

DAVID MOON
RICHARD M. STALLMAN
DANIEL WEINREB

CHINESE NUAL

Lisp Machine Manual

Fifth Edition, System Version 92

January 1983

Daniel Weinreb
David Moon
Richard Stallman

This report describes research done at the Artificial Intelligence Laboratory of the Massachusetts Institute of Technology. Support for the laboratory's artificial intelligence research is provided in part by the Advanced Research Projects Agency of the Department of Defense under Office of Naval Research Contract number N00014-80-C-0505.

Preface

The Lisp Machine manual describes both the language and the "operating system" of the Lisp Machine. The language, a dialect of Lisp called Zetalisp, is completely documented by this manual. The software environment and operating-system-like parts of the system contain many things which are still in a state of flux. This manual confines itself primarily to the stabler parts of the system, and does not address the window system and user interface at all. That documentation will be released as a separate volume at a later time.

Any comments, suggestions, or criticisms will be welcomed. Please send Arpa network mail to `BUG-IMMAN@MIT-OZ`.

Those not on the Arpanet may send U.S. mail to
Richard M. Stallman
545 Technology Square, Room 793
Cambridge, Mass. 02139

Note

The Lisp Machine is a product of the efforts of many people too numerous to list here and of the unique environment of the M.I.T. Artificial Intelligence Laboratory.

Portions of this manual were written by Mike McMahon and Alan Bawden. The chapter on the LOOP iteration macro is a reprint of Laboratory for Computer Science memo TM-169, by Glenn Burke.

Summary Table of Contents

1. Introduction	1
2. Primitive Object Types	7
3. Evaluation	14
4. Flow of Control.	39
5. Manipulating List Structure.	60
6. Symbols	96
7. Numbers.	102
8. Arrays	121
9. Strings	143
10. Functions	154
11. Closures.	180
12. Stack Groups	186
13. Locatives	197
14. Subprimitives	200
15. Areas	223
16. The Compiler	228
17. Macros	248
18. The LOOP Iteration Macro	274
19. Defstruct	298
20. Objects, Message Passing, and Flavors	321
21. The I/O System	362
22. Naming of Files	453
23. The Chaosnet	484
24. Packages	506
25. Maintaining Large Systems	520
26. Processes	538
27. Errors and Debugging	552
28. How to Read Assembly Language	602
29. Querying the User	620
30. Initializations	624
31. Dates and Times.	628
32. Miscellaneous Useful Functions	635
Concept Index.	663
Flavor Index.	666
Operation (Message Name) Index	667
Keyword Index	669
Object Creation Options.	673
Meter Index.	674
Variable Index.	675
Function Index	679
Condition Name Index	692

Table of Contents

1. Introduction	1
1.1 General Information	1
1.2 Structure of the Manual	1
1.3 Notational Conventions and Helpful Notes	2
2. Primitive Object Types	7
2.1 Data Types	7
2.2 Predicates	8
3. Evaluation	14
3.1 Variables	15
3.2 Functions	21
3.3 Some Functions and Special Forms	26
3.4 Tail Recursion	32
3.5 Multiple Values	33
3.6 Evaluator Errors	36
4. Flow of Control	39
4.1 Conditionals	39
4.2 Iteration	45
4.3 Non-Local Exits	54
4.4 Mapping	57
5. Manipulating List Structure	60
5.1 Conses	61
5.2 Lists	62
5.3 Alteration of List Structure	70
5.4 Cdr-Coding	72
5.5 Tables	74
5.6 Lists as Tables	75
5.7 Association Lists	78
5.8 Stack Lists	80
5.9 Property Lists	81
5.10 Hash Tables	83
5.10.1 Hash Table Functions	85
5.10.2 Hash Tables and the Garbage Collector	87
5.10.3 Hash Primitive	87
5.11 Sorting	89
5.12 Resources	91
5.12.1 Defining Resources	92
5.12.2 Allocating Resource Objects	94
5.12.3 Accessing the Resource Data Structure	95
6. Symbols	96
6.1 The Value Cell	96
6.2 The Function Cell	98
6.3 The Property List	99
6.4 The Print Name	99
6.5 The Package Cell	100

6.6 Creating Symbols	100
7. Numbers	102
7.1 Numeric Predicates	105
7.2 Numeric Comparisons	106
7.3 Arithmetic	107
7.4 Complex Number Functions	111
7.5 Transcendental Functions	112
7.6 Numeric Type Conversions	113
7.7 Logical Operations on Numbers	114
7.8 Byte Manipulation Functions	116
7.9 Random Numbers	118
7.10 24-Bit Numbers	119
7.11 Double-Precision Arithmetic	119
8. Arrays	121
8.1 Extra Features of Arrays	124
8.1.1 Displaced Arrays	125
8.2 Basic Array Functions	126
8.3 Getting Information About an Array	130
8.4 Changing the Size of an Array	132
8.5 Arrays Overlaid With Lists	133
8.6 Adding to the End of an Array	133
8.7 Copying an Array	134
8.8 Order of Array Elements	137
8.9 Matrices and Systems of Linear Equations	138
8.10 Planes	139
8.11 Maclisp Array Compatibility	141
9. Strings	143
9.1 Characters	144
9.2 Upper and Lower Case Letters	144
9.3 Basic String Operations	145
9.4 String Searching	148
9.5 I/O to Strings	150
9.6 Maclisp-Compatible Functions	152
10. Functions	154
10.1 What Is a Function?	154
10.2 Function Specs	154
10.3 Simple Function Definitions	157
10.4 User Operations on Functions	159
10.5 Kinds of Functions	160
10.5.1 Interpreted Functions	161
10.5.2 Lambda Macros	162
10.5.3 Compiled Functions	163
10.5.4 Other Kinds of Functions	163
10.5.5 Special Forms and Functions	164
10.6 Function-Defining Special Forms	165
10.7 Lambda-List Keywords	168
10.8 How Programs Manipulate Function Specs	169

10.9	How Programs Examine Functions	172
10.10	Encapsulations	175
10.10.1	Rename-Within Encapsulations	179
11.	Closures	180
11.1	What a Closure Is	180
11.2	Examples of the Use of Closures	182
11.3	Closure-Manipulating Functions	183
11.4	Entities	185
12.	Stack Groups	186
12.1	Resuming of Stack Groups	187
12.2	Stack Group States	188
12.3	Stack Group Functions	189
12.4	Analyzing Stack Frames	191
12.5	Input/Output in Stack Groups	194
12.6	An Example of Stack Groups	194
13.	Locatives	197
13.1	Cells and Locatives	197
13.2	Functions That Operate on Locatives	198
13.3	Mixing Locatives with Lists	198
14.	Subprimitives	200
14.1	Data Types	201
14.2	Forwarding	202
14.3	Pointer Manipulation	204
14.4	Analyzing Structures	204
14.5	Creating Objects	205
14.6	Copying Data	206
14.7	Returning Storage	206
14.8	Locking Subprimitive	207
14.9	I/O Device Subprimitives	207
14.10	Special Memory Referencing	208
14.11	Storage Layout Definitions	210
14.12	Function-Calling Subprimitives	211
14.13	Special-Binding Subprimitive	212
14.14	The Paging System	212
14.15	Closure Subprimitives	215
14.16	Microcode Variables	215
14.17	Meters	219
15.	Areas	223
15.1	Area Functions and Variables	224
15.2	Interesting Areas	226
15.3	Errors Pertaining to Areas	227
16.	The Compiler	228
16.1	The Basic Operations of the Compiler	228
16.2	How to Invoke the Compiler	228
16.3	Input to the Compiler	230
16.4	Compiler Declarations	233

16.5 Using Compiler Warnings	238
16.5.1 Controlling Compiler Warnings	239
16.5.2 Recording Warnings	241
16.6 Compiler Source-Level Optimizers	243
16.7 Files that Maclisp Must Compile	244
16.8 Putting Data in QFASL Files	245
16.9 Analyzing QFASL Files	247
16.10 Compiler Interlocking	247
17. Macros	248
17.1 Introduction to Macros	248
17.2 Aids for Defining Macros	250
17.2.1 Defmacro	250
17.2.2 Backquote	251
17.3 Substitutable Functions	255
17.4 Hints to Macro Writers	256
17.4.1 Name Conflicts.	257
17.4.2 prog-context Conflicts	259
17.4.3 Macros Expanding into Many Forms	260
17.4.4 Macros that Surround Code.	261
17.4.5 Multiple and Out-of-order Evaluation.	263
17.4.6 Nesting Macros.	265
17.4.7 Functions Used During Expansion	266
17.5 Aids for Debugging Macros	267
17.6 Displacing Macros.	267
17.7 Advanced Features of Defmacro.	268
17.8 Functions to Expand Macros	269
17.9 Generalized Variables.	270
18. The LOOP Iteration Macro	274
18.1 Introduction.	274
18.2 Clauses	275
18.2.1 Iteration-Driving Clauses.	276
18.2.2 Bindings	278
18.2.3 Entrance and Exit	280
18.2.4 Side Effects.	280
18.2.5 Values	281
18.2.6 Endtests	282
18.2.7 Aggregated Boolean Tests	283
18.2.8 Conditionalization	283
18.2.9 Miscellaneous Other Clauses	285
18.3 Loop Synonyms.	286
18.4 Data Types	286
18.5 Destructuring	287
18.6 The Iteration Framework	288
18.7 Iteration Paths.	289
18.7.1 Pre-Defined Paths	291
18.7.1.1 The Interned-Symbols Path	291
18.7.1.2 Sequence Iteration.	291
18.7.2 Defining Paths	293

18.7.2.1 An Example Path Definition	295
19. Defstruct.	298
19.1 Introduction to Structure Macros.	298
19.2 How to Use Defstruct	300
19.3 Options to Defstruct	301
19.4 Using the Constructor and Alterant Macros.	308
19.4.1 Constructor Macros.	308
19.4.2 By-position Constructor Macros.	309
19.4.3 Alterant Macros.	310
19.5 Byte Fields	310
19.6 Grouped Arrays	312
19.7 Named Structures	312
19.8 The si:defstruct-description Structure.	315
19.9 Extensions to Defstruct	316
19.9.1 An Example	316
19.9.2 Syntax of defstruct-define-type	317
19.9.3 Options to defstruct-define-type.	317
20. Objects, Message Passing, and Flavors	321
20.1 Objects	321
20.2 Modularity	322
20.3 Generic Operations	325
20.4 Generic Operations in Lisp.	326
20.5 Simple Use of Flavors	327
20.6 Mixing Flavors	331
20.7 Flavor Functions.	334
20.8 Defflavor Options	342
20.9 Flavor Families	348
20.10 Vanilla Flavor	348
20.11 Method Combination.	350
20.12 Implementation of Flavors	355
20.12.1 Order of Definition	356
20.12.2 Changing a Flavor	357
20.12.3 Restrictions	357
20.13 Entities.	357
20.14 Useful Editor Commands.	358
20.15 Property List Operations	359
20.16 Printing Flavor Instances Readably	360
21. The I/O System	362
21.1 The Character Set	362
21.2 Printed Representation.	367
21.2.1 What the Printer Produces.	367
21.2.2 What The Reader Accepts.	371
21.2.3 Macro Characters.	374
21.2.4 Sharp-sign Constructs.	374
21.2.5 Special Character Names	378
21.2.6 The Readtable	379
21.3 Input Functions	382

21.4	Output Functions	388
21.5	I/O Streams.	391
21.5.1	Standard Input Stream Operations	391
21.5.2	Standard Output Stream Operations	393
21.5.3	Asking Streams What They Can Do.	394
21.5.4	Operations for Interactive Streams	395
21.5.5	Cursor Positioning Stream Operations.	396
21.5.6	Operations for Efficient Grinding.	397
21.5.7	Random Access File Operations	397
21.5.8	Buffered Stream Operations	398
21.5.9	Standard Streams.	399
21.5.10	Obtaining Streams to Use	400
21.5.11	Implementing Streams.	403
21.5.12	Implementing Streams with Flavors	403
21.5.13	Implementing Streams Without Flavors	408
21.6	Formatted Output.	411
21.6.1	The Format Function.	411
21.6.2	The Output Subsystem	422
21.6.3	Formatting Lisp Code	426
21.7	Rubout Handling	427
21.8	The :read and :print Stream Operations	430
21.9	Accessing Files	431
21.9.1	Loading Files.	437
21.9.2	File Attribute Lists	438
21.9.3	File Stream Operations.	441
21.10	Accessing Directories.	442
21.11	Errors in Accessing Files	447
21.12	File Servers	450
21.12.1	Errors in Communication with File Servers	452
22.	Naming of Files	453
22.1	Pathnames	453
22.2	Pathname Components	455
22.2.1	Raw Components and Interchange Components.	457
22.2.2	Pathname Component Operations	458
22.2.3	Canonical Types	459
22.3	Defaults and Merging	462
22.4	Pathname Functions.	465
22.5	Generic Pathnames	467
22.6	Pathname Operations	469
22.7	Host File Systems Supported	472
22.7.1	ITS	473
22.7.2	TOPS-20 (Twenex), Tenex, and VMS.	474
22.7.3	Unix pathnames	475
22.7.4	Lisp Machine File Systems	476
22.7.5	Logical Pathnames	477
22.7.6	Editor Buffer Pathnames	479
22.8	Hosts	480
22.8.1	Parsing Hostnames	480

22.8.2 Host Object Operations	482
22.9 Maclisp Conversion	483
23. The Chaosnet	484
23.1 Chaosnet Overview	484
23.2 Conns	487
23.3 Opening and Closing Connections	488
23.3.1 User-Side.	488
23.3.2 Server-Side	489
23.4 Stream Input and Output	490
23.5 Packet Input and Output.	491
23.6 Connection Interrupts	494
23.7 Chaosnet Errors	494
23.7.1 Local Problems	495
23.7.2 Problems Involving Other Machines' Actions	495
23.8 Information and Control.	496
23.9 Higher-Level Protocols.	497
23.9.1 Status.	498
23.9.2 Telnet and Supdup	499
23.9.3 File Access	499
23.9.4 Mail	499
23.9.5 Send	500
23.9.6 Name.	500
23.9.7 Time	501
23.9.8 Arpanet Gateway	501
23.9.9 Host Table	501
23.9.10 Dover	502
23.9.11 Remote Disk.	503
23.9.12 The Eval Server	503
23.10 Using Higher Level Protocols	504
24. Packages	506
24.1 The Organization of Name Spaces	507
24.2 Shared Programs.	508
24.3 Declaring Packages	509
24.4 Packages and Writing Code	510
24.5 Shadowing.	511
24.6 Packages and Interning.	512
24.7 Status Information.	515
24.8 Packages, Loading, and Compilation	516
24.9 Subpackages.	516
24.10 Initialization of the Package System	518
24.11 Initial Packages.	519
25. Maintaining Large Systems.	520
25.1 Defining a System	520
25.2 Transformations	523
25.3 Making a System.	526
25.4 Adding New Keywords to make-system	527
25.5 Adding New Options for defsystem.	529

25.6 More Esoteric Transformations	530
25.7 The Patch Facility	531
25.7.1 Defining a System	533
25.7.2 Patch files	534
25.7.3 Loading Patches	535
25.7.4 Making Patches	535
25.7.5 System Status	536
26. Processes	538
26.1 The Scheduler.	539
26.2 Locks	542
26.3 Creating a Process.	544
26.4 Process Generic Operations	545
26.4.1 Process Attributes	546
26.4.2 Run and Arrest Reasons	547
26.4.3 Bashing the Process.	548
26.5 Process Flavors	549
26.6 Other Process Functions.	550
27. Errors and Debugging	552
27.1 Conditions	552
27.2 Handling Conditions	554
27.3 Standard Condition Flavors	558
27.4 Condition Operations	559
27.4.1 Condition Operations for the Debugger.	560
27.5 Signaling Conditions	561
27.5.1 Convenience Functions for Signaling	562
27.5.2 Creating Condition Instances	565
27.5.3 Signaling a Condition Instance	568
27.6 Proceeding	569
27.6.1 Proceeding and the Debugger.	571
27.6.2 How Signalers Provide Proceed Types	574
27.6.3 Nonlocal Proceed Types	575
27.7 The Debugger.	578
27.7.1 Entering the Debugger	578
27.7.2 How to Use the Debugger	579
27.7.3 Debugger Commands	581
27.7.4 Summary of Commands	586
27.7.5 Deexposed Windows and Background Processes	587
27.8 Tracing Function Execution	588
27.9 Breakon.	591
27.10 Advising a Function	592
27.10.1 Designing the Advice	594
27.10.2 :around Advice	595
27.10.3 Advising One Function Within Another	595
27.11 Stepping Through an Evaluation	596
27.12 Evalhook.	598
27.13 The MAR	599
27.14 Variable Monitoring	601

28. How to Read Assembly Language	602
28.1 Introduction	602
28.2 A More Advanced Example	605
28.3 The Rest of the Instructions	608
28.4 Function Entry	611
28.5 Special Class IV Instructions	613
28.6 Estimating Run Time	616
29. Querying the User	620
30. Initializations	624
30.1 System Initialization Lists	626
31. Dates and Times	628
31.1 Getting and Setting the Time.	629
31.1.1 Elapsed Time in 60ths of a Second.	629
31.1.2 Elapsed Time in Microseconds	630
31.2 Printing Dates and Times	630
31.3 Reading Dates and Times	631
31.4 Reading and Printing Time Intervals	632
31.5 Time Conversions	633
31.6 Internal Functions	633
32. Miscellaneous Useful Functions	635
32.1 Hardcopy	635
32.2 Metering	637
32.3 Poking Around in the Lisp World	640
32.4 Utility Programs	642
32.5 The Lisp Top Level	644
32.6 The Garbage Collector.	646
32.7 Logging In.	648
32.8 Dribble Files.	649
32.9 Status and SStatus	650
32.10 Booting and Disk Partitions.	651
32.10.1 Manipulating the Label	651
32.10.2 Updating Software.	653
32.10.3 Garbage Collecting to Compress Bands.	654
32.10.4 Installing New Software	655
32.10.5 Installing New Microcode	656
32.11 Site Options and Host Table	657
32.11.1 Updating Site Information	660
32.11.2 Accessing Site Options.	660
32.11.3 The LMLOCS File.	661
32.11.4 Initializing a Band at a New Site	661
Concept Index	663
Flavor Index	666
Operation (Message Name) Index	667

Keyword Index	669
Object Creation Options	673
Meter Index	674
Variable Index	675
Function Index	679
Condition Name Index	692

**MA
MA**