Chapter 5

Executing a C* Program

Once a C* program has been compiled and linked, you can execute the output file on a CM-5. This chapter gives an overview of how to execute a program on a CM-5:

- Section 5.1 describes how to execute the program directly.
- Section 5.2 describes how to execute the program in batch mode.
- Section 5.3 describes how to execute a C* program on a Sun-4.

You can also execute C* programs within Prism, the CM's programming environment. See Chapter 6 for information on Prism.

5.1 Executing the Program Directly

To execute a program directly on a CM-5, you must gain access to it first. To do this, you must know the name of one of its partition managers. You can find out the names of partition managers from your system administrator; the system administrator can also tell you if you have permission to use a particular partition manager.

From your terminal or workstation on the network, you can gain access to the partition manager via the UNIX command **`rlogin`** or **`rsh`**. For example, to log in to the partition manager Mars, issue the command

```
% rlogin mars
```


Copyright © 1990–1995 Thinking Machines Corporation

You can then execute your program on the partition just as you would any UNIX command or program. For example,

```
% a.out
```

Use the **rsh** command to execute the program without logging in to the partition manager. Simply specify the name of the partition manager, followed by the name of the executable program, on the **rsh** command line. For example,

```
% rsh mars a.out
```

You are then returned to your local UNIX shell.

5.2 Executing in Batch Mode

You can also run your program in batch mode. You can use the Distributed Job Manager (DJM), the NQS batch system, or standard UNIX batch commands.

5.2.1 UNIX Batch Commands

The CM-5 supports the standard UNIX batch commands **at** and **batch**. For example,

```
% at 0815am Jan 24
at>  primes
at>  Ctrl-d
```

This causes the program **primes** to be executed at 8:15 a.m. on January 24th. *Ctrl-d* means "press the *d* key while holding down the *Control* key." This is the way to signal the end of information you are providing to the standard input.

5.2.2 DJM (Distributed Job Manager)

DJM (Distributed Job Manager) is a batch queuing system and interactive job manager. In it, you submit a request to a queue for execution. A request

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

consists of one or more programs. When your request reaches the head of the queue, it is executed. Queues can be configured in various ways: for example, so that they are active only at certain times, or so that they accept requests only from certain users.

NOTE: If DJM is running at your site, you should typically use it to run your CM-5 programs. Otherwise, you might have difficulty obtaining the resources to run your program.

Use the **jsub** or **jrun** command to submit a job. You can specify the name of the program from the standard input, or in a script file. For example, if the script file's name is **/myname/primes_script**, you would issue the command as follows:

```
% jsub /myname/primes_script
```

If you don't include the appropriate options (see below), DJM then responds by asking questions about the job:

```
Number of processors (32)?
Estimated CPU time (5min)?
Estimated memory (128M)?
```

(The defaults in parentheses may differ at your site.)

Make your best estimates of the resources your job will need to run. If your estimates are too low, DJM will not effectively schedule your job; if they are too high, the job may be delayed unnecessarily, waiting for the resources to be available. The actual resources consumed by your job are printed at the end of the output file when the job finishes. You can use this information to provide better resource estimates the next time you run your job.

When you have answered DJM's questions, it prints a job ID:

```
Job submitted successfully. Job id is 43.
```

Output from the job is written to the file *myjob*`.o`*nnn*, where *myjob* is the name of the program that ran and *nnn* is its job ID. You can override this by specifying the **-stdout** option; this sends output to the standard output.

Use the **-queue** option to specify the queue to which the job is to be submitted. If you omit the option, DJM submits the job to the first queue that meets the requirements for your job. You can also use the **-nproc**, **-cputime**, and **-mem** options to specify your job's resources on the command line, rather than having to respond to DJM's questions.

Copyright © 1990–1995 Thinking Machines Corporation

For complete information on DJM, see the manual *Using DJM on the CM-5*.

5.2.3 The NQS Batch System

NQS is a batch system that your system administrator can set up to control the execution of programs on the CM-5. As in DJM, you submit a request to a queue, which can be configured in various ways. When your request reaches the head of the queue, it is executed.

Use the **qsub** command to submit a request to a queue. Specify the name of the queue on the command line, using the **-q** option. You can specify the name of the program from the standard input, as with the **at** command. For example, if the queue's name is **b_queue**, you could type this:

```
% qsub -q b_queue
  primes
  Ctrl-d
```

Alternatively, you can put the program's name in a script file. This is especially useful if you have multiple programs to run, or you have CMOST commands you want to issue along with the program. If the script file's name is **/myname/primes_script**, you would issue the command as follows:

```
% qsub -q b_queue /myname/primes_script
```

Use the **-I** option to specify your account ID, if one is required; obtain the account ID from your system administrator.

For a complete discussion of NQS, see the manual *NQS for the CM-5*.

Use the **-p** option, followed by an integer from 0 through 63, to set a priority for this request in its queue. 63 is the highest priority, and 0 is the lowest priority. The priority determines the request's position in the queue. If you don't set a priority, the request is assigned a default priority.

Specifying Output Files

Use the **-o** option, followed by a pathname, to specify the file to which output of the batch request is to be sent. Use the **-e** option to specify the pathname for the standard error output. If you omit these options, the output is sent to default files based on an ID number assigned to the request by the batch system.

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

5.3 Executing a C* Program on a Sun-4

If you compiled your program with the **`-cmsim`** option, you can run your program on a Sun-4. You run the program just as you would any other program on a Sun-4. Performance, of course, will be much worse than on a CM-5.

Copyright © 1990–1995 Thinking Machines Corporation