Table of Contents

Introduction ..... **xv**

Chapter 1: Overview of System Software **1**

1.1 Software 2
1.1.1 Software Characteristics 2
1.2 Software Hierarchy 3
1.3 System Programming 4
1.4 Machine Structure 5
1.4.1 Von Neumann Architecture 6
1.4.2 Harvard Computer Architecture 7
1.4.3 Comparison between Harvard and Von Neumann Architectures 7
1.5 Interfaces 8
1.5.1 Software Interface 8
1.5.2 Hardware Interface 9
1.5.3 User Interface 9
1.6 Address Space 10
1.6.1 Types of Address Space 10
1.7 Computer Languages 11
1.7.1 Classification of Computer Languages 11
1.7.2 Low Level Languages 11
1.7.3 High Level Languages 13
1.7.4 Advanced High Level Languages 13
1.8 Life Cycle of a Source Program 14
1.9 System Software Development 15
1.9.1 System Software Development Process 15
1.9.2 Recent Trends in Software Development 17

1.10 Levels of System Software 18
Summary 19
Key Terms 21
Review Exercise 21
True or False 21
Multiple Choice Questions 22
Descriptive Questions 22

Chapter 2: Overview of Language Processors 25

2.1 Programming Languages and Language Processors 27
2.2 Language Processing Activities 30
2.2.1 Program Generation Activities 30
2.2.2 Program Execution Activities 31
2.3 Fundamentals of Language Processing 34
2.3.1 Analysis Phase 35
2.3.2 Synthesis Phase 36
2.3.3 Language Processor Pass 36
2.4 Symbol Tables 37
2.4.1 Uses of Symbol Table 38
2.4.2 Contents of the Symbol Table 38
2.5 Data Structures for Language Processing 40
2.5.1 Choice of Data Structures 41
2.5.2 Search Data Structures 43
2.5.3 Allocation Data Structures 50
Summary 54
Key Terms 56
Review Exercise 57
True or False 57
Multiple Choice Questions 57
Descriptive Questions 58

Chapter 3: Assemblers 63

3.1 Assemblers 64
3.1.1 Advantages and Disadvantages of Assembly Language Programming 65
3.2 Elements of Assembly Language Programming 66
3.2.1 Composition of an Assembly Language Program 66
3.2.2 General Format of an Assembly Language Statement 67
3.2.3 Types of Assembly Language Statements 71
3.3 Design of Assembler 76
3.3.1 Analysis and Synthesis 77
3.3.2 Tasks Performed during Analysis and Synthesis Phases of Assembly Process 78
3.4 Assembler Design Criteria 79
3.5 Types of Assemblers 79
3.5.1 Two-Pass Assemblers 80
3.5.2 Single-Pass Assemblers 81
3.5.3 Comparison between Single-Pass and Two-Pass Assemblers 81
3.6 Assembler for Intel x86 82
3.6.1 Intel 8086/8088 Architecture 82
3.6.2 Addressing Modes and Instruction Formats 84
3.6.3 Instruction Decoding during Assembly 86
3.6.4 Assembler Directives 88
3.6.5 Writing Segment-Based Program 89
3.6.6 Advanced Intel Processors 90
3.7 Algorithm for Single-Pass and Multi-Pass Assemblers 92
3.7.1 Algorithm for Single-Pass Assembler 92
3.7.2 Algorithm for Multi-Pass Assembler 94
3.8 Advanced Assembly Process 96
3.9 Variants of Assemblers 98
3.9.1 MASM32 99
3.9.2 The Netwide Assembler (NASM) 99
3.9.3 SPASM – The Specific Assembler 100

3.10 Design of Two-pass Assembler 100
3.10.1 Pass I of Assembler 100
3.10.2 Pass II of Assembler 108
Summary 111
Key Terms 112
Review Exercise 113
True or False 113
Multiple Choice Questions 113
Descriptive Questions 114

Chapter 4: Macro and Macro Processors 119

4.1 Macro and Macro Processors 120
4.1.1 Differences between Macro and Subroutine 121
4.2 Macro Definition and Call 123
4.3 Macro Expansion 124
4.3.1 Lexical Expansion and Parameter Substitution 125
4.3.2 Flow of Control during Expansion 129
4.4 Nested Macro Facility 130
4.4.1 Nested Macro Definitions 130
4.4.2 Nested Macro Expansions 131
4.4.3 Recursive Macro Calls 133
4.5 Advanced Macro Facilities 134
4.5.1 SET Directive and Symbols 134
4.5.2 Facilities to Alter Flow of Control 135
4.5.3 Facilities to Generate Expansion Time Variables 136
4.5.4 Attributes of Macro Arguments 137
4.5.5 Some More Advanced Directives 139
4.6 Design of Macro Preprocessor 141
4.7 Design of Macro Assembler 143
4.8 Functions of Macro Processor 143
4.9 Basic Tasks of Macro Processor 144
4.9.1 Handling Macro Definition 144
4.9.2 Handling Macro Expansion 144

4.10 Design Features and Issues of Macro Processor 145
4.11 Macro Processor Design Options 146
4.11.1 Recursive Macro Expansion 146
4.11.2 General-purpose Macro Processors 146
4.11.3 Macro Processing within Language Translators 147
4.11.4 Line-by-Line Macro Processor 147
4.12 Two-pass Macro Processors 147
4.12.1 Databases 147
4.12.2 Design of Two-pass Macro Preprocessor (Pass 0 of Assembler) 149
4.12.3 Design of Nested Macro Expansion (Pass 0 of Assembler) 152
4.13 One-pass Macro Processors 153
Summary 154
Key Terms 156
Review Exercise 156
True or False 156
Multiple Choice Questions 157
Descriptive Questions 157

Chapter 5: Linkers and Loaders 161

5.1 Basic Linker and Loader Functions 162
5.2 Relocation and Linking Concepts 163
5.2.1 Types of Program Address 164
5.2.2 Performing Relocation 165
5.2.3 Program Linking 166
5.2.4 Object Files 169
5.3 Design of Linker 171
5.3.1 Relocation of Object Module 171
5.3.2 Linking of Object Modules 172
5.4 Relocating and Self-Relocating Programs 174
5.5 Linking in MS DOS 175
5.5.1 Two-pass Linking 177
5.6 Linking of Overlay Structured Programs 180

5.7 Dynamic Linking Loaders 181
5.7.1 Dynamic Linking 181
5.7.2 Dynamic Loading 182
5.8 Different Loading Schemes 183
5.8.1 Compile-and-Go Loading Scheme 183
5.8.2 General Loading Scheme 184
5.8.3 Absolute Loading Scheme 185
5.8.4 Relocating Loaders 186
5.8.5 Practical Relocating Loaders 187
5.8.6 Linking Loaders 188
5.8.7 Relocating Linking Loaders 188
5.8.8 Binder Program 189
5.9 Design of Absolute Loaders 189
5.10 Design of Direct-Linking Loaders 190
5.10.1 Passes Structure of Direct-Linking Loaders 191
5.10.2 Linker Command Languages 194
Summary 194
Key Terms 196
Review Exercise 196
True or False 196
Multiple Choice Questions 197
Descriptive Questions 197

Chapter 6: Scanning and Parsing 203

6.1 Programming Language Grammar 204
6.1.1 Elements of Programming Languages 204
6.2 Classification of Grammar 206
6.2.1 Chomsky Classification of Grammar 206
6.2.2 Simple Grammar and Operator Grammar 209
6.2.3 Derivation Tree 210
6.2.4 Left and Right Derivations 213
6.3 Ambiguity in Grammar Specifications 214
6.3.1 Removing Ambiguity 216

6.4 Scanning 218
6.4.1 Specification of Tokens 219
6.4.2 Regular Expression 219
6.4.3 Arden's Rule 220
6.4.4 Recognition of Tokens 222
6.4.5 Transition Diagrams 223
6.4.6 Finite-State Automata 225
6.4.7 Finite Automata from Regular Expressions 228
6.4.8 Converting NFA to DFA 230
6.4.9 Designing a Lexical Analyzer Generator 234
6.4.10 Optimizing DFA 236
6.5 Parsing 240
6.5.1 Types of Parsing 241
6.6 Top-down Parsing 241
6.6.1 Top-down Parsing Based on Prediction and Backtracking 243
6.6.2 Recursive Productions 245
6.6.3 Left Factoring 246
6.6.4 Recursive Descent and Predictive Parser 247
6.6.5 LL Parsers 249
6.7 Bottom-up Parser 255
6.7.1 Shift-reduce Parsers 256
6.7.2 Precedence Parsers 258
6.7.3 Simple Precedence Parsing 259
6.7.4 Operator Precedence Parsing 267
6.8 Language Processor Development Tools 276
6.8.1 LEX 277
6.8.2 YACC 280
Summary 282
Key Terms 284
Review Exercise 284
True or False 284
Multiple Choice Questions 285
Descriptive Questions 285

Chapter 7: Compilers..... 291

7.1 Compilers..... 292
7.1.1 Stages of Compiler..... 293
7.1.2 Types of Compilers..... 299
7.1.3 Front-End and Back-End of Compiler..... 302
7.1.4 The Pass Structure of Compiler..... 304
7.1.5 Compiler Writing Tools..... 305
7.2 Causes of Large Semantic Gap..... 307
7.3 Binding and Binding Time..... 308
7.4 Scope Rules..... 309
7.4.1 Static Scope Rules..... 309
7.4.2 Dynamic Scope Rules..... 310
7.5 Data Structures Used in Compiling..... 311
7.5.1 Tokens..... 311
7.5.2 Syntax Tree..... 311
7.5.3 Symbol Table..... 313
7.5.4 Literal Table..... 314
7.5.5 Temporary Files..... 314
7.5.6 Intermediate Code..... 314
7.6 Memory Allocation..... 315
7.6.1 Static Allocation..... 316
7.6.2 Dynamic Allocation..... 316
7.6.3 Allocation of Simple Data Objects..... 317
7.6.4 Activation and Activation Trees..... 319
7.6.5 Activation Records..... 320
7.6.6 Activation Record for Procedure Calls..... 321
7.6.7 Handling Recursive Calls..... 322
7.7 Compilation of Expressions..... 323
7.7.1 Abstract Syntax Trees..... 324
7.7.2 Direct Acyclic Graphs..... 325
7.7.3 Postfix Code..... 326
7.7.4 Three-address Code..... 328
7.7.5 Three-Address Code Generation..... 333

7.8 Compilation of Control Structure 341
7.8.1 Conditional and Iterative Control Flow 341
7.8.2 Code Generation for Function Calls 344
7.9 Code Optimization 345
7.9.1 Classification of Compiler Optimization 346
7.9.2 Important Optimizing Transformations 347
7.9.3 Global Data Flow Analysis 354
7.9.4 Optimizations Based on Data Flow Analysis 357
Summary 364
Key Terms 366
Review Exercise 366
True or False 366
Multiple Choice Questions 366
Descriptive Questions 367

Chapter 8: Interpreters and Debuggers 371

8.1 Overview of Interpretation 372
8.1.1 Comparison between Compilers and Interpreters 372
8.1.2 Comparing the Performance of Compilers and Interpreters 374
8.2 Benefits of Interpretation 374
8.3 Java Language Environment 375
8.3.1 The Java Compiler: javac 376
8.3.2 Overview of Compilation 377
8.3.3 Different Approaches of Java Compilation 380
8.3.4 A Look into Java Bytecode 380
8.4 Java Virtual Machine 382
8.5 Types of Errors 383
8.5.1 Syntax Error 384
8.5.2 Semantic Error 384
8.5.3 Logical Error 385
8.5.4 Errors Detected by Compiler and Runtime Errors 386
8.5.5 Error Analysis 386

8.6 Debugging Procedures 387
8.6.1 Debug Monitors 387
8.6.2 Assertions 389
8.7 Classification of Debugging 390
8.7.1 Static Debugging 390
8.7.2 Dynamic Analysis or Interactive Debugging 390
Summary 392
Key Terms 393
Review Exercise 394
True or False 394
Multiple Choice Questions 394
Descriptive Questions 395

Lab 399

Index 417