

CHAPTER 1. OS-9 - THE OPERATING SYSTEM	1
1.1. WHAT IS AN OPERATING SYSTEM?	1
1.2. WHY IS AN OPERATING SYSTEM IMPORTANT?	2
1.3. WHAT IS MULTI-TASKING?	3
1.4. PROGRAMS IN ROM	4
1.5. THE FUNCTIONS OF AN OPERATING SYSTEM	4
1.6. A COMPARISON OF OPERATING SYSTEMS	6
1.7. THE MAIN PROPERTIES OF OS-9	6
1.7.1. Multi-tasking	6
1.7.2. Real Time	6
1.7.3. ROMmable	8
1.7.4. Modular	9
1.7.5. Unified Device Independent I/O System	9
1.7.6. Inter-process Communication Functions	9
1.7.7. High Performance	10
1.7.8. Adaptation to New Hardware	10
1.7.9. Complete Set of Functions	10
1.7.10. Broad Spectrum of Applications	11
1.7.11. The Future	11
1.8. THE PARTS OF OS-9	12
1.8.1. Utilities	12
1.8.2. Language compilers	12
1.8.3. Kernel	12
1.8.4. File managers	12
1.8.5. Device drivers	13
1.8.6. Device descriptors	13
1.8.7. Program support modules (trap handlers)	13
1.8.8. Customization modules	14
1.9. SPECIAL FEATURES OF OS-9	14
CHAPTER 2. USING OS-9	17
2.1. BOOTING OS-9	17

2.2. SHELL - THE COMMAND LINE INTERPRETER	21
2.3. SHELL BUILT-IN COMMANDS	24
2.4. ENVIRONMENT VARIABLES	25
2.5. PATHLISTS	26
2.6. CURRENT DIRECTORIES	27
2.7. INPUT LINE EDITING	28
2.8. OTHER SPECIAL KEYS	29
2.9. MULTIPLE PROCESSES ACCESSING THE TERMINAL	30
2.10. A TYPICAL DIRECTORY STRUCTURE	30
CHAPTER 3. OS-9 MODULES, MEMORY, AND PROCESSES	33
3.1. THE OS-9 MEMORY MODULE	33
3.2. A PROGRAM MODULE	35
3.2.1. Sync Word	35
3.2.2. System Revision ID	36
3.2.3. Module Size	37
3.2.4. User and Group	37
3.2.5. Offset to Module Name	37
3.2.6. Access Permissions	37
3.2.7. Type and Language	39
3.2.8. Attributes and Revision	40
3.2.9. Edition Number	43
3.2.10. Other Fields	43
3.2.11. Header Parity	43
3.2.12. Offset to Program Entry Point	44
3.2.13. Offset to Default Trap Entry Point	44
3.2.14. Minimum Program Data Space	45
3.2.15. Minimum Program Stack Size	45
3.2.16. Offset to Data Initialization Table	45
3.2.17. Offset to Data Pointers Initialization Table	46
3.2.18. Module CRC	47
3.3. MODULES IN FILES	48

3.3.1. Module Groups	49
3.4. OS-9 MEMORY	50
3.4.1. Coloured Memory	51
3.4.2. Memory Allocation	53
3.4.3. Inter-task Memory Protection	55
3.5. PROCESSES AND MULTI-TASKING	57
3.5.1. A Dead Process	60
3.5.2. System State Processes	60
CHAPTER 4. THE OS-9 UTILITIES	63
4.1. WHAT IS A UTILITY?	63
4.2. UTILITY SYNTAX	64
4.2.1. Formal Syntax Notation	65
4.3. UTILITIES FOR OPERATING SYSTEM FUNCTIONS	66
4.4. SYSTEM MANAGEMENT UTILITIES	68
4.5. GENERAL UTILITIES	68
CHAPTER 5. SYSTEM MANAGEMENT	71
5.1. THE SYSTEM MANAGER	71
5.2. THE FILING SYSTEM	72
5.3. THE PASSWORD FILE	72
5.4. SYSTEM STARTUP	73
5.5. THE .LOGIN FILE	74
5.6. DISK FORMATTING	74
5.7. INSTALLING A BOOT FILE	75
5.8. ARCHIVING	78
CHAPTER 6. C COMPILER, ASSEMBLER, LINKER, AND DEBUGGER	81
6.1. THE DEVELOPMENT SYSTEM	81
6.2. THE C COMPILER	82
6.3. FILE NAMING CONVENTIONS	83
6.4. CC OPTIONS	84
6.5. THE ASSEMBLER	87

CONTENTS

6.5.1. The psect Directive	88
6.5.2. The vsect Directive	89
6.5.3. External Symbols	90
6.6. THE LINKER	91
6.6.1. Linker Options	92
6.7. THE PROGRAM DEBUGGER	93
CHAPTER 7. THE OS-9 I/O SYSTEM	97
7.1. I/O SUB-SYSTEMS AND DEVICES	98
7.2. FILE MANAGERS AND DEVICE DRIVERS	102
7.3. DEVICE DESCRIPTORS	103
7.4. PATHS, PATHLISTS, AND FILES	105
7.5. PERMISSIONS, ATTRIBUTES, AND MODES	108
7.6. THE I/O SYSTEM CALLS	110
7.6.1. I\$Attach: Add Device to Device Table	111
7.6.2. I\$Detach: Remove Device from Device Table	113
7.6.3. I\$Dup: Duplicate a Path	113
7.6.4. I\$Create: Create a File	114
7.6.5. I\$Open: Open a Path	115
7.6.6. I\$MakDir: Create a New Directory	116
7.6.7. I\$ChgDir: Change Current Directory	117
7.6.8. I\$Delete: Delete a File	117
7.6.9. I\$Seek: Set the File Pointer	117
7.6.10. I\$Read: Read Data	118
7.6.11. I\$Write: Write Data	118
7.6.12. I\$ReadLn: Read Line	118
7.6.13. I\$WritLn: Write Line	118
7.6.14. I\$GetStt: Get Status	119
7.6.15. I\$SetStt: Set Status	120
7.6.16. I\$Close: Close a Path	120
7.6.17. I\$SGetSt: Get Status on System Path	120
7.7. PATH DESCRIPTOR OPTIONS	121
7.7.1. RBF Options Section	123

7.7.2. SCF Options Section	125
7.7.3. SBF Options Sections	128
7.8. MAKING A NEW DEVICE DESCRIPTOR	129
7.9. SPECIAL FEATURES	134
7.9.1. RBF Disk Caching	134
7.9.2. SCSI Device Driver System	135
7.9.3. Ethernet support	136
7.9.4. The X Window System	136
CHAPTER 8. INTER-PROCESS COMMUNICATION	137
8.1. WHY USE MULTI-TASKING?	137
8.2. WHAT IS INTER-PROCESS COMMUNICATION?	138
8.3. OS-9 INTER-PROCESS COMMUNICATION FACILITIES	139
8.4. FORKING A PROCESS	140
8.5. SIGNALS	142
8.5.1. Masking Signals	145
8.5.2. Signals - Cautions	149
8.6. EVENTS	149
8.6.1. Using Events	154
8.6.2. Pulsing an event	154
8.6.3. Interlocked handshake	155
8.6.4. Buffered handshake	155
8.6.5. One to many synchronization	156
8.6.6. Rendezvous	158
8.6.7. Semaphore	158
8.7. PIPES	159
8.7.1. Using Unnamed Pipes	161
8.7.2. Using Named Pipes	163
8.8. DISK FILES	164
8.8.1. RAM Disks	166
8.9. DATA MODULES	169
8.10. SHARED EXTERNAL MEMORY	173
8.11. ALARMS	175

8.11.1. System State Alarms	178
CHAPTER 9. MULTI-TASKING	183
9.1. OS-9 PROCESS SCHEDULING	183
9.2. THE SCHEDULER FEATURES	185
9.3. ACTIVATING A PROCESS	186
9.4. AUTOMATIC SCHEDULING	188
9.5. AN EXAMPLE OF SCHEDULING	189
9.6. SCHEDULING PRE-EMPTION MECHANISMS	190
9.6.1. Minimum Priority	190
9.6.2. Maximum Age	192
9.6.3. Seizing Control	193
9.6.4. The Precedence of the Mechanisms	194
9.7. SCHEDULING IN REAL TIME APPLICATIONS	194
CHAPTER 10. EXCEPTION HANDLING	197
10.1. EXCEPTION HANDLING UNDER OS-9	198
10.2. USER AND SYSTEM STATE RETURN	199
10.3. SYSTEM CALLS - TRAP #0	200
10.4. TRAP HANDLER MODULES - TRAPS #1 TO #15	202
10.4.1. The Trap Handler Routine	204
10.4.2. Installing Trap Handlers	206
10.4.3. Terminating Trap Handlers	208
10.4.4. Writing a Trap Handler in C	208
10.5. HARDWARE EXCEPTIONS	212
10.5.1. Hardware Exceptions in User State	213
10.5.2. Example - Bus Error Handler	217
10.5.3. 'move from sr' and 'move from ccr'	219
10.5.4. Hardware Exceptions in System State	220
10.6. INTERRUPTS	222
10.6.1. How 68000 Interrupts Work	222
10.6.2. Using Interrupts Under OS-9	223
10.6.3. Interrupts OS-9 Cannot Handle	227

10.6.4. The Level 7 Interrupt	228
10.7. THE EXCEPTION VECTOR TABLE	229
10.8. THE EXCEPTION JUMP TABLE	230
CHAPTER 11. OS-9 SYSTEM CALLS	233
11.1. THE SYSTEM CALL MECHANISM	233
11.2. SYSTEM CALL PARAMETERS	234
11.3. CUSTOM SYSTEM CALLS	235
11.4. USER AND SYSTEM STATE CALLS	236
11.5. THE SYSTEM CALLS	236
11.5.1. F\$AllTsk System Call	242
11.5.2. F\$CCtl System Call	243
11.5.3. F\$ChkMem System Call	245
11.5.4. F\$DatMod System Call	245
11.5.5. F\$DelTsk System Call	246
11.5.6. F\$FModul System Call	246
11.5.7. F\$GBlkMp System Call	247
11.5.8. F\$GProcP System Call	248
11.5.9. F\$Gregor System Call	248
11.5.10. F\$GSPUMp System Call	249
11.5.11. F\$IODel System Call	250
11.5.12. F\$Load System Call	250
11.5.13. F\$Permit System Call	251
11.5.14. F\$Protect System Call	253
11.5.15. F\$SysDbg System Call	254
11.5.16. F\$SysID System Call	255
CHAPTER 12. DEVICE DRIVERS	257
12.1. THE FUNCTION OF A DEVICE DRIVER	257
12.2. DEVICE STATIC STORAGE	258
12.3. PATH DESCRIPTOR	263
12.3.1. RBF Path Descriptor	263
12.3.2. SCF Path Descriptor	268

12.4. SYMBOLIC DEFINITIONS	271
12.5. REGISTER USAGE	272
12.6. DEVICE DRIVER ROUTINES	273
12.6.1. Initialize	274
12.6.2. Terminate	276
12.6.3. Read	277
12.6.4. Write	281
12.6.5. Get Status and Set Status	284
12.7. INTERRUPTS	293
12.7.1. Solicited Interrupts	298
12.7.2. Unsolicited Interrupts	301
12.7.3. Choosing Interrupt Levels	304
12.8. A SKELETON DEVICE DRIVER	305
12.9. CLOCK DRIVERS	308
CHAPTER 13. FILE MANAGERS	313
13.1. THE FUNCTION OF A FILE MANAGER	313
13.2. FILE MANAGER ROUTINES	315
13.3. KERNEL ACCESS TO THE FILE MANAGER	316
13.4. PARAMETER CONVENTION	317
13.5. PATHLISTS	318
13.6. CREATE AND OPEN	318
13.6.1. SCF	318
13.6.2. RBF	319
13.6.3. The File Descriptor Sector	320
13.6.4. The Allocation Bit Map	322
13.6.5. Access to the Whole Disk	324
13.7. CHANGE DIRECTORY	324
13.8. MAKE DIRECTORY	325
13.9. DELETE	327
13.10. SEEK	328
13.11. READ AND WRITE	328

13.11.1. RBF	328
13.11.2. SCF	329
13.12. READ LINE AND WRITE LINE	329
13.12.1. RBF	330
13.12.2. SCF	330
13.13. GET STATUS AND SET STATUS	331
13.14. CLOSE	333
13.15. CALLING THE DEVICE DRIVER	333
13.16. RESOURCE CONTROL	334
13.17. A SKELETON FILE MANAGER	336
CHAPTER 14. OS-9 INTERNAL STRUCTURE	341
14.1. THE SYSTEM GLOBALS	342
14.2. THE OTHER SYSTEM MEMORY STRUCTURES	343
14.2.1. Process Descriptor	345
14.2.2. Path Descriptor	345
14.2.3. Module Directory	347
14.2.4. Device Table	348
14.2.5. Device Static Storage	348
14.2.6. Process Descriptor Table	350
14.2.7. Path Descriptor Table	350
14.2.8. Interrupt Polling Table	351
14.2.9. Event Table	351
14.2.10. Service Dispatch Tables	352
14.3. SYSTEM GLOBALS STRUCTURE	353
14.4. PROCESS DESCRIPTOR STRUCTURE	361
CHAPTER 15. MICROWARE C AND ASSEMBLY LANGUAGE	373
15.1. MICROWARE C	373
15.2. CIO AND MATH TRAP HANDLERS	374
15.3. THE REMOTE DIRECTIVE	374
15.4. PROGRAM STARTUP	376

CONTENTS

15.5. C WITH ASSEMBLY LANGUAGE	376
15.6. REGISTER VARIABLES	377
15.7. CODING FOR SPEED	378
15.8. THE 'LINK' INSTRUCTION	380
15.9. A FUNCTION IN ASSEMBLY LANGUAGE	381
15.10. STRUCTURE RETURN	382
15.11. CALLING C FROM ASSEMBLY LANGUAGE	384
15.12. A DEVICE DRIVER IN C	387
15.13. A FILE MANAGER IN C	396
15.14. HINTS ON C PROGRAMMING	403
15.14.1. Declarations and Definitions	403
15.14.2. Pointers and Arrays	404
15.14.3. Pointers and Functions	406
15.14.4. Pointers and Structures	408
APPENDIX A GLOSSARY	411
APPENDIX B SBF DEFINITIONS	417