Macmillan Computer Science Series

Consulting Editor:

Professor F. H. Sumner, University of Manchester

- A. Abdellatif, J. Le Bihan, M. Limame, *Oracle – A User's Guide*
S. T. Allworth and R.N. Zobel, *Introduction to Real-time Software Design, second edition*
Ian O. Angell, *High-resolution Computer Graphics Using C*
Ian O. Angell and Gareth Griffith, *High-resolution Computer Graphics Using FORTRAN 77*
Ian O. Angell and Gareth Griffith, *High-resolution Computer Graphics Using Pascal*
M. Azmoodeh, *Abstract Data Types and Algorithms, second edition*
C. Bamford and P. Curran, *Data Structures, Files and Databases, second edition*
Philip Barker, *Author Languages for CAL*
P. Beynon-Davies, *Information Systems Development, second edition*
G. M. Birtwistle, *Discrete Event Modelling on Simula*
B. G. Blundell, C. N. Daskalakis, N. A. E. Heyes and T. P. Hopkins, *An Introductory Guide to Silvar Lisco and HILO Simulators*
Richard Bornat, *Understanding and Writing Compilers*
Linda E. M. Brackenbury, *Design of VLSI Systems – A Practical Introduction*
Alan Bradley, *Peripherals for Computer Systems*
G. R. Brookes and A. J. Stewart, *Introduction to occam 2 on the Transputer*
P. C. Capon and P. J. Jinks, *Compiler Engineering Using Pascal*
J. C. Cluley, *Introduction to Low Level Programming for Microprocessors*
Robert Cole, *Computer Communications, second edition*
Eric Davalo and Patrick Naïm, *Neural Networks*
S. M. Deen, *Principles and Practice of Database Systems*
C. Delannoy, *Turbo Pascal Programming*
Tim Denvir, *Introduction to Discrete Mathematics for Software Engineering*
D. England et al., *A Sun User's Guide, second edition*
J. S. Florentin, *Microprogrammed Systems Design*
A. B. Fontaine and F. Barrand, *80286 and 80386 Microprocessors*
Michel Gauthier, *Ada – A Top-down Course*
J. B. Gosling, *Design of Arithmetic Units for Digital Computers*
M. G. Hartley, M. Healey and P. G. Depledge, *Mini and Microcomputer Systems*
J. A. Hewitt and R. J. Frank, *Software Engineering in Modula-2 – An Object-oriented Approach*
Roger Hutton, *COBOL 85 Programming*
Roland N. Ibbett and Nigel P. Topham, *Architecture of High Performance Computers, Volume I*
Roland N. Ibbett and Nigel P. Topham, *Architecture of High Performance Computers, Volume II*
Patrick Jaulent, *The 68000 - Hardware and Software*
P. Jaulent, L. Baticle and P. Pillot, *68020-30 Microprocessors and their Coprocessors*
M. J. King and J. P. Pardoe, *Program Design Using JSP – A Practical Introduction, second edition*
V. P. Lane, *Security of Computer Based Information Systems*
Bernard Leguy, *Ada – A Programmer's Introduction*
M. Léonard, *Database Design Theory*
David Lightfoot, *Formal Specification Using Z*
A. M. Lister and R. D. Eager, *Fundamentals of Operating Systems, fifth edition*
Elizabeth Lynch, *Understanding SQL*
Tom Manns and Michael Coleman, *Software Quality Assurance, second edition*
B. A.E. Meekings, T. P. Kudrycki and M. D. Soren, *A Book on C, third edition*
R. J. Mitchell, *Microcomputer Systems Using the STE Bus*
R. J. Mitchell, *Modula-2 Applied*
Y. Nishinuma and R. Espesser, *UNIX – First contact*
Pham Thu Quang and C. Chartier-Kastler, *MERISE in Practice*
A. J. Pilavakis, *UNIX Workshop*

continued overleaf

E. J. Redfern, *Introduction to Pascal for Computational Mathematics*
Gordon Reece, *Microcomputer Modelling by Finite Differences*
F. D. Rolland, *Programming with VDM*
W. P. Salman, O. Tisserand and B. Toulout, *FORTH*
L. E. Scales, *Introduction to Non-Linear Optimization*
A. G. Sutcliffe, *Human-Computer Interface Design, second edition*
C. J. Theaker and G. R. Brookes, *Concepts of Operating Systems*
M. Thorin, *Real-time Transaction Processing*
M. R. Tolhurst et al., *Open Systems Interconnection*
A. J. Tyrrell, *COBOL from Pascal*
M. J. Usher, *Information Theory for Information Technologists*
I. R. Wilson and A. M. Addyman, *A Practical Introduction to Pascal with BS6192, second edition*

Other titles

Roy Anderson, *Management, Information Systems and Computers*
I. O. Angell, *Advanced Graphics on VGA and XGA Cards Using Borland C++*
B. V. Cordingley and D. Chamund, *Advanced BASIC Scientific Subroutines*
N. Frude, *A Guide to SPSS/PC+, second edition*
Percy Mett, *Introduction to Computing*
Tony Royce, *COBOL – An Introduction*
Tony Royce, *Structured COBOL – An Introduction*
Barry Thomas, *A PostScript Cookbook*

Concepts of Operating Systems

Colin J. Theaker

Staffordshire University

and

Graham R. Brookes

Hull University

M
MACMILLAN

© Colin J. Theaker and Graham R. Brookes 1993

All rights reserved. No reproduction, copy or transmission of this publication may be made without written permission.

No paragraph of this publication may be reproduced, copied or transmitted save with written permission or in accordance with the provisions of the Copyright, Designs and Patents Act 1988, or under the terms of any licence permitting limited copying issued by the Copyright Licensing Agency, 90 Tottenham Court Road, London W1P 9HE.

Any person who does any unauthorised act in relation to this publication may be liable to criminal prosecution and civil claims for damages.

First published 1993 by
THE MACMILLAN PRESS LTD
Houndmills, Basingstoke, Hampshire RG21 2XS
and London
Companies and representatives
throughout the world

This text is a revised and expanded edition of
A Practical Course on Operating Systems, by the same authors,
first published in 1983

ISBN 978-0-333-52366-7 ISBN 978-1-349-11511-2 (eBook)
DOI 10.1007/978-1-349-11511-2

A catalogue record for this book is available
from the British Library.

Contents

<i>Preface</i>	x
1 Overview	1
1.1 What is an operating system?	1
1.2 Why are operating systems needed?	1
1.3 Must there always be an operating system?	2
1.4 Where does an operating system 'live'?	3
1.5 How do operating systems differ from one another?	4
1.6 How large and complex is the operating system software?	4
1.7 Historical development	5
1.8 References and bibliography	10
2 Basic Concepts	11
2.1 Processes	11
2.2 Principles of multiprogramming	12
2.3 The use of interrupts	13
2.4 Simple I/O buffering	14
2.5 Spooling	15
2.6 Concepts of time-sharing systems	22
2.7 Functional units of a multiprocessor system	25
2.8 Some multiprocessing system structure designs	26
2.8.1 Master-slave	26
2.8.2 Separate executive	27
2.8.3 Symmetrical systems	27
2.9 Summary	28
2.10 References and bibliography	29
3 I/O Buffering	30
3.1 Simple buffering	30
3.2 More complex I/O buffering	32
3.3 Memory mapped systems	35
3.4 Window management systems	37
3.5 Summary	39
3.6 References and bibliography	40

4	Scheduling	41
4.1	Preemptive and non-preemptive scheduling	41
4.2	Time-slicing	43
4.3	Choice of a time slice period	46
4.4	Estimation of response time	47
4.5	Scheduling algorithms	49
4.6	Summary	52
4.7	References and bibliography	53
5	Memory Management - Basic Principles	54
5.1	Swapping strategies	55
5.1.1	Simple swapping system	55
5.1.2	A more complex swapping system	57
5.1.3	Further developments of the swapping system	58
5.2	Memory protection	58
5.3	Virtual addressing	60
5.4	Segmentation	62
5.4.1	Fragmentation	62
5.4.2	Program locality	63
5.4.3	Sharing of code and data	65
5.4.4	Multiple base-limit register machines	66
5.4.5	Address translation on a multiple base-limit register machine	68
5.4.6	Shared segments	71
5.5	Summary	75
5.6	References and bibliography	75
6	Memory Management - Paging Algorithms and Performance	77
6.1	Paging	77
6.1.1	Address translation in a paged machine	78
6.2	Paged segmented machines	79
6.2.1	Memory management in a paged segmented machine	80
6.3	Algorithms and performance	83
6.4	Page replacement algorithms	90
6.4.1	Least-recently-used algorithm (LRU)	90
6.4.2	First-in-first-out algorithm (FIFO)	91
6.4.3	Not-recently-used algorithm (NRU)	92
6.4.4	Stack algorithms	93
6.5	Multiprogramming	93
6.5.1	Thrashing	95
6.6	Storage allocation techniques	98
6.6.1	First fit algorithm	99

6.6.2	Next fit algorithm	99
6.6.3	Best fit algorithm	100
6.6.4	Worst fit algorithm	100
6.6.5	Buddy system	100
6.7	Summary	100
6.8	References and bibliography	101
7	File Management	103
7.1	Requirements of a file system	103
7.2	Directories and access rights	104
7.2.1	Single-level directory structure	105
7.2.2	Hierarchical directory structure	106
7.3	Dumping and archiving	107
7.4	Secure file updating	107
7.5	File system of UNIX	109
7.6	Summary	114
7.7	References and bibliography	114
8	Resource Management - Deadlocks	115
8.1	Allocation of resources	115
8.2	Deadlocks	116
8.2.1	Deadlock prevention	117
8.2.2	Deadlock detection	118
8.2.3	Deadlock avoidance	119
8.2.4	The Banker's Algorithm	120
8.3	Summary	123
8.4	References and bibliography	124
9	Resource Management - Protection	125
9.1	Introduction to protection systems	125
9.2	A general model of protection systems	126
9.2.1	Defining and representing domains	127
9.3	Inter-domain transitions	130
9.3.1	Domain changes only on process changes	130
9.3.2	Restricted domain change systems	131
9.3.3	Arbitrary domain changes	133
9.4	An example of a capability system	135
9.4.1	Types and access amplification	136
9.4.2	Creation of new types	137
9.5	Summary	139
9.6	References and bibliography	139

10 Process Synchronisation - Basic Principles	141
10.1 Introduction	141
10.1.1 Process synchronisation	142
10.1.2 Process competition	142
10.2 Flags	144
10.3 Semaphores	146
10.3.1 Mutual exclusion by semaphores	147
10.3.2 Process communication using semaphores	147
10.4 Implementation of semaphores	148
10.5 Semaphore implementation for a multiprocessor system	149
10.5.1 Hardware implementation	150
10.5.2 Software implementation	151
10.6 Semaphore implementation for multiprogramming systems	152
10.7 The use of semaphores - the reader's and writer's problem	152
10.7.1 Readers have priority	153
10.7.2 Writers have priority	154
10.8 Summary	156
10.9 References and bibliography	156
11 Process Synchronisation - Language-Based Approaches	157
11.1 Introduction	157
11.2 Monitors	158
11.2.1 Encapsulation	159
11.2.2 Mutual exclusion	160
11.2.3 Communication	160
11.3 Modula-2	161
11.3.1 Modules	161
11.3.2 Processes - coroutines	162
11.3.3 Devices and interrupts	163
11.4 Ada	163
11.4.1 Rendezvous	164
11.5 occam	167
11.5.1 Processes and channels	167
11.5.2 Inter-process communication	170
11.6 Summary	170
11.7 References and bibliography	171
12 Process Synchronisation - Message-Based Approaches	172
12.1 Introduction	172
12.2 Message-passing operations	173
12.3 Practical considerations	176
12.3.1 Form of the message	176

12.3.2	Addressing of processes	177
12.3.3	Type of message queuing discipline	177
12.3.4	Message validation / protection	178
12.3.5	Message deadlocks	178
12.4	Solution of the reader's and writer's problem using messages	178
12.4.1	Design of the controlling process	179
12.5	Summary	180
12.6	References and bibliography	181
13	Networks and Interworking	182
13.1	Introduction to networks	182
13.2	File transfers and mail systems	184
13.3	Networked terminals	185
13.3.1	Simple character terminals	186
13.3.2	Screen based terminals	187
13.3.3	Virtual terminals	187
13.3.4	Bit mapped terminals	189
13.4	ISO models	189
13.5	Summary	194
13.6	References and bibliography	194
14	Operating System Performance	196
14.1	Functions and constraints of a system	196
14.2	Performance of a simple system	197
14.3	Performance evaluation	199
14.4	Performance monitoring	200
14.5	Usability of a simple system	201
14.6	Summary	202
14.7	References and bibliography	202
	Appendix - An example system	203
	<i>Index</i>	222

Preface

An operating system is one of the most important pieces of software in any modern computer system, irrespective of its size or application, yet much mystique still surrounds its design. Operating systems themselves seek to be transparent to users, which adds to their perception as a black box. As computer systems increase in complexity, particularly at the microprocessor level, more complexity appears in the system software. It is easy to forget that the operating system is not an end in itself; it is there solely to provide efficient support for other activities. With this in mind, the primary objective of this book is to provide a greater practical understanding of the concepts of the design and implementation of operating systems, the problems they address and the constraints imposed on them.

This book has been produced as an update on an earlier text *A Practical Course on Operating Systems* by the same authors. The subject of operating systems has advanced substantially since the earlier text, as the ideas of what operating systems should do are constantly being modified by changing user expectations and requirements, and through the increased potential of the system hardware due to technological developments. Whilst this book focuses on the basic concepts of operating systems, many of which are long standing, it has been necessary to make substantial revision of the earlier material in producing this book. The ordering of much of the material has been changed, some aspects such as treatment of traditional Job Control Languages have been deleted, and new topics such as Networking have been introduced.

The contents of this book are derived from a number of courses of lectures given to undergraduate students of computer science. In its entirety, it provides suitable material for a full course on operating systems for students who have a basic grounding in computer science or who have some practical experience of computing. Less detailed courses may be derived by choosing a sub-section of chapters. For example, the first two chapters provide an overview of the structure of an operating system, and can therefore be used as material for a lower level course that seeks to provide a general introduction to the nature and role of operating systems. By focusing on concepts rather than the current flavour of any particular operating system, the text is also of interest to

software engineers involved in the development of embedded systems, as the problems faced at the hardware interface are identical, albeit in different guises.

This book differs significantly from other operating systems texts in the way in which the material is presented. The intention has been to provide a sound understanding of the underlying concepts, particularly from the point of view of the problems that the operating system is trying to solve. A typical subject will therefore be considered initially by examining the problem area and then introducing possible solutions. These may reflect chronological developments, changes in technology or levels of sophistication within the systems.

Although this approach may appear as being 'bottom-up', the authors feel that it has strength in providing an appreciation of basic concepts and rationale. It provides an understanding of the constraints, interfaces and problems faced by operating system software, without losing sight of the end objectives. In contrast, the more traditional way of examining the subject is from the 'outside looking in', where one might see a complete operating system, with all its sophistication and vagaries. Although good for examining particular systems, to stand back and admire the handiwork of others might not provide a sound understanding of the rationale.

The first two chapters of this book provide essential background on which further detailed consideration of topics follow. Chapter 1 identifies why operating systems are necessary, provides a brief historical view of their developments and surveys major types of operating system from the point of view of their application. Chapter 2 introduces basic concepts relating to the structure of operating systems. The notion of a process is described, together with the use of interrupts and multiprogramming. An outline of a simple operating system design is developed, initially using a simple spooling system as the vehicle, and subsequently extending the ideas to the realisation of a time sharing system. This system provides the platform on which other more advanced topics are developed within the text. Further coverage of this system design is also provided in the appendix.

Chapter 3 provides an examination of input/output management techniques. Although not fundamental to the overall structure and operation of a system, it picks up on topics introduced in chapter 2 to complete the coverage of this material. This is followed in chapter 4 by an examination of scheduling principles and the algorithms associated with scheduling. In particular, time slicing is considered in more depth, including its effect on the responsiveness of the system.

A comprehensive treatment of memory management strategies follows in chapters 5 and 6. Starting from the practicalities of supporting the simple

system discussed in chapter 2, the notions of protection and virtual memory are introduced, using a simple base-limit register scheme for address translation. Segmentation is described as a means of resolving issues of program locality and sharing, and paging is presented as a solution to fragmentation problems. The chapters conclude with a consideration of memory management algorithms and their associated performance.

Chapter 7 considers the issues that relate to the design of file management systems. The chapter also examines the design of the UNIX file system, being one of the more distinguishing features of that particular operating system.

The chapters so far have largely been concerned with the major functional components or processes within an operating system. Chapters 8 to 12 relate to techniques that are applicable to many different processes, and indeed to supporting the overall process structure. Chapter 8 examines the problems of resource management from the point of view of allocation of resources, competition for their control and the potential for deadlocks. This is followed by a formal review of protection systems, identifying where and how they might be implemented within an operating system.

The problems of concurrency are then examined, and approaches for achieving co-operation and synchronisation of processes are described. Chapter 10 identifies the low level issues that affect co-operating processes. As a primitive solution, the semaphore mechanism is presented, including examples of its use and how it might be implemented. Chapters 11 and 12 discuss higher level approaches to the problem. Firstly language-based approaches are discussed. The functions provided are related to those of the more primitive semaphore mechanism, without addressing the wider issues of their implementation for distributed computing. Other language-based facilities to support the development of system software, such as process management, i/o handling, etc. are also discussed briefly. Message-based approaches are then examined, including the practical considerations that affect their application. The use of messages in a distributed context are also considered, particularly in the implementation of a remote procedure calling mechanism.

The theme of distributed processing is carried further in chapter 13, where issues of networking and interworking are considered. Although networking is playing a more significant role in the use of computers generally, it can be viewed as an enabling technology to support the more fundamental aspects of operating systems. The topic is therefore considered in this light, and readers wishing a more comprehensive coverage of networking issues are referred to specialist texts on the subject.

The final chapter addresses issues relating to the overall performance of the system. As with the engineering of any product, it is important to be able to quantify the effectiveness of any design and implementation. This is not always easy to achieve in the case of an operating system owing to its primary role in supporting other applications. Chapter 14 surveys techniques for the monitoring and evaluation of system performance and reviews the use of such information.

An appendix provides a more detailed specification of a simple operating system. It provides simple time sharing facilities, similar to the system introduced in chapter 2, and employs input/output, scheduling, memory and file management techniques, as discussed in chapters 3-7. The system design has deliberately been kept simple for illustrative purposes, and exercises are provided to allow readers to develop more sophisticated implementations.

Throughout the book, specific cases and examples of systems are considered to illustrate general points made in the text. At the end of chapters, there are references to which readers can find more detailed information on specific topics.

Colin J Theaker

Graham R Brookes