Chapter 2

CM-5 File System Interfaces

The CM programming languages have a variety of interfaces to the Connection Machine file systems; these are introduced in the following sections:

- For information about CM-5 I/O using C, see Section 2.2.1.
- For information about CM-5 I/O using C*, see Section 2.2.2.
- For information about CM-5 I/O using FORTRAN 77, see Section 2.3.1.
- For information about CM-5 I/O using CM Fortran, see Section 2.3.2.

The interfaces to the file systems are described fully in the following Thinking Machines documentation:

- The CMFS library and CMOST I/O commands and system calls are described in Part II of this manual.
- The CMMD library (Connection Machine message-passing routines) is described in the *CMMD User's Guide* and the *CMMD Reference Manual*.
- The C* I/O functions are described in the *C* Release Notes*, Version 7.1.1.
- The CM Fortran utility library and the language's I/O statements are described in the CM Fortran documentation set, which includes the *CM Fortran User's Guide*, the *CM Fortran Reference Guide*, and the *CM Fortran Libraries Reference Manual*, V. 2.1.

This manual assumes knowledge of the UNIX operating system and its OS and I/O commands and routines. If you do not have a UNIX background, we recommend the following SunOS manuals:

- *Getting Started With SunOS: Beginner's Guide*

Copyright © 1994 Thinking Machines Corporation

- *Doing More with SunOS: Beginner's Guide*
- *Setting Up Your SunOS Environment: Beginner's Guide*

2.1 CMFS Software vs. UNIX Software: What's the Difference?

Regardless of the language in which you program, you can use both the CMFS software and the UNIX software on which it is modeled to perform serial I/O, non-I/O file operations (such as creating, opening, seeking, and closing), and file system maintenance operations (such as making a directory and copying a file). The CMFS commands and routines were originally developed for use on the CM-2 and CM-200 systems (on which UNIX commands and routines were not supported), and they have been ported to the CM-5 system for your convenience.

For many CM-5 application tasks it doesn't matter whether you use a UNIX command/routine or the comparable CMFS command/routine. Although you may prefer to use UNIX more often because it may be more familiar to you, do not discount the CMFS routines:

- Some CMFS commands and routines — printed in boldface in Table 4 and Table 5 (see Chapter 4) — supplement the standard UNIX set or are enhanced for CM-5 functionality and/or performance.
- CMFS commands and routines are useful for porting from the CM-5 to the CM-2/200 and vice versa.
- UNIX commands and routines can operate on CMFS files (DataVault files, for example) only if that file system is NFS-mounted.
- CMFS commands and routines may achieve better performance than UNIX commands/routines on an NFS-mounted CMFS file system.

Note that not all CMFS routines have both a C interface and a Fortran interface, although most do.

Copyright © 1994 Thinking Machines Corporation

2.2 Accessing the File Systems from C-Based Languages

The information in Sections 2.2.1 and 2.2.2 is summarized in Table 1.

Table 1. The C and C* interfaces to the Connection Machine file systems.

Interface	Language(s)	File System	I/O Process†
CMMD library	C, C*	SFS, CMFS, and UNIX	Parallel
CMOST I/O system calls	C, C*	SFS and UNIX	Parallel
CMFS library	C, C*	SFS, CMFS, and UNIX	Serial
CMFS library	C*	SFS, CMFS, and UNIX	Parallel
UNIX routines	C, C*	SFS, NFS-mounted CMFS, and UNIX	Serial
C* I/O functions	C*	SFS and UNIX	Parallel and serial

† Recall that parallel I/O is data transfer between a file and one or more PNs. Serial I/O is data transfer between a file and serial-computer memory, or non-I/O file operations or file system operations.

2.2.1 Accessing the File Systems from C

To access any Connection Machine file system from a C program running on the CM-5, use the following interfaces:

- When performing parallel I/O, the program must use the CMMD library or the CMOST I/O system calls. (The CMMD library is also supported from C++.)
- When performing serial I/O, the program can use the CMFS library or the UNIX routines. Note that UNIX routines can access a CMFS file system only if it is NFS-mounted.

Copyright © 1994 Thinking Machines Corporation

2.2.2 Accessing the File Systems from C*

NOTE

C* programs that use the CMFS library must include **<cm/cmfs.h>** and link with **-lcmfs_cs** and **-lcmfs_cm5**.

CM-5 C* programs can access the CM-5 file systems via the following interfaces:

- Parallel I/O:
  - If using an SFS or UNIX file, the program can use the CMMD library, the CMOST I/O system calls, the CMFS library, or the C* I/O functions.
  - If using a CMFS file, the program must use the CMMD library or the CMFS library.
- Serial I/O: The program can use either the CMFS library serial I/O routines or the UNIX I/O routines. (To use UNIX routines to access a CMFS file system, it must be NFS-mounted.)
  - If using an SFS or UNIX file, the program can use the CMFS library, the UNIX routines, or the C* I/O functions.
  - If using a CMFS file, the program must use the CMFS library or the UNIX routines.

2.3 Accessing the File Systems from Fortran-Based Languages

The information in Sections 2.3.1 and 2.3.2 is summarized in Table 2 and Table 3, respectively.

Copyright © 1994 Thinking Machines Corporation

2.3.1 Accessing the File Systems from FORTRAN 77

To access a CMFS file system or the SFS file system (or other UNIX file system) from a FORTRAN 77 program running on the CM-5, use the CMMD library, the CMFS library, or the UNIX routines:

- When performing parallel I/O, the program must use the CMMD library.
- When performing serial I/O, the program must use either the CMFS library's serial routines or the UNIX routines. (To use UNIX routines to access a CMFS file system, it must be NFS-mounted.)

Table 2. The FORTRAN 77 interfaces to the Connection Machine file systems.

Mechanism	File System	I/O Process†
CMMD library	SFS, CMFS, and UNIX	Parallel
CMFS library	SFS, CMFS, and UNIX	Serial
UNIX routines	SFS, NFS-mounted CMFS, and UNIX	Serial

† Recall that parallel I/O is data transfer between a file and one or more PNs. Serial I/O is data transfer between a file and serial-computer memory, or non-I/O file operations or file system operations.

2.3.2 Accessing the File Systems from CM Fortran

To access the SFS or CMFS file system from a CM Fortran program running on the CM-5, use the CM Fortran utility library (preferred), the language's I/O statements, the CMFS library, or UNIX routines:

- When performing parallel I/O, the program must use either the CM Fortran utility library, the language's I/O statements, or the CMFS library's synchronous I/O routines, **`CMFS_READ_FILE_ALWAYS`** and **`CMFS_WRITE_FILE_ALWAYS`**; see Section 7.3 (in Chapter 7, Part II of this manual).

Copyright © 1994 Thinking Machines Corporation

- When performing serial I/O, the program can use the CM Fortran utility library, the language's I/O statements, the CMFS library, or, if the file system is NFS-mounted, UNIX routines.

To access a standard implementation of a UNIX file system via serial I/O only, use the CM Fortran utility library, the language's I/O statements, the CMFS library, or UNIX routines.

Although you should use the CM Fortran utility library or the Fortran I/O statements whenever possible because these interfaces are designed specifically for I/O from CM Fortran, the CMFS library's routines provide greater flexibility and power for those applications that require it.

Table 3. CM Fortran interfaces to the Connection Machine file systems.

Mechanism	File System	I/O Process†
CMF Utility Library	SFS and CMFS	Parallel and serial
CMF Utility Library	UNIX	Serial
I/O statements	UNIX	Serial
I/O statements	SFS and CMFS	Parallel and serial
CMFS library	SFS, CMFS, and UNIX	Parallel and serial
UNIX routines	SFS, NFS-mounted CMFS, and UNIX	Serial

† Recall that parallel I/O is data transfer between a file and one or more PNs. Serial I/O is data transfer between a file and serial-computer memory, or non-I/O file operations or file system operations.

Copyright © 1994 Thinking Machines Corporation