Chapter 1

Introduction

The Prism programming environment is an integrated graphical environment within which users can develop, execute, debug, and analyze the performance of programs written for the CM-2, CM-200, or CM-5 Connection Machine system. It provides an easy-to-use, flexible, and comprehensive set of tools for performing all aspects of Connection Machine programming. Prism operates on terminals or workstations running the X Window System. In addition, a commands-only option allows you to operate on any terminal, but without the graphical interface.

This chapter introduces Prism. Subsequent chapters discuss specific aspects of it.

1.1 Overview

You can either load an executable program into Prism, or start from scratch by calling up an editor and a UNIX shell within Prism and using them to write and compile the program.

Once an executable program is loaded into Prism, you can (among other things):

- execute the program
- debug the program
- analyze the program's performance
- visualize data from the program

Copyright © 1991–1994 Thinking Machines Corporation

There are slightly different versions of Prism for the CM-2/200 and the CM-5. There is also a version, called *Node Prism*, designed specifically for working with message-passing programs on the CM-5.

1.2 The Look and Feel of Prism

Figure 1 shows the main window of Prism (CM-2 version) with a program loaded. It is within this window that you debug and analyze your program. You can operate with a mouse, use keyboard equivalents of mouse actions, or issue keyboard commands.

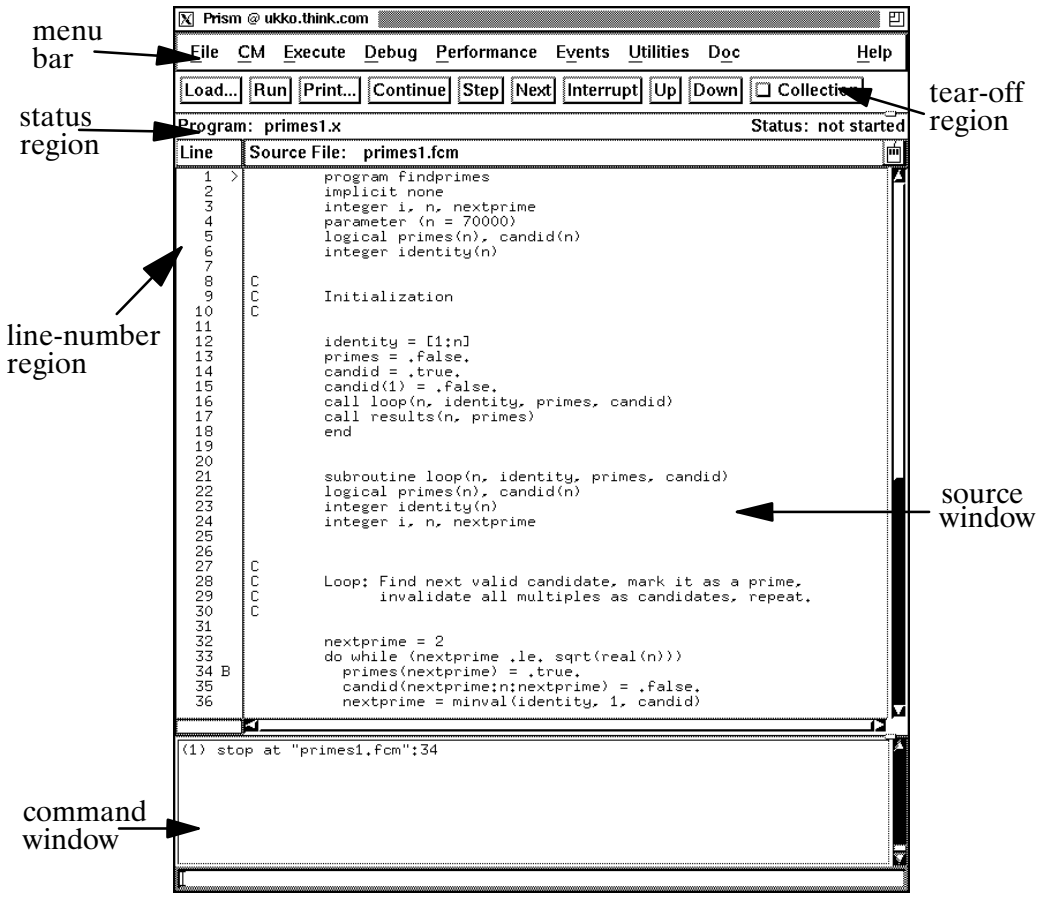

Figure 1. Prism's main window.

Version 2.0, April 1994
Copyright © 1991–1994 Thinking Machines Corporation

Clicking on items in the **menu bar** displays pulldown menus that provide access to most of Prism's functionality.

You can add frequently used menu items and commands to the **tear-off region**, below the menu bar, to make them more accessible.

The **status region** displays the program's name and messages about the program's status.

The **source window** displays the source code for the executable program. You can scroll through this source code and display any of the source files used to compile the program. When a program stops execution, the source window updates to show the code currently being executed. You can select variables or expressions in the source code and print their values or obtain other information about them.

The **line-number region** is associated with the source window. You can click to the right of a line number in this region to set a breakpoint at that line. In Figure 1, a breakpoint is set at line 34.

The **command window** at the bottom of the main Prism window displays messages and output from Prism. You can also type commands in the command window rather than use the graphical interface.

General aspects of using these areas are discussed in Chapter 2.

1.3 Loading and Executing Programs

You can load an executable program into Prism when you start it up, or any time afterward. Once the program is loaded, you can run the program or step through it. You can also interrupt execution at any time. CM-2 and CM-200 users can attach to and detach from a CM from within Prism.

You can also attach to a running program or associate a core file with a program.

Chapter 3 discusses these topics in more detail. See Section 10.4 for aspects of these topics that are unique to Node Prism.

Copyright © 1991–1994 Thinking Machines Corporation

1.4 Debugging

Prism allows you to perform standard debugging operations such as setting breakpoints and traces, and displaying and moving through the call stack. Chapter 4 discusses these topics. See Section 10.5 for a discussion of debugging in Node Prism.

1.5 Visualizing Data

In parallel computing, it is often important to obtain a visual representation of the data elements that make up an array or parallel variable. In Prism, you can create *visualizers* that provide standard representations of variables or expressions. For example:

- In the *text* representation, the data is shown as numbers or characters.
- In the *colormap* representation, each data element is mapped to a color, based on a range of values and a color map you specify. (This representation is available only on color workstations.)
- In the *threshold* representation, each data element is mapped to either black or white, based on a cutoff value that you can specify.

A *data navigator* lets you manipulate the display window relative to the data being visualized. Options are available that let you update a visualizer or save a snapshot of it.

See Chapter 5 for a discussion of visualizing data. Section 10.6 covers aspects of visualization unique to Node Prism.

1.6 Analyzing Program Performance

Prism provides performance data essential for effectively analyzing and tuning data parallel programs. The data is slightly different for the different versions of Prism. For the CM-5 version, for example, the data includes:

- processing time on the partition manager
- processing time on the nodes

Version 2.0, April 1994
Copyright © 1991–1994 Thinking Machines Corporation

- time spent doing various forms of communication
- time spent performing I/O

The performance data is displayed as histograms and percentages for each computing resource. For each resource, you can also obtain data on usage for each procedure and source line in the program. You can save the performance data in a file and redisplay it at a later time.

In addition, a performance advisor provides information about and analysis of the data that Prism collects.

Performance data is available for both C* and CM Fortran programs.

See Chapter 6 for a discussion of performance analysis. Section 10.7 discusses aspects of performance analysis unique to Node Prism.

1.7 Editing and Compiling

You can call up the editor of your choice within Prism to edit source code (or anything else). If you change your source code and want to recompile, Prism also provides an interface to the UNIX **`make`** utility. See Chapter 7.

1.8 Obtaining On-Line Help and Documentation

Prism features a comprehensive on-line help system. Help is available for each menu, window, and dialog box in Prism. These help screens provide links into an on-line version of the *Prism User's Guide* and *Prism Reference Manual*. Users can follow these links to obtain complete information about a topic.

In addition to help on Prism itself, an interface is provided to CMview, Thinking Machines Corporation's on-line documentation system. CMview provides hyperlinked access to user documentation for the Connection Machine system, as well as full-text searching capabilities through the documentation set.

On-line help and documentation are described in more detail in Chapter 8.

Copyright © 1991–1994 Thinking Machines Corporation

1.9 Customizing Prism

Prism provides various ways in which you can change aspects of the way it operates. They are discussed in Chapter 9.

Version 2.0, April 1994
Copyright © 1991–1994 Thinking Machines Corporation