

CONTENTS IN DETAIL

PREFACE

xv

1		
THE BASICS REVISITED		1
Getting Environment Variables into GNU make		1
Setting Variables from Outside the Makefile		3
The Environment Used by Commands		6
The \$(shell) Environment		7
Target-Specific and Pattern-Specific Variables		9
Target-Specific Variables		10
Pattern-Specific Variables		11
Version Checking		13
MAKE_VERSION		13
.FEATURES		14
Detecting \$(eval)		16
Using Boolean Values		16
Undefined Variables in Conditionals		17
Consistent Truth Values		18
Logical Operations Using Boolean Values		19
User-Defined Logical Operators		19
Built-in Logical Operators (GNU make 3.81 and Later)		20
Command Detection		21
Delayed Variable Assignment		22
Simple List Manipulation		24
User-Defined Functions		25
The Basics		26
Argument-Handling Gotchas		26
Calling Built-in Functions		27
Recent GNU make Versions: 3.81, 3.82, and 4.0		29
What's New in GNU make 3.81		29
What's New in GNU make 3.82		34
What's New in GNU make 4.0		38
What's New in GNU make 4.1		42
2		
MAKEFILE DEBUGGING		43
Printing the Value of a Makefile Variable		43
Dumping Every Makefile Variable		45
Tracing Variable Values		47
Tracing Variable Use		47
How the Variable Tracer Works		48
Tracing Rule Execution		51
An Example		51
The SHELL Hack		52
An Even Smarter SHELL Hack		53
GNU make 4.0 Tracing		54

Makefile Assertions	55
assert.	55
assert_exists	56
assert_target_directory.	57
An Interactive GNU make Debugger	58
The Debugger in Action	58
Breakpoints in Patterns.	60
Breakpoints in Makefiles	61
Debugger Internals	62
Dynamic Breakpoints in the GNU make Debugger	65
Dynamic Breakpoints in Action	65
The Easy Part	67
The Trick	68
Rocket Science	69
An Introduction to remake	69
Just Print and Trace	69
Debugging	72
Targets, Macro Values, and Expansion	74

3 BUILDING AND REBUILDING 77

Rebuilding When CPPFLAGS Changes	77
An Example Makefile	78
Changing Our Example Makefile	79
How Signature Works	81
Limitations	82
Rebuilding When a File's Checksum Changes	82
An Example Makefile	83
Digesting File Contents	83
The Modified Makefile.	84
The Hack in Action	85
Improving the Code.	86
Automatic Dependency Generation	86
An Example Makefile	87
makedepend and make depend	88
Automating makedepend and Removing make depend	89
Making Deleted Files Disappear from Dependencies	90
Doing Away with makedepend.	91
Using gcc -MP	92
Atomic Rules in GNU make.	92
What Not to Do	93
Using Pattern Rules	93
Using a Sentinel File	94
Painless Non-recursive make	96
A Simple Recursive Make	97
A Flexible Non-recursive make System.	98
Using the Non-recursive make System	103
What About Submodules?	104

4

PITFALLS AND PROBLEMS

109

GNU make Gotcha: <code>ifndef</code> and <code>?=</code>	110
What <code>?=</code> Does	110
What <code>ifndef</code> Does	111
<code>\$(shell)</code> and <code>:=</code> Go Together	111
<code>\$(shell)</code> Explained	111
The Difference Between <code>=</code> and <code>:=</code>	112
The Hidden Cost of <code>=</code>	113
<code>\$(eval)</code> and Variable Caching	115
About <code>\$(eval)</code>	115
An <code>\$(eval)</code> Side Effect	116
Caching Variable Values	116
Speed Improvements with Caching	117
A Caching Function	118
Wrapping Up	119
The Trouble with Hidden Targets	120
An Unexpected Error if the Hidden Target Is Missing	121
The <code>-n</code> Option Fails	121
You Can't Parallelize <code>make</code>	121
<code>make</code> Does the Wrong Work if the Hidden Target Is Updated	122
You Can't Direct <code>make</code> to Build <code>foo.o</code>	122
GNU <code>make</code> 's Escaping Rules	122
Dealing with <code>\$</code>	123
Playing with <code>%</code>	123
Wildcards and Paths	123
Continuations	124
Comments	124
I Just Want a Newline!	124
Function Arguments: Spaces and Commas	125
The Twilight Zone	126
The Trouble with <code>\$(wildcard)</code>	127
<code>\$(wildcard)</code> Explained	127
Unexpected Results	128
Unexpected Results Explained	130
Making Directories	131
An Example Makefile	132
What Not to Do	132
Solution 1: Build the Directory When the Makefile Is Parsed	133
Solution 2: Build the Directory When all Is Built	134
Solution 3: Use a Directory Marker File	134
Solution 4: Use an Order-Only Prerequisite to Build the Directory	135
Solution 5: Use Pattern Rules, Second Expansion, and a Marker File	136
Solution 6: Make the Directory in Line	137
GNU <code>make</code> Meets Filenames with Spaces	137
An Example Makefile	137
Escape Spaces with <code>\</code>	138
Turn Spaces into Question Marks	140
My Advice	140

Path Handling	141
Target Name Matching	142
Working with Path Lists	142
Lists of Paths in VPATH and vpath	143
Using / or \	143
Windows Oddity: Case Insensitive but Case Preserving	144
Built-in Path Functions and Variables	145
Useful Functions in 3.81: abspath and realpath	146
Usman's Law	147
The Human Factor	147
Poor Naming	147
Silent Failure	148
Recursive Clean	148
Pitfalls and Benefits of GNU make Parallelization	148
Using -j (or -jobs)	149
Missing Dependencies	150
The Hidden Temporary File Problem	151
The Right Way to Do Recursive make	153
Amdahl's Law and the Limits of Parallelization	154
Making \$(wildcard) Recursive	157
Which Makefile Am I In?	158

5 PUSHING THE ENVELOPE 161

Doing Arithmetic	161
Addition and Subtraction	162
Multiplication and Division	165
Using Our Arithmetic Library: A Calculator	167
Making an XML Bill of Materials	170
An Example Makefile and BOM	170
How It Works	171
Gotchas	172
Advanced User-Defined Functions	174
Getting Started Modifying GNU make	174
Anatomy of a Built-In Function	176
Reverse a String	177
GNU make 4.0 Loadable Objects	179
Using Guile in GNU make	180
Self-Documenting Makefiles	182
Documenting Makefiles with print-help	185
The Complete help-system.mak	185

6 THE GNU MAKE STANDARD LIBRARY 187

Importing the GMSL	188
Calling a GMSL Function	189
Checking the GMSL Version	189

Example Real-World GMSL Use	190
Case-Insensitive Comparison	190
Finding a Program on the Path	190
Using Assertions to Check Inputs	191
Is DEBUG Set to Y?	192
Is DEBUG Set to Y or N?	193
Using Logical Operators in the Preprocessor	193
Removing Duplicates from a List	194
Automatically Incrementing a Version Number	194
GMSL Reference	196
Logical Operators	196
Integer Arithmetic Functions	198
Integer Comparison Functions	203
Miscellaneous Integer Functions	204
List Manipulation Functions	205
String Manipulation Functions	210
Set Manipulation Functions	213
Associative Arrays	216
Named Stacks	218
Function Memoization	220
Miscellaneous and Debugging Facilities	221
Environment Variables	223

INDEX

225