up processing data instead of managing information.

Data includes such things as invoice numbers and customer addresses, price lists and commodity codes. **Information** answers questions; Is the company making or losing money? Do the manufacturing divisions have enough material to produce what is needed for next month? How well is a particular product selling? Does the company have anyone who is over 35 years, has a degree in engineering and can speak French? The answers to these questions are information that can be used to run a commercial organisation.

However, on most computer systems, machine needs and not user needs predominate. Instead of helping the user to squeeze information out of the real world, the computer tends to trap the user in an unreal world conditioned by its own structure. This unreal structure is all too readily reinforced by dedicated DP staff.

In an ideal world (or if the computer were designed to match the needs of the commercial user) the structure of the query put to the computer would match

the structure of the problem. The user would be able to think in terms of questions and answers and transmit them to the computer without having to translate them into a language of files and fields, record lengths and search keys.

Unfortunately, reality is not static. Not only does the data change, but the relationships change too. New ways of doing things are developed, new people join the business, responsibilities change, even the rules and laws that govern the operation may change. The computer's metastructure should adapt to these changes as they occur.

The PICK operating system represents one attempt to meet this challenge. PICK is a virtual memory multi-user operating system specifically designed to help the user obtain information from the system.

Although it is an extremely sophisticated system, PICK is easy for non-programmers to use. In a few minutes the PICK English-like query language can be learnt well enough to begin answering complex questions about data stored in the built-in relational database.

An overview of the PICK structure

Although PICK is technically an operating system, it contains much more than the regular operating system functions. As part of the operating system, PICK has a wealth of features that are applications programs or add-ons in other systems.

The heart of the PICK system is its relational database. In PICK, the database management system is an inherent part of the operating system and not a utility or an add-on. Other parts of the operating system include an English non-procedural query language called ACCESS; a compiled version of BASIC with major enhancements for business and data management; PROC, a stored-procedure processor that allows the user to write and store elaborate command sequences, much like scripts in UNIX; a command processor called Terminal Control Language (TCL); a print spooler and a number of other utilities.

However, most of the utilities asociated with microcomputer operating systems are either invisible or non-existent in PICK; as much as possible, tasks like file handling and memory allocation are performed by the operating system. This allows the user freedom to concentrate on the information that can be obtained from the system. In addition PICK is rich in utilities related to information handling.

Underpinning all of this is an elaborate and complex software structure. PICK is implemented as a virtual machine with limited, carefully optimised connections to the underlying hardware through a section of code called the monitor. PICK's designers began by coming up with the best possible paper computer they could imagine for managing information. Then they wrote an operating system to emulate their design on real computers. The PICK virtual machine has its own internal structure and even its own pseudo-assembly language, all designed for data management tasks.

On a microcomputer, these features are implemented in software, right down to the virtual memory management system. One result of having a virtual

machine implemented in software is a high degree of applications portability. Any PICK machine will run almost any piece of PICK software. The only difference between an IBM PC XT and an IBM 4300 mainframe is that the application will probably run slower on the PC XT.

The usual penalty for having such an elaborate software superstructure is a loss of speed. One would expect PICK to be slow, especially on something like the 8088 microprocessor in an IBM PC XT. But that is not the case. While the PC XT implementation is noticeably slower than PICK on 68000-based microcomputers, it does not drag. To someone used to microcomputers, PICK can appear somewhat slow because most PICK implementations are designed to work with terminals via serial I/O ports rather than the kind of memory mapped video used on most microcomputers.

Processing speed, however, is good especially on things like searches and sorts. The PICK operating system contains many commonly used functions, such as database management and the high-level BASIC language. It is because these functions are so tightly integrated with the overall operating system that they run so fast.

There are other reasons why the processing speeds are so impressive. One is that PICK executes a machine language code on whatever machine it is running on. Other operating systems which use the virtual machine concept are usually interpreted or compiled/interpreted. Another reason is that PICK is carefully optimised for each implementation to be as fast as possible.

It is important to note, however, that PICK is not a number crunching system. There are better operating systems for scientific and engineering work or for anything else that requires a lot of calculation. PICK will score poorly on a computational benchmark such as the Sieve of Eratosthenes.

On the other hand, PICK will shine in an information management test, particularly one with a lot of complex data manipulation. This is because the database is relational and because of the way the system stores information; it can usually retrieve any piece of data with one or two disc accesses.

Strength in the file structure

There is tremendous strength in the file structure of the PICK operating system. Everything is contained in a file and the files are organised hierarchically.

There are three kinds of file in PICK, but only two are of immediate use – these are **data files** and **dictionary files**. The third, **binary files**, hold compiled code and stored lists. Both the data and the dictionary files are structured in the same manner. The distinction between data and dictionary files is that data files hold the data and dictionary files establish and maintain relationships between the data.

In PICK, there is only one type of data file, but three types of dictionary files. The **master dictionary** for each user defines the user's vocabulary, while **data dictionaries** describe the data and define the relationships (see Fig. 1.1).

One of the beauties of this separation of data and dictionary files is that it makes it easy to change relationships without affecting the data, and vice versa.
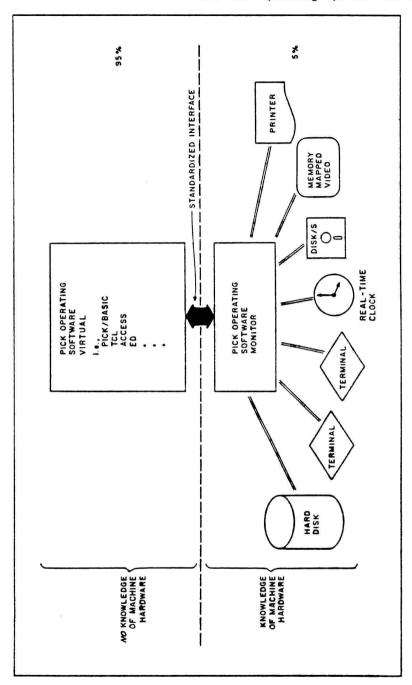
Figure 1.1. The parts of the PICK operating system and some supporting hardware. The interface between virtual, which contains most of the PICK operating system, and the monitor, which connects to the hardware, is standardised. Only about 5 per cent of the code in PICK is in the monitor.

Dictionary files can be edited, updated or changed completely without disturbing the data entries to which they refer. In the same fashion, data can be edited and updated without making any modifications to the dictionaries or disturbing the relationships already created. This gives flexibility to change the system as the real world changes.
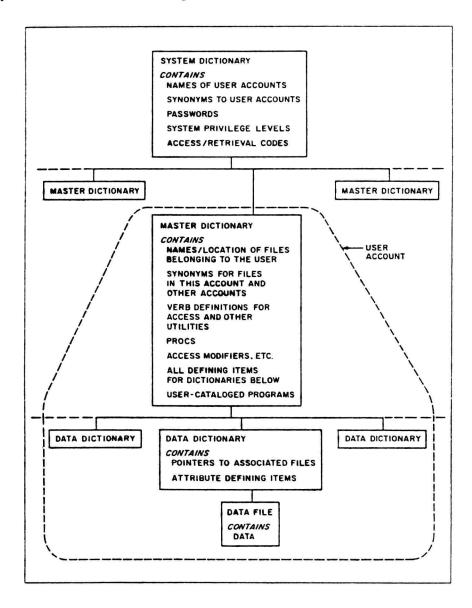

Figure 1.2. The PICK file hierarchy. There is only one system dictionary on the PICK operating system. Beneath it are any number of master dictionaries, one for each account on the system. Each master dictionary can have any number of data dictionaries beneath it, and each data dictionary points to one or more data files. A master dictionary and the subsidiary dictionaries and data files make up a user account on PICK.

The ACCESS query language

One of the genuine user friendly aspects of the PICK operating system is the ACCESS query language. ACCESS is designed to make it simple for the user to turn data into information. It is easy to use and allows the user to obtain almost immediate answers to even complex questions.

The simplicity of ACCESS stems from three things; it makes use of words or universally recognised symbols (such as $+$, $>$, $<$, $=$) to express concepts, it requires the minimum of extraneous information from the user, and most importantly it works in the way in which the user thinks.

One result of these powerful concepts is that the inexperienced user can be taught to use ACCESS very quickly and feel at home with the system in a short space of time. The fact that users can formulate questions in a perfectly natural way and then ask those same questions in almost the same format develops enthusiasm for the system and therefore ensures that the power is used to develop the business.

Commands in ACCESS are called **verbs**. They are action words like SORT or LIST. The usual sequence of an ACCESS command is verb, file-name, selection criteria, sequence criteria, report output attributes and modifiers. However, the user can develop any sequence that seems comfortable as long as it starts with a verb.

These facilities enable both the experienced and novice user to turn the data in the system into information, which is very powerful and in many cases unusual. This is because many computer systems can turn information into data by telling the user more than he wishes to know; confusion by facts. Computers are expert at this because they are so unwieldy; they are programmed to produce reams of paper with all the information contained within the pages.

Everyone in a company wishes to obtain information for decision making or whatever. It might be how many parts are in stock, or the cost of a particular item, or the rate at which an item is being used, or the amount of an item on order, or the name of the suppliers of a particular item, or the amount owed by the company to the supplier of a particular item, or when the next shipment is due, and perhaps one hundred other things. All this information is in the database, but to give all of it in answer to any of the questions would be unnecessary and counter productive. This is why many traditional computer systems are not used or are not useful. They make work; they don't make work easier.

PICK solves this problem by allowing the user, through ACCESS, to specify selection and output criteria. The system will then display the information that is of interest to the user. The rest of the information remains invisible and, in fact, may not be accessible to a user as a security measure.

Virtual memory keeps the action on the boil

There are other aspects to the PICK operating system that speed up the system, and virtual memory is one of them. The operating system allows the user to

address the entire range of available disc space as if it were memory. Consequently, any number of users may execute multiple processes without regard to the actual amount of system memory. The memory is arranged in 512 byte pages or frames. When a frame is needed for processing, the firmware determines if it is already in memory. If it is not, then the monitor automatically transfers the frame from disc, with the least recently used frame being backed out of disc if there is a need to create a space. Since these operations are also hidden from the user there is no requirement for the user or the system builders to concern themselves with optimisation, 'locality of reference' or any other virtual memory handling technique. The system takes the strain and allows the user to be an information worker and not a computer programmer, boffin or whatever else.

Multi-user capacity

The PICK system is designed as an interactive system capable of communicating with multiple users simultaneously. Users communicate with the system through terminals. Each terminal creates a **process**, an interruptable operation that is controlled by the operating system. There are are two types of process – **virtual** and **monitor**. Each process has a primary control block (PCB) which contains its software CPU – accumulator, registers, etc. The monitor PCB is permanently memory resident, while the virtual process PCBs – which are, after all, only data elements – are paged into memory as required.

When running, PICK assumes the availablity of a CPU per process, using its PCB. The machine must maintain the pretence by manipulating dedicated fields within each PCB as if they were address registers or accumulators, etc., thus providing the environment in which the processes can be executed. Each process attached to the system has a process identification block (PIB), which contains information about the status of the process with which it is associated.

Memory map makes it work

Virtual memory is configured into five primary areas, plus an overflow pool. These areas are as follows:

(1) Executive area
(2) Work area
(3) Low workspace area
(4) Preallocated terminal workspace
(5) File area

The executive area starts at the first frame and extends up to 511 or 1024 frames (manufacturer dependant); it contains base system software and user/manufacturer assembly language programs. The work area extends over 32 frames reserved for each process and the PCB. Beginning immediately after the work

area is the low workspace area which consists of a set of numbers of workspace frames per process and is used for additional process workspace. The file area is the remainder of virtual memory and is made up of user data held within the relational database. Any unused storage is allocated to the overflow pool and is used for additional process workspace when requested. The allocation of this space is controlled by the overflow space management processor. As files expand and contract or terminals log on and log off, frames are taken from or returned to this space. Obviously, the space becomes fragmented over time and a logical file save/restore process is necessary to recreate a contiguous overflow area.

In the PICK operating system, disc space is treated as one contiguous volume. All references to virtual space under PICK are by frame numbers (or FID) and a relative displacement from the first byte in the frame. Frames may be linked together to accommodate larger records.

The languages of PICK

Traditional PICK implementations support only three languages, which are:

(1) PROC
(2) ENGLISH
(3) DATA/BASIC

The first two are unique to the PICK operating system; the other, being BASIC, is a misnomer since it is an enhanced version of the standard Dartmouth BASIC. It is probably only as a result of the micro boom that BASIC is becoming more widely acceptable. The larger mainframe and minicomputer market has been COBOL-orientated for so long that PICK's lack of a COBOL compiler has reduced its acceptability. However, now that this market is widening its view and recognising the position of BASIC, the limitation will not seem so severe – not that it stopped the spread of PICK during the mid seventies to mid eighties.

Terminal Control Language

The Terminal Control Language (TCL) forms the main interface between user and PICK. Hundreds of standard system utilities, menus and procedures may be invoked from the TCL environment, as well as an unlimited number of user-defined procedures. TCL allows the copy, addition, deletion and modification of any predefined system procedure name, thus enabling each user to create an application environment or installation-specific 'vocabulary'.

The syntax is simple. The first word in a TCL sentence has to be a verb or a command. The system software handling that verb or command will then parse the rest of the sentence for anything that it requires, for example file-name, and process the command if it can.

Summary

This brief overview is an introduction to the many features offered by the PICK operating system that changes the complexion of a computer environment. The change is very fundamental and should not be underestimated. The PICK operating system allows the end user, however inexperienced, to use the computer as an information tool without being bogged down in the clumsy mystique of traditional computing. PICK makes the computer work for the non-DP user without the need to involve traditional DP staff. Freedom at last!