Chapter 1

Introduction

C* is an extension of the C programming language designed for the Connection Machine data parallel computing system. This chapter presents an overview of the process of developing, executing, and debugging a C* program on a CM-5 or CM-5E system. The rest of this manual goes into the process in more detail.

1.1 Developing a C* Program

Develop C* source code in one or more files. Use the suffix **.cs** if the file contains parallel code or any other features that C* adds to Standard C (for example, the new **<?=** and **>?=** operators). Chapter 2 describes facilities for developing a C* program. It also describes how to call functions in CM libraries, as well as CM Fortran subroutines.

1.2 Performing I/O in a C* Program

C* provides several methods for performing I/O, including parallel interfaces to the UNIX **fread** and **fwrite** functions, and functions for creating and using an NMO buffer, so that you can use CM system calls such as **CM_read** and **CM_write** to transfer data. Chapter 3 describes these and other methods of performing I/O.

Copyright © 1990–1995 Thinking Machines Corporation

1.3 Compiling a C* Program

Compile the files by issuing the command **cs**. The command can take various options, some of which are identical to options for the C compiler **cc**. Chapter 4 describes the compiler options and the compiling process in detail.

1.4 Executing a C* Program

You execute a C* program on a CM-5 partition manager, just as you would any UNIX program or command. You can also submit the program as a batch request to DJM, the CM's distributed job manager.

Executing a C* program is discussed in more detail in Chapter 5.

1.5 Debugging a C* Program

You can debug and analyze the performance of your C* program using Prism, the CM programming environment. Prism is described briefly in Chapter 6.

Version 7.2, May 1995
Copyright © 1990–1995 Thinking Machines Corporation