

Contents

	<i>Page</i>
Preface	xi
Related Publications	xi
Ordering Cray Research publications	xii
Conventions	xii
Reader comments	xiv
Introduction [1]	1
Terminology	2
Hardware	3
Tape interfaces	3
Tape subsystem features	4
Tape subsystem architecture	4
Tape label support	6
Tape positioning	7
Front-end servicing	7
User end-of-volume processing	7
Multifile volume allocation	7
Concatenated tape files	7
Tape performance	8
System buffering	8
Kernel interface	9
Tape multilevel security	9
Tape Formats [2]	11
IBM compatible tape format	11
SG-2051 10.0	iii

	<i>Page</i>
Nonlabeled tapes	11
Two tape mark tapes	11
Single tape mark tapes	12
Labeled tapes	13
IBM compatible tape format summary	14
Tape label fields	16
VOL1 label	17
HDR1, EOV1, and EOF1 labels	19
HDR2, EOV2, and EOF2 labels	21
ER90 volumes	23
Tape Subsystem Tutorial [3]	25
Getting started	25
Obtaining tape status	28
Tape status commands	28
Tape log file	30
Messages to operator	31
Using standard commands	32
Using the <code>cp(1)</code> command	32
Using the <code>dd(1)</code> command	33
Using the <code>tar(1)</code> command	33
Procedure 1: Example 1	33
Procedure 2: Example 2	34
Using the <code>cpio(1)</code> command	35
Using the <code>tpmnt(1)</code> command to read concatenated tape files	35
Using the <code>tpmnt(1)</code> command to read or write multifile tapes	36
Example 1	37
Example 2	37

	<i>Page</i>
Mounting ER90 volumes	38
Using MLS	38
Writing Fortran Applications Using Tapes [4]	41
IBM compatible tape processing	41
Reading and writing to tape	41
Reading and writing tape marks	43
Positioning a tape by blocks	44
Positioning a tape by using the SETTP(3) library call	46
Example 1	46
Example 2	49
Reading and writing tapes containing foreign data	50
Converting foreign data explicitly	50
Example 1	52
Example 2	54
Example 3	55
Converting foreign data implicitly	56
Using the bad data recovery routines	58
Example 1	59
Example 2	61
Example 3	63
Using end-of-volume processing requests	65
Example 1	67
Example 2	70
Example 3	72
ER90 tape processing	75
Using pure data mode	76
Using COS blocking mode	79

	<i>Page</i>
Writing C Applications Using Tapes [5]	81
C flexible file I/O library routines	81
System call I/O	89
Cray Research systems	89
Transparent I/O	91
Transparent buffered I/O	91
Transparent unbuffered I/O	92
Tape information requests	96
Tape information table	96
Tape daemon requests	99
ioctl(2) requests	107
ER90 TPC_EXTSTS request	107
ER90 read of the buffer log using TPC_RDLOG	113
IBM compatible read of the buffer log using TPC_RDLOG	115
Tape positioning requests	116
End-of-volume requests	116
Tape control requests	116
ER90 set data block size request	116
ER90 synchronize request	117
Using the Character-Special Tape Interface [6]	121
Using character-special tapes	121
Writing C applications	122
Opening files	122
Closing files	122
Using I/O	122
Using ioctl(2) requests	123
MTIOCKERR call	124

	<i>Page</i>
MTIOCATTR call	124
MTIOCGET call	125
MTIOCTOP call	127
Hardware error codes	144
Appendix A Interpreting System Messages	147
Index	203
Figures	
Figure 1. Tape subsystem architecture	5
Figure 2. Communication between the user, tape driver, and tape daemon	6
Figure 3. Nonlabeled, two tape mark formats	12
Figure 4. Nonlabeled, single tape mark formats	13
Figure 5. Labeled tape formats	14
Figure 6. Single-volume file	15
Figure 7. Multifile, single-volume tape	15
Figure 8. Multivolume, single-file tape	15
Figure 9. Multifile, multivolume tape	16
Figure 10. VOL1 label	18
Figure 11. HDR1/EOV1/EOF1 labels	21
Figure 12. HDR2/EOV2/EOF2 labels	23
Figure 13. Creating a tape	26
Figure 14. Reading an existing tape file	26
Figure 15. Adding a new file to an existing tape	27
Figure 16. NQS tape job	28
Figure 17. tprst(1) status display	28
Figure 18. tpstat(1) status display	29
Figure 19. tplist(1) display	30

	<i>Page</i>
Figure 20. <code>tape.msg</code>	31
Figure 21. Writing an unlabeled tape	42
Figure 22. Reading an unlabeled tape	43
Figure 23. Reading and writing tape marks	44
Figure 24. Positioning by blocks	46
Figure 25. SETTP(3) positioning, example 1	48
Figure 26. SETTP(3) positioning, example 2	50
Figure 27. Converting data to an IBM format	53
Figure 28. Reading an unknown number of records	55
Figure 29. Reading mixed data types	56
Figure 30. Converting foreign data	58
Figure 31. Using the SKIPBAD(3) routine	60
Figure 32. Using the ACPTBAD(3) routine	62
Figure 33. Using the ISHELL(3) routine	64
Figure 34. Using Fortran library routines for EOVS processing	68
Figure 35. Using EOVS processing when writing a file	71
Figure 36. Using EOVS processing when reading a multivolume file	74
Figure 37. Using pure data mode	78
Figure 38. Using COS blocking mode	80
Figure 39. C library routine usage	83
Figure 40. Executing <code>cexam.c</code>	84
Figure 41. Executing <code>cexam2.c</code>	85
Figure 42. Using C library routines for EOVS processing	86
Figure 43. Reading from an IBM compatible device (unbuffered I/O)	93
Figure 44. Reading from an ER90 device (unbuffered blocked I/O)	94
Figure 45. Reading from an ER90 device (unbuffered byte stream I/O)	94
Figure 46. Writing to an IBM compatible device (unbuffered I/O)	95

	<i>Page</i>
Figure 47. Writing to an ER90 device (unbuffered byte stream I/O)	96
Figure 48. Tape information table header	97
Figure 49. Using the tape information table	98
Figure 50. Using the TR_INFO request	101
Figure 51. TR_INFO information	105
Figure 52. ct1_extstts structure	108
Figure 53. Using the ER90 TPC_EXTSTTS request (tape path)	110
Figure 54. Using the ER90 TPC_EXTSTTS request (pseudo device)	112
Figure 55. ct1_rdlog structure	113
Figure 56. Using the ER90 TPC_RDLOG request	114
Figure 57. Using the TPC_RDLOG request (IBM compatible)	115
Figure 58. Setting data block size	117
Figure 59. dmn_comm structure (synchronizing request)	118
Figure 60. Synchronizing your program with a tape	119
Figure 61. Block identifiers	130

Tables

Table 1. VOL1 label values	17
Table 2. HDR1/EOV1/EOF1 labels	19
Table 3. HDR2/EOV2/EOF2 labels	22

