

Contents

Preface v

1 Preliminaries 1

- 1.1 Introduction 1
- 1.2 Memory hardware 2
 - 1.2.1 General considerations 2
 - 1.2.2 The main memory system 2
 - 1.2.3 Indexing hardware 7
 - 1.2.4 Virtual memory 8
- 1.3 Software 10
 - 1.3.1 Statement labels and program control 13
 - 1.3.2 Constants and variables 13
 - 1.3.3 Declarations, data types, and referencing variables 14
 - 1.3.4 Procedures 17
 - 1.3.5 Procedure invocation 18
 - 1.3.6 Parameter passage 19
 - 1.3.7 Functions and subroutines 20
- 1.4 An algorithmic language 21
 - 1.4.1 Symbols 21
 - 1.4.2 Evaluation of expressions 21
 - 1.4.3 Executable statements 22
 - 1.4.4 Non-executable statements 27
 - 1.4.5 Allocation of variables 33
 - 1.4.6 Designational expressions 34
 - 1.4.7 Address extraction 39

2 Linear structures 42

- 2.1 Introduction 42
- 2.2 Sequential memory representations 44
 - 2.2.1 Implementing sequential lists in data structures provided by high-level languages 44
 - 2.2.2 Manipulating data in sequentially stored lists 51

viii Contents

2.2.3	Sets, stacks, and queues	55
2.2.4	Sequential allocation with more than one list	68
2.3	Linked memory representations	71
2.3.1	Singly linked lists	71
2.3.2	Linked stacks and queues	74
2.3.3	Multi-linked linear structures	76
2.4	String representation of linear lists	83
2.5	Comparison of methods	84
2.6	Summary	85
3	String processing	91
3.1	Introduction	91
3.2	Character strings	92
3.3	String processing operations	92
3.4	Implementation techniques	95
3.4.1	Sequential string processing techniques	96
3.4.2	Moving characters from one field to another	97
3.4.3	Implementing string processing routines using string identifiers	99
3.4.4	An efficient implementation of INDEX	104
3.4.5	Implementing string processing routines using string descriptors	112
3.4.6	Implementing strings using linked structures	118
3.5	Summary	120
4	Trees and graphs	130
4.1	Introduction	130
4.2	Trees	130
4.2.1	Binary trees	134
4.2.2	Implementing general trees	144
4.2.3	Some applications of trees	147
4.3	Graphs and digraphs	186
4.3.1	Matrix representations of graphs	186
4.3.2	Linked representations of graphs	192
4.4	Summary	201
5	Arrays, matrices, and tables	208
5.1	Introduction	208
5.2	Multi-dimensional arrays	211
5.3	Symmetric and triangular matrices	219

- 5.4 Extendible arrays 222
 - 5.5 Sparse matrices 225
 - 5.5.1 Row and column indexing 226
 - 5.5.2 Bit maps 233
 - 5.5.3 Address maps 235
 - 5.5.4 Linked memory representations 236
 - 5.5.5 Hash table techniques 236
 - 5.6 Tables 236
 - 5.6.1 Sequential representations 237
 - 5.6.2 Hash tables 239
 - 5.6.3 Hashing functions 240
 - 5.6.4 Hash table organizations 242
 - 5.6.5 External tables 248
 - 5.7 Summary 254
- 6 Dynamic memory management using sequential allocation and compaction 262**
- 6.1 Introduction 262
 - 6.2 Memory regeneration 263
 - 6.3 Dynamic memory management using indirect addressing 265
 - 6.3.1 Memory allocation 268
 - 6.3.2 Memory regeneration 270
 - 6.4 Dynamic memory management using direct addressing 275
 - 6.5 Marking algorithms 280
 - 6.6 Memory compaction 294
 - 6.7 Summary 299
- 7 Dynamic memory management using lists of available space 303**
- 7.1 Introduction 303
 - 7.2 Memory management techniques used by list processing systems 306
 - 7.2.1 Allocating cells 306
 - 7.2.2 Dangling references 308
 - 7.2.3 Garbage collection 309
 - 7.2.4 Reference counts 312
 - 7.2.5 Memory management in SLIP 314
 - 7.2.6 Maintaining lists of available space in real time 323
 - 7.2.7 Systems that use cells of several sizes 325
 - 7.3 Memory management techniques used by operating systems 333
 - 7.3.1 First-fit allocation 336
 - 7.3.2 Best-fit allocation 339

x Contents

- 7.3.3 Deallocation 339
- 7.3.4 The boundary tag method 343
- 7.3.5 The buddy system 345
- 7.3.6 Comparison of methods 347
- 7.4 Summary 347

8 Recursion 353

- 8.1 Introduction 353
- 8.2 Recursive definitions and recursive procedures 353
- 8.3 Implementing non-recursive procedures 359
- 8.4 Implementing recursive procedures 369
- 8.5 Data structures as actual procedure parameters 379
- 8.6 Recursive procedures for manipulating data structures 381
- 8.7 Recursive definitions and LISP 385
 - 8.7.1 Preliminary procedures 395
 - 8.7.2 Sets in LISP 397
 - 8.7.3 Tables in LISP 398
 - 8.7.4 Property lists and structures in LISP 399
 - 8.7.5 Comments about the efficiency of these techniques 406
- 8.8 Summary 407

9 Some systems that use interesting data structures 410

- 9.1 Introduction 410
- 9.2 IMS: A hierarchical database system 411
 - 9.2.1 The IMS structure 411
 - 9.2.2 The IMS query language 413
 - 9.2.3 Implementations 415
- 9.3 ESP³: A graphics system 417
 - 9.3.1 The ESP³ language 418
 - 9.3.2 The picture data structure 420
 - 9.3.3 Plotting a picture 422
- 9.4 GROPE: A graph processing system 429
 - 9.4.1 GROPE entities 429
 - 9.4.2 GROPE objects and values 433
 - 9.4.3 Relationships and structures 433
 - 9.4.4 GROPE functions and predicates 435
 - 9.4.5 Implementation of GROPE structures 437
- 9.5 Summary 446

General References 448

Author index 451

Procedure index 455

General index 459

