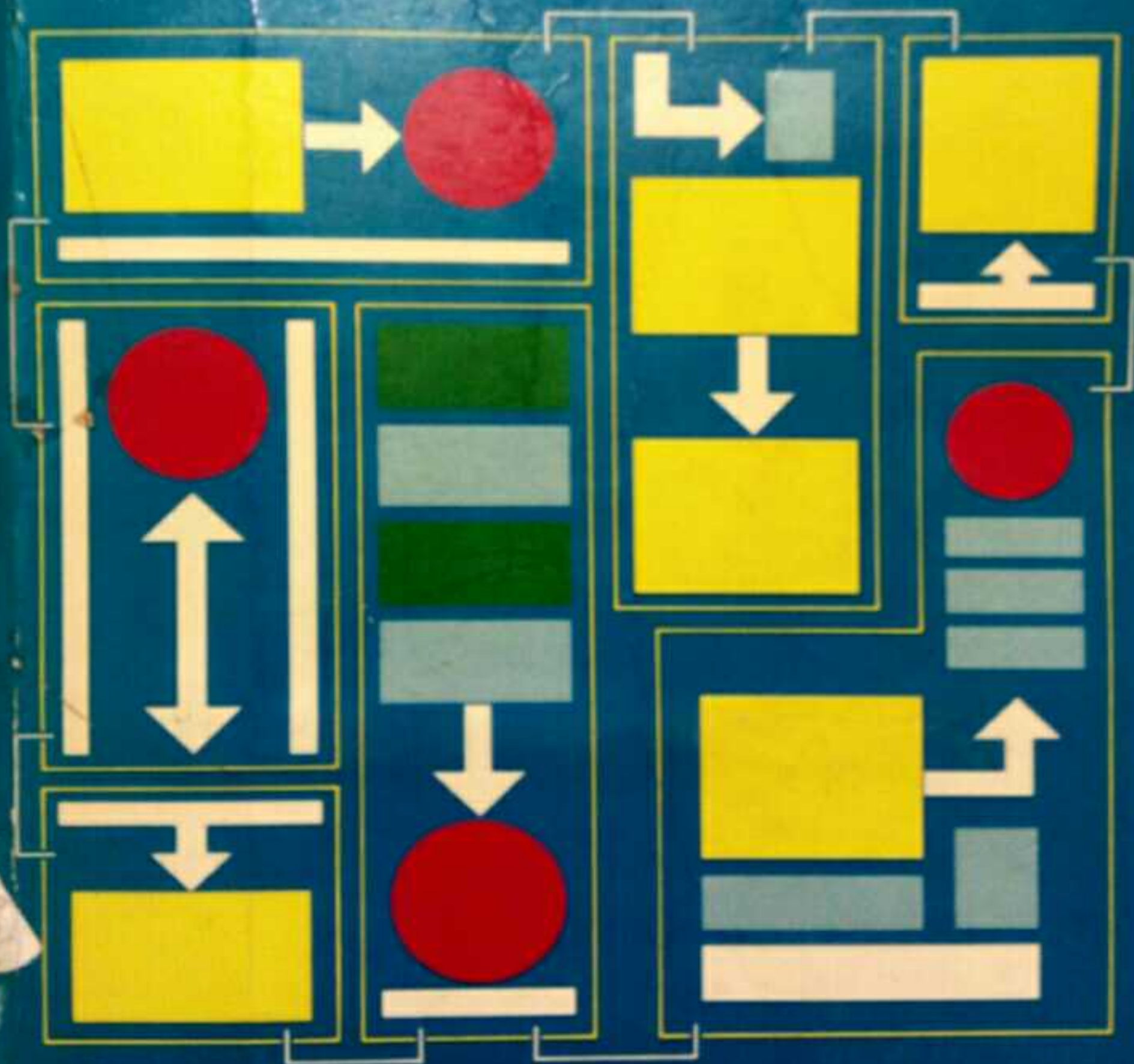


OPERATING SYSTEMS

COMMUNICATING WITH AND CONTROLLING THE COMPUTER



LAURIE S KELLER

INC-NO 103

RECEIVED
Department of Computer Science
UNIVERSITY OF MICHIGAN
5-6-74

OPERATING SYSTEMS

Communicating with and Controlling the Computer

LAURIE S. KELLER

The Open University, Milton Keynes, UK



PRENTICE HALL

New York • London • Toronto • Sydney • Tokyo

4 Virtual storage systems 96

- 4.1 Performance and virtual systems 97
- 4.2 Concepts of virtual storage 100
- 4.3 Implementing virtual systems 105
- 4.4 Managing virtual storage 115
- 4.5 System uses 117
- 4.6 Summary 119

5 Multicomputer and multiprocessor systems 122

- 5.1 Definitions and classifications 124
- 5.2 Principles of multiprocessor and multicomputer systems 131
- 5.3 Multiprocessing operating systems 136
- 5.4 Multiprocessing system performance 140
- 5.5 Network and distributed systems 143
- 5.6 Summary 147

6 A brief look into the future 149**PART II CASE STUDIES 155****7 CP/M: a microcomputer operating system
(by Gordon Davies) 157**

- 7.1 Structure 157
- 7.2 Storage allocation 159
- 7.3 File organization 161
- 7.4 Device mapping 164
- 7.5 Portability 165
- 7.6 MP/M 165
- 7.7 CP/Net 167

8 MS-DOS and PC-DOS 170

- 8.1 Structure 171
- 8.2 Storage allocation 172
- 8.3 File organization 174
- 8.4 Portability 177

9 The p-System (by Mark Woodman) 178

- 9.1 The universal operating system? 178
- 9.2 A brief history 179
- 9.3 Using menu-driven commands 181
- 9.4 The file system 186
- 9.5 Architecture of the p-machine 190

- 4.1 Performance and virtual systems 97
- 4.2 Concepts of virtual storage 100
- 4.3 Implementing virtual systems 105
- 4.4 Managing virtual storage 115
- 4.5 System uses 117
- 4.6 Summary 119

5 Multicomputer and multiprocessor systems 122

- 5.1 Definitions and classifications 124
- 5.2 Principles of multiprocessor and multicomputer systems 131
- 5.3 Multiprocessing operating systems 136
- 5.4 Multiprocessing system performance 140
- 5.5 Network and distributed systems 143
- 5.6 Summary 147

6 A brief look into the future 149

PART II CASE STUDIES 155

7 CP/M: a microcomputer operating system (by Gordon Davies) 157

- 7.1 Structure 157
- 7.2 Storage allocation 159
- 7.3 File organization 161
- 7.4 Device mapping 164
- 7.5 Portability 165
- 7.6 MP/M 165
- 7.7 CP/Net 167

8 MS-DOS and PC-DOS 170

- 8.1 Structure 171
- 8.2 Storage allocation 172
- 8.3 File organization 174
- 8.4 Portability 177

9 The p-System (by Mark Woodman) 178

- 9.1 The universal operating system? 178
- 9.2 A brief history 179
- 9.3 Using menu-driven commands 181
- 9.4 The file system 186
- 9.5 Architecture of the p-machine 190
- 9.6 P-code program execution 193

9.7	The p-machine emulator	196
9.8	Overcoming weaknesses and restrictions	198
9.9	p-System components	200
9.10	Device independence	201
9.11	Networking the p-System	202
9.12	Variations on a theme	208
9.13	Summary	209
10	Unix (by Gordon Davies)	211
10.1	The shell command language	212
10.2	Files on Unix	215
10.3	Implementation of the file system	220
10.4	Shell programming	226
10.5	The Unix kernel	228
10.6	Primary storage	236
10.7	Communication between processes	237
10.8	An evaluation of Unix	238
11	TPF/II: a high-performance system	246
11.1	History of TPF/II	248
11.2	Functional requirements	249
11.3	Meeting system requirements	252
11.4	What about support functions?	269
11.5	Cohabitation and the hypervisor	271
11.6	Case study conclusions	273
12	Virtual Machine/SP: an operating system	276
12.1	Requirements	276
12.2	Overview of VM/SP	278
12.3	The virtual machine	280
12.4	Managing virtual resources	283
12.5	Managing virtual I/O	292
12.6	The user's view	295
12.7	Communication between virtual machines	296
12.8	Case study conclusions	300
13	OS/VS2 MVS	304
13.1	Functional requirements	306
13.2	The computer system MVS controls	310
13.3	The supervisor	312
13.4	The master scheduler	317
13.5	Data management	318
13.6	Interactive system use - TSO	320
13.7	Miscellaneous system elements	321
13.8	Job entry	323

- 13.9 Error recovery 329
- 13.10 System integrity 331

14 General conclusions 335

- Glossary 338
- Index 359

Preface

The material for this book was originally gathered and written in a different form as the operating systems' half of a module, *PMT601: Computer Architectures and Operating Systems*, from the Open University's postgraduate course in the *Industrial Applications of Computers*.

I would like to thank the people who had a hand in producing it: Gordon Davies, who wrote the original versions of the CP/M and Unix case studies (Chapters 7 and 10) and Mark Woodman, who wrote the p-System case study (Chapter 9). Their work remains substantially as they wrote it. I would also like to thank Jonathan Blandon of Geophysical Service International (who is also my husband) for the benefit of his experiences as 'systems man' on the older IBM/360 systems and MS-DOS, Mike Cowlshaw of the IBM (UK) Scientific Centre at Winchester who took the time to read and comment on the material on VM and to demonstrate to me the latest bells and whistles, Geoff Gardiner of PA Technology (Cambridge) who had a hand in the computer architectures side of the original course and taught me a lot about array processors and multiprocessors, and Derrick Robinson of IBM (UK) Ltd who read the material on TPF/II.

Special thanks go to Prof. D.J. Howarth of Imperial College, University of London and Patricia Endacott – former boss and mentor – now friend and at Prime Computers (Europe) who read and commented on the original material.

ORGANIZATION AND ORIENTATION OF THE BOOK

This work is divided into two major parts. The first part describes the nature of and fundamental principles which govern operating systems. Part I answers the questions: what is an operating system and what does it do? Part II contains case studies of specific operating systems, ranging from the simple one-user microcomputer's operating system to the complex mainframe operating system. The range of case studies given is intended to impart a flavor of the variety of computing needs and how those needs are met by the different operating systems.