

Table of Contents

About the Authors	iii
Introduction	xvii
Chapter 1: System Software	1
1.1 Concept of System Software	2
Types of Software Programs.....	2
Software Programs and the Computing Machine.....	2
Introduction to Language Processing.....	3
1.2 Introduction to Various Systems Programs	5
1.3 Assembler.....	5
1.4 Loader and Linker.....	7
1.5 Macro Processors	8
1.6 Compiler	9
Analysis Phase: The Front End of the Compiler.....	9
Synthesis Phase: The Back-End of the Compiler	10
1.7 Interpreters	10
1.8 Operating System	12
1.9 Device Drivers	13
Writing Device Drivers	14
Interfacing Functions between User Space and Kernel Space	15
Summary.....	16
Review Exercise.....	18
True and False	18
Multiple Choice Questions	19
Descriptive Questions	20
Chapter 2: Assemblers.....	23
2.1 Basic Functions of Assemblers	24
2.2 General Design Procedure.....	25
Assembly Language Programming.....	25
General Format of an Assembly Language Statement	26
Types of Assembly Language Statements.....	30
Advantages and Disadvantages of Assembly Language Programming	34

2.3	Design of Assembler and Overview of Assembly Process	35
	Analysis and Synthesis.....	35
	Task Performed by the Analysis and Synthesis Phases of Assembly Process	37
	Single Pass and Multi pass Assemblers.....	37
2.4	Single Pass Assembler for IBM PC	38
	Intel 8086/8088 Architecture.....	38
	Addressing Modes and Instruction Formats	40
	Instruction Decoding during Assembly.....	42
	Assembler Directives	44
	Writing Segment-Based Program	45
	Advanced Intel Assemblers.....	46
	Registers in a 64-bit Architecture	48
2.5	Design of Multipass Assembler	48
	Pass I of Assembler	48
	Pass II of Assembler	55
2.6	IBM 360/370 Processors.....	58
	Statement of the Problem	70
	Data Structure and Databases	72
	Format of Databases	73
	Algorithms	75
2.7	Comparison of Single-Pass and Two-Pass Assemblers.....	77
2.8	Output Interface of Assembler.....	77
	Summary.....	79
	Review Exercise.....	81
	True and False	81
	Multiple Choice Questions	82
	Descriptive Questions	83
Chapter 3: Macro and Macro Processors		85
3.1	Macro and Macro Processors	86
	Differences between Macro and Subroutines	87
3.2	Macro Instruction-Definition and Call	88
	Macro Expansion	89
	Handling Macro Calls.....	90

3.3	Features of Macro Facility.....	91
	Lexical Expansion and Parameter Substitution.....	91
	Flow of Control during Expansion.....	95
	Nested Macro Facility.....	96
	Recursive Macro Calls.....	99
	Advanced Macro Facilities.....	99
	Facilities to Alter Flow of Control.....	100
	Facilities to Generate Expansion Time Variables.....	102
	Attributes of Macro Arguments.....	102
	Some More Advanced Directives.....	105
3.4	Design of Two-pass Macro Processors.....	106
	Functions of a Macro Preprocessor.....	107
	Databases for Macro Processing.....	107
	Design of Two-pass Macro Preprocessor (Pass-0 of Assembler).....	109
	Design of Nested Macro Expansion (Pass-0 of Assembler).....	111
	Design of Macro Assemblers.....	112
	Summary.....	113
	Review Exercise.....	115
	True and False.....	115
	Multiple Choice Questions.....	115
	Descriptive Questions.....	116
	Chapter 4: Loaders and Linkers.....	120
4.1	Loader and Linkers.....	120
	Functions of Loader.....	120
4.2	Loader Schemes.....	121
	Compile and Go Loaders.....	121
	General Loading Scheme.....	122
	Absolute Loading Scheme.....	123
	Relocating Loader Scheme.....	125
	Linking Loader Scheme.....	126
	Other Loading Schemes.....	126
4.3	More into Relocation and Linking.....	127
	Types of Program Address.....	127

	Relocating Programs	128
	Performing Relocation	129
	Program Linking	130
	Object Files.....	132
4.4	Design of Loader and Linker	134
	Relocation of Object Module	134
	Linking of Object Modules.....	135
	Linker for MS DOS.....	137
	Two-Pass Linking.....	138
	Linking for Overlays.....	141
4.5	Design of Absolute Loaders	142
4.6	Design of Direct-Linking Loaders	143
	Passes Structure of Direct-Linking Loaders	144
	Linker Command Languages.....	146
	Summary.....	147
	Review Exercise.....	149
	True and False	149
	Multiple Choice Questions	149
	Descriptive Questions	151
Chapter 5: Software Tool		153
5.1	Steps Involved in Program Development.....	153
5.2	Software Tools	154
	Functions Provided by Software Tools	155
5.3	Tools Used in Program Development.....	155
	Tools Used for Designing and Coding	155
	Tools Used for Program Entry and Editing	157
	Tools Used for Program Testing.....	157
	Tools Used for Debugging.....	160
	Tools Used for Performance Tuning.....	164
5.4	Editors	165
	Types of Editors	165
	Features Provided by an Editor	169
	Design of an Editor	171
	Collaborative Editors	173

Issues Related to Editors	174
Developing Collaborative Software.....	176
Summary	177
Review Exercise.....	177
True and False	177
Multiple Choice Questions	178
Descriptive Questions	179
Chapter 6: Compiler.....	181
6.1 Overview of Translation Process	181
6.2 Introduction to Compiler	183
Types of Compilers.....	184
Analysis-Synthesis Model Compilation	187
The Pass Structure of Compiler.....	188
Compiler Writing Tools	189
6.3 The Phases of a Compiler	191
Lexical Analysis Phase.....	192
Syntax Analysis Phase.....	193
Semantic Analysis and Interpretation Phase	194
Code Optimization Phase.....	196
Storage Allocation Phase.....	197
Code Generation Phase	197
Code Assembly Phase	198
Symbol Table and Error Handling Phase	198
The Grouping of Phases: Front End and Back End of Compiler	200
6.4 Difference between an Interpreter, Assembler, and Compiler	202
6.5 Cousins of the Compiler.....	205
Preprocessors.....	205
Assembler	207
Loader and Linkers.....	208
Summary	209
Review Exercise.....	211
True and False	211
Multiple Choice Questions	212
Descriptive Questions	213

Chapter 7: Lexical Analysis	215
7.1 Role of Lexical Analyzer	216
7.2 Input Buffering	217
7.3 Specification of Tokens	218
Tokens, Patterns, and Lexemes	218
Elements of Token Design	220
Chomsky Classification of Grammar	222
Regular Expression	224
7.4 Recognition of Tokens	227
Transition Diagrams	228
Finite-State Automata	230
Finite Automata from Regular Expressions	233
Converting NFA to DFA	236
Building DFAs	239
7.5 Designing a Lexical Analyzer Generator	239
Recognizer for Identifiers	240
Language for Specifying Lexical Analyzer	241
7.6 Pattern Matching Using NFA	244
DFAs for Lexical Analyzers	245
7.7 Optimizing of DFA	246
Summary	250
Review Questions	251
True and False	251
Multiple Choice Questions	252
Descriptive Questions	253
Chapter 8: Syntax Analysis	255
8.1 Role of Parser	255
Types of Parsing: Top-Down and Bottom-Up Parsing Algorithms	256
8.2 Top-Down Parsing	257
Top-Down Parsing Based on Prediction and Backtracking	258
Recursive Productions	260
Left Factoring	261
8.3 Recursive Descent and Predictive Parser	262
8.4 LL Parsers	264

	LL(1) Parsers	265
8.5	Bottom-Up Parser.....	269
	Shift-Reduce Parsers	270
	Precedence Parsers	272
8.6	Simple Precedence Parsing.....	273
	Computing Simple Precedence Relations--Method I	274
	Computing Simple Precedence Relations --Method II	276
	Bottom-Up Parsing Algorithm Using SPM	278
8.7	Operator Precedence Parsing	280
8.8	Precedence Functions.....	289
	Computing Precedence Function—Method I	289
	Computing Precedence Functions—Method II.....	291
8.9	LR Parsers	295
	LR Configuration and Tracing	297
	LR Parsing Algorithm	297
	Types of LR Parsers.....	300
8.10	LR(0) Parsing	301
	Computing CLOSURE set (or configuration set)	302
	Creating Complete CLOSURE Set C.....	303
	Construction of LR(0) Parsing Table.....	304
8.11	SLR(1) Parsers.....	307
	Limitations of SLR(1) Parsers.....	311
8.12	LR(1) Parsers	311
	Construction of Canonical LR Parsing Table.....	312
8.13	LALR(1) Parser	316
	Construction of LALR(1) Parsing Table	316
8.14	Parser Generators.....	319
8.15	Using Ambiguous Grammar with LR Parsers.....	322
	Summary	327
	Review Exercise.....	329
	True and False	329
	Multiple Choice Questions	330
	Descriptive Questions	331

Chapter 9: Syntax-Directed Translation	337
9.1 Introduction to Syntax-Directed Translation	338
9.2 Syntax-Directed Definitions	338
9.3 Inherited and Synthesized Attributes	339
9.4 Evaluation Order of SDD	343
Dependency Graphs	343
Type Expressions	345
Construction of Syntax Trees	345
S-Attributed Definitions	349
Bottom-Up Evaluation of S-Attributed Definitions	349
9.5 L-Attributed Definitions	350
9.6 Syntax-Directed Translation Scheme	352
9.7 Error Detection and Recovery	354
Error Detection	355
Error Reporting	355
Adhoc and Systematic Methods	356
Summary	358
Review Exercise	359
True and False	359
Multiple Choice Questions	360
Descriptive Questions	361
Chapter 10: Intermediate Code Generation	363
10.1 Intermediate Languages	364
High-Level IRs	365
Low-Level IRs	370
10.2 Intermedia Code for Assignment Statements	374
10.3 Intermediate Code Generation for Arrays	376
10.4 Intermediate Code for Boolean Expressions	378
10.5 Back Patching	379
10.6 Conditional and Iterative Control Flow	382
10.7 Code for Switch Statements	385
10.8 Computing l- and r-Values	386
10.9 Procedure Calls	386
10.10 Translation of Mixed-Mode Expressions	388
Summary	389

Review Exercise.....	390
True or false.....	390
Multiple Choice Questions.....	391
Descriptive Questions.....	392
Chapter 11: Code Generation.....	395
11.1 Major Tasks in Code Generation.....	396
11.2 Design Issues of a Code Generator.....	397
Input for Code Generators.....	397
Output from the Code Generator.....	398
Memory and Reference Management.....	398
Choice of Instruction to Generate.....	399
Register Allocation.....	401
Choice of Evaluation Order.....	402
11.3 Generating Code for the Target Machine.....	402
Addressing Modes.....	402
Program and Instruction Cost.....	403
11.4 Runtime Storage Management for Code Generation.....	404
Stack Allocation.....	404
Caller Calling Callee.....	405
Callee Returning.....	405
11.5 Basic Blocks and Flow Graphs.....	405
Basic Blocks.....	405
Flow Graphs.....	407
Next-Use Information.....	408
11.6 Code generation Algorithm.....	410
The Code Generation Algorithm.....	411
Register Allocation and Assignments.....	411
Machines Implementation of Conditional Jumps.....	412
11.7 The DAG Representation of Basic Blocks.....	413
11.8 Generating Code from DAGs.....	415
Rearranging Order of the Code.....	415
A More Practical Code Generation from DAG.....	416
11.9 Dynamic Programming Code-Generation Algorithm.....	419
11.10 Code-Generator Generators.....	420

Summary	421
Review Exercise.....	422
True and False	422
Multiple Choice Questions	422
Descriptive Questions	424
Chapter 12: Code Optimization	427
12.1 Classification of Compiler Optimization.....	428
12.2 Principal Sources of Optimization for Compiler	429
Compile-Time Evaluation.....	429
Common Sub-Expression Elimination	430
Variable Propagation.....	430
Code Movement	431
Dead Code Elimination.....	432
Loop Optimization.....	432
Strength Reduction	435
12.3 Global Data Flow Analysis	435
Available Expressions	436
Reaching Definitions.....	436
Data-Flow Equations	437
Use-Definition (U-D) Chaining.....	438
12.4 Loops in Flow Graphs.....	438
12.5 Optimizations of Basic Blocks	439
Invariant Code Motion	439
Common Sub-Expression Elimination and Flow Analysis	440
Dead Code Elimination.....	441
Loop Unrolling	441
Induction Variable Elimination	442
Busy Expressions.....	442
12.6 Peephole Optimization.....	443
Eliminating Redundant Loads and Stores	444
Identifying Unreachable Instructions.....	445
Simplify Algebraic Expressions.....	446
Strength Reduction	446

Use Faster Machine Instructions	447
Elimination of Multiple Jumps	447
Summary	448
Review Exercise.....	450
True and False	450
Multiple Choice Questions	450
Descriptive Questions	451
Chapter 13: Run-Time Storage.....	455
13.1 Source Language Issues	456
13.2 Storage Organization.....	457
The Run-Time Storage Organization	457
Allocation of Simple Data Objects	458
Activation and Activation Trees.....	459
Activation Records	461
Activation Record for Procedure Calls	462
13.3 Storage Allocation Strategies	463
Static Allocation	463
Stack Allocation	464
Heap Allocation.....	465
Handling Recursive Calls	465
13.4 Management of Variable-Length Blocks.....	467
Run-Time Allocation Using Stacks	467
Run-Time Allocation Using Heaps.....	469
13.5 Passing Parameters	469
13.6 Symbol Table.....	470
Uses of Symbol Table.....	471
Content of the Symbol Table.....	471
Data Structures for Symbol Table.....	472
13.7 Introduction to Garbage Collection and Compaction.....	475
Dangling References	475
Garbage	475
Garbage Collection Algorithms.....	475
Compacting Collectors	476
Generational Collectors	478

Advantages and Disadvantages of Garbage Collection.....	478
13.8 Dynamic Storage Allocation Techniques.....	479
Managing Implicit Deallocation	479
Explicit Allocation of Fixed-Sized Blocks	479
Explicit Allocations of Variable-Sized Blocks	479
Summary.....	480
Review Exercise.....	482
True and False	482
Multiple Choice Questions	482
Descriptive Questions	484
Chapter 14: Compiler-Compilers.....	485
14.1 Java Compiler Environment	485
The Java Compiler: javac.....	486
Overview of Compilation.....	487
Different Approaches of Java Compilation	489
A Look into Java Bytecode	490
14.2 YACC Compiler-Compiler.....	492
Working of Yacc.....	492
The Yacc File Format	493
Summary.....	494
Review Exercise.....	495
True and False	495
Multiple Choice Questions	495
Descriptive Questions	497
Lab	499
Index	549