INTEL CORPORATION
15201 N.W. Greenbrier Parkway
Beaverton, OR 97006
(503) 629-7600

January 15, 1992

Dear iPSC® System Update Customer:

This package contains your Release 3.3.1 system software. With this software installed on your iPSC® Supercomputer[1], your system is ready for use. Please read through the documentation and distribute it to anyone intending to use the system.

The R3.3.1 system software runs on iPSC®/2 (with Intel386™-microprocessor-based CX nodes) and iPSC®/860 systems (with i860™-microprocessor-based RX nodes or with a combination of CX and RX nodes). It includes a broad selection of software tools such as the Fortran and C programming language compilers, the Interactive Parallel Debugger, Network Queueing System and the Performance Analysis Tools. The Network File System, node TCP/IP, and X Window System software packages are offered as options.

Before using your iPSC System:

- **Read this letter completely.**
- **Verify the contents of this package.**
- **Read the *iPSC®/2 and iPSC®/860 System Software Release 3.3.1 Product Release Notes.***

1. The terms "iPSC Supercomputer" and "iPSC system" refer to any of the following SSD products: iPSC®/2 and iPSC®/2S, iPSC®/860 and iPSC®/860S.

PN: 312255

Package Contents

Your iPSC system software package is shipped in one box. Please verify that they include the following items:

Media

iPSC®/2 System Software R3.3.1
cartridge tape (part number 312251-001)
iPSC®/2/860 Optional Software R3.3.1
cartridge tape (part number 312252-001)
iPSC®/860 Extension Software R3.3
cartridge tape (part number 312253-001)

TCP/IP R3.1 cartridge tape
(part numbers 312245-001)

R3.3.1 Documentation

Refer to the "Documentation Included With iPSC® System Release 3.3.1" section for a complete list of documentation supplied.

If items are missing, or if you have any questions, contact Intel Supercomputer Systems Division immediately. Refer to "Comments and Assistance" for information about how to contact Intel Supercomputer Systems Division.

What is in this Release?

Release 3.3.1 contains iPSC system software for both iPSC/2 and iPSC/860 systems. Please refer to the *iPSC®/2 and iPSC®/860 System Software Release 3.3.1 Product Release Notes* for a complete listing of product content.

New Documentation

Please refer to the *iPSC®/2 and iPSC®/860 System Software Release 3.3.1 Product Release Notes.*

Restrictions and Limitations of Release 3.3.1

Every effort has been made to assure the quality of this release, but at shipping time we are aware of a few problems. Please refer to the *iPSC®/2 and iPSC®/860 System Software Release 3.3.1 Product Release Notes* for known restrictions, limitations, and their workarounds.

Bugs Fixed by Release 3.3.1

Please refer to the *iPSC®/2 and iPSC®/860 System Software Release* 3.3.1 *Product Release Notes* for a complete listing of all bugs fixed by this release.

Installation

Your Intel Service Representative will install your new iPSC system software. If you wish to reinstall all or any part of your system software, refer to the installation instructions in the *iPSC®/2 and iPSC®/860 System Software Release 3.3 Product Release Notes*. This document has separate sections describing how to install the UNIX operating system, the iPSC system software, and the TCP/IP software. Refer to the release notes for optional products (such as CIO Ethernet) for their installation instructions.

NOTE

Adding or removing any boards or components from your iPSC system can damage the system and may invalidate your warranty. Please contact Supercomputer Systems Division Customer Support for assistance in answering your questions.

Comments and Assistance

Supercomputer Systems Division is eager to hear of your experiences with our new software product. Please call us if you need assistance, have questions, or otherwise want to comment on your iPSC system.

1-800-421-2823 (Customer Support Response Center)
Your Local Intel Sales Office (in Europe)
support@ssd.intel.com (Internet address)

Supercomputer Systems Division is trying to produce the best documentation for your needs. If you have comments about the iPSC manuals, please fill out and mail the enclosed Comment Card. You can also send your comments electronically to the following address:

techpubs@ssd.intel.com (Internet address)

The Intel Supercomputer Users' Group promotes the exchange of information among users. Intel strongly supports the Users' Group and encourages participation in its activities, which include: Special Interest Groups (SIGs), an annual international users' conference, an e-mail task force, and a "freeware" library of user-contributed software, available electronically to all members of the Intel Supercomputer Users' Group. For membership information contact:

JoAnne Wold (503-629-7737)
joanne@ssd.intel.com (Internet address)

Sincerely,

Elliot Swan

Manager, Technical and Product Marketing
Intel Supercomputer Systems Division

January 1992
Order Number: 312250

iPSC®/2 AND iPSC®/860
RELEASE 3.3.1 SOFTWARE
PRODUCT RELEASE NOTES

intel® Corporation

REV.	REVISION HISTORY	DATE
---	Original Issue	01/92

RESTRICTED RIGHTS

Use, duplication or disclosure by the U.S. Government is subject to restrictions as set forth in subparagraph (c) (1) (ii) of the rights in Technical Data and Computer Software clause at 52.227-7013. Intel Corporation, 2200 Mission College Drive, Santa Clara, California 95052-8126.

Preface

These release notes provide the latest information on features, limitations, workarounds, and installation for the Release 3.3.1 system software for the following Intel Supercomputer Systems Division products: iPSC®/2, iPSC®/2S, iPSC®/860, and iPSC®/860S.

NOTE

In the remainder of the manual, the term "iPSC system(s)" is used to refer to these products.

These release notes assume that you are an application programmer, familiar with the C or Fortran language and the UNIX operating system.

For more information, refer to the Release 3.3.1 system software customer letter that accompanied your software.

For iPSC®/2 system installation instructions, refer to Chapter 2. For iPSC®/860 system installation instructions, refer to Chapter 3.

Organization

Chapter 1	Describes the features of Release 3.3 and Release 3.3.1 system software.
Chapter 2	Provides system software installation information for iPSC®/2 systems.
Chapter 3	Provides system software installation information for iPSC®/860 systems.
Chapter 4	Describes known limitations and their workarounds.
Chapter 5	Describes software enhancements in this release.

Notational Conventions

This manual uses the following notational conventions:

Bold Identifies command names and switches, system call names, reserved words, and other items that must be used exactly as shown.

Italic Identifies variables, filenames, directories, processes, user names, and writer annotations in examples. Italic type style is also occasionally used to emphasize a word or phrase.

`Plain-Monospace`
Identifies computer output (prompts and messages), examples, and values of variables. Some examples contain annotations that describe specific parts of the example. These annotations (which are not part of the example code or session) appear in *italic* type style and flush with the right margin.

Bold-Italic-Monospace
Identifies user input (what you enter in response to some prompt).

`Bold-Monospace`
Identifies the names of keyboard keys (which are also enclosed in angle brackets). A dash indicates that the key preceding the dash is to be held down *while* the key following the dash is pressed. For example:

`<Break>` **`<s>`** **`<Ctrl-Alt-Del>`**

[ ] (Brackets) Surround optional items.

. . . (Ellipsis dots) Indicate that the preceding item may be repeated.

| (Bar) Separates two or more items of which you may select only one.

{ } (Braces) Surround two or more items of which you must select one.

Applicable Documents

For more information, refer to the following manuals:

iPSC® System Manuals

(NEW) *Intel® Supercomputer Systems Division System Log Book*
311927-003
Provides forms on which to keep system maintenance logs.

(NEW) *iPSC® System Technical Documentation Guide*
312026-002
Describes the technical documentation that supports the iPSC System and tells how to use the various documents.

(REV) *iPSC®/2 and iPSC®/860 C Commands and Routines Quick Reference*
311610-004
Summarizes all C routines and commands for the iPSC system.

iPSC®/2 and iPSC®/860 C Language Reference Manual
311567-004
Describes the Green Hills C compiler for the iPSC/2 and iPSC/860 systems.

iPSC®/2 and iPSC®/860 FORGE User's Guide
311866-001
Tells how to use the FORGE tool set to analyze Fortran programs and to port them to a parallel machine.

(REV) *iPSC®/2 and iPSC®/860 Fortran Commands and Routines Quick Reference*
311615-004
Summarizes all Fortran routines and commands and for the iPSC system.

(REV) *iPSC®/2 and iPSC®/860 Hardware Installation Manual*
311461-003
(Replaces 311461-001 and 313990-001)
Describes installation and powering up of all iPSC system configurations.

(NEW) *iPSC®/2 and iPSC®/860 Interactive Parallel Debugger Commands Quick Reference*
312042-001
(Replaces 311798-001)
Summarizes all iPSC system IPD commands.

(NEW) *iPSC®/2 and iPSC®/860 Interactive Parallel Debugger Manual*
312043-001
(Replaces 311569-002 and 311826-001)
Tells how to use IPD, the iPSC system concurrent debugger.

iPSC®/2 and iPSC®/860 Math Libraries Reference Manual
311868-001
Describes the math libraries available on the iPSC system.

(NEW) *iPSC®/2 and iPSC®/860 Network Queueing System Manual*
312061-002
Tells how to use the network queueing system software.

(REV) *iPSC®/2 and iPSC®/860 Programmer's Reference Manual*
311708-004
(Replaces 311071-003, 311019-003, and 311831-001)
Provides detailed information on all C and Fortran routines and commands for the iPSC system.

(REV) *iPSC®/2 and iPSC®/860 Site Preparation Guide*
312028-001
Tells the customer how to prepare a site for the installation of an iPSC system.

iPSC®/2 and iPSC®/860 System Acceptance Test User's Guide
311870-001
Tells how to use the System Acceptance Test.

iPSC®/2 and iPSC®/860 System Administrator's Guide
311014-006
(Replaces 311842-001 and 311833-001)
Describes the system administration tasks related to operating and maintaining an iPSC system.

iPSC®/2 and iPSC®/860 User's Guide
311532-007
Overviews the iPSC system, including hardware and software architectures. Tells how to develop and run programs.

iPSC®/2 and iPSC®/860 VME Interface Reference Manual
311785-002
Describes the installation and development of software drivers for the VME Interface Adapter board.

iPSC®/2 Ada Commands and Routines Quick Reference
311759-002
Summarizes all Ada routines and commands for the iPSC/2 system.

iPSC®/2 Ada Program Development Guide
311768-001
Describes and tells how to use the tools for developing Ada programs for the iPSC/2 system.

iPSC®/2 Ada Program Development Guide Change Notice
311929-001
Adds information on how to use the Ada cross-debugger for developing Ada programs for the iPSC/2 system.

(REV) *iPSC®/2 Ada Programmer's Reference Manual*
311767-002
Describes all Ada routines and commands for the iPSC/2 system.

iPSC®/2 Ada Programmer's Reference Manual Change Notice
312045-001
Changes the **getcube()** manual page.

iPSC®/2 Fortran Language Reference Manual
311020-004
Describes the Green Hills Fortran compiler for the iPSC/2 system.

iPSC®/2 Lisp Language Reference Manual
311630-001
Describes the iPSC/2 implementation of Common Lisp. It explains iPSC/2 Lisp language features and extensions.

iPSC®/2 Lisp Language Reference Manual Change Notice
311799-001
Adds The Flavors System documentation to the *iPSC®/2 Lisp Language Reference Manual.*

iPSC®/2 Lisp Programmer's Reference Manual
311629-001
Provides detailed information on the iPSC/2 Lisp User Interface and iPSC/2 Lisp-unique Concurrent Constructs.

iPSC®/2 Lisp Programming Quick Reference
311631-002
Summarizes all Lisp constructs for the iPSC/2 system.

iPSC®/2 Simulator Manual
311534-003
Tells how to use the iPSC/2 Simulator for software development.

(NEW) *iPSC®/860 Basic Math Library User's Guide*
312128-001
Describes the basic linear algebra subroutines for the iPSC/860 systems.

(NEW) *iPSC®/860 C Compiler User's Guide*
312130-001
(Replaces 312006-001)
Describes the C cross-compiler and compiler driver for iPSC/860 systems.

(NEW) *iPSC®/860 Fortran Compiler User's Guide*
312131-001
(Replaces 312006-001)
Describes the Fortran cross-compiler and compiler driver for iPSC/860 systems.

(NEW) *iPSC®/860 Parallel Performance Analysis Tools Manual*
312139-001
Tells how to use the performance analysis software for the iPSC/860 system.

(NEW) *iPSC®/860 ProSolver™-SES Manual*
312141-002
Describes the skyline equation solver application software.

Intel® Manuals

UNIX System V Release 3.2 NFS User's/System Administrator's Guide and Reference
465725-001
Describes the NFS programming environment and provides user and system administration information.

UNIX System V Release 3.2 NFS Programmer's Guide and Reference
465726-001
Describes the NFS programming environment and tools.

UNIX System V Release 3.2 TCP/IP Administrator's Guide and Reference
465728-001
(Replaces Excelan TCP/IP documentation)
Describes TCP/IP Network administration.

UNIX System V Release 3.2 TCP/IP Programmer's Guide and Reference
465729-001
(Replaces Excelan TCP/IP documentation)
Describes the TCP/IP Network programming environment and provides information on programming tools.

UNIX System V Release 3.2 TCP/IP User's Guide and Reference
465727-001
(Replaces Excelan TCP/IP documentation)
Describes the TCP/IP Network programming environment and provides user information.

i860™ 64-Bit Microprocessor Assembler and Linker Reference Manual
240436-003
Tells how to use the i860 microprocessor assembler and linker. When you order this manual, you also receive the following manuals:

i860™ 64-Bit Microprocessor Object File Utilities Reference Manual
464410-002
Provides reference information for using the i860 microprocessor object file utilities.

i860™ 64-Bit Microprocessor Simulator and Debugger Reference Manual
240437-003
Describes the i860 microprocessor debugger and simulator.

i860™ 64-Bit Microprocessor Programmer's Reference Manual
240329-002
Tells how to use the i860 microprocessor.

SYP301 Installation and User's Guide
451684-001
Tells how to install and start the System Resource Manager. Also provides hardware technical data.

UNIX Release R3.2 Manuals Literature Kit, UNXSYS386R3.2
Consists of the following documents:

(NEW) *UNIX System V Integrated Software Development Guide*
465274-001
Supplies information needed to write application software and installable drivers for new hardware additions for UNIX.

(NEW) *UNIX System V Introduction to UNIX System V*
465273-001
Introduces you to UNIX System V Release 3.2 on PC AT compatible computers using Intel386™ and Intel486™ microprocessors.

(NEW) *UNIX System V Network Programmer's Guide*
465282-001
Describes the UNIX System network programming environment, and provides detailed descriptions of programming tools.

(NEW) *UNIX System V Programmer's Guide, Vol. I*
465277-001
Describes the UNIX System programming environment, and provides detailed descriptions of programming tools.

(NEW) *UNIX System V Programmer's Guide, Vol. II*
465278-001
Describes the UNIX System programming environment, and provides detailed descriptions of programming tools.

(NEW) *UNIX System V Programmer's Reference Manual*
465276-001
Contains descriptions of commands, system calls, subroutines, libraries, file formats, macro packages, and character set tables.

(NEW) *UNIX System V Software Development Set*
465255-001
Provides UNIX System V Release 3.2 release notes.

(NEW) *UNIX System V Streams Primer*
465283-001
Introduces the UNIX System streams programming environment.

(NEW) *UNIX System V Streams Programmer's Guide*
465279-001
Describes the UNIX System streams programming environment, and provides detailed descriptions of programming tools.

(NEW) *UNIX System V System Administrator's Guide*
465280-001
Describes system maintenance tasks performed on the System Resource Manager under UNIX.

(NEW) *UNIX System V System Administrator's Reference Manual*
465281-001
Describes the UNIX System commands used by system administrators.

(NEW) *UNIX System V User's Guide*
465275-001
Provides a general description of UNIX.

Other Manuals

C: A Reference Manual - Harbison and Steele
480628-001
Describes the C programming language.

(NEW) *Effective Fortran 77* - Michael Metcalf
312201-001
Describes the Fortran 77 programming language.

Reference Manual For The Ada Programming Language - ANSI/MIL-STD-1815A-1983
311795-001
Describes the Ada programming language.

TCP/IP for UNIX Software Reference Guide (Release 2.6 Man Pages) - CMC
312171-001
Provides software reference information for the TCP/IP Network programming environment associated with the CMC Ethernet board.

(NEW) *TCP/IP for UNIX Release 2.6 User's Guide* - CMC
312170-001
Describes the TCP/IP Network programming environment associated with the CMC Ethernet board and provides user information.

The C Programming Language - Kernighan and Ritchie
122008-002
Describes the C programming language.

The X Window System Manual Set
311953-001
Consists of the following documents:

Volume 0 - X Protocol Reference Manual - O'Reilly & Associates, Inc.
ISBN 0-937175-40-0
Provides reference information on X Network Protocol, the language for communication between the X server and X client.

Volume 1 - Xlib Programming Manual - O'Reilly & Associates, Inc.
ISBN 0-937175-26-9
Tells how to program using the X library, the lowest level programming interface to the X window system.

Volume 2 - Xlib Reference Manual - O'Reilly & Associates, Inc.
ISBN 0-937175-12-9
Provides reference information for programming using the X library.

Volume 4 - X Toolkit Intrinsics Programming Manual - O'Reilly & Associates, Inc.
ISBN 0-937175-34-X
Tells how to program using the X Toolkit.

Volume 5 - X Toolkit Intrinsics Reference Manual - O'Reilly & Associates, Inc.
ISBN 0-937175-35-8
Provides reference information for programming using the X Toolkit.

UNIX V - The Quick Reference Guide
311533-003
Summarizes UNIX commands, buzzwords, C shell hints and standard directory layout.

Table of Contents

Chapter 1
Product Features

Chapter 2
iPSC®/2 Software Installation

Chapter 3
iPSC®/860 Software Installation

Chapter 4
Limitations and Workarounds

Chapter 5
Software Enhancements

List of Tables

Product Features 1

Introduction

This chapter describes all of the new product features provided in Release 3.3.1 and Release 3.3.

New Features in Release 3.3.1

The following features are provided in the iPSC Release 3.3.1 system software:

1. **comminfo Command**
   comminfo is a new command that has been added to display the queues of pending messages on the SRM. **comminfo** has two optional arguments:

-a	Returns message information for the cubes that you own on the SRM from which you invoked the command.
-s	Returns message information for all cubes on the SRM from which you invoked the command.

   Executing the **comminfo** command with no arguments returns message information for the current attached cube.

   For example, when the system is running and usable with no user activity taking place, the **comminfo -s** command will show the following message information:

```
% comminfo -s
                Pending RECV messages

  Waiting pid   Message Type   Blocked   Username   Cubename      ID
  ===========   ============   =======   ========   ===========   ==
         -34     1000000015       yes   root       iocube         1
         -64     1000000020       yes   root       iocube         1
        2329     2000000041       yes   root       iocube         1
                All Host Send Message Queues Are Empty
```

In the previous example, under "Pending RECV messages":

- PID -34 (*loader.srm*) is waiting for a message of type 1000000015 (**LOAD_REQUEST**)
- PID -64 (*fserver*) is waiting for a message of type 1000000020 (**FILE_REQUEST**)
- PID 2329 (*comminfo*'s UNIX PID) is waiting for a type 2000000041 (**COMMSER_REPLY**).

All of these message are associated with cube ID 1 (*iocube*).

Note that negative PIDs are system processes.

Here is an example of the message information returned for a **comminfo -s** command when the system is busy:

```
                Pending RECV messages

  Waiting Pid   Message Type   Blocked   Username   Cubename      ID
  ===========   ============   =======   ========   ===========   ==
         -39             -1       yes   jclark     joel           5
         -34     1000000015       yes   root       iocube         1
         -64     1000000020       yes   root       iocube         1
         101              6       yes   soetrisn   HANA          10
       23727     2000000041       yes   root       iocube         1

                      Pending SEND messages
```

```
      For               From              Type       Size    Username Cubename    ID
================ ================ ========== ========= ======== ========== ==
(host:      200) (  48:         0)        300         8 weeratun ARC3D       6
(host:      -64) (  25:         0) 1000000020        45 weeratun ARC3D       6
(host:      -65) (  40:         0) 1000000020        45 weeratun ARC3D       6
(host:      -65) (  64:       -62) 1000000020       100 weeratun ARC3D       6
(host:      -65) (  64:       -62) 1000000020       100 weeratun ARC3D       6
(host:      -64) (  64:       -62) 1000000020       100 weeratun ARC3D       6
(host:      -65) (  64:       -62) 1000000020       100 weeratun ARC3D       6
(host:      -64) (  10:         0) 1000000020        52 soetrisn HANA       10
(host:      -64) (   2:         0) 1000000020        52 soetrisn HANA       10
(host:      -64) (   3:         0) 1000000020        52 soetrisn HANA       10
(host:      -64) (   9:         0) 1000000020        52 soetrisn HANA       10
```

In the previous example, under "Pending RECV messages":

- PID -39 (*tohost*) is waiting for any type message for the remote host cube *joel* owned by user *jclark*, cube ID 5.
- PID -34 (*loader.srm*) is waiting for a message of type 1000000015 (**LOAD_REQUEST**).
- PID -64 (*fserver*) is waiting for a message of type 1000000020 (**FILE_REQUEST**) for the *iocube*.
- PID 101 associated with cube ID 10 owned by user *soetrisn*, is waiting for a message of type 6 (an application-specific message type).
- PID 23727 (*comminfo*'s UNIX PID) is waiting for a message of type 2000000041 (**COMMSER_REPLY**).

Under "Pending SEND messages":

- In cube 6, owned by user *weeratun* there are several messages:
  - The first one is a type 300 (non-system) message sent from node 48 to the host PID 200.
  - The next two are file service requests (which may be **printf()**s) from nodes 25 and 40 to host PIDs -64 and -65 respectively. -64 and -65 are both *fserver* PIDs. Notice that PID -64 only serves nodes 0 through 31. PID -65 serves nodes 32 through 63, PID -66 would serve nodes 64 through 95, and PID -67 would serve nodes 96 through 127.
  - The next four messages are messages of type 1000000020 from PID -62 (*adminproc*) destined for the *fservers*. These are most likely messages notifying the *fservers* that processes have terminated in cube 6.

- The next four messages, type 1000000020, are file service requests from various nodes in cube 10, owned by user *soetrisn*.

If you have the iPSC®/2 or iPSC®/860 Source Product Code, the iPSC source files */usr/ipsc/src/h/pid.h* and */usr/ipsc/src/h/msgtype.h* define the use of system PIDs and message types. If you have Remote Host installed, the source files */usr/ipsc/rhost/h/pid.h* and */usr/ipsc/rhost/h/msgtype.h* define the use of Remote Host PID's and message types, although the **comminfo** command is not supported on Remote Host.

Use this information to help in determining how the system buffers are used and if there is a problem with the system. Users may be asked to cleanup the buffers by receiving the pending messages or killing and releasing their cubes.

2. Tuning Your Network
Although the UNIX kernel for the SRM has been tuned to give good performance in most cases, the default configuration may not be optimal for your computing environment. Once your system has been up and running in a typical usage pattern for a month or so, you may want to consider additional Unix kernel tuning to get the most out of your SRM.

Information on tuning the UNIX kernel can be found in Chapter 5 of the "Operations/System Administrator's Guide" of the SVR3.2 manual set. SSD Technical Support can supply additional information on tuning your SRM. You can contact SSD Technical Support at:

1-800-421-2823 (Customer Support Response Center)
Your Local Intel Sales Office (in Europe)
support@ssd.intel.com (Internet address)

3. **cfschk** Command
Each time bootcube runs, it performs a consistency check on all CFS files. This check is part of the rc3 startup run script (initializing the CIO system) in step 23. If problems are found, bootcube tries to fix them. If no problems remain in CFS, bootcube finishes normally. If file system inconsistencies are found that cannot be fixed, however, the errors are reported to stdout before bootcube finishes.

Any file system inconsistency errors reported by **bootcube** should be noted for future reference. A history file showing failure trends can be useful in isolating problems with specific disks and controllers. The error message format indicates the file exhibiting the error, the volume containing the error, and the block within the volume.

If file system inconsistency errors are reported, you can use the **cfschk** utility to temporarily repair the file system. At the next opportunity (e.g., when the iPSC system is powered down for maintenance) you can use the **mkcfs** command to remake the entire file system. If neither the **cfschk** nor the **mkcfs** procedures fix the file system, contact the SSD Customer Support Response Center.

The following example shows the procedure to follow when using the **cfschk** utility to repair file system inconsistency errors reported by **bootcube**:

File system inconsistency errors are reported during a **bootcube** as follows:

```
# bootcube
  .
  .
23.  Execute startup run file: /usr/ipsc/lib/rc3  -Q /usr/ipsc/conf/cubeconf

Found 56 volumes of file system CIT

Checking 56 volumes with 511 nodes
Starting Phase I, Checking directories and files
Starting Phase II
ERROR[0]: usr/i860/bin/dchk,     FHDR: [10, 35]
      fchk: bad magic number in file header:  header block duplicated
ERROR[1]: usr/ipsc/log/process.log,     FHDR: [47, 30]
      fchk: duplicate data blocks block [10, 35] in blist [47, 30] index 19

771 files checked with: 2 errors and 1 duplicated disk blocks.
Run cfschk after system is booted.
CFS initialization complete
#
```

The errors reported in this example provide enough information to locate the problem area. For example, error #0 detected a bad magic number in the file header for file */usr/i860/bin/dchk* at volume #10, block #35.

Run the **cfschk** utility to repair any file system inconsistency errors reported by **bootcube**, as follows:

```
# cfschk
Found 56 volumes of file system CIT

Checking 56 volumes with 511 nodes
Starting Phase I, Checking directories and files
FIXED[0]: usr/i860/bin/dchk,     FHDR: [10, 35]
      fchk: bad magic number in file header:  header block duplicated
FIXED[1]: usr/ipsc/log/process.log,     FHDR: [47, 30]
      fchk: duplicate data blocks block [10, 35] in blist [47, 30] index 19

771 files checked
CFS initialization complete
```

The **cfschk** utility will try to rename any files that have bad file headers. The utility appends a "#" character to the file name of any suspect files, and changes permissions to read/write for all users. These changes make it easy for the system administrator to identify the corrupted files and remove them from CFS.

If the **cfschk** utility fixes the files, it returns with no errors. If errors are still reported, make the file system using the **mkcfs** command. If the file system still is not fixed, contact the SSD Customer Support Response Center.

New Features in Release 3.3

The following features were provided in the iPSC Release 3.3 system software:

1. **Improved iPSC®/860 C and Fortran compilers**
   The iPSC/860 C and Fortran compilers have been improved over those delivered with Release 3.2. The new compilers are called **icc** and **if77**, and can be run on Sun-3 and Sun-4 workstations as well as on the SRM.

2. **Interactive Parallel Debugger**
   The new Interactive Parallel Debugger (IPD) replaces DECON as the source-level application debugger for the iPSC/2 and iPSC/860 systems. IPD offers conventional debugging facilities, such as breakpoints and the ability to display and modify memory, as well as features for examining the state of the message passing network. Differences from DECON include improved reliability, i860 machine-level debugging, and enhanced iPSC/860 Fortran support.

3. **Parallel Performance Analysis Tools**
   Release 3.3 includes an early release of the new Performance Analysis Tools (PAT). These tools help you analyze your iPSC/860 application programs and improve their performance. PAT consists of three utilities: an execution profiler that monitors the time spent in individual routines, a communication trace tool that assesses the time spent in communication, and an event trace tool that shows the interactions between processors and can monitor user-specified events. Performance data collected by the PAT utilities can be displayed graphically on Sun-3 and Sun-4 workstations. When using the Performance Analysis Tools, all nodes allocated with the **getcube** command must be loaded with the same compiled user program that has Performance Analysis turned on. The Performance Analysis Code expects to communicate with each of the nodes in the cube and will appear to hang while waiting for communications from nodes without the same user program. Although the use of PAT was discouraged in Release 3.3 due to some product limitations, this version of PAT is ready for use.

4. **Profiler for iPSC®/860 node programs**
   The new **prof860** command lets you profile execution of RX node programs. It is based on the UNIX System V **prof** command and runs on Sun-3 and Sun-4 workstations as well as the SRM.

5. **Basic Math Library**
   The new Basic Math Library is an implementation of the BLAS (Basic Linear Algebra Subprograms) levels 1, 2, and 3 for iPSC/860 programs. It includes some one-dimensional Fast Fourier Transform (FFT) operations as well as the vector-vector operations, matrix-matrix operations, and vector-matrix operations of BLAS.

6. **Network Queueing System**
   The Network Queueing System (NQS), the *de facto* standard for batch queueing on supercomputers, is now supported on the SRM for iPSC/860 systems. It is completely compatible with the COSMIC version of NQS from NASA, and has been modified so that it can load cube jobs.

7. **More memory on RX nodes**
   The system software now supports RX nodes with 32 and 64M bytes of memory on iPSC/860 systems, as well as the existing 8 and 16M byte configurations.

8. **Node Shell on service node**
   The Node Shell (**nsh**) now executes on any I/O node without a SCSI interface or a customized VME interface (such as an ENET node or service node) rather than on the currently attached cube. This lets you invoke a shell on your iPSC/2 or iPSC/860 system without tying up any compute nodes. The new **-s** option to **nsh** makes the shell execute on the current cube, for backwards compatibility.

9. **Improved remote host security**
   A new file, */usr/ipsc/lib/ruser*, lists the workstations and users or groups who are allowed to access the iPSC system's remote host facilities. This new feature prevents unauthorized users from using your iPSC system over the network; it is available on both iPSC/2 and iPSC/860 systems. All remote host user's are required to have an entry in the */etc/passwd* file.

10. **NX information structure**
    NX for the iPSC/860 system stores performance and status information (about message passing, interrupts, CPU usage, and so on) in a special read-only structure called *nxinfo*. This structure has now been made accessible to application programmers, who can use the information in it for performance improvements and checking the status of the nodes.

11. **Enhanced accounting and security**
    The **getcube** command and *commser* process have been enhanced to provide hooks for user-developed accounting packages on both the iPSC/2 and iPSC/860 systems. You can use these hooks to keep track of system activity and control system access with a higher level of detail than the standard iPSC system administration tools provide. The **plogon** command and call also record more information than they did in Release 3.2, and the **plogon** command now has an **-i** option to report the pathname of the current log file.

NOTE

The **getcube** command now contains a new *account* parameter. Please refer to the *iPSC®/2 and iPSC®/860 Programmer's Reference Manual* for more information on how to use the *account* parameter.

12. **Non-IEEE arithmetic for iPSC®/860 programs**
This release supports a faster, less accurate non-IEEE mode for divide and square root, and the flushing of denormal data values to zero for RX node programs, in addition to the IEEE operations provided in Release 3.2.

13. **NX message buffering modification on iPSC®/860 systems**
The **getcube** command has been enhanced to optionally disable message buffering on RX nodes. This provides a performance boost for some programs which overlap communication with computation. In addition, the default message buffering algorithm for RX nodes has been optimized for better performance.

iPSC®/2 Software Installation 2

Introduction

This chapter describes how to install the Release 3.3.1 software. The R3.3.1 system software runs on iPSC®/2 (with Intel386™-microprocessor-based CX nodes) and iPSC®/860 systems (with i860™-microprocessor-based RX nodes or with a combination of CX and RX nodes). To install Release 3.3.1 software on an iPSC®/2 system, follow the steps outlined below. To install Release 3.3.1 software on an iPSC®/860 system, please follow the steps outlined in Chapter 3.

Installing Release 3.3.1 Software on an iPSC®/2 System

CAUTION

If your software release is 3.2 or earlier and if you use CIO Ethernet or Ada, *do not* install this Release 3.3.1 software on your system until you have the updated version of each package. Earlier versions of these software packages are not compatible with Release 3.3.1 of the system software unless otherwise noted.

The correct versions of these packages for use with Release 3.3.1 are as follows:

- CIO Ethernet Release 1.1 and the diskette labeled *iPSC/860 CIO Ethernet Update*
- Ada Release 1.2
- Lisp Release 1.4

If you have any questions, please contact SSD Customer Support:

1-800-421-2823 (Customer Support Response Center)
Your Local Intel Sales Office (in Europe)
support@ssd.intel.com (Internet address)

There are three ways to use the procedures in this chapter:

- You are upgrading your system from Release 3.3 to Release 3.3.1. In this case, begin with "Installation Instructions for Update Customers" on page 2-4. You *do not* have to reinstall the UNIX operating system.

- You are upgrading your system from a release earlier than Release 3.3. In this case, you must reinstall all system software from scratch. Begin with "Installing UNIX System V/386 Release 3.2 Version 2.1" on page 2-13.

- You are reinstalling your Release 3.3.1 system software because it has become damaged. In this case, begin with "Installing UNIX System V/386 Release 3.2 Version 2.1" on page 2-13.

If the software on your system becomes damaged, please contact SSD Customer Support before performing any installations.

NOTE

Before performing any procedure in this chapter, read through the procedure completely to make certain that you understand what the procedure involves. If you have any questions before or during any procedure, please call SSD Customer Support for help.

If you perform the procedures in this chapter exactly as described, you will create a software system that is identical to the one originally installed by SSD.

The procedures in this chapter use the conventions described in the preface. You should also be aware of the following conventions used in these procedures:

- The instruction "Enter ***character(s)***" means type the indicated character(s), and then press the **<Enter>** key. For example, "Enter **y**" means type the letter `y`, and then press the key labeled `Enter`.

- In prompts, square brackets surround a default value. Pressing **<Enter>** selects the indicated default value.

- Some steps in these procedures cause a lot of information to be displayed on the system monitor. However, the step usually shows only the last message displayed. Do not be concerned if the indicated message does not appear immediately. Be patient. Some steps take several minutes to complete.

Software Dependencies

Table 2-1 describes the dependencies between the various software packages in this release.

Table 2-1. Software Dependencies for Release 3.3.1 Installation

If you want to install this:	You must install this first:
TCP - Intel Corp. Release 3.0.1	UNIX System V Release 3.2 Cartridge Tape Utilities Version 2.1
Ethernet Drivers Package for Intel TCP/IP Version 3.0.1	UNIX System V Release 3.2 TCP - Intel Corp. Release 3.0.1
iPSC System Software Release 3.3.1	UNIX System V Release 3.2 Cartridge Tape Utilities Version 2.1 TCP Intel Corp. Release 3.0 Ethernet Drivers Package for Intel TCP/IP Version 3.0
iPSC Remote Host Software Release 3.3.1	UNIX System V Release 3.2 Cartridge Tape Utilities Version 2.1
iPSC Simulator Software Release 3.0	UNIX System V Release 3.2 Cartridge Tape Utilities Version 2.1
iPSC NQS software Release 3.0	UNIX System V Release 3.2 Cartridge Tape Utilities Version 2.1

Table 2-1 describes only the dependencies to *install* the named package. Further dependencies may exist in order to *use* the software in the package.

Disk Space Requirements

Table 2-2 lists the disk space required by each iPSC software package in this release.

Table 2-2. Disk Space Requirements for Release 3.3.1 (1 of 2)

Package	Size	Difference from 3.3
TCP - Intel Corp. Release 3.0.1	3.0M bytes	0
Ethernet Drivers Package for Intel TCP/IP Version 3.0.1	1.1M bytes	+ or -
iPSC System Software Release 3.3.1	22.2M bytes	+2.9M bytes
iPSC Remote Host Software Release 3.3.1	1.3M bytes	+ 0.4M bytes
iPSC Simulator Software Release 3.0	0.7M bytes	0 [1]

Table 2-2. Disk Space Requirements for Release 3.3.1 (2 of 2)

Package	Size	Difference from 3.3
iPSC NQS Software Release 1.0	2.6M bytes	0 [1]
1. This package has not changed from Release 3.3.		

To determine the free disk space you will need to upgrade your system from Release 3.3 to Release 3.3.1, add the numbers in the "Difference from 3.3" column for the packages you are installing.

Installation Instructions for Update Customers

The following instructions must be followed if you have Release 3.3, Update 9/16 or 10/16 installed on your iPSC®/2 system. Please follow the instructions, in sequence, in order to successfully install Release 3.3.1 software.

Removing Update 9/16 or 10/16 and Release 3.3 Software

Removal Time: Approximately 30 minutes.

Information you need: Which update is installed on your system.

---

1. Login as *root*.

   Make sure that the users are not logged in and the system is not NFS mounted.

2. Shutdown the cube by entering:

   ```
   shutdown -D1
   ```

3. Shutdown to single user mode by entering:

   ```
   shutdown -is
   ```

   The following message is displayed:

   ```
   Type Ctrl-d to proceed with normal startup
   (or give root password for system maintenance):
   ```

Enter the root password.

4. The single-user shutdown does not automatically mount the user partition. The **removepkg** command resides in this partition. Mount the "usr" partition by entering the following command:

```
/etc/mount  /usr
```

5. Enter ***displaypkg***.

   This will identify which software update package is installed on the system. If you have release 3.3 installed, you may also have either Update 9/16 or Update 10/16 installed. If you have Update 9/16 installed, the following message will appear:

```
iPSC S1 System Software Update Release 3.3
```

   If you have Update 10/16 installed, the following messages will appear:

```
iPSC S2 System Software Update Release 3.3
iPSC R2 Remote Host Software Update Release 3.3
iPSC T2 TCP - Intel Corp. Update Release 3.0
```

   The Remote Host software is optional and may not be installed.

Removing Update 9/16

6. If you have Update 9/16 installed, you need to remove the system software update using the command **removepkg**. When you enter the ***removepkg*** command, the system displays all software packages installed on the system.

   Enter the number corresponding to the `iPSC S1 System Software Update Release 3.3` software package.

   When the following message appears:

```
Confirm

Do you really want to remove iPSC S1 System Software Update
Release 3.3?

Strike ENTER when ready
or ESC to stop.
```

   Press **<Enter>**.

When the following message appears:

```
Confirm

To complete the install/removal process a shutdown is now
being initiated automatically.

Make sure your floppy drive is empty.  If you are installing
or removing controller boards, you may power down the system
after the shutdown has completed.

Strike ENTER when ready
or ESC to stop.
```

Press **<Esc>**.

NOTE

If you press any key other than **<Esc>**, the system will shutdown. If the system does shutdown, you must first reboot the system, shutdown into single user mode, and mount the usr partition to continue the software removal process. This process is explained in the beginning of this section. You can then continue with the removal process with the step after the last one you completed.

The software for this step has now been successfully removed.

NOTE

Now, to continue the software removal process, please go to "Removing iPSC® Software Release 3.3" on page 2-9.

Removing Update 10/16

7. If you have Update 10/16 installed, you need to remove the system software update using the command **removepkg**. When you enter the ***removepkg*** command, the system displays all software packages installed on the system.

   Enter the number corresponding to the `iPSC S2 System Software Update Release 3.3` software package.

When the following message appears:

```
Confirm

Do you really want to remove iPSC S2 System Software Update
Release 3.3?

Strike ENTER when ready
or ESC to stop.
```

Press **<Enter>.**

When the following message appears:

```
Confirm

To complete the install/removal process a shutdown is now
being initiated automatically.

Make sure your floppy drive is empty.  If you are installing
or removing controller boards, you may power down the system
after the shutdown has completed.

Strike ENTER when ready
or ESC to stop.
```

Press **<Esc>.**

NOTE

If you press any key other than **<Esc>**, the system will shutdown. If the system does shutdown, you must first reboot the system, shutdown into single user mode, and mount the usr partition to continue the software removal process. This process is explained in the beginning of this section. You can then continue with the removal process with the step after the last one you completed.

The software for this step has now been successfully removed.

8. If you do not have the Remote Host Update package installed on your system, skip to instruction number 9. At the # prompt, enter the **removepkg** command.

   Enter the number corresponding to the `iPSC R2 Remote Host Software Update Release 3.3` software package (if this package is installed on your system.)

When the following message appears:

```
Confirm

Do you really want to remove iPSC R2 Remote Host Software
Update Release 3.3?

Strike ENTER when ready
or ESC to stop.
```

Press **<Enter>**.

When the following message appears:

```
Confirm

To complete the install/removal process a shutdown is now
being initiated automatically.

Make sure your floppy drive is empty.  If you are installing
or removing controller boards, you may power down the system
after the shutdown has completed.

Strike ENTER when ready
or ESC to stop.
```

Press **<Esc>**.

9. At the # prompt, enter the **removepkg** command.

Enter the number corresponding to the `iPSC T2 TCP - Intel Corp. Update Release 3.0` software package.

When the following message appears:

```
Confirm

Do you really want to remove iPSC T2 TCP - Intel Corp. Update
Release 3.0?

Strike ENTER when ready
or ESC to stop.
```

Press **<Enter>**.

When the following message appears:

```
Confirm

To complete the install/removal process a shutdown is now
being initiated automatically.

Make sure your floppy drive is empty.  If you are installing
or removing controller boards, you may power down the system
after the shutdown has completed.

Strike ENTER when ready
or ESC to stop.
```

Press **<Esc>**

Removing iPSC® Software Release 3.3

Do not remove the following software packages: (you may or may not have them installed)

```
iPSC NQS Software Release 1.0
iPSC Simulator Software Release 3.0
```

10. Remove the Release 3.3 software packages by entering the **removepkg** command:

    Enter the number corresponding to the `iPSC System Software Release 3.3` software package.

    When the following message appears:

```
Confirm

Do you really want to remove iPSC System Software Release 3.3?

Strike ENTER when ready
or ESC to stop.
```

    Press **<Enter>.**

    When the following message appears:

```
Confirm

To complete the install/removal process a shutdown is now
being initiated automatically.
```

```
Make sure your floppy drive is empty.  If you are installing
or removing controller boards, you may power down the system
after the shutdown has completed.

Strike ENTER when ready
or ESC to stop.
```

Press **<Esc>**.

11. If you do not have the Remote Host software package installed, proceed to instruction number 12. Otherwise, enter the **removepkg** command:

    Enter the number corresponding to the `iPSC Remote Host Software Release 3.3` software package.

    When the following message appears:

```
Confirm

Do you really want to remove iPSC Remote Host Software Release
3.3?

Strike ENTER when ready
or ESC to stop.
```

    Press **<Enter>**.

    When the following message appears:

```
Confirm

To complete the install/removal process a shutdown is now
being initiated automatically.

Make sure your floppy drive is empty.  If you are installing
or removing controller boards, you may power down the system
after the shutdown has completed.

Strike ENTER when ready
or ESC to stop.
```

    Press **<Esc>**

12. Enter the **removepkg** command:

    Enter the number corresponding to the `Ethernet Drivers Package for Intel TCP/IP Version 3.0` software package.

When the following message appears:

```
Confirm

Do you really want to remove Ethernet Drivers Package for
Intel TCP/IP Version 3.0?

Strike ENTER when ready
or ESC to stop.
```

Press **<Enter>.**

When the following message appears:

```
Confirm

To complete the install/removal process a shutdown is now
being initiated automatically.

Make sure your floppy drive is empty.  If you are installing
or removing controller boards, you may power down the system
after the shutdown has completed.

Strike ENTER when ready
or ESC to stop.
```

Press **<Esc>**

CAUTION

The *remove* and *install* scripts for TCP/IP remove/overwrite the following configuration files:

/etc/hosts
/etc/networks
/etc/gateways
/etc/hosts.equiv
/usr/lib/named/named.hosts
/usr/lib/named/named.local
/usr/lib/named/named.rev
/usr/lib/named/named.soa
/usr/lib/named/root.cache

If you are updating a previous release of Intel TCP/IP, you may want to rename or backup these files *before* removing the previous release. After you successfully install TCP/IP Release 3.0.1, you can restore these files.

13. Enter the **removepkg** command:

    Enter the number corresponding to the `TCP - Intel Corp. Release 3.0` software package.

    When the following message appears:

```
Confirm

Do you really want to remove TCP - Intel Corp. Release 3.0?

Strike ENTER when ready
or ESC to stop.
```

    Press **<Enter>**.

    When the following message appears:

```
Confirm

To complete the install/removal process a shutdown is now
being initiated automatically.

Make sure your floppy drive is empty.  If you are installing
or removing controller boards, you may power down the system
after the shutdown has completed.

Strike ENTER when ready
or ESC to stop.
```

    Press **<Esc>**.

14. Enter the **removepkg** command:

    Enter the number corresponding to the `PC586 Ethernet Driver - Version 1.0` software package.

    When the following message appears:

```
Confirm

Do you really want to remove the PC586 Ethernet Driver -
Version 1.0?

Strike ENTER when ready
or ESC to stop.
```

When the following message appears:

```
Confirm

To complete the install/removal process a shutdown is now
being initiated automatically.

Make sure your floppy drive is empty.  If you are installing
or removing controller boards, you may power down the system
after the shutdown has completed.

Strike ENTER when ready
or ESC to stop.
```

15. Press **<Esc>**.

NOTE

Go to the section "Installing TCP/IP and Ethernet Drivers" on page 2-25 to continue the upgrade process.

Complete Installation Process (not an update)

Installing UNIX System V/386 Release 3.2 Version 2.1

Use these installation procedures to perform a new and/or complete installation.

CAUTION

Installing the UNIX software reformats the hard disk. To prevent loss of data, save all user files and any system files that have been modified *before* installing the UNIX software. The cpio utility is the preferred method of doing this.

If you are upgrading from UNIX Release 3.0, *do not replace the new system files with the modified system files* that you saved. Doing so downgrades those system files to UNIX Release 3.0. Instead, after installing the new UNIX software, make the same modifications to the new system files.

There are six major steps to installing the UNIX software:

1. Install the root and user filesystems
2. Install the base UNIX operating system
3. Install the Cartridge Tape Utilities
4. Install the UNIX add-on packages
5. Set up user and system logins
6. Install the remote terminal package

To install the UNIX software, you must perform all parts in the order specified.

Installing the Root and User Filesystems

Installation Time:	Approximately 30 minutes.
Installation Medium:	Floppy diskette (1) labeled "Boot Floppy."
Information you need:	Size of your hard disk (140MB or 380MB).

---

1. If the system is on, turn it off as described in Chapter 2 in the *iPSC®/2 and iPSC®/860 System Administrator's Guide*, in the sections "Stopping the UNIX Operating System" and "Removing Power."
2. Insert the installation diskette into the diskette drive.
3. Turn the system on.
4. When the following message appears:

   ```
   Strike ENTER to install the UNIX System on your hard disk.
   ```

   Press **<Enter>**.

5. When the following message appears:

```
WARNING: A new installation of the UNIX System will destroy
all files currently on the system.  Do you wish to continue
(y or n)?
```

Enter *y*.

6. When the following message appears:

```
SELECT ONE OF THE FOLLOWING:

   1.  Create a partition
   2.  Change Active (Boot from) partition
   3.  Delete a partition
   4.  Exit (Update disk configuration and exit)
   5.  Cancel (Exit without updating disk configuration)
Enter Selection:
```

Enter *5* if the partitioning is 100% UNIX, Active, with no other partitions. (The partitioning was probably done when your system was originally installed.)

If the partitioning is not 100% UNIX, Active, with no other partitions,then:

Enter *3* to delete the default partition

Enter *1* to create the desired partition.

7. When the following message appears:

```
A surface analysis will now be done.
This will destroy all data on the hard disk.
Strike ENTER to continue or DEL to abort.
```

Press **<Enter>**.

8. Following a message describing a suggested partitioning, this message appears:

```
Is this allocation acceptable to you (y/n)?
```

Before responding to this message, make a note of the suggested number of cylinders for the root file system, and for the swap/paging area. You will use these to answer the questions in steps 11 and 12 below.

Then, to respond to the allocation question, enter *n*. This keeps the partitioning that was originally sent to you.

9. When the following message appears:

```
Do you wish to have separate root and usr filesystems (y/n)?
```

Enter ***y***.

10. When the following message appears:

```
Do you want an additional /usr2 filesystem (y/n)?
```

Enter ***n***.

11. When the following message appears:

```
How many cylinders would you like for swap/paging (1-xxx)?
```

Enter the number of swap/paging cylinders you noted in step 8 above. Typical numbers are ***109*** for a 140M-byte drive, and ***78*** for a 380M-byte drive.

12. When the following message appears:

```
How many cylinders would you like for root (1-xxx)?
```

Enter the number of root cylinders you noted in step 8 above. Typical numbers are ***201*** for a 140M-byte drive, and ***143*** for a 380M-byte drive.

13. When the following message appears:

```
Is this allocation acceptable to you (y/n)?
```

Enter ***y*** if the cylinder quantities for root and swap/paging are what you entered.

14. When the following message appears:

```
Reboot the system now.
```

The installation of the root and user filesystems is complete.

15. Remove the diskette from the diskette drive.

To complete the installation of the UNIX software, install the base UNIX operating system, as described in the next procedure.

Installing the Base UNIX Operating System

NOTE

You must install the root and user filesystems (as described in the previous procedure) *before* you install the base UNIX operating system.

Installation Time:	Approximately 20 minutes.
Installation Medium:	Cartridge tape labeled "UNIX System V/386, R3.2 V2.1". Diskette labeled "Cartridge Tape Utilities"
Information you need:	*root* password. *install* password.

---

16. Press **<Ctrl-Alt-Del>** to reboot the system.

17. When the following message appears:

```
Installation from Cartridge Tape is available using Interrupt
#5 and Address Range 300 through 301.
Are you installing from tape (y/n)?
```

    Enter *y*.

18. When the following message appears:

```
Please make sure your Cartridge Tape hardware is configured
correctly. Insert System Installation Tape in drive and press
<RETURN>
```

    A. Insert the installation tape into the tape drive (label up, exposed tape to the left; push until the tape cartridge locks into place).

    B. Press **<Enter>**.

19. When the system prompts for a *root* password, enter the password that you have chosen for the *root* login.

20. When the system prompts for an *install* password, enter the password that you have chosen for the *install* login.

21. When the following message appears:

```
Confirm

Do you want to install the Cartridge Tape Drive?
Press ENTER for YES or ESC for NO

Strike ENTER when ready
or ESC to stop.
```

    Press **<Esc>**.

22. When the following message appears:

```
Reboot the system now
```

    Press **<Ctrl-Alt-Del>** to reboot the system.

23. When the following message appears:

```
Console login:
```

24. Login as *root*.

25. Enter ***installpkg***.

26. Insert the Cartridge Tape Utilities installation diskette into the diskette drive and turn the handle down.

27. When the following message appears:

```
Strike ENTER when ready
or ESC to stop
```

    Press **<Enter>**.

28. When the following message appears:

```
Type the interrupt number and strike the ENTER key or
type Q to cancel installation.
```

    Enter *5*.

29. When the following message appears:

```
Strike ENTER when ready
or ESC to stop
```

Press **<Enter>**.

30. Again, when the following message appears:

```
Strike ENTER when ready
or ESC to stop
```

Press **<Enter>**.

31. When the following message appears:

```
Reboot the system now.
```

A. Remove the Cartridge Tape Utilities diskette from the diskette drive.

B. Leave the tape in the tape drive for the next procedure.

C. Press **<Ctrl-Alt-Del>** to reboot the system.

32. When the following message appears:

```
Console login:
```

The installation of the base UNIX operating system is complete.

To complete the installation of the UNIX software, install the UNIX add-on packages, as described in the next procedure.

Installing the UNIX Add-On Packages

NOTE

You must install the base UNIX operating system (as described in the previous procedure) *before* you install the UNIX add-on packages.

Also, if any of the UNIX Add-On packages are currently installed, you must remove them before installing the new software. Refer to the section "Remove Add-On Software Packages" in the *AT&T UNIX System V/386 Release 3.2 System Administrator's Guide.*

Installation Time:	Approximately 45 minutes.
Installation Medium:	Cartridge tape labeled "UNIX System V/386, R3.2 V2.1."
Information you need:	*root* password.

33. Login as *root*.

34. Enter ***installpkg***.

35. When the following message appears:

```
Are you installing from tape (y/n)?
```

    Enter ***y***.

36. When the following message appears:

```
Insert Installation Tape in drive and press <RETURN>.
```

    (If you removed the tape after the last procedure, reinsert the tape into the tape drive.)

    Press **<Enter>**.

37. When the following message appears:

```
Do you want to install all of the above packages? <y/[n]>:
```

    Press **<Enter>** (to accept the default ***n***).

38. The system then prompts you to select the packages to install. Enter ***y*** after each of the following packages, and ***n*** after all the other packages (or press **<Enter>** to accept the default ***n***):

```
Editing Package Version 2.0
Extended Terminal Interface Package Version 2.0
C Software Development Set 4.1.6
Network Support Utilities Package (1.2) Version 2.0
2 Kilobyte File System Utility Package Version 2.0
Kernel Debugger(s) - Version 2.0
System Administration Software
```

A. When the Kernel Debugger package is being installed, the following appears:

```
Which kernel debugger(s) do you want to install?

   1) DEBUGGER (polish calculator style)
   2) GDEBUGGER (traditional)
   3) both DEBUGGER and GDEBUGGER

Choose 1, 2, or 3:
```

Enter *1*.

B. When the System Administration Software package is being installed, the following appears:

```
Do you want to give passwords to administration login?
(y/n) [n]
```

Press **<Enter>** (to accept the default *n*).

39. When the following message appears:

```
Confirm

To complete the install/removal process a shutdown is now
being initiated automatically.

Make sure your floppy drive is empty.  If you are installing
or removing controller boards, you may power down the system
after the shutdown has completed.

Strike ENTER when ready
or ESC to stop.
```

Press **<Enter>**.

40. When the following message appears:

```
Reboot the system now.
```

Press **<Ctrl-Alt-Del>** to reboot the system.

41. When the following message appears:

```
Console login:
```

The installation of the UNIX Add-on packages is complete.

42. Remove the tape from the tape drive (push tape in to release catch).

To complete the installation of the UNIX software, set up the user and system logins, as described in the next procedure.

Setting Up User and System Logins

NOTE

You must install the UNIX Add-on packages (as described in the previous procedure) *before* you set up the user and system logins.

Installation Time:	Approximately 10 minutes, depending on number of user logins being set up.
Installation Medium:	None.
Information you need:	Name of your time zone. Whether you observe daylight savings time. User login names, IDs, passwords, etc. Administrative passwords. System passwords. *node name* (name by which other machines know this machine).

---

43. Login as *setup*.

44. Answer the questions that **setup** asks.

NOTE

One of the questions asks you to select your time zone from a list. If your time zone is not included in this list, select GMT, and then enter the correct local time when a later question asks for the time.

To complete the installation of the UNIX software, install the remote terminal package, as described in the next procedure.

Installing the Remote Terminal Package

Installation Time:	Approximately 5 minutes.
Installation Medium:	Floppy diskette (1) labeled "Remote Terminal Package."
Information you need:	*root* password.

45. Login as *root*.
46. Enter ***installpkg***.
47. When the following message appears:

```
Are you installing from tape (y/n)?
```

    Enter ***n***.

48. Insert the installation diskette into the diskette drive, and turn the handle down.
49. When the following message appears:

```
Strike ENTER when ready
or ESC to stop.
```

    Press **<Enter>**.

50. When the following message appears:

```
Enter option:
```

    Enter ***1***.

51. When the following message appears:

```
Enter a file name, 'all', 'done', or 'files':
```

    Enter ***all***.

52. When the following message appears:

```
Enter a file name, 'all', 'done', or 'files':
```

Enter ***done***.

53. When the following message appears:

```
Enter option:
```

Enter ***0***.

54. Remove the diskette from the diskette drive.

55. If you are not planning to install the TCP/IP and Ethernet drivers (as described in the next section), press **<Ctrl-D>** to log out. If you are planning to continue the installation, enter:

```
shutdown -is
```

The following message is displayed:

```
Type Ctrl-d to proceed with normal startup
(or give root password for system maintenance):
```

Enter the root password.

NOTE

The TCP/IP and Ethernet Drivers packages are prerequisites for the iPSC System software.

56. The single-user shutdown does not automatically mount the user partition. The **installpkg** command resides in this partition. Mount the "usr" partition by entering the following command:

```
/etc/mount  /usr
```

This completes the installation of the UNIX software.

Installing TCP/IP and Ethernet Drivers

NOTE

The UNIX software must be installed *before* you can install the TCP/IP software. Refer to the discussion of "Installing UNIX System V/386 Release 3.2 Version 2.1" on page 2-13 of this release note.

Installation Time: Approximately 30 minutes.

Installation Media: Cartridge Tape labeled "Intel TCP/IP R3.0.1"

Information you need:
root password.
Internet address.
Network name.
Whether network is subnetted.
Domain name.
Broadcast Address

CAUTION

The *remove* and *install* scripts for TCP/IP remove/overwrite the following configuration files:

/etc/hosts
/etc/networks
/etc/gateways
/etc/hosts.equiv
/usr/lib/named/named.hosts
/usr/lib/named/named.local/usr/lib/named/named.rev
/usr/lib/named/named.soa
/usr/lib/named/root.cache

If you are updating a previous release of Intel TCP/IP, you may want to rename or backup these files *before* removing the previous release. After you successfully install TCP/IP Release 3.0.1, you can restore these files.

If you are continuing the installation process from a previous section, begin with step number 4.

The procedure for installing the TCP/IP packages is as follows:

1. Login as *root*.

   Make sure that the users are not logged in and the system is not NFS mounted.

2. Shutdown to single user mode by entering:

   ```
   shutdown -is
   ```

   The following message is displayed:

   ```
   Type Ctrl-d to proceed with normal startup
   (or give root password for system maintenance):
   ```

   Enter the root password.

3. The single-user shutdown does not automatically mount the user partition. The **installpkg** command resides in this partition. Mount the "usr" partition by entering the following command:

   ```
   /etc/mount /usr
   ```

4. Enter ***installpkg***.

5. When the following message appears:

   ```
   Are you installing from tape (y/n)?
   ```

   Enter *y*.

6. When the following message appears:

   ```
   Insert Installation Tape in drive and press <RETURN>.
   ```

   insert the tape labeled "Intel TCP/IP R3.0.1" in the tape drive.

   Press **<Enter>**.

7. When the following message appears:

   ```
   Do you want to install all of the above packages? <y/[n]>:
   ```

   Press *y* to install all packages.

8. When the following message appears:

```
Will you be installing the Ethernet Drivers? <y/[n]>:
```

   Press *y*.

At this point, you will be asked for you network configuration information.

9. When the following message appears:

```
Confirm

To complete the install/removal process a shutdown is now
being initiated automatically.

Make sure your floppy drive is empty.  If you are installing
or removing controller boards, you may power down the system
after the shutdown has completed.

Strike ENTER when ready
or ESC to stop.
```

   Press **<Esc>**.

   The installation is complete.

10. Remove the tape from the tape drive (push tape in to release catch).

NOTE

If you are installing NFS, that installation process should take place at any time after the TCP/IP and Ethernet Drivers software have been installed.

Installing the Release 3.3.1 System Software

NOTE

The install script modifies the three standard *crontab* files. If you want to keep your existing *crontab* files, save them before performing this procedure, and then restore them afterwards. The *crontab* files are stored in the directory */usr/spool/cron/crontabs*.

Installation Time:	Approximately 30 minutes.
Installation Medium:	1 cartridge tape labeled "iPSC®/2 System Software R3.3.1"
Information you need:	*root* password.

If you are continuing the installation process from a previous section, begin with step number 4.

1. Login as *root*.

2. Enter single user mode. This prevents other users from logging in.

   ```
   shutdown -is
   ```

   The following message is displayed:

   ```
   Type Ctrl-d to proceed with normal startup
   (or give root password for system maintenance):
   ```

   Enter the root password.

3. Mount the user partition. The **removepkg** command resides in this partition. The single-user shutdown does not automatically mount the user partition.

   ```
   /etc/mount  /usr
   ```

4. Enter ***installpkg***.

5. When the following message appears:

```
Are you installing from tape (y/n)?
```

Enter *y*.

6. When the following message appears:

```
Insert Installation Tape in drive and press <RETURN>.
```

Insert the installation tape labeled "iPSC®/2 System Software R3.3.1" into the tape drive (label up, exposed tape to the left; push until the tape cartridge locks into place).

7. Press **<Enter>**.

8. Eventually, the following message appears:

```
Do you want to install all of the above packages? <y/[n]>:
```

Enter *y*.

9. When the following message appears:

```
Strike ENTER when ready or
ESC to stop
```

Press **<Enter>**.

10. When the following message appears:

```
Reboot the system now.
```

A. Remove the tape from the tape drive (push the tape in to release the catch).

B. Press **<Ctrl-Alt-Del>** to reboot the system.

11. When the following message appears:

```
Console login:
```

log in as *root* again.

12. Enter the following command to remove permission for ordinary users to use the **rebootcube** command:

```
chmod 700 /usr/bin/rebootcube
```

This **chmod** command means that only the superuser can reboot the cube, making it less likely that someone will reboot the cube without understanding the consequences of their actions.

The installation of the Release 3.3.1 system software is now complete.

NOTE

If your hardware configuration includes an I/O subsystem, then create and run the following device configuration file:

```
bootcube)
cd /usr/ipsc/conf
cp devconf devconf.old
getcube -t0
../diag/mkdevconf
killcube
relcube
```

Installing the Release 3.3.1 Optional Software

Installation Time:	Approximately 30 minutes.
Installation Medium:	1 cartridge tape labeled "iPSC®/2/860 Optional Software R3.3.1"
Information you need:	*root* password.

---

1. Login as *root*.
2. Enter ***installpkg***.
3. When the following message appears:

   ```
   Are you installing from tape (y/n)?
   ```

   Enter ***y***.

4. When the following message appears:

```
Insert Installation Tape in drive and press <RETURN>.
```

   Insert the installation tape labeled "iPSC®/2/860 Optional Software R3.3.1" into the tape drive (label up, exposed tape to the left; push until the tape cartridge locks into place).

5. Press **<Enter>**.

6. Eventually, the following message appears:

```
Do you want to install all of the above packages? <y/[n]>:
```

   What you install depends on what you use. To save disk space, do not install the Remote Host, Simulator or NQS software on your system unless you need it. You can install this software later if desired.

   If you don't want to install all of the packages, press **<Enter>** (to accept the default ***n***). The system then prompts you to select the packages to install. Enter ***y*** for the packages you want and ***n*** for the ones you don't want.

When the prompt returns, the installation of the iPSC/2/860 Release 3.3.1 Optional software is complete.

Configuring NQS Software

NOTE

In order to make NQS usable, you need to compile and link NQS with Release 3.3.1 software, even if NQS is already built.

The NQS software is on the iPSC®/2/860 Optional Software R3.3.1 tape; you choose whether or not to install it when you install this tape. If you do install the NQS package, you need to configure it. This section describes a procedure that configures the NQS software for use with a basic iPSC system. The procedure describes the configuration steps necessary at the SRM.

After the procedures in this section are complete, the SRM will be configured as machine number one (mid=1) in a "locally functional" NQS environment with a single batch queue named `bq`. The locally functional NQS environment will be able to accept batch jobs submitted from the SRM, but it will not yet be able to accept jobs submitted from other machines. In order to use NQS with other machines or in more complex systems, you must perform the more detailed configuration procedures described in the *iPSC®/2 and iPSC®/860 Network Queueing System Manual*. Refer to Appendix A, "Reconfiguring NQS" in the *iPSC®/2 and iPSC®/860 Network Queueing System Manual* for more information.

Installation Time:	Approximately 30 minutes.
Installation Medium:	None.
Information you need:	Root password.

NOTE

All pathnames must be absolute pathnames (i.e., they must start with `/`).

Perform the following steps at the iPSC SRM:

1. Login as *root* on the SRM.
2. If you already have NQS running from a previous release, then kill the following processes:

   nqsdaemon
   netdaemon
   logdaemon

3. Change directories to the NQS source directory, */usr/src/nqs*, as follows:

   ```
   cd /usr/src/nqs
   ```

4. Make sure that your umask is set to 0 by entering the following:

   ```
   umask 0
   ```

5. Compile NQS code as follows:

   ```
   make
   ```

6. Install the NQS executable code by entering the following:

   ```
   make install
   ```

   If NQS is already built, a number of **mkdir** error messages are generated. You may safely ignore these error messages.

Before the NQS daemon can be started, the host machine must be in the NQS database. Once the daemon is running, the **qmgr** command is functional and you can use the **qmgr** command to create a queue for job requests. There is an entry point in the *make* file that will allow you to do this automatically. The automatic method will accomplish the following tasks:

1. Insert the SRM machine name as machine #1.
2. Start the NQS daemon.
3. Create a single batch queue by the name of `bq`.
4. Make the `bq` batch queue the default queue for the **qsub** command.

If you want your machine to accept requests from other machines on the network or accomplish other tasks, you must set up the machine IDs and the queues by hand. Refer to Appendix A, "Reconfiguring NQS," of the *iPSC®/2 and iPSC®/860 Network Queueing System Manual* for instructions on setting up the machine IDs and queues by hand.

NOTE

If you already have NQS configured from a previous release, then start NQS with this command:

```
/usr/lib/nqs/nqsdaemon > /usr/lib/nqs/logfile
```

NQS configuration is now complete. Continue the installation process by proceeding to "Installing Online Documentation in a Remote Host Environment" on page 2-34.

If NQS is not configured from a previous release, then perform the following steps to start NQS using the automatic method:

1. Login as *root* on the SRM
2. Change directories to the source directory, */usr/src/nqs* as follows:

```
cd /usr/src/nqs
```

3. Make sure that your umask is set to 0 by entering the following:

```
umask 0
```

4. Start the make file at the start-up tag as follows:

```
make startup
```

You are now able to submit local jobs to the batch queue bq by using the **qsub** command. In order to submit jobs from remote locations in the network, you must set each remote machine up using the detailed procedures presented in Appendix A, "Reconfiguring NQS," of the *iPSC®/2 and iPSC®/860 Network Queueing System Manual.*

NOTE

In order to completely remove the NQS software package and all of its daemons and configuration files, you need to do a ***make remove*** under the */usr/src/nqs* directory prior to using the **removepkg** command.

Installing Online Documentation in a Remote Host Environment

These are the basic instructions for installing the on-line documentation in a remote host environment on an NFS network. This should be done after all the optional iPSC software packages have been installed on the SRM. If you are doing a complete installation process (not an update) the NFS package should be installed before installing the Online Documentation in a Remote Host Environment.

Installation Time:	Approximately 25 minutes.
Installation Medium:	None.
Information you need:	Root password for the file server.

---

1. Have the system administrator of the network create an SSD tools root directory. There needs to be about 40 megabytes of disk space on the NFS file server that has this directory. The directory should be owned by *root* and writable only by *root* but readable and executable by all.

   For example, if the SSD tools root directory is to be */vol/tools/ssd*, perform the following commands as *root* on the file server:

```
mkdir /vol/tools/ssd/man
chmod 755 /vol/tools/ssd/man
```

2. Set the environment variable **IPSC_XDEV** to the full pathname of the SSD tools root directory. For example:

```
setenv IPSC_XDEV /vol/tools/ssd
```

3. Copy all the files and directories from the directory */usr/ipsc/XDEV/i860/man* on the SRM to the SSD tools root directory on the file server.

   For example, using **rcp** from a Sun network:

```
rcp -p -r SRM:/usr/ipsc/XDEV/i860/man $IPSC_XDEV/man
```

   Or using **tar** from a Sun network:

```
cd $IPSC_XDEV/man
rsh SRM "cd /usr/ipsc/XDEV/i860/man;tar cf - ."|tar xf -
```

   In both of the above commands, *SRM* represents the network name of the SRM.

The installation of the Online Documentation in the cross-development environment is now complete.

Installing Remote Host Software on a Diskful Host

NOTE

The remote host software package must be installed on the SRM *before* you can install the remote host software on a remote host. Refer to the section "Installing the Release 3.3.1 Optional Software" on page 2-30 of this manual.

Use the following procedure to install the remote host software on a remote host that has its own disk. If installing on a diskless host, see "Installing Remote Host Software on a Diskless Host" on page 2-38.

Installation Time:	Approximately 30 minutes.
Installation Medium:	None.
Information you need:	*root* password.

1. Login as *root* on the remote host.
2. Create a directory on the remote host to contain the remote host source code. This directory can be located wherever you choose.
3. Copy the remote host source code from the directory */usr/ipsc/rhost* on the SRM to the directory you created in the previous step. For example, this can be accomplished with the following command line:

   ```
   rcp -p -r SRMname:/usr/ipsc/rhost newdir
   ```

   where:

   SRMname is the internal name of the SRM.
   newdir is the name of the directory created in Step 2 of this procedure.
4. On the remote host, edit the makefile in the top-level remote host source code directory so that the installation directories are correct for your system:

NOTE

All pathnames must be absolute pathnames (i.e., they must start with `/`).

IPSCDIR Contains daemons and named sockets. The default is */usr/ipsc* and the subdirectories *lib* and *log* will be created.

BINDIR Contains cube binaries. On the SRM these files are in */usr/bin*. Any directory can be used for the binaries, as long as the cube users have it in their search path. The default directory on the remote host is *IPSCDIR/bin*.

LIBDIR Contains *libhost.a*. The default location for the library is in */usr/lib*. Again, any directory can be used as long as the user application makefiles reflect the correct directory.

INCDIR Contains the iPSC/2 include files *cube.h* and *fcube.h*. The default is *IPSCDIR/include*.

5. Kill any existing *commser* and *fserver* daemons on the remote host.

6. Enter ***make*** to build binaries on the remote host.

NOTE

On some systems, umask is declared in a standard header file, and this step will fail during serve_io.c compilation. If this happens, edit *./fileio/serve_io.c* and comment out (using C-style comments) the umask declaration. Then, invoke the **make** command again.

7. Enter ***make install*** to install the binaries in the prescribed directories.

8. Edit the *srms* file (*IPSCDIR/lib/srms*) to include the names of all SRMs that the remote host may access.

9. Edit the */etc/hosts* file on the SRM to include the names of all the remote workstations that may access the SRM.

10. Edit the file */usr/ipsc/lib/ruser* on the SRM to include the host and user names for all hosts and users that may access the SRM using remote host software. (The format of this file is the same as that of */etc/hosts.equiv*).

11. Edit the user's *.cshrc* file to include the following line:

```
setenv TTY 'tty' >& /dev/null
```

This completes the instllatiobn of the remote host software on a remote host.

Installing Remote Host Software on a Diskless Host

NOTE

The remote host software package must be installed on the SRM *before* you can install the remote host software on a remote host. Refer to the section "Installing the Release 3.3.1 Optional Software" on page 2-30 of this manual.

Use the following procedure to install the remote host software on a remote host that does not have its own disk. If installing on a diskful host, see "Installing Remote Host Software on a Diskful Host" on page 2-35.

Installation Time: Approximately 30 minutes.

Installation Medium: None.

Information you need: *root* password.

NFS security features conflict with the installation procedure for remote host software described in the previous section. NFS does not allow *root* access to remotely-mounted filesystems. The remote host software requires the builder and installer to have *root* access to the filesystems on which it is installed. Since a diskless client cannot have *root* access, a different installation procedure must be followed.

NOTE

Some NFS mounted file systems also do not allow **setuid** programs to be installed on them. The remote host software executables *must* be installed on a mounted file system that allows a **setuid** program to execute.

In the examples in this procedure, the NFS file server (called server) has a directory called */export/sun386/local* which its diskless clients (one is called **client**) mount as */usr/local*. The remote host software for the clients is to be installed in a directory called */usr/local/ipsc* from the client's point of view. This path will be compiled into several of the remote host programs.

1. Become *root* on the NFS file server and cd to the parent directory that will contain the subdirectory where the remote host software is to be installed. For example:

```
{server:49} cd /export/sun386/local
```

2. Create the directory to contain the remote host source code, and set its protection so anyone can write to this directory: For example:

```
{server:50} mkdir ipsc
{server:51} chmod a+w ipsc
```

3. Become *root* on the client machine and cd to the directory you created in the previous step. For example:

```
{client:17} cd /usr/local/ipsc
```

4. Copy the remote host source code from the directory */usr/ipsc/rhost* on the SRM to the current directory.

5. Edit the makefile in the current directory so that the installation directories are correct for your system:

NOTE

All pathnames must be absolute pathnames (i.e., they must start with /) and must be from the client's view of the file system (i.e., */usr/local/ipsc* instead of */export/sun386/local/ipsc*).

IPSCDIR Contains daemons and named sockets. The default is */usr/ipsc* and the subdirectories *lib* and *log* will be created.

BINDIR Contains cube binaries. On the SRM these files are in */usr/bin*. Any directory can be used for the binaries, as long as the cube users have it in their search path. The default directory on the remote host is *IPSCDIR/bin*.

LIBDIR Contains *libhost.a*. The default location for the library is in */usr/lib*. Again, any directory can be used as long as the user application makefiles reflect the correct directory.

INCDIR Contains the iPSC/2 include files *cube.h* and *fcube.h*. The default is *IPSCDIR/include*.

6. Kill any existing *commser* and *fserver* daemons on the remote host.

7. Type ***make***. Some directories will be created in */usr/local/ipsc* and some header files installed. The remote host software libraries and executables will be created.

NOTE

On some systems, umask is declared in a standard header file, and this step will fail during serve_io.c compilation. If this happens, edit *./fileio/serve_io.c* and comment out (using C-style comments) the umask declaration. Then, invoke the **make** command again.

8. When the **make** is completed, become *root* on the NFS file server again and remove the world write permissions from the directory created in step 2. For example:

```
{server:52} cd /export/sun386/local
{server:53} chmod o-w ipsc
```

9. **cd** to the directory containing the remote host source. For example:

```
{server:54} cd /export/sun386/local/ipsc
```

10. Edit the **makefile** and set **IPSCDIR** correctly for the server's view of the filesystem (in this example, this would be */usr/sun386/local/ipsc*).

11. Install the software by typing ***make install.***

12. Edit the *srms* file (*IPSCDIR/lib/srms*) to include the names of all SRMs that the remote host may access.

13. Edit the */etc/hosts* file on the SRM to include the names of all the remote workstations that may access the SRM.

14. Edit the file */usr/ipsc/lib/ruser* on the SRM to include the host and user names for all hosts and users that may access the SRM using remote host software. (The format of this file is the same as that of */etc/hosts.equiv*).

15. Edit the user's *.cshrc* file to include the following line:

```
setenv TTY `tty` >& /dev/null
```

This completes the installation of the remote host software on a remote host.

iPSC®/860 Software Installation 3

Introduction

This chapter describes how to install the Release 3.3.1 software. The R3.3.1 system software runs on iPSC®/2 (with Intel386™-microprocessor-based CX nodes) and iPSC®/860 systems (with i860™-microprocessor-based RX nodes or with a combination of CX and RX nodes). To install release 3.3.1 software on an iPSC®/860 system, follow the steps outlined below.

Installing Release 3.3.1 Software on an iPSC®/860 System

CAUTION

If your software release is 3.2 or earlier and if you use CIO Ethernet, *do not* install this Release 3.3.1 software on your system until you have the updated version of CIO Ethernet. Earlier versions of this software package are not compatible with Release 3.3.1 of the system software.

The correct version of this package for use with Release 3.3.1 is as follows:

- CIO Ethernet Release 1.1 and the diskette labeled *iPSC/860 CIO Ethernet Update.*

If you have any questions, please contact SSD Customer Support:

1-800-421-2823 (Customer Support Response Center)
Your Local Intel Sales Office (in Europe)
support@ssd.intel.com (Internet address)

There are three reasons to use the procedures in this chapter:

- You are upgrading your system from Release 3.3 to Release 3.3.1. In this case, begin with "Installation Instructions for Update Customers" on page 3-4. You *do not* have to reinstall the UNIX operating system.
- You are upgrading your system from a release earlier than Release 3.3. In this case, you must reinstall all system software from scratch. Begin with "Installing UNIX System V/386 Release 3.2 Version 2.1" on page 3-18.
- You are reinstalling your Release 3.3.1 system software because it has become damaged. In this case, begin with "Installing UNIX System V/386 Release 3.2 Version 2.1" on page 3-18.

If the software on your system becomes damaged, please contact SSD Customer Support before performing any installations.

NOTE

Before performing any procedure in this chapter, read through the procedure completely to make certain that you understand what the procedure involves. If you have any questions before or during any procedure, please call SSD Customer Support for help.

If you perform the procedures in this chapter exactly as described, you will create a software system that is identical to the one originally installed by SSD.

The procedures in this chapter use the conventions described in the preface. You should also be aware of the following conventions used in these procedures:

- The instruction "Enter ***character(s)***" means type the indicated character(s), and then press the **<Enter>** key. For example, "Enter *y*" means type the letter `y`, and then press the key labeled `Enter`.
- In prompts, square brackets surround a default value. Pressing **<Enter>** selects the indicated default value.
- Some steps in these procedures cause a lot of information to be displayed on the system monitor. However, the step usually shows only the last message displayed. Do not be concerned if the indicated message does not appear immediately. Be patient. Some steps take several minutes to complete.

Software Dependencies

Table 3-1 describes the dependencies between the various software packages in this release.

Table 3-1. Software Dependencies for Release 3.3.1 Installation

If you want to install this:	You must install this first:
TCP - Intel Corp. Release 3.0.1	UNIX System V Release 3.2 Cartridge Tape Utilities Version 2.1
Ethernet Drivers Package for Intel TCP/IP Version 3.0.1	UNIX System V Release 3.2 TCP - Intel Corp. Release 3.0.1
iPSC System Software Release 3.3.1	UNIX System V Release 3.2 Cartridge Tape Utilities Version 2.1 TCP Intel Corp. Release 3.0 Ethernet Drivers Package for Intel TCP/IP Version 3.0
iPSC Compiler Libraries System i860 Release 3.3.1	UNIX System V Release 3.2 Cartridge Tape Utilities Version 2.1
iPSC Performance Analysis Tools i860 Release 1.1	iPSC Compiler Libraries System i860 UNIX System V Release 3.2 Cartridge Tape Utilities Version 2.1
iPSC i860 Extension Software Release 3.3.1	iPSC System Software Release 3.3.1
iPSC C Compiler i860 Release 2.0	iPSC Compiler Libraries System i860 UNIX System V Release 3.2 Cartridge Tape Utilities Version 2.1
iPSC FORTRAN Compiler i860 Release 2.0	iPSC Compiler Libraries System i860 UNIX System V Release 3.2 Cartridge Tape Utilities Version 2.1
iPSC Remote Host Software Release 3.3.1	UNIX System V Release 3.2 Cartridge Tape Utilities Version 2.1
iPSC Simulator Software Release 3.0	UNIX System V Release 3.2 Cartridge Tape Utilities Version 2.1
iPSC NQS software Release 3.0	UNIX System V Release 3.2 Cartridge Tape Utilities Version 2.1

Table 3-1 describes only the dependencies to *install* the named package. Further dependencies may exist in order to *use* the software in the package.

Disk Space Requirements

Table 3-2 lists the disk space required by each iPSC software package in this release.

Table 3-2. Disk Space Requirements for Release 3.3.1

Package	Size	Difference from 3.3
TCP - Intel Corp. Release 3.0.1	3.0M bytes	0
Ethernet Drivers Package for Intel TCP/IP Version 3.0.1	1.1M bytes	0
iPSC System Software Release 3.3.1	22.2M bytes	+ 2.9M bytes
iPSC Compiler Libraries System i860 Release 3.3.1	3.8M bytes	0-.8M bytes
iPSC Remote Host Software Release 3.3.1	1.3M bytes	+ 0.4M bytes
iPSC Simulator Software Release 3.0	0.7M bytes	0 [1]
iPSC NQS Software Release 1.0	2.6M bytes	0 [1]
iPSC Performance Analysis Tools i860 Release 1.1	6.8M bytes	+ 6.8M bytes
iPSC i860 Extension Software Release 3.3.1	7.9M bytes	– 4.3M bytes
iPSC C Compiler i860 Release 2.0	7.2M bytes	0M bytes
iPSC FORTRAN Compiler i860 Release 2.0	7.9M bytes	0M bytes
Both i860 compilers together	11.2M bytes	0M bytes
1. This package has not changed from Release 3.3.		

To determine the free disk space you will need to upgrade your system from Release 3.3 to Release 3.3.1, add the numbers in the "Difference from 3.3" column for the packages you are installing.

Installation Instructions for Update Customers

The following instructions must be followed if you have Update 9/16 or 10/16 installed on your iPSC®/860 system. Please follow the instructions, in sequence, in order to successfully install release 3.3.1 software.

Removing Update 9/16 or 10/16 and Release 3.3 Software

Removal Time: Approximately 30 minutes.

Information you need: Which update is installed on your system.

---

1. Login as *root*.

   Make sure that the users are not logged in and the system is not NFS mounted.

2. Shutdown the cube by entering:

   ```
   shutdown -D1
   ```

3. Shutdown to single user mode by entering:

   ```
   shutdown -is
   ```

   The following message is displayed:

   ```
   Type Ctrl-d to proceed with normal startup
   (or give root password for system maintenance):
   ```

   Enter the root password.

4. The single-user shutdown does not automatically mount the user partition. The **removepkg** command resides in this partition. Mount the "usr" partition by entering the following command:

   ```
   /etc/mount  /usr
   ```

5. Enter ***displaypkg***.

   This will identify which software update package is installed on the system. If you have release 3.3 installed, you will have either Update 9/16 or Update 10/16 installed. If you have Update 9/16 installed, the following message will appear:

   ```
   iPSC E1 i860 Extension Software Update Release 3.3
   iPSC S1 System Software Update Release 3.3
   ```

If you have Update 10/16 installed, the following messages will appear:

```
iPSC E2 i860 Extension Software Update Release 3.3
iPSC S2 System Software Update Release 3.3
iPSC R2 Remote Host Software Update Release 3.3
iPSC T2 TCP - Intel Corp. Update Release 3.0
iPSC L2 Compiler Libraries System Update Release 3.3
```

The Remote Host software is an optional software package and may not be installed on your system.

Removing Update 9/16

6. If you have Update 9/16 installed, you need to remove the software using the **removepkg** command . When you enter the **removepkg** command, the system displays all software packages installed on the system.

   Enter the number corresponding to the `iPSC S1 System Software Update Release 3.3` software package.

   When the following message appears:

```
Confirm

Do you really want to remove iPSC S1 System Software Update
Release 3.3?

Strike ENTER when ready
or ESC to stop.
```

   Press **<Enter>**.

   When the following message appears:

```
Confirm

To complete the install/removal process a shutdown is now
being initiated automatically.

Make sure your floppy drive is empty.  If you are installing
or removing controller boards, you may power down the system
after the shutdown has completed.

Strike ENTER when ready
or ESC to stop.
```

   Press **<Esc>**.

NOTE

If you press any key other than **<Esc>**, the system will shutdown. If the system does shutdown, you must first reboot the system, shutdown into single user mode, and mount the usr partition to continue the software removal process. This process is explained in the beginning of this section on page 3-4. You can then continue with the removal process with the step after the last one you completed.

The software for this step has now been successfully removed.

7. At the # prompt, enter the **removepkg** command.

   Enter the number corresponding to the `iPSC E1 i860 Extension Software Update Release 3.3` software package.

   When the following message appears:

```
Confirm

Do you really want to remove iPSC E1 i860 Extension Software
Update Release 3.3?

Strike ENTER when ready
or ESC to stop.
```

   Press **<Enter>**.

   When the following message appears:

```
Confirm

To complete the install/removal process a shutdown is now
being initiated automatically.

Make sure your floppy drive is empty.  If you are installing
or removing controller boards, you may power down the system
after the shutdown has completed.

Strike ENTER when ready
or ESC to stop.
```

   Press **<Esc>**.

NOTE

Now, to continue the software removal process, please go to the section "Removing iPSC Software Release 3.3" on page 3-12.

Removing Update 10/16

8. If you have Update 10/16 installed, you need to remove the software using the **removepkg** command. When you enter the **removepkg** command, the system displays all software packages installed on the system.

   Enter the number corresponding to the `iPSC S2 System Software Update Release 3.3` software package.

   When the following message appears:

```
Confirm

Do you really want to remove iPSC S2 System Software Update
Release 3.3?

Strike ENTER when ready
or ESC to stop.
```

   Press **<Enter>**.

   When the following message appears:

```
Confirm

To complete the install/removal process a shutdown is now
being initiated automatically.

Make sure your floppy drive is empty.  If you are installing
or removing controller boards, you may power down the system
after the shutdown has completed.

Strike ENTER when ready
or ESC to stop.
```

   Press **<Esc>**.

NOTE

If you press any key other than **<Esc>**, the system will shutdown. If the system does shutdown, you must first reboot the system, shutdown into single user mode, and mount the usr partition to continue the software removal process. This process is explained in the beginning of this section on page 3-4. You can then continue with the removal process with the step after the last one you completed.

The software for this step has now been successfully removed.

9. At the # prompt, enter the **removepkg** command.

Enter the number corresponding to the `iPSC E2 i860 Extension Software Update Release 3.3` software package.

When the following message appears:

```
Confirm

Do you really want to remove iPSC E2 i860 Extension Software
Update Release 3.3?

Strike ENTER when ready
or ESC to stop.
```

Press **<Enter>**.

When the following message appears:

```
Confirm

To complete the install/removal process a shutdown is now
being initiated automatically.

Make sure your floppy drive is empty.  If you are installing
or removing controller boards, you may power down the system
after the shutdown has completed.

Strike ENTER when ready
or ESC to stop.
```

Press **<Esc>**.

10. If you do not have the Remote Host Update package installed on your system, skip to instruction number 11. At the # prompt, enter the **removepkg** command.

    Enter the number corresponding to the `iPSC R2 Remote Host Software Update Release 3.3` software package.

    When the following message appears:

    ```
    Confirm

    Do you really want to remove iPSC R2 Remote Host Software
    Update Release 3.3?

    Strike ENTER when ready
    or ESC to stop.
    ```

    Press **<Enter>**.

    When the following message appears:

    ```
    Confirm

    To complete the install/removal process a shutdown is now
    being initiated automatically.

    Make sure your floppy drive is empty.  If you are installing
    or removing controller boards, you may power down the system
    after the shutdown has completed.

    Strike ENTER when ready
    or ESC to stop.
    ```

    Press **<Esc>**.

11. At the # prompt, enter the **removepkg** command.

    Enter the number corresponding to the `iPSC L2 Compiler Libraries System Update Release 3.3` software package.

    When the following message appears:

    ```
    Confirm

    Do you really want to remove iPSC L2 Compiler Libraries System
    Update Release 3.3?
    ```

```
Strike ENTER when ready
or ESC to stop.
```

Press **<Enter>**.

When the following message appears:

```
Confirm

To complete the install/removal process a shutdown is now
being initiated automatically.

Make sure your floppy drive is empty.  If you are installing
or removing controller boards, you may power down the system
after the shutdown has completed.

Strike ENTER when ready
or ESC to stop.
```

Press **<Esc>**.

12. At the # prompt, enter the **removepkg** command.

Enter the number corresponding to the `iPSC T2 TCP - Intel Corp. Update Release 3.0` software package.

When the following message appears:

```
Confirm

Do you really want to remove iPSC T2 TCP - Intel Corp. Update
Release 3.0?

Strike ENTER when ready
or ESC to stop.
```

Press **<Enter>**.

When the following message appears:

```
Confirm

To complete the install/removal process a shutdown is now
being initiated automatically.

Make sure your floppy drive is empty.  If you are installing
or removing controller boards, you may power down the system
after the shutdown has completed.
```

```
Strike ENTER when ready
or ESC to stop.
```

Press **<Esc>**.

Removing iPSC Software Release 3.3

NOTE

Do not remove the following software packages: (you may or may not have them installed)

```
iPSC NQS Software Release 1.0
iPSC Simulator Software Release 3.0
```

13. Remove the following software packages by entering the **removepkg** command.

Enter the number corresponding to the `iPSC System Software Release 3.3` software package.

When the following message appears:

```
Confirm

Do you really want to remove iPSC System Software Release 3.3?

Strike ENTER when ready
or ESC to stop.
```

Press **<Enter>**.

When the following message appears:

```
Confirm

To complete the install/removal process a shutdown is now
being initiated automatically.

Make sure your floppy drive is empty.  If you are installing
or removing controller boards, you may power down the system
after the shutdown has completed.

Strike ENTER when ready
or ESC to stop.
```

Press **<Esc>**.

14. Enter the **removepkg** command.

    Enter the number corresponding to the `iPSC i860 Extension Software Release 3.3` software package.

    When the following message appears:

```
Confirm

Do you really want to remove iPSC i860 Extension Software
Release 3.3?

Strike ENTER when ready
or ESC to stop.
```

    Press **<Enter>**.

    When the following message appears:

```
Confirm

To complete the install/removal process a shutdown is now
being initiated automatically.

Make sure your floppy drive is empty.  If you are installing
or removing controller boards, you may power down the system
after the shutdown has completed.

Strike ENTER when ready
or ESC to stop.
```

    Press **<Esc>**.

15. If you do not have the Remote Host software package installed on your system, go to instruction number 16, otherwise enter the **removepkg** command.

    Enter the number corresponding to the `iPSC Remote Host Software Release 3.3` software package if it is installed on your system.

    When the following message appears:

```
Confirm

Do you really want to remove iPSC Remote Host Software Release
3.3?

Strike ENTER when ready
or ESC to stop.
```

Press **<Enter>**.

When the following message appears:

```
Confirm

To complete the install/removal process a shutdown is now
being initiated automatically.

Make sure your floppy drive is empty.  If you are installing
or removing controller boards, you may power down the system
after the shutdown has completed.

Strike ENTER when ready
or ESC to stop.
```

Press **<Esc>**.

16. Enter the **removepkg** command.

Enter the number corresponding to the `iPSC Compiler Libraries System i860 Release 3.3` software package.

When the following message appears:

```
Confirm

Do you really want to remove iPSC Compiler Libraries System
i860 Release 3.3?

Strike ENTER when ready
or ESC to stop.
```

Press **<Enter>**.

When the following message appears:

```
Confirm

To complete the install/removal process a shutdown is now
being initiated automatically.

Make sure your floppy drive is empty.  If you are installing
or removing controller boards, you may power down the system
after the shutdown has completed.

Strike ENTER when ready
or ESC to stop.
```

Press **<Esc>**.

17. If you do not have the Performance Analysis Tools software package installed, skip to instruction number 18. Otherwise, enter the **removepkg** command.

    Enter the number corresponding to the `iPSC Performance Analysis Tools i860 Release 1.0` software package if this package is installed on your system. Verify your choice and press **<Enter>** when prompted.

    When the following message appears:

    ```
    Confirm

    Do you really want to remove iPSC Performance Analysis Tools
    i860 Release 1.0?

    Strike ENTER when ready
    or ESC to stop.
    ```

    Press **<Enter>**.

    When the following message appears:

    ```
    Confirm

    To complete the install/removal process a shutdown is now
    being initiated automatically.

    Make sure your floppy drive is empty.  If you are installing
    or removing controller boards, you may power down the system
    after the shutdown has completed.

    Strike ENTER when ready
    or ESC to stop.
    ```

    Press **<Esc>**.

18. Enter the **removepkg** command.

    Enter the number corresponding to the `Ethernet Drivers Package for Intel TCP/IP Version 3.0` software package.

    When the following message appears:

    ```
    Confirm

    Do you really want to remove the Ethernet Drivers Package for
    Intel TCP/IP Version 3.0?
    ```

```
Strike ENTER when ready
or ESC to stop.
```

Press **<Enter>**.

When the following message appears:

```
Confirm

To complete the install/removal process a shutdown is now
being initiated automatically.

Make sure your floppy drive is empty.  If you are installing
or removing controller boards, you may power down the system
after the shutdown has completed.

Strike ENTER when ready
or ESC to stop.
```

Press **<Esc>**.

CAUTION

The *remove* and *install* scripts for TCP/IP remove/overwrite the following configuration files:

/etc/hosts
/etc/networks
/etc/gateways
/etc/hosts.equiv
/usr/lib/named/named.hosts
/usr/lib/named/named.local
/usr/lib/named/named.rev
/usr/lib/named/named.soa
/usr/lib/named/root.cache

If you are updating a previous release of Intel TCP/IP, you may want to rename or backup these files *before* removing the previous release. After you successfully install TCP/IP Release 3.0.1, you can restore these files.

19. Enter the **removepkg** command:

Enter the number corresponding to the `Tcp - Intel Corp. Release 3.0` software package.

When the following message appears:

```
Confirm

Do you really want to remove the TCP - Intel Corp. Release 3.0?

Strike ENTER when ready
or ESC to stop.
```

Press **<Enter>**.

When the following message appears:

```
Confirm

To complete the install/removal process a shutdown is now
being initiated automatically.

Make sure your floppy drive is empty.  If you are installing
or removing controller boards, you may power down the system
after the shutdown has completed.

Strike ENTER when ready
or ESC to stop.
```

Press **<Esc>**.

20. Enter the **removepkg** command:

Enter the number corresponding to the `PC586 Ethernet Driver - Version 1.0` software package.

```
Confirm

Do you really want to remove PC586 Ethernet Driver - Version
1.0?

Strike ENTER when ready
or ESC to stop.
```

Press **<Enter>**.

When the following message appears:

```
Confirm

To complete the install/removal process a shutdown is now
being initiated automatically.

Make sure your floppy drive is empty.  If you are installing
or removing controller boards, you may power down the system
after the shutdown has completed.

Strike ENTER when ready
or ESC to stop.
```

Enter **<Esc>**.

NOTE

Go to the paragraph labeled "Installing TCP/IP and Ethernet Drivers" on page 3-30 to continue the installation process

Complete Installation Process (not an update)

Installing UNIX System V/386 Release 3.2 Version 2.1

Use these installation procedures to perform a new and/or complete installation.

CAUTION

Installing the UNIX software reformats the hard disk. To prevent loss of data, save all user files and any system files that have been modified *before* installing the UNIX software.

If you are upgrading from UNIX Release 3.0, *do not replace the new system files with the modified system files* that you saved. Doing so downgrades those system files to UNIX Release 3.0. Instead, after installing the new UNIX software, make the same modifications to the new system files.

There are six major steps to installing the UNIX software:

1. Install the root and user filesystems
2. Install the base UNIX operating system
3. Install the Cartridge Tape Utilities
4. Install the UNIX add-on packages
5. Set up user and system logins
6. Install the remote terminal package

To install the UNIX software, you must perform all parts in the order specified.

Installing the Root and User Filesystems

Installation Time:	Approximately 30 minutes.
Installation Medium:	Floppy diskette (1) labeled "Boot Floppy."
Information you need:	Size of your hard disk (140MB or 380MB).

---

1. If the system is on, turn it off as described in Chapter 2 in the *iPSC®/2 and iPSC®/860 System Administrator's Guide*, in the sections "Stopping the UNIX Operating System" and "Removing Power."
2. Insert the installation diskette into the diskette drive.
3. Turn the system on.
4. When the following message appears:

   ```
   Strike ENTER to install the UNIX System on your hard disk.
   ```

   Press **<Enter>**.

5. When the following message appears:

```
WARNING: A new installation of the UNIX System will destroy
all files currently on the system.  Do you wish to continue
(y or n)?
```

Enter *y*.

6. When the following message appears:

```
SELECT ONE OF THE FOLLOWING:

   1.  Create a partition
   2.  Change Active (Boot from) partition
   3.  Delete a partition
   4.  Exit (Update disk configuration and exit)
   5.  Cancel (Exit without updating disk configuration)
Enter Selection:
```

Enter *5* if the partitioning is 100% UNIX, Active, with no other partitions. (The partitioning was probably done when your system was originally installed.)

If the partitioning is not 100% UNIX, Active, with no other partitions,then:

Enter *3* to delete the default partition

Enter *1* to create the desired partition.

7. When the following message appears:

```
A surface analysis will now be done.
This will destroy all data on the hard disk.
Strike ENTER to continue or DEL to abort.
```

Press **<Enter>**.

8. Following a message describing a suggested partitioning, this message appears:

```
Is this allocation acceptable to you (y/n)?
```

Before responding to this message, make a note of the suggested number of cylinders for the root file system, and for the swap/paging area. You will use these to answer the questions in steps 11 and 12 below.

Then, to respond to the allocation question, enter *n*. This keeps the partitioning that was originally sent to you.

9. When the following message appears:

```
Do you wish to have separate root and usr filesystems (y/n)?
```

Enter *y*.

10. When the following message appears:

```
Do you want an additional /usr2 filesystem (y/n)?
```

Enter *n*.

11. When the following message appears:

```
How many cylinders would you like for swap/paging (1-xxx)?
```

Enter the number of swap/paging cylinders you noted in step 8 above. Typical numbers are *109* for a 140M-byte drive, and *78* for a 380M-byte drive.

12. When the following message appears:

```
How many cylinders would you like for root (1-xxx)?
```

Enter the number of root cylinders you noted in step 8 above. Typical numbers are *201* for a 140M-byte drive, and *143* for a 380M-byte drive.

13. When the following message appears:

```
Is this allocation acceptable to you (y/n)?
```

Enter *y* if the cylinder quantities for root and swap/paging are what you entered.

14. When the following message appears:

```
Reboot the system now.
```

The installation of the root and user filesystems is complete.

15. Remove the diskette from the diskette drive.

To complete the installation of the UNIX software, install the base UNIX operating system, as described in the next procedure.

Installing the Base UNIX Operating System

NOTE

You must install the root and user filesystems (as described in the previous procedure) *before* you install the base UNIX operating system.

Installation Time:	Approximately 20 minutes.
Installation Medium:	Cartridge tape labeled "UNIX System V/386, R3.2 V2.1". Diskette labeled "Cartridge Tape Utilities"
Information you need:	*root* password. *install* password.

---

16. Press **<Ctrl-Alt-Del>** to reboot the system.

17. When the following message appears:

```
Installation from Cartridge Tape is available using Interrupt
#5 and Address Range 300 through 301.
Are you installing from tape (y/n)?
```

    Enter *y*.

18. When the following message appears:

```
Please make sure your Cartridge Tape hardware is configured
correctly. Insert System Installation Tape in drive and press
<RETURN>
```

    A. Insert the installation tape into the tape drive (label up, exposed tape to the left; push until the tape cartridge locks into place).

    B. Press **<Enter>**.

19. When the system prompts for a *root* password, enter the password that you have chosen for the *root* login.

20. When the system prompts for an *install* password, enter the password that you have chosen for the *install* login.

21. When the following message appears:

```
Confirm

Do you want to install the Cartridge Tape Drive?
Press ENTER for YES or ESC for NO

Strike ENTER when ready
or ESC to stop.
```

    Press **<Esc>**.

22. When the following message appears:

```
Reboot the system now
```

    Press **<Ctrl-Alt-Del>** to reboot the system.

23. When the following message appears:

```
Console login:
```

24. Login as *root*.

25. Enter ***installpkg***.

26. Insert the Cartridge Tape Utilities installation diskette into the diskette drive and turn the handle down.

27. When the following message appears:

```
Strike ENTER when ready
or ESC to stop
```

    Press **<Enter>**.

28. When the following message appears:

```
Type the interrupt number and strike the ENTER key or
type Q to cancel installation.
```

    Enter *5*.

29. When the following message appears:

```
Strike ENTER when ready
or ESC to stop
```

Press **<Enter>**.

30. Again, when the following message appears:

```
Strike ENTER when ready
or ESC to stop
```

Press **<Enter>**.

31. When the following message appears:

```
Reboot the system now.
```

A. Remove the Cartridge Tape Utilities diskette from the diskette drive.

B. Leave the tape in the tape drive for the next procedure.

C. Press **<Ctrl-Alt-Del>** to reboot the system.

32. When the following message appears:

```
Console login:
```

The installation of the base UNIX operating system is complete.

To complete the installation of the UNIX software, install the UNIX add-on packages, as described in the next procedure.

Installing the UNIX Add-On Packages

NOTE

You must install the base UNIX operating system (as described in the previous procedure) *before* you install the UNIX add-on packages.

Also, if any of the UNIX Add-On packages are currently installed, you must remove them before installing the new software. Refer to the section "Remove Add-On Software Packages" in the *AT&T UNIX System V/386 Release 3.2 System Administrator's Guide.*

Installation Time:	Approximately 45 minutes.
Installation Medium:	Cartridge tape labeled "UNIX System V/386, R3.2 V2.1."
Information you need:	*root* password.

33. Login as *root*.

34. Enter ***installpkg***.

35. When the following message appears:

```
Are you installing from tape (y/n)?
```

    Enter *y*.

36. When the following message appears:

```
Insert Installation Tape in drive and press <RETURN>.
```

    (If you removed the tape after the last procedure, reinsert the tape into the tape drive.)

    Press **<Enter>**.

37. When the following message appears:

```
Do you want to install all of the above packages? <y/[n]>:
```

    Press **<Enter>** (to accept the default *n*).

38. The system then prompts you to select the packages to install. Enter *y* after each of the following packages, and *n* after all the other packages (or press **<Enter>** to accept the default *n*):

```
Editing Package Version 2.0
Extended Terminal Interface Package Version 2.0
C Software Development Set 4.1.6
Network Support Utilities Package (1.2) Version 2.0
2 Kilobyte File System Utility Package Version 2.0
Kernel Debugger(s) - Version 2.0
System Administration Software
```

A. When the Kernel Debugger package is being installed, the following appears:

```
Which kernel debugger(s) do you want to install?

   1) DEBUGGER (polish calculator style)
   2) GDEBUGGER (traditional)
   3) both DEBUGGER and GDEBUGGER

Choose 1, 2, or 3:
```

Enter *1*.

B. When the System Administration Software package is being installed, the following appears:

```
Do you want to give passwords to administration login?
(y/n) [n]
```

Press **<Enter>** (to accept the default *n*).

39. When the following message appears:

```
Confirm

To complete the install/removal process a shutdown is now
being initiated automatically.

Make sure your floppy drive is empty.  If you are installing
or removing controller boards, you may power down the system
after the shutdown has completed.

Strike ENTER when ready
or ESC to stop.
```

Press **<Enter>**.

40. When the following message appears:

```
Reboot the system now.
```

Press **<Ctrl-Alt-Del>** to reboot the system.

41. When the following message appears:

```
Console login:
```

The installation of the UNIX Add-on packages is complete.

42. Remove the tape from the tape drive (push tape in to release catch).

To complete the installation of the UNIX software, set up the user and system logins, as described in the next procedure.

Setting Up User and System Logins

NOTE

You must install the UNIX Add-on packages (as described in the previous procedure) *before* you set up the user and system logins.

Installation Time: Approximately 10 minutes, depending on number of user logins being set up.

Installation Medium: None.

Information you need:
Name of your time zone.
Whether you observe daylight savings time.
User login names, IDs, passwords, etc.
Administrative passwords.
System passwords.
node name (name by which other machines know this machine).

---

43. Login as *setup*.

44. Answer the questions that **setup** asks.

NOTE

One of the questions asks you to select your time zone from a list. If your time zone is not included in this list, select GMT, and then enter the correct local time when a later question asks for the time.

To complete the installation of the UNIX software, install the remote terminal package, as described in the next procedure.

Installing the Remote Terminal Package

Installation Time:	Approximately 5 minutes.
Installation Medium:	Floppy diskette (1) labeled "Remote Terminal Package."
Information you need:	*root* password.

45. Login as *root*.
46. Enter ***installpkg***.
47. When the following message appears:

```
Are you installing from tape (y/n)?
```

    Enter ***n***.

48. Insert the installation diskette into the diskette drive, and turn the handle down.
49. When the following message appears:

```
Strike ENTER when ready
or ESC to stop.
```

    Press **<Enter>**.

50. When the following message appears:

```
Enter option:
```

    Enter ***1***.

51. When the following message appears:

```
Enter a file name, 'all', 'done', or 'files':
```

    Enter ***all***.

52. When the following message appears:

```
Enter a file name, 'all', 'done', or 'files':
```

Enter ***done***.

53. When the following message appears:

```
Enter option:
```

Enter ***0***.

54. Remove the diskette from the diskette drive.

55. If you are not planning to install the TCP/IP and Ethernet drivers (as described in the next section), press **<Ctrl-D>** to log out. If you are planning to continue the installation process, enter:

```
shutdown -is
```

The following message is displayed:

```
Type Ctrl-d to proceed with normal startup
(or give root password for system maintenance):
```

Enter the root password.

56. The single-user shutdown does not automatically mount the user partition. The **installpkg** command resides in this partition. Mount the "usr" partition by entering the following command:

```
/etc/mount  /usr
```

This completes the installation of the UNIX software.

Installing TCP/IP and Ethernet Drivers

NOTE

The UNIX software must be installed *before* you can install the TCP/IP software. Refer to the discussion of "Installing UNIX System V/386 Release 3.2 Version 2.1" on page 3-18 of this release note

---

Installation Time: Approximately 30 minutes.

Installation Media: Cartridge Tape labeled "Intel TCP/IP R3.0.1"

Information you need:
root password.
Internet address.
Network name.
Whether network is subnetted.
Domain name.
Broadcast Address

CAUTION

The *remove* and *install* scripts for TCP/IP remove/overwrite the following configuration files:

/etc/hosts
/etc/networks
/etc/gateways
/etc/hosts.equiv
/usr/lib/named/named.hosts
/usr/lib/named/named.local
/usr/lib/named/named.rev
/usr/lib/named/named.soa
/usr/lib/named/root.cache

If you are updating a previous release of Intel TCP/IP, you may want to rename or backup these files *before* removing the previous release. After you successfully install TCP/IP Release 3.0.1, you can restore these files.

If you are continuing the installation process from a previous section, begin with instruction number 4.

The procedure for installing the TCP/IP packages is as follows:

1. Login as *root*.

   Make sure that the users are not logged in and the system is not NFS mounted.

2. Shutdown to single user mode by entering:

   shutdown -is

   The following message is displayed:

   ```
   Type Ctrl-d to proceed with normal startup
   (or give root password for system maintenance):
   ```

   Enter the root password.

3. The single-user shutdown does not automatically mount the user partition. The **installpkg** command resides in this partition. Mount the "usr" partition by entering the following command:

   /etc/mount /usr

4. Enter ***installpkg***.

5. When the following message appears:

   ```
   Are you installing from tape (y/n)?
   ```

   Enter *y*.

6. When the following message appears:

   ```
   Insert Installation Tape in drive and press <RETURN>.
   ```

   insert the tape labeled "Intel TCP/IP R3.0.1" in the tape drive.

   Press **<Enter>**.

7. When the following message appears:

   ```
   Do you want to install all of the above packages? <y/[n]>:
   ```

   Press *y* to install all packages.

8. When the following message appears:

```
Will you be installing the Ethernet Drivers? <y/[n]>:
```

   Press **y**.

At this point, you will be asked for your network configuration information.

9. When the following message appears:

```
Confirm

To complete the install/removal process a shutdown is now
being initiated automatically.

Make sure your floppy drive is empty.  If you are installing
or removing controller boards, you may power down the system
after the shutdown has completed.

Strike ENTER when ready
or ESC to stop.
```

   Press **<Esc>**.

   The installation is complete.

10. Remove the tape from the tape drive (push tape in to release catch).

NOTE

If you are installing NFS, that installation process should take place at any time after the TCP/IP and Ethernet Drivers software have been installed.

Installing the Release 3.3.1 System Software

NOTE

The install script modifies the three standard *crontab* files. If you want to keep your existing *crontab* files, save them before performing this procedure, and then restore them afterwards.

Installation Time:	Approximately 30 minutes.
Installation Medium:	1 cartridge tape labeled "iPSC®/2 System Software R3.3.1"
Information you need:	*root* password.

If you are continuing the installation process from a previous section, begin with instruction number 4.

1. Login as *root*.

2. Enter single user mode. This prevents other users from logging in.

   shutdown -is

   The following message is displayed:

   ```
   Type Ctrl-d to proceed with normal startup
   (or give root password for system maintenance):
   ```

   Enter the root password.

3. Mount the user partition. The **removepkg** command resides in this partition. The single-user shutdown does not automatically mount the user partition.

   /etc/mount /usr

4. Enter ***installpkg***.

5. When the following message appears:

```
Are you installing from tape (y/n)?
```

   Enter *y*.

6. When the following message appears:

```
Insert Installation Tape in drive and press <RETURN>.
```

   Insert the installation tape labeled "iPSC®/2 System Software R3.3.1" into the tape drive (label up, exposed tape to the left; push until the tape cartridge locks into place).

7. Press **<Enter>**.

8. Eventually, the following message appears:

```
Do you want to install all of the above packages? <y/[n]>:
```

   Enter *y*.

9. When the following message appears:

```
Confirm

To complete the install/removal process a shutdown is now
being initiated automatically.

Make sure your floppy drive is empty.  If you are installing
or removing controller boards, you may power down the system
after the shutdown has completed.

Strike ENTER when ready or
ESC to stop
```

   Press **<Enter>**.

10. When the following message appears:

```
Reboot the system now.
```

   A. Remove the tape from the tape drive (push the tape in to release the catch).

   B. Press **<Ctrl-Alt-Del>** to reboot the system.

11. When the following message appears:

```
Console login:
```

    log in as *root* again.

12. Enter the following command to remove permission for ordinary users to use the **rebootcube** command:

```
chmod 700 /usr/bin/rebootcube
```

    This **chmod** command means that only the superuser can reboot the cube, making it less likely that someone will reboot the cube without understanding the consequences of their actions.

The installation of the Release 3.3.1 system software is now complete.

Installing the Release 3.3.1 Extension Software

NOTE

You *must* install the iPSC/860 Release 3.3.1 Compiler Libraries *before* you install the iPSC/860 Performance Analysis Tools package from the iPSC/860 Extension Software R3.3.1 tape. Both packages are on the Extension tape.

If you have a 140M-byte disk on your SRM, please see "Disk Space Requirements" on page 3-4 to determine how much disk space is required to install the extension software. You may need to upgrade to a larger disk to install all the software you need.

Installation Time:	Approximately 30 minutes.
Installation Medium:	1 cartridge tape labeled "iPSC®/860 Extension Software R3.3.1"
Information you need:	*root* password.

1. Login as *root*.
2. Enter ***installpkg***.
3. When the following message appears:

   ```
   Are you installing from tape (y/n)?
   ```

   Enter ***y***.
4. When the following message appears:

   ```
   Insert Installation Tape in drive and press <RETURN>.
   ```

   Insert the installation tape labeled "iPSC®/860 Extension Software R3.3.1" into the tape drive (label up, exposed tape to the left; push until the tape cartridge locks into place).
5. Press **<Enter>**.
6. Eventually, the following message appears:

   ```
   Do you want to install all of the above packages? <y/[n]>:
   ```

   What you install depends on what you use. To save disk space, do not install the PAT software on your system unless you need it. You can install this software later if desired. However, you *must* install the iPSC i860 extension software and the Compiler Libraries System before you can use your iPSC/860 system.

   If you don't want to install all of the packages, press **<Enter>** (to accept the default ***n***). The system then prompts you to select the packages to install. Enter ***y*** for the packages you want and ***n*** for the ones you don't want.

When the prompt returns, the installation of the iPSC/860 Release 3.3.1 extension software is complete.

- If you installed PAT, you must install the cross-development environment library tape, as described under "Installing the Cross-Development Environment" on page 3-42, then perform the configuration procedure described under "Configuring PAT Software" on page 3-44 to make PAT usable.

NOTE

If your hardware configuration includes an I/O subsystem, then create and run the following development configuration file:

```
bootcube
cd /usr/ipsc/conf
cp devconf devconf.old
getcube -t0
../diag/mkdevconf
killcube
relcube
```

Installing the Release 3.3.1 iPSC®/860 Compilers

If you are an update customer, please skip this section and proceed to the section titled "Installing the Release 3.3.1 Extension Software" on page 3-35.

NOTE

You need to remove R1.0 compilers prior to the installation of R2.0 compilers.

At this point, you install the iPSC/860 Fortran and/or C compilers on the SRM. See the *iPSC®/860 Fortran Compiler Release 2.0 Software Product Release Notes* and/or *iPSC®/860 C Compiler Release 2.0 Software Product Release Notes* for instructions on removing the old compilers and installing the new compilers, then return to this document to complete your software installation.

NOTE

The Compiler Release Notes describe only how to install the compilers on the SRM. Installation of the cross-compilers on Sun-3 or Sun-4 workstations is performed later, as part of the procedure described under "Installing the Cross-Development Environment" on page 3-42.

Installing the Release 3.3.1 Optional Software

Installation Time:	Approximately 30 minutes.
Installation Medium:	1 cartridge tape labeled "iPSC®/2/860 Optional Software R3.3.1"
Information you need:	*root* password.

1. Login as *root*.
2. Enter ***installpkg***.
3. When the following message appears:

   ```
   Are you installing from tape (y/n)?
   ```

   Enter ***y***.
4. When the following message appears:

   ```
   Insert Installation Tape in drive and press <RETURN>.
   ```

   Insert the installation tape labeled "iPSC®/2/860 Optional Software R3.3.1" into the tape drive (label up, exposed tape to the left; push until the tape cartridge locks into place).
5. Press **<Enter>**.
6. Eventually, the following message appears:

   ```
   Do you want to install all of the above packages? <y/[n]>:
   ```

   What you install depends on what you use. To save disk space, do not install the Remote Host, Online Documentation, Simulator or NQS software on your system unless you need it. You can install this software later if desired. The Simulator, and NQS software have not changed from Release 3.3 though it is necessary to reconfigure the NQS as described in the Configuring NQS Software section. If you have any of these packages already installed, the content is the same as the packages in Release 3.3.1.

If you don't want to install all of the packages, press **<Enter>** (to accept the default *n*). The system then prompts you to select the packages to install. Enter *y* for the packages you want and *n* for the ones you don't want.

When the prompt returns, the installation of the iPSC/2/860 Release 3.3.1 optional software is complete.

Configuring NQS Software

NOTE

In order to make NQS usuable, you need to compile and link NQS with Release 3.3.1 software, even if NQS is already built.

The NQS software is on the iPSC®/2/860 Optional Software R3.3.1 tape; you choose whether or not to install it when you install this tape. If you do install the NQS package, you need to configure it. This section describes a procedure that configures the NQS software for use with a basic iPSC system. The procedure describes the configuration steps necessary at the SRM.

After the procedures in this section are complete, the SRM will be configured as machine number one (mid=1) in a "locally functional" NQS environment with a single batch queue named `bq`. The locally functional NQS environment will be able to accept batch jobs submitted from the SRM, but it will not yet be able to accept jobs submitted from other machines. In order to use NQS with other machines or in more complex systems, you must perform the more detailed configuration procedures described in the *iPSC®/2 and iPSC®/860 Network Queueing System Manual*. Refer to Appendix A, "Reconfiguring NQS" in the *iPSC®/2 and iPSC®/860 Network Queueing System Manual* for more information.

Installation Time:	Approximately 30 minutes.
Installation Medium:	None.
Information you need:	Root password.

NOTE

All pathnames must be absolute pathnames (i.e., they must start with `/`).

Perform the following steps at the iPSC SRM:

1. Login as *root* on the SRM.
2. If you already have NQS running from a previous release, then kill the following processes:
   nqsdaemon
   netdaemon
   logdaemon
3. Make sure that your umask is set to 0 by entering the following:

   ```
   umask 0
   ```

4. Change the current directory to */usr/src/nqs* by entering the following:

   ```
   cd /usr/src/nqs
   ```

5. Compile NQS code as follows:

   ```
   make
   ```

6. Install the NQS executable code by entering the following:

   ```
   make install
   ```

   If NQS is already built, a number of **mkdir** error messages are generated. You may safely ignore these error messages.

Before the NQS daemon can be started, the host machine must be in the NQS database. Once the daemon is running, the **qmgr** command is functional and you can use the **qmgr** command to create a queue for job requests. There is an entry point in the *make* file that will allow you to do this automatically. The automatic method will accomplish the following tasks:

1. Insert the SRM machine name as machine #1.
2. Start the NQS daemon.
3. Create a single batch queue by the name of `bq`.
4. Make the `bq` batch queue the default queue for the **qsub** command.

If you want your machine to accept requests from other machines on the network or accomplish other tasks, you must set up the machine IDs and the queues by hand. Refer to Appendix A, "Reconfiguring NQS," of the *iPSC®/2 and iPSC®/860 Network Queueing System Manual* for instructions on setting up the machine IDs and queues by hand.

NOTE

If you already have NQS configured from a previous release, then start NQS with this command:

```
/usr/lib/nqs/nqsdaemon > /usr/lib/nqs/logfile
```

NQS configuration is now complete. Continue the installation process by proceeding to "Installing the Cross-Development Environment" on page 3-42.

If NQS is not configured from a previous release, then perform the following steps to start NQS using the automatic method:

1. Login as *root* on the SRM
2. Change directories to the source directory, */usr/src/nqs* as follows:

```
cd /usr/src/nqs
```

3. Make sure that your umask is set to 0 by entering the following:

```
umask 0
```

4. Start the make file at the start-up tag as follows:

```
make startup
```

You are now able to submit local jobs to the batch queue bq by using the **qsub** command. In order to submit jobs from remote locations in the network, you must set each remote machine up using the detailed procedures presented in Appendix A, "Reconfiguring NQS," of the *iPSC®/2 and iPSC®/860 Network Queueing System Manual.*

NOTE

In order to completely remove the NQS software package and all of its daemons and configuration files, you need to do a ***make remove*** under the */usr/src/nqs* directory prior to using the **removepkg** command.

Installing the Cross-Development Environment

These are the basic instructions for installing the cross-development environment on an NFS network. This should be done after all the optional iPSC software packages have been installed on the SRM. If you are doing a complete installation process (not an update) the NFS package should be installed before installing the Online Documentation and the Cross-Development Environment.

Installation Time:	Approximately 25 minutes.
Installation Medium:	None.
Information you need:	Root password for the file server.

---

1. Have the system administrator of the network create an SSD tools root directory. There needs to be about 40 megabytes of disk space on the NFS file server that has this directory. The directory should be owned by *root* and writable only by *root* but readable and executable by all.

   For example, if the SSD tools root directory is to be */vol/tools/ssd*, perform the following commands as *root* on the file server:

   ```
   mkdir /vol/tools/ssd
   chmod 755 /vol/tools/ssd
   ```

2. Set the environment variable **IPSC_XDEV** to the full pathname of the SSD tools root directory. For example:

   ```
   setenv IPSC_XDEV /vol/tools/ssd
   ```

   This environment variable is used by all of SSD's cross-development tools.

3. Copy all the files and directories from the directory */usr/ipsc/XDEV/i860* on the SRM to the SSD tools root directory on the file server.

   For example, using **rcp** from a Sun network:

   ```
   rcp -p -r SRM:/usr/ipsc/XDEV/i860 $IPSC_XDEV
   ```

   Or using **tar** from a Sun network:

   ```
   cd $IPSC_XDEV
   rsh SRM "cd /usr/ipsc/XDEV/i860;tar cf - ."|tar xf -
   ```

   In both of the above commands, *SRM* represents the network name of the SRM.

The installation of the cross-development environment is now complete.

Contents of the Cross-Development Environment

The subdirectories in the **$IPSC_XDEV** directory created by the procedure in the previous section are as follows:

$IPSC_XDEV/i860/bin	Executables for the SRM
$IPSC_XDEV/i860/bin.sun3	Executables for the Sun3
$IPSC_XDEV/i860/bin.sun4	Executables for the Sun4
$IPSC_XDEV/i860/include	Standard include files
$IPSC_XDEV/i860/include-ipsc	iPSC include files
$IPSC_XDEV/i860/lib-coff	COFF libraries
$IPSC_XDEV/i860/lib	Files for SRM executables (ipd daemon)
$IPSC_XDEV/i860/lib.sun3	Files/libraries for Sun3 executables
$IPSC_XDEV/i860/lib.sun4	Files/libraries for Sun4 executables

Using the Cross-Development Environment

Once you have installed the cross-development environment, the users must set the environment variable **IPSC_XDEV** to the full pathname of the SSD tools root directory and then add the appropriate "bin" directory to their execution search paths. For example, C-shell users could place the following commands in their *.cshrc* files:

```
setenv IPSC_XDEV /vol/tools/ssd
set path=( $path $IPSC_XDEV/i860/bin.`arch` )
```

Configuring PAT Software

The PAT software is on the iPSC®/2/860 Extension Software R3.3.1 tape; you choose whether or not to install it when you install this tape. If you do install the PAT package, you need to configure it. This section describes a procedure that configures the PAT software for use with a basic iPSC system. The procedure describes the configuration steps necessary at a Sun workstation.

Installation Time:	Approximately 10 minutes.
Installation Medium:	None.
Information you need:	Pathnames to the cross-development directories.

NOTE

All pathnames must be absolute pathnames (i.e., they must start with /). The cross-development environment must also be installed on your Sun workstation, as described under "Installing the Cross-Development Environment" on page 3-42, before you can configure the PAT software.

The software is installed in the following cross-development directories on the SRM.

/usr/ipsc/XDEV/i860/bin.sun4
/usr/ipsc/XDEV/i860/bin.sun3
/usr/ipsc/XDEV/i860/lib.sun4
/usr/ipsc/XDEV/i860/lib.sun3

Perform the following steps to configure your Sun workstation for use with the PAT utilities.

1. After installing the cross-development environment on your Sun workstation, edit the following two files:

   $IPSC_XDEV/i860/lib.sun4/express.cst
   $IPSC_XDEV/i860/lib.sun3/express.cst

2. In each of the above files, change the following line:

   ```
   PARASOFT:=/usr/ipsc/XDEV/i860
   ```

   to reflect the location of the cross-development directories on your Sun system. For example, if the cross-development directories on your Sun system are installed under */vol/tools/ssd*, change the line to:

   ```
   PARASOFT:=/vol/tools/ssd/i860
   ```

NOTE

You must use the full path name of the directory.

3. In your .login or .cshrc file, set the environment variable EXPRESS to point to the file express.cst. For example:

   ```
   setenv EXPRESS $IPSC_XDEV/i860/lib.'arch'/express.cst
   ```

Your Sun workstation should now be configured properly to use any of the PAT utilities. Refer to the *iPSC®/860 Parallel Performance Analysis Tools Manual* for instructions on using the PAT utilities.

Installing Remote Host Software on a Diskful Host

NOTE

The remote host software package must be installed on the SRM *before* you can install the remote host software on a remote host. Refer to the section "Installing the Release 3.3.1 Extension Software" on page 3-35 of this manual.

Use the following procedure to install the remote host software on a remote host that has its own disk. If installing on a diskless host, see "Installing Remote Host Software on a Diskless Host" on page 3-48.

Installation Time:	Approximately 30 minutes.
Installation Medium:	None.
Information you need:	*root* password.

1. Login as *root* on the remote host.
2. Create a directory on the remote host to contain the remote host source code. This directory can be located wherever you choose.
3. Copy the remote host source code from the directory */usr/ipsc/rhost* on the SRM to the directory you created in the previous step. For example, this can be accomplished with the following command line:

   ```
   rcp -p -r SRMname:/usr/ipsc/rhost newdir
   ```

   where:

   SRMname is the internal name of the SRM.
   newdir is the name of the directory created in Step 2 of this procedure.
4. On the remote host, edit the makefile in the top-level remote host source code directory so that the installation directories are correct for your system:

NOTE

All pathnames must be absolute pathnames (i.e., they must start with `/`).

IPSCDIR Contains daemons and named sockets. The default is */usr/ipsc* and the subdirectories *lib* and *log* will be created.

BINDIR Contains cube binaries. On the SRM these files are in */usr/bin*. Any directory can be used for the binaries, as long as the cube users have it in their search path. The default directory on the remote host is *IPSCDIR/bin*.

LIBDIR Contains *libhost.a*. The default location for the library is in */usr/lib*. Again, any directory can be used as long as the user application makefiles reflect the correct directory.

INCDIR Contains the iPSC/2 include files *cube.h* and *fcube.h*. The default is *IPSCDIR/include*.

5. Kill any existing *commser* and *fserver* daemons on the remote host.

6. Enter ***make*** to build binaries on the remote host.

NOTE

On some systems, umask is declared in a standard header file, and this step will fail during serve_io.c compilation. If this happens, edit *./fileio/serve_io.c* and comment out (using C-style comments) the umask declaration. Then, invoke the **make** command again.

7. Enter ***make install*** to install the binaries in the prescribed directories.

8. Edit the *srms* file (*IPSCDIR/lib/srms*) to include the names of all SRMs that the remote host may access.

9. Edit the */etc/hosts* file on the SRM to include the names of all the remote workstations that may access the SRM.

10. Edit the file */usr/ipsc/lib/ruser* on the SRM to include the host and user names for all hosts and users that may access the SRM using remote host software. (The format of this file is the same as that of */etc/hosts.equiv*).

11. Edit the user's *.cshrc* file to include the following line:

```
setenv TTY `tty` >& /dev/null
```

This completes the installation of the remote host software on a remote host. In order to compile programs for RX nodes on the remote host, you must also install the cross-development environment, as described under "Installing the Cross-Development Environment" on page 3-42 This will make the cross-compilers **icc** and **if77** available to you on your Sun-3 or Sun-4 workstation.

Installing Remote Host Software on a Diskless Host

NOTE

The remote host software package must be installed on the SRM *before* you can install the remote host software on a remote host. Refer to the section "Installing the Release 3.3.1 Extension Software" on page 3-35 of this manual.

Use the following procedure to install the remote host software on a remote host that does not have its own disk. If installing on a diskful host, see "Installing Remote Host Software on a Diskful Host" on page 3-45.

Installation Time: Approximately 30 minutes.

Installation Medium: None.

Information you need: *root* password.

NFS security features conflict with the installation procedure for remote host software described in the previous section. NFS does not allow *root* access to remotely-mounted filesystems. The remote host software requires the builder and installer to have *root* access to the filesystems on which it is installed. Since a diskless client cannot have *root* access, a different installation procedure must be followed.

NOTE

Some NFS mounted file systems also do not allow **setuid** programs to be installed on them. The remote host software executables *must* be installed on a mounted file system that allows a **setuid** program to execute.

In the examples in this procedure, the NFS file server (called **server**) has a directory called */export/sun386/local* which its diskless clients (one is called **client**) mount as */usr/local*. The remote host software for the clients is to be installed in a directory called */usr/local/ipsc* from the client's point of view. This path will be compiled into several of the remote host programs.

1. Become *root* on the NFS file server and **cd** to the parent directory that will contain the subdirectory where the remote host software is to be installed. For example:

```
{server:49} cd /export/sun386/local
```

2. Create the directory to contain the remote host source code, and set its protection so anyone can write to this directory: For example:

```
{server:50} mkdir ipsc
{server:51} chmod a+w ipsc
```

3. Become *root* on the client machine and **cd** to the directory you created in the previous step. For example:

```
{client:17} cd /usr/local/ipsc
```

4. Copy the remote host source code from the directory */usr/ipsc/rhost* on the SRM to the current directory.

5. Edit the makefile in the current directory so that the installation directories are correct for your system:

NOTE

All pathnames must be absolute pathnames (i.e., they must start with /) and must be from the client's view of the file system (i.e., */usr/local/ipsc* instead of */export/sun386/local/ipsc*).

IPSCDIR Contains daemons and named sockets. The default is */usr/ipsc* and the subdirectories *lib* and *log* will be created.

BINDIR Contains cube binaries. On the SRM these files are in */usr/bin*. Any directory can be used for the binaries, as long as the cube users have it in their search path. The default directory on the remote host is *IPSCDIR/bin*.

LIBDIR Contains *libhost.a*. The default location for the library is in */usr/lib*. Again, any directory can be used as long as the user application makefiles reflect the correct directory.

INCDIR Contains the iPSC/2 include files *cube.h* and *fcube.h*. The default is *IPSCDIR/include*.

6. Kill any existing *commser* and *fserver* daemons on the remote host.

7. Type ***make***. Some directories will be created in */usr/local/ipsc* and some header files installed. The remote host software libraries and executables will be created.

NOTE

On some systems, umask is declared in a standard header file, and this step will fail during serve_io.c compilation. If this happens, edit *./fileio/serve_io.c* and comment out (using C-style comments) the umask declaration. Then, invoke the **make** command again.

8. When the **make** is completed, become *root* on the NFS file server again and remove the world write permissions from the directory created in step 2. For example:

```
{server:52} cd /export/sun386/local
{server:53} chmod o-w ipsc
```

9. **cd** to the directory containing the remote host source. For example:

```
{server:54} cd /export/sun386/local/ipsc
```

10. Edit the **makefile** and set **IPSCDIR** correctly for the server's view of the filesystem (in this example, this would be */usr/sun386/local/ipsc*).

11. Install the software by typing ***make install***.

12. Edit the *srms* file (*IPSCDIR/lib/srms*) to include the names of all SRMs that the remote host may access.

13. Edit the */etc/hosts* file on the SRM to include the names of all the remote workstations that may access the SRM.

14. Edit the file */usr/ipsc/lib/ruser* on the SRM to include the host and user names for all hosts and users that may access the SRM using remote host software. (The format of this file is the same as that of */etc/hosts.equiv*).

15. Edit the user's *.cshrc* file to include the following line:

```
setenv TTY `tty` >& /dev/null
```

This completes the installation of the remote host software on a remote host. In order to compile programs for RX nodes on the remote host, you must also install the cross-development environment, as described under "Using the Cross-Development Environment" on page 3-44. This will make the cross-compilers **icc** and **if77** available to you on your Sun-3 or Sun-4 workstation.

Limitations and Workarounds 4

Introduction

This chapter describes known limitations and suggested workarounds for the following Release 3.3.1 system software components:

- Documentation
- NX Node Executive
- Node Shell
- Concurrent File System™
- Concurrent File System™ Tape
- Intel386™ Assembler and Linker
- iPSC®/2 C Compiler
- iPSC®/2 Fortran Compiler
- Interactive Parallel Debugger
- Compiler Libraries
- RX Vector Library
- Basic Math Library (*libkmath.a*)
- Remote Host
- Network Queueing System (NQS)
- iPSC®/2 Simulator
- UNIX Software
- TCP/IP on the SRM
- Cube Diagnostic Program (CDP)
- Performance Analysis Tool (PAT)

NOTE

Read the following sections carefully. Report any problems you encounter while using your iPSC system to SSD Customer Support Response Center at:

1-800-421-2823 (Customer Support Response Center)
Your Local Intel Sales Office (in Europe)
support@ssd.intel.com (Internet address)

NOTE

Most of the following Limitations and Workarounds contain a number in brackets at the end of each Limitation. That number represents Intel's internal tracking system. Please refer to that number when refering to a problem documented in this chapter.

Documentation

1. **Remove some i860 manuals from your manual set**
   Four Intel i860 manuals are shipped as part of the iPSC documentation package. These four manuals are shrink-wrapped in a single package. Only two of these manuals apply to the iPSC/860 systems. The two manuals that apply are the *i860™ 64-Bit Microprocessor Assembler and Linker Reference Manual* and the *i860™ 64-Bit Microprocessor Simulator and Debugger Reference Manual.*

   There are two manuals in the package that are not supported and should not be used:

   - *i860™ 64-Bit Microprocessor Math Library Reference Manual*
     The iPSC system does not support the i860 math libraries as documented in this manual. The iPSC supports the math libraries described in the UNIX System V/386 documentation.

   - *i860™ 64-Bit Microprocessor Object File Utilities Reference Manual.* The object file utilities described in this manual have much the same functions as those that the iPSC supports, but the names are different. iPSC object file utilities are described in the *iPSC®/2 and iPSC®/860 User's Guide.*

2. **The Intel386 assembler is not documented**
   as is documented only as **as(1)** in the UNIX System V/386 manuals. The actual assembly instructions are not documented.

3. **Discussion of account_hook() is unclear**
   The discussion of the **account_hook()** routine in the "Accounting Software Support" section in chapter 6 of the *iPSC®/2 and iPSC®/860 System Administrator's Guide* does not make clear that **account_hook()** is a routine that is *written* by the user and *called* by the system. The file */usr/ipsc/lib/commser.bin/hooks.c* is not the source code for the system **account_hook()**; it is an

example of the kind of **account_hook()** function you might use. Most of this file is surrounded by an **#ifdef DEBUG/#endif** block; to use this example file, compile and install it with the command **make DBG=-DDEBUG.**

4. **rebootcube is no longer usable by ordinary users**
Intel SSD strongly recommends that you remove permission for ordinary users to use the **rebootcube** command. However, the documentation still describes the **rebootcube** command as a command that can be used by ordinary users.

5. **b switch in getcube (getcube -tb)**
The intent of this switch is to allow users to easily try their programs with buffering disabled to see if it makes any performance difference. Use **b** in the cube type specifier to disable message buffering, which is normally enabled by default. Message buffering is required in some programs, especially simple ones which interchange messages. However, it can be demonstrated that it adds a penalty of about 20% in some kinds of ring programs, due to the extra overhead of copying a message when it would have been better to wait before sending it.

6. **Improving the performance of exchange message passing**
The following Fortran code uses an exchange message passing technique among the processes. In Release 3.3.1, it runs about 30% slower than Release 3.2 due to some synchronization. This type of exchange is used in applications such as 2-dimensional FFTs, matrix transpose, matrix-vector multiply, and so on.

```
                .
                .
                .
          do 10 i = 1, p-1
                .
                .
                .
              call csend(i, buf1, size, xor(me, i), pid)
              call crecv(i, buf2, size)
                .
                .
                .
10      continue
```

To improve the performance of this code, you can use force type in addition to some synchronization, as follows:

```
                .
                .
                .
          do 10 i = 1, p-1
                .
                .
                .
              id = irecv(forcetype, buf2, size)
              call csend(type, 0, 0, xor(me, i), pid)
              call crecv(type, 0, 0)
```

```
                call csend(forcetype, buf1, size, xor(me, i), pid)
                call msgwait(id)
                 •
                 •
                 •
10      continue
```

7. **Return values for estat() and festat() in PRM are incorrect**
   In the *iPSC®/2 and iPSC®/860 Programmer's Reference Manual* the "Return Values" description states that **estat()** and **festat()** return -1 when an error occurs. This is incorrect. When an error occurs, **estat()** and **festat()** display an error message and terminate the user's application, which is consistent with the rest of the iPSC system calls. The functions **_estat()** and **_festat()** return -1 upon error.
   [2199]

8. **offl option for mt() command not documented**
   The **mt** command on the node shell supports the **offl** command option. The **offl** option is not documented in *iPSC®/2 and iPSC®/860 Programmer's Reference Manual.*
   [2115]

9. **ps cube command -s option not documented**
   The iPSC implementation of the **ps** command supports the **-s** option, which provides information about system processes that are running in the cube. The System/V.3.2 **ps** utility does not recognize the **-s** option. This difference is not documented in the *iPSC®/2 and iPSC®/860 Programmer's Reference Manual.*
   [2048]

10. **etool label information is lost**
    None of the **eprof_label()** information is visible when **etool** is viewing the second dump file. Before viewing a second dump file the labels need to be redefined and all of the toggles need to be reinitialized. This needs to be clarified in the *iPSC®/2 and iPSC®/860 Programmer's Reference Manual.*
    [2040]

11. **icc -u switch ignored**
    The **-u** switch that is documented on page 2-2 of the *iPSC®/860 C Compiler User's Guide* is not supported, which explains why **icc** responds to **-u** with the following message: `icc: -u ignored`.
    [1764]

12. **cwrite() command is used on cfs and srm files**
    In the *iPSC®/2 and iPSC®/860 Programmer's Reference Manual* the section explaining **cwrite()** claims that **cwrite()** is used only for CFS files. This is incorrect. The **cwrite()** function may be used for both CFS and SRM files. (This is also true for **cread()**.)
    [1760]

13. **restrictvol() documentation does not match software functionality**
The documentation for **restrictvol()** states that if the *nvol* parameter is negative, then **restrictvol()** returns the volume numbers that a file is restricted to. This is true only if *nvol* is -1.

    Negative numbers other than -1 are considered invalid by the software.
    This problem affects the *iPSC®/2 and iPSC®/860 Programmer's Reference Manual.*
    [1747]

MAN PAGES

1. **The man manpage does not show the -d option**
The **-d** option selects debug information. This feature is not documented in the on-line manpage.
[1757]

2. **man-M option does not work correctly with relative pathnames**
Use absolute pathnames with the **-M** option.
[1773]

3. **PAT call eprof_toginit manpage filename too long**
The manpage for the routine **eprof_toginit** is accessable as **eprof_togini.**
[1782]

4. ***more* pager formats bold lines incorrectly**
Use the default *less* pager with man. If *more* is required, set the environmental variable as follows:

```
setenv PAGER = "col -b | more"
```

    [1793]

5. **man pages cannot be accessed when request is in uppercase letters**
Specify call names in all lower case.
[1888]

6. **Manpath usage message is incorrect**
The **-h** option supercedes the **-q** option. Use the **-h** option separately.
[2050]

7. **cbackup man page implies that -s 2.2G should work**
The **cbackup -s** switch does not accept decimal points. Use either of the following:

```
cbackup -s 2G
cbackup -s 2200M
```

    [2071]

NX Node Executive

1. **Use of a node PID number other than 0** ***(RX nodes only)***
   If you send a message to an RX node using a PID (process id) other than 0, the message will be delivered to PID 0 and no error message will be returned.

2. **Mixing long and short messages may result in messages being received out of order**
   Long and short messages are queued separately on the iPSC. (A "long" message is one that is greater than about 100 bytes in length.) As a result, if a series of long and short messages of the same type are sent from one node (or the host) to one or more other nodes (or the host), the messages may not always arrive in the same order as they were sent. For example, if the receiving node has several received long messages pending but no short ones, and a short message arrives, the node will process the short message immediately. This can cause unexpected results.

   Workaround: To enforce message order, use unique message types for messages of different sizes.
   [1399]

3. **Can't use csend() or crecv() within a signal handler**
   A call to **csend()** or **crecv()** within a signal handler may fail with *errno* **EINVAL** if there are pending **csend()**s or **crecv()**s. Note that some library calls, such as **printf()**, generate **csend()**s.

   Workaround: Keep your signal handlers extremely simple. Have them set flags and let the surrounding application handle the condition.
   [1126]

4. **Invalid buffer pointers**
   Invalid buffer pointers in C programs will sometimes be accepted as valid and not return an error message.

5. **Node may deadlock with file I/O**
   If a node's message buffers are filled but no receives are posted, attempts to do file I/O may fail. To avoid this failure, you must make sure that the node receives pending messages.

6. **Host program hangs when sending messages to nonexistent nodes**
   If a host program sends messages to nodes that are not part of the cube, the host program hangs. Use **killcube** to flush all the messages.
   [1006]

7. **cubeinfo system call and command give slightly different results**
   The **cubeinfo()** system call returns the system and domain name for the *srmname* and *hostname* fields. The **cubeinfo** command truncates the domain name, and displays just the system name.

8. **Prints originating within a handler**
   Prints originating from within a handler routine may overwrite or merge with the prints from other nodes.

9. **Sending long messages to RX nodes**
   If you send a long message to an RX node other than yourself, the send will not return until the receiving process has been loaded.

10. **Differences between RX and CX nodes when using hrecv()**
    On RX nodes, using **hrecv()** to receive a message that is too long to fit in the buffer causes the program to terminate with an error message. On CX nodes, the receive completes, the message is lost, and no error is returned.

11. **Handler routine gets different node value for global calls on RX and CX nodes**
    If the **hsend()** system call is used on an RX node with a negative destination node value, the handler receives the given value in its *node* parameter. If the same **hsend()** call is used on a CX node, the handler always receives -1 in its *node* parameter.
    [1219]

12. **Calling sleep() and hsend(), hrecv(), or hsendrecv() in a node program**
    Calling **sleep()** and **hsend()**, **hrecv()**, or **hsendrecv()** in a node program may cause the program to fail to return from the handler routine.

    Workaround: Don't call **sleep()**. Instead, you can use **mclock()** in a loop.

13. **pipe(), fork(), and exec() calls not fully supported**
    UNIX-compatible calls **pipe()**, **fork()**, the **exec()** family, **setuid()**, and **setgid()** for the nodes are only partially implemented. They are used by the node shell (**nsh**) and may be useful in porting programs from the UNIX environment to run under **nsh**. Using them in node programs is *strongly* discouraged.

14. **setenv TTY 'tty' in .cshrc file**
    The **TTY** environment variable is used by several cube commands and must be set before you use the cube. This variable should be set in *.cshrc*, not in *.login*. However, for the remote copy command (**rcp**) to work properly, you must redirect the standard error from the **setenv TTY 'tty'** to null as follows:

    ```
    setenv TTY `tty` >& /dev/null
    ```

15. **archcube -c gives spurious error message if no cube is attached**
    If you do not have a current cube (for example, if you have just done a **relcube** without a following **attachcube**), the command **archcube -c** will give the error message "(host) attachcube: cubename does not exist." This message can safely be ignored.
    [1360]

16. **relcube -c does not reattach to original cube**
If you are attached to a cube and you use the command **relcube -c** to release a different cube, it leaves you unattached to any cube.
[1406]

17. **Rebooting the cube stops process logging**
When the cube is rebooted, any previous process logging (started with **plogon**) is stopped.

Workaround: Write a short C program that uses the routine **system()** to call **bootcube**, then **plogon -a**. Use this program instead of the command **bootcube** to boot the cube.
[1208,1384]

18. **syslog hangs when node sends many messages to the host**
If a lot of messages are being sent to the host and have not yet been received, the host's buffers can fill up. If you attempt to use the **syslog** command or system call while the buffers are full, it can hang.

Workaround: To unlock **syslog**, flush or receive all messages on the host.
[1317]

19. **waitcube and killcube**
The **waitcube** and **killcube** commands occasionally fail to return. To recover, wait at least 30 seconds, press your interrupt key (default **<Del>**) to kill the command, and then do a **relcube**. If the **relcube** fails to return after 30 seconds, a **bootcube** may be required.

20. **Host pid 0 may cause cube commands to fail**
If there is a host program in the background that has done a **setpid(0)**, some cube commands will die early with the following message:

```
(host) ___: Pid already in use
```

This is because **waitcube**, **startcube**, **load**, and **killcube** try to use **setpid(0)**.
[736]

21. **Nodes marked unusable**
If you get a message saying,

```
nn node(s) not responding
```

(where *nn* is the number of nodes), use **bootcube** to reset the nodes. If problems persist, use the Cube Diagnostic Program, **cdp**, to check for bad node boards. The *iPSC®/2 and iPSC®/860 System Administrator's Guide* describes **cdp**.

22. **Incorrect cubeconf settings can cause red LEDs to flash continuously**
If *cubeconf* slot fields are set to EMPTY but contain CX nodes, the red LEDs flash continuously after a **bootcube**. See the *iPSC®/2 and iPSC®/860 System Administrator's Guide* for more information on the *cubeconf* file.

23. **Host long messages**
Host programs cannot use **csend()** or **isend()** followed by **crecv()** to send long messages (greater than 100 bytes) to themselves. Also, a node program sending a message of greater than 100 bytes to the host will not complete the send operation until the host application posts a receive for the message.

Workaround: Use **irecv()**, **csend()**, **msgdone()**, or **msgwait()**.
[1009]

The following example shows a code fragment from a host program:

```
id = irecv ( 1, buf, sizeof(buf));
.
.
.
csend ( 1, msg, sizeof(msg), myhost(), mypid());
msgwait(id);
```

The following host code works only if the send and receive buffers are different:

```
id = isend ( 1, msg, sizeof(msg), myhost(), mypid());
.
.
.
crecv ( 1, buf, sizeof(buf));
msgwait(id);
```

24. **Receiving process must be alive at time of send for long message buffering to work**
Long message buffering on the nodes only works if the receiving process is alive at the time of the send.

25. **setsyslog()**
Calling **setsyslog()** when the host program is already piping the output through **syslog** causes the host program to hang. To recover, wait 10 seconds and then press your interrupt key (default **<Del>**).

26. **newserver()**
A host program cannot call **newserver()** if its output is being piped through **syslog**. Doing so causes the host program to hang or be killed. To recover, wait 10 seconds and then press your interrupt key (default **<Del>**).

27. **killcube() doesn't flush file server messages**
If a host program calls **killcube()** before nodes complete file I/O, some output may be lost.

Workaround: Call **waitcube()** before calling **killcube()**.

28. **Cube allocated by bootcube**
When **bootcube** is executed on an SRM, a zero-node cube named "iocube" is allocated. This is necessary for all systems. It reduces the number of cubes that users can allocate from 10 to 9.

29. **Maximum number of outstanding irecv()s varies between SRM and remote host**
A maximum of 11 outstanding **irecv()**s per cube may be posted on an SRM by a host program. If you attempt to post more than this, an error message displays, and the process aborts. The maximum for the remote host is 12 outstanding **irecv()**s per cube.

30. **Running bootcube under rsh never returns**
The command **rsh** *srmname* **bootcube** on a remote workstation appears to hang, because the **rsh** process waits for the **bootcube** process and *all* its children to finish, and some of the child processes created by **bootcube** do not finish until the next time the cube is rebooted.

Workaround: To use **bootcube** from a remote workstation, don't use **rsh**; log into the SRM with **rlogin**.
[348]

31. **Using floating point operations in the hrecv() handler for CX applications requires special care**
If your program is running on iPSC/2, your **hrecv()** handler must take special measures if any Intel387 floating point operations are being performed. This is to ensure that the state of the Intel387 floating point hardware is properly preserved and restored.

Workaround: Include the following lines inside your **hrecv()** handler immediately after any local variable declarations:

```
asm(" .data);
asm("save_space:  .=.+200");
asm(".text");
asm(" fnsave save_space");
```

At the very end of your **hrecv()** handler, include the line:

```
asm(" frstor save_space");
```

Node Shell

1. **Don't use load command under nsh**
   If you use the **load** command under **nsh**, it may fail with the message "invalid node" or "pid already in use." After the "pid already in use" error, **bootcube** may be required in order to use those nodes.

   Workaround: The system administrator is advised to remove read and execute permissions from the file */usr/i860/ipsc/bin/load*. Executable files on the CFS can be loaded using the **load** command on the SRM.
   [1375]

2. **nsh standard output can go to the wrong window**
   If you are doing a **getcube** in one window and running **nsh** on a service node in another window (or using two terminals), under some circumstances the standard output of **nsh** can go to the **getcube** window. (Standard input comes from the correct window.)

   Workaround: To prevent this problem, be sure the **TTY** variable is properly set in each window. To do this, add the following line to your *.cshrc* (not *.login*) file:

   ```
   setenv TTY `tty` >& /dev/null
   ```

   [1352]

3. **nsh will not start up if ipd is running in another window**
   If you are logged into the same SRM twice with the same user ID in two different windows (or on two different terminals), and you are running the Interactive Parallel Debugger in one window, any attempt to start **nsh** on a service node in the other window fails with the error "setpid: PID already in use."

   Workaround: To prevent this problem, be sure the **TTY** variable is properly set in each window, as described in the previous item, or get a small cube and use **nsh -s** to run **nsh** on the cube instead of the service node.
   [1353]

4. **At least four nodes are required to run the node shell *(nsh -s on RX nodes only)***
   To run the node shell on your current cube (**nsh -s**), if your current cube is made up of RX nodes it must have at least four nodes. The node shell can be run on one service node (which is what you get if you do not specify the **-s** option) or one CX node. If you use **nsh -s** in a cube consisting of a single RX node, you cannot use any pipes (|) or non-built-in commands (such as **ls** or **cat**) in the node shell. To use pipes and non-built-in commands in the node shell, you must have additional nodes for the additional processes that result from pipes and non-built-in commands.

5. **star command can fail with "permission denied" if there is no tape in the SRM tape drive**
The **star** command defaults to the tape device */dev/tape*, which is the tape drive on the SRM. If you do not specify an output device when you use **star** under **nsh** and there is no tape in the SRM's tape drive, you will get the error message "permission denied." (This error just means "cannot open */dev/tape*.")
[1315]

6. **Changing directory names**
The **mv** command does not work on directories, even to change the name of the directory. The only way to do this is to copy the whole directory structure to the new name and remove the old structure.

7. **Simultaneous tar commands in the background hang nsh**
Simultaneous **tar** commands in the background may hang **nsh**. If this occurs, kill **nsh** with **<Ctrl-\>**. If you were using **nsh -s**, you may also have to use **killcube** and **relcube** to recover.

8. **star cv gives "directory checksum error"**
If you use the **c** and **v** switches of the **star** command together, the resulting tape will give you a "directory checksum error" message when you read it with a **star x** command.

   Workaround: Do not use these two switches in a single **star** command.
   [1309]

9. **cbackup and crestore now require root permissions**
The **cbackup** and **crestore** commands have been changed so that the user must have a numeric user ID of zero (*root*) in order to run them.

Concurrent File System™

1. **CFS can be "full" even if there is space on some volumes**
If even one of the physical disks in the CFS is full, you will not be allowed to allocate any space on the CFS. This can lead to "disk full" errors even when there is still space left on the CFS.

   Workaround: Use **restrictvol()** to restrict your files to disks that still have space left.

2. **Allocating several small areas of disk is less efficient that allocating one large area**
If you repeatedly increase the size of a file by a small amount, the disk space is allocated in small chunks. This can create a lot of overhead and fragmentation, causing the CFS to fill up faster. When possible, use **esize()** to pre-allocate all the disk space you expect you will need in one large chunk.

   For example, suppose the CFS has just over 200M bytes free. If you allocate 200M bytes in one **esize()** call, you'll get it. But if you allocate 100K bytes at a time, the disk might fill up after you get only 180M bytes.
   [1004]

3. **Deleting large CFS files can take a while**
   Because the information on the CFS is spread over multiple disks, it can take a long time (up to several minutes) for a very large file to be deleted.
   [1030]

4. **Bad blocks are not automatically mapped out**
   If a disk drive detects a media error, the driver does not automatically remap the block. If a disk block goes bad (which causes hardware-related I/O errors and possible disk hangs), you must manually reformat the disk to get rid of the bad block.
   [1065]

5. **lsize() returns more than what is available on the file system**
   If you attempt to increase the size of a file by more than the remaining free space on the CFS, **lsize()** returns the size you asked for rather than the space left on the file system or -1.
   [1094]

6. **Cannot use open() on a directory name**
   You cannot use **open()** on a directory name in CFS. Use **opendir()** instead.
   [278]

7. **bootcube must be done after mkcfs for showvol to give correct results**
   After creating a new CFS with the **mkcfs** command, you must reboot the cube with the **bootcube** command. If you do not do this, the bytes free count shown by **showvol** will be incorrect.
   [279]

8. **umask() works only on CFS**
   The **umask()** routine on the nodes works only on CFS. It is not supported for SRM files. ***[313]***

9. **Cannot modify permissions or owner of devices created by mkdev**
   There is no way to change the permissions or owner of links that are created by **mkdev**. This prevents giving restricted access to some devices.

10. **Changing disk configuration between a cbackup and crestore**
    If a disk is added between a full **cbackup** and **crestore**, the **crestore** damages the file system, but the damage isn't visible until the next **bootcube**.
    [594]

11. **iowait() returns incorrect error message**
    If an error occurs when you are using **iread()** (such as reading past EOF), **iowait()** may print:

```
Error 0
```

12. **iread()/iowait() on empty file does not give any error message**
If **iread()** is used to read an empty file, the following **iowait()** returns without any errors (unlike **cread()**, which prints an "I/O error" message if you attempt to read from an empty file). The buffer retains the value it had before the **iread()**.

    Workaround: Use **iseof()** to detect an empty file before calling **iread()**.
    [1396]

13. **No way to tell how much iread() or cread() actually read**
The **iread()** and **cread()** system calls do not return the amount of data that was actually read.

    Workaround: If you need to know how much data was read, use **read()**.
    [1268]

14. **lsize() may set errno incorrectly**
lsize() sets *errno* to an incorrect value when it encounters an `insufficient space` error or an invalid `whence` value error.

15. **File date information not updated until file is closed**
The date indicating when a file was last modified is not updated until the file is closed. Therefore, checking this date on a file that is currently open and being written to produces misleading information.
    [416]

16. **File system permissions are not always handled correctly**
In some cases, file system permissions are not handled correctly. In particular, you can not remove the files of others from your directory.
    [682]

17. **Programs compiled with the PAT software enabled (-Mperf) will hang if CFS I/O mode 1 is used.**

    Workaround: Use I/O mode 0, 2, or 3.
    [2216]

Concurrent File System™ Tape

1. **Tape device file must exist before using CFS tape drive**
Refer to the **mkdev** command in the *iPSC®/2 and iPSC®/860 Programmer's Reference Manual*.

2. **Tape devices must be remade after mkcfs**
The **mkcfs** command destroys the entire contents of the CFS, including any tape devices (such as */cfs/tape*). You must use the **mkdev** command after **mkcfs** to recreate each tape device.
    [1029]

3. **Don't reposition the tape head when the tape device is in fixed block mode**
   Using the **mt** command or **ioctl()** system call to reposition the tape head is not recommended when the tape device is in fixed block mode. This is because the CFS reads ahead, copying tape data to memory buffers, so the tape head position will not correspond to the position of the file pointer. If you read from the tape after repositioning the head, the resulting data will not be what was expected.

4. **Only one node at a time can access a CFS tape device**
   A tape device on CFS, such as */cfs/tape*, cannot be opened by more than one process at a time.
   [713]

5. **Fortran tape control statements are not supported for tape devices**
   Tape control is not supported in Fortran. This means that if you open a tape device with the Fortran **open** statement, Fortran tape control statements will not work. For example, the statement **rewind 1** will have no effect if unit 1 is opened as a tape device.

   Workaround: Instead of **rewind**, you can close and re-open the tape device.
   [1031]

6. **Multiple tar commands in nsh script gives tape read error**
   Using the **tar** command several times within a shell script under **nsh** can cause a "tape read error" to occur.
   [1229]

7. **Writing large blocks to tape in variable block mode causes I/O errors**
   When the tape drive is in variable block mode, writing a block that is larger than half the available memory on the controlling I/O node can cause an I/O error. The available memory on the I/O node is the memory size of the node minus approximately 3M bytes, so if you use 4M-byte I/O nodes, the largest block that can be written in variable block mode is approximately 0.5M bytes.
   [1329, 1343]

8. **CFS tape device file entry date is not updated**
   The date field in a CFS tape device file entry does not reflect the last write time. This is because it represents the tape device itself, not the tape loaded in that device.

9. **Multiple names for a tape drive treated the same**
   If there is more than one name for a tape drive, any changes to one name affect the other as well. For example, if the Volume 1 drive is named both */cfs/tape1* and */cfs/tape2*, the command **tapemode -n /cfs/tape1** sets both */cfs/tape1* and */cfs/tape2* to "no rewind."

10. **Attempting to write on a write-protected tape in fixed-block mode**
    Writing to a write-protected tape that is in fixed-block mode may not return an error. (This is because data may be buffered before writing to the tape.) A second attempt to write to a write-protected tape in fixed-block mode may hang the tape driver. To recover, use **bootcube**.

11. **Accessing an off-line tape drive**
Attempts to access a tape drive that is off-line cause CFS to hang. If this occurs, try closing the tape door. If this does not allow CFS to continue, recover using **bootcube**.

12. **Must read same size blocks as were written**
If you use **cread()** to read data from a tape device whose block size is set larger than the block size of the device that was used to write the data onto the tape, you see the message

```
Attempt to read past end of file
```

even when this is not the case.

Workaround: Use the **tapemode** command to reduce the block size of your tape device.

13. **Using crestore with the CFS tape device in variable block mode**
Using the node **crestore** command with the CFS tape device in variable block mode damages the CFS file system.

Workaround: Use the **cbackup** and **crestore** commands with the CFS tape device set to a fixed block mode.

14. **Certain mt commands work only in variable block mode**
The commands **mt bsf** and **mt bsr** work only when the tape device is in variable block mode, and not when it is in fixed block mode.

Intel386™ Assembler and Linker

1. **Directive incorrectly used**
In assembling data objects declared with the **.data** instruction, the Intel386 assembler creates a large temporary file on the SRM. This file can exhaust all available file space on the SRM and abort the assembly.

Workaround: Such data objects will not create large files during assembly if they are declared using the equivalent **.bss** instruction.

iPSC®/2 C Compiler

The iPSC/2 C compiler, **gcc**, is located in */usr/bin* on the SRM and is linked to */usr/bin/cc*. (The UNIX-provided C compiler is in */bin/pcc*.) Use this compiler to compile and link host programs and CX node programs written in C.

The gcc compiler cannot be used to compile programs for RX nodes, and no longer accepts the **-i860** switch. The **-i386** switch is the default, and should be used to compile host programs that will run on the SRM. To compile programs for RX nodes, use the optional iPSC/860 C compiler, **icc**. See the *iPSC®/860 C Compiler Software Product Release Notes* for release note information on **icc**.

The following limitations and workarounds apply to the iPSC/2 C compiler:

1. **Deeply nested macros and -ansi switch cause internal compiler error**
   When macros are deeply nested (greater than 4 levels) and the undocumented and unsupported **-ansi** switch is used, the compiler quits with the following error message:

   ```
   Internal Compiler Error
   ```

2. **asm386 directive support**
   The C compiler is missing some support for the **asm386** directives.

3. **-C compiler switch**
   The **-C** compiler switch is not implemented.

iPSC®/2 Fortran Compiler

The iPSC/2 Fortran compiler, **f77**, is located in */usr/bin* on the SRM. You use this compiler to compile and link host programs and CX node programs written in Fortran.

The **f77** compiler cannot be used to compile programs for RX nodes, and no longer accepts the **-i860** switch. The **-i386** switch is the default, and should be used to compile host programs that will run on the SRM. To compile programs for RX nodes, use the optional iPSC/860 Fortran compiler, **if77**. See the *iPSC®/860 Fortran Compiler Software Product Release Notes* for release note information on **if77**.

The following limitations and workarounds apply to the iPSC/2 Fortran compiler:

1. **Output lines from write or print statement**
   Instead of using the first character in a **write** or **print** statement as a printer control character, the character is printed. Also, an extra space is added for each continuation line in these statements. Use the **-vms** compiler switch to make the **write** and **print** statements use the first character as a printer control character and to eliminate the extra space on continuation lines.

2. **Largest binary record is 65536 values**
   The largest binary record that can be written is 2**16 values, or 65536 values.
   [1162]

3. **Error messages for the following are not provided:**

   - Extra characters on a **DO** statement
   - **DATA** statements used with a variable declared in a common block, but not in **BLOCK DATA** (VMS extensions)
   - Trigonometric function whose arguments are greater than 2**63

- Same formal argument name used in a subroutine statement more than once
- Logical variable used as array index
- Integer variable used as logical (VMS extension)
- Mix of character and any other type within a common block (VMS extension)

4. **-i2 compiler switch**
   The **-i2** compiler switch (integers default to 2 bytes) returns an incorrect error message for large integer arrays, and the compilation aborts.

5. **Invalid invocation line switches**
   Invalid compiler switches are silently ignored by the compiler.

6. **VAX/VMS extensions are not enabled by default**
   Normally, the **f77** compiler does not accept VAX/VMS extensions. Compiling with the **-vms** switch makes VAX/VMS Fortran extensions available during compilation. For more information, refer to the *VAX-11 Fortran User's Guide and Language Reference Manual*, available from Digital Equipment Corporation.

7. **Running out of node board memory**
   If you run out of node board memory while running a Fortran application, you may receive one or more of the following messages:

```
Fortran runtime error on external file "" (106): Buffer too large
Stack overflow
Memory fault
```

   This can happen when making the first write to a file, as the program tries to allocate a buffer for the file.

8. **I = MOD(J, 0) causes error message**
   The expression **I = MOD(J, 0)** in a host program causes a runtime divide-by-zero exception. This generates illegal assembler instructions, causing the assembler to issue a warning as follows:

```
WARNING: Shift count isn't in 0-31 range
```

9. **Formatted print statements on multiple nodes**
   Using Fortran formatted print statements that contain carriage return control characters ("/") on multiple nodes may cause prints from multiple nodes to merge together. That is, a message line from one node may have portions of a message from another node mixed in.

10. **Node and SRM versions of rewind**
    The node and SRM versions of the Fortran **rewind()** routine only reset the file pointer to the beginning of the file. They do not shrink the file, when appropriate, as does the standard **rewind()** routine.

11. **Character string padding in Fortran etos() and stoe() routines**
The Fortran **etos()** routine pads a string with NULL characters (ASCII 0) instead of blanks (ASCII 32). Fortran **stoe()** expects strings to be padded with NULL characters, not blanks. This may cause the following unexpected behavior:

    - Using a string padded with blanks in **stoe()** may cause the following error:

      ```
      stoe: Invalid size
      ```

      Workaround: Use strings generated by **etos()** in **stoe()**.

    - String comparisons may fail. A string padded with NULLs will not match a string padded with blanks.

12. **New names for Fortran unnamed and scratch files**
The implementation of Fortran unnamed files and scratch files has changed from Release 3.2.

    - When you open a file without specifying a name in the **open** statement, the file is created in your current working directory on the host system with the name *fort.ddd*, where *ddd* is the Fortran unit number in decimal.

      Because the naming convention for unnamed files is based on the unit number alone, if multiple nodes open unnamed files on the same unit number, they will all be connected to the same file (each with its own file pointer).

    - When you open a scratch file (**status='scratch'**), the file is created with the name *FTNpppppppp.uu*, where *pppppppp* is the process number that opened the file in hexadecimal and *uu* is the Fortran unit number in decimal. If there is a CFS file system, the scratch file is created in */cfs* (or the location indicated by the *CFS_MOUNT* environment variable); if there is no CFS, the file is created in */usr/tmp* on the host file system.

      Because the naming convention for scratch files is based on the process number and unit number, if multiple nodes open scratch files on the same unit number, each has its own file.

Interactive Parallel Debugger

1. **No dimension information is available for some array arguments**
   The iPSC/2 Fortran compiler does not provide symbol table information for assumed-size array arguments (such as **array(*)**) and adjustable array arguments (such as **array(*n*)**) concerning the size of the dimension. Therefore, when debugging CX node programs written in Fortran, a size of 1 is assumed with a lower bound of 1. A warning message is displayed whenever the debugger detects this.

   With the iPSC/860 Fortran compiler, only assumed-size arrays (**array(*)**) have no dimension size information available. Adjustable arrays (**array(*n*)**) are sized correctly.

   Workaround: Use the **type** command to determine what the debugger thinks the size of an array is for a particular scope. Use a count specification to look beyond the end of the assumed size of the array and thus get access to all of the elements.

2. **Incorrect symbol table information is generated for some adjustable array arguments**
   The iPSC/860 Fortran compiler provides incorrect symbol table information for array arguments that have adjustable dimensions and only one bound that is a variable (such as **array(5:*n*)**). Using **display**, **assign** or data breakpoints will have unpredictable results.

   Workaround: Use the **type** command to determine what the debugger thinks the dimension and lower bound are.
   [2178]

3. **Incorrect results when accessing large arrays in CX node programs**
   The iPSC/2 compilers generate incorrect symbol table information for arrays larger than 64K bytes. If the array has no dimensions with more than 64K elements, but the array size is greater than 64K bytes, a warning is printed and the array size is calculated correctly. If the array has any dimensions with more than 64K elements, the condition may not be detected and any access to the array may be incorrect.

   Workaround: Don't trust results of accessing any array in a CX node program declared with more than 64K elements in any dimension.

4. **Killing program being debugged can abort debugger**
   If a program under debug is killed via **killproc()** or **killcube()**, any subsequent attempt to debug that process will cause the debugger to abort.

   Workaround: Either do not call **killproc()** or **killcube()**, or set breakpoints on these routines to intercept any calls to them.
   [1264]

5. **Display and assignment of addresses in CX node programs is inconsistent**
All application addresses on CX nodes are offset in the linear address space by 0x40000000 by NX. The debugger does not perform this translation consistently. When displaying a C language pointer variable, the offset is not included (e.g., 0x004046f4) while displaying the address of a C language variable using an ampersand (&) the offset is included (e.g., 0x404046f4).

   Workaround: When displaying the address of a variable using the ampersand syntax, subtract 0x40000000 from the value displayed. When assigning a pointer value, do not add 0x40000000. ***[1395]***

6. **Problems displaying and assigning C character strings**
The debugger has the following problems displaying and assigning character strings in C programs:

   - Using **-string** when displaying a pointer to a character string results in the address of the character string being displayed rather than the string itself.

     Workaround: Display the pointer using a subscript (such as **ptr[0]**), include a count specification indicating the number of characters in the string to display, and omit the **-string** switch.
     [2180]

   - When **-string** is used to display an array of pointers to strings, only the first string is displayed, the subscript value displayed is incorrect, and any count specification is ignored.

     Workaround: Display each element individually.
     [2181]

   - Assigning a string value to a pointer to a character string works incorrectly.

     Workaround: Access the pointer as an array and assign each element individually.
     [2182]

   - The display command's **-string** switch is only significant when displaying C character arrays or pointers to characters. It is ignored otherwise.

   - Assigning a character to a C **char** array has different results from assigning a string.

     If a character value (denoted by single quotes) is assigned to a string, the entire string (or *count* elements) will be filled with that character. If a string value (denoted by double quotes) is assigned, a single copy of the null terminated string is assigned, and any count specification is ignored.

7. **Limitations in display command**
The **display** command has the following limitations:

   - The **-complex** and **-dcomplex** switches are unsupported.
   [2183]

   - Radix conversion is limited. The switches **-hexadecimal**, **-octal**, and **-decimal** are recognized only for integer variables.
   [2184]

8. **Can't display C variables of type unsigned char**
Display of C variables whose type is **unsigned char** is unsupported.

   Workaround: Get the address of the variable using the ampersand syntax (&) and display the address. This will result in a hex dump with alphanumeric equivalent values on the right.
   [2185]

9. **type, assign, and display commands don't support C bitfields**
If a structure contains bitfield elements, any attempt to use the **type**, **display**, or **assign** command on the bitfield will result in a "member of struct/union not found" error.
[1408]

10. **Data breakpoints on RX node programs may corrupt floating point results register**
If a process hits a data breakpoint on a core instruction while a floating point operation is in progress, the floating point result register may be invalid. However, the program will run correctly. This is a problem for RX nodes only.

    Workaround: Don't trust the contents of the floating point result register when stopped at a data breakpoint.
    [2186]

11. **Extra lines in log files for commands containing semicolons**
Command lines that contain semicolons (;) are echoed to log files an extra time for every semicolon.

    Workaround: Ignore extra lines in log file.
    [1263]

12. **Braces delimiting C statements are non-executable**
Breakpoints cannot be set on non-executable lines; thus, a line which contains only an opening brace ({) or closing brace (}) in a C program cannot have a breakpoint set on it. The braces which delimit a routine are an exception to this. Setting a breakpoint on the opening brace is equivalent to setting a breakpoint on the first executable line of the routine. The closing brace is treated as an executable statement and therefore may have a breakpoint on it.

13. **A disassemble command erroneously flags some opcodes as invalid**
Some valid opcodes are not recognized by the disassembler and are displayed as invalid opcodes. The instructions which we are aware of that cause this problem are **calli**, **fmov.ds**, and **fmov.sd**.
[1394]

14. **Context limitation for CX nodes**
When debugging on CX nodes, there is a limitation when specifying a context: Every process id listed must exist for every node listed. For example, if process 99 existed on node 0 only and process 100 existed on node 0 and 1, a context of **(all:all)** would be invalid. There is no process 99 on node 1. A valid context would be **(0:all)**, all processes on node 0. Process 100 on node 1 would have to be accesses as **(1:100)** or **(1:all)**.
[1809]

15. **No alternate entry support for Fortran**
The debugger does not support Fortran alternate entry points. Once alternate entry code has been entered the debugger loses an understanding of its scope and location so that commands such as **step**, **list** and **display** cannot function.

Workaround: If you enter such code, set a breakpoint on the line which follows the call which caused you to enter the alternate entry point and use the continue command to get back into known territory.
[1826]

16. **Problem displaying Fortran assumed size character strings**
A Fortran character string which is declared with an assumed size (i.e., *CHARACTER NAME*(*)*) is displayed as a single character. A count value cannot be specified to get more characters printed as this is not considered an array.

Workaround: Get the address of the variable (e.g. **display &NAME**) and then display the address with a count value. This will result in a hex dump with alphanumeric equivalents displayed to the right.
[1976]

17. **Problems displaying contents of string pointers**
Displaying the contents of a pointer to a character string using a count value does not work. For example, a variable declared as *char *name* cannot be displayed as **display *name,4**.

Workaround: Use array syntax to specify the pointer variable (i.e., **display name [0],4**).
[1977]

18. **Displaying a variable in a multi-process context may give unexpected results**
If you are debugging in a context with multiple processes and try to display a variable which is not within the scope of all the processes in the context, you may get an unexpected result. Depending on which process(es) is within the scope of the variable, you will see either the value displayed for each process that is in the proper scope and a node error message indicating that the process is not active for those processes that are not. Or, you will see an error message that the variable cannot be found.

Workaround: Specify only the processes which are within the scope of the variable in the context of the **display** command.
[1979]

19. **Exceeding run command line length causes fatal debugger error**
When using the run command within an exec file, a list of command line arguments which exceeds 132 characters (including separating spaces) causes a fatal internal error for the debugger. Normally, it would just truncate the command line at 132 characters and ignore the rest of the line.

Workaround: Use an input file containing the data to be read if you need to provide data to the program which exceeds the 132 character limit.
[2131]

20. **Problem with setting data breakpoint on a Fortran COMPLEX number**
If you set a data breakpoint on a Fortran COMPLEX variable, the break will not occur if only the imaginary portion of the number is accessed. This is also true for the DOUBLE COMPLEX data type.

Workaround: You can set a data breakpoint on the imaginary half of a COMPLEX variable by using an EQUIVALENCE, or by calculating the address of the imaginary part using an "&" and the display command.
[2034]

21. **run command no longer allows ";"**
A syntax error will occur if the command separator ";" follows a **run** command.
[2223]

Compiler Libraries

1. **Standard I/O library may have trouble printing very large numbers**
   The standard I/O library may produce incorrect results when printing very large numbers. The only known example is 2.0**53, which does not print as an integer.
   [1369]

2. **Exceptions enabled using fpsetmask(3C) may not be reported correctly on RX nodes**
   If you use the fpsetmask(3C) function to enable ANSI/IEEE Standard 754-1985 for Binary Floating-Point Arithmetic exceptions, any exceptions that occur may not be reported correctly.
   [1165, 1381]

3. **Spurious warnings from time.h header file**
   An erroneous **typedef** in the header file *time.h* causes warning messages about "more than one type specified" and "useless typedef declaration" when programs that use this file are compiled with the iPSC/860 compilers. These warnings can safely be ignored.

RX Vector Library

1. **RX vector library is not optimized**
   The vector library for the iPSC/860 system is included in the software to provide a migration path from the iPSC/2 vector product to the iPSC/860 system. These libraries are not currently optimized to run at their peak speed on the iPSC/860 system.

Basic Math Library (libkmath.a)

1. **IPD errors when debugging programs using Basic Math Library**
   When using the Interactive Parallel Debugger (IPD) to debug a program that uses the Basic Math Library, you may encounter the following problems:

   - Inability to set breakpoints within Basic Math Library subroutines written in assembly language
   - Inability to trace the call chain to Basic Math Library subroutines written in assembly language.

2. **Two problems exist with IZAMAX**
   The **IZAMAX** routine in the Basic Math Library has the following problems:

   - If called with a length of zero or less, **IZAMAX** returns 1 instead of 0.
   - If called with an increment of other than one, **IZAMAX** returns an incorrect answer.

Remote Host

1. **Compile programs for RX nodes with icc and if77, not rcc and rf77**
   Currently, the **rcc** and **rf77** commands can only be used to compile programs for the SRM and CX nodes. You can now cross-compile programs for RX nodes using the **icc** and **if77** commands, which run locally on your Sun-3 or Sun-4 workstation or on the SRM. If your remote host is not a Sun-3 or Sun-4 workstation, you must currently rlogin to the SRM to compile programs for RX nodes.

2. **relcube may return before cube is completely released**
   If you release a cube with the **relcube** command, and then immediately try to get the same cube or an overlapping cube with the **getcube** command, you will sometimes get the message:

   ```
   (host) getcube: No SRM that matched your request was found.
   ```

   Workaround: Wait a few seconds after a **relcube** before doing a **getcube**.
   [1326]

3. **Node program and directory must be world-executable on NFS-mounted file systems**
   For the **load** command or **load()** system call to load a file on the remote host, both the file and the directory containing it must be executable by "other" if the file system containing the file is NFS-mounted.
   [1325]

4. **Cube ownership**
   Cubes that you own are not automatically released when you log out from the remote workstation. You must use **relcube** to release the cube.

5. **Maximum of six cube partitions per remote host**
   Some remote host operating systems (for example, Sun OS 3.5) support a maximum of only six cube partitions per remote host. Attempting to allocate a seventh cube may cause the remote host to hang. To recover, reboot the remote host.

6. **setenv TTY 'tty' on Sun workstations**
   The **TTY** environment variable is used by several cube commands and must be set before you use the cube. Because each newly created window on a Sun workstation inherits its parent's shell variables, you should set **TTY** in *.cshrc* and not in *.login*. However, for the remote copy command (**rcp**) to work properly, you must redirect the standard error from the **setenv TTY 'tty'** to null as follows:

   ```
   setenv TTY `tty` >& /dev/null
   ```

7. **cubeinfo's SRM field does not contain complete name**
   On a remote host, the **cubeinfo** command's *srmname* field contains the SRM name specified by **getcube -h**, which may not be the complete name.

8. **FORCE TYPE range messages**
   Message types in the FORCE TYPE range (types that are greater than 40000000 hexadecimal), sent from a node to a remote host, must be received using a **csendrecv()** or **isendrecv()** call in the host program. Forced messages sent from a node to a **crecv()** or **irecv()** on a remote host will be lost. Force messages *can* be sent from a host program to a host program, a node program to a node program, or from a remote host program to a node program and received using **crecv()** or **irecv()**.

9. **Merged error messages**
   On a remote host, iPSC/860 error messages may be merged with output from user applications.

10. **Invalid buffer pointers**
    Invalid buffer pointers in a C host program are sometimes accepted as valid and the system does not return an error message. This may cause unpredictable behavior.

11. **SSD syslog command clashes with SunOS syslog command**
    The SSD Remote Host command **syslog** has the same name as the SunOS command **syslog**. This can lead to confusion about which **syslog** command is being used.

    Workaround: If you want to be sure you are using the SSD **syslog** command, but don't want to overwrite the SunOS **syslog** command: create a special directory for SSD commands such as */usr/ipsc/bin*, install the SSD **syslog** command there, and put that directory ahead of */usr/bin* and */usr/ucb* in your search path.
    [345]

12. **Loading setuid() or setgid() calls from remote host is no longer allowed**
    Since the UID on the remote is not necessarily the same as the UID on the SRM, these calls are no longer allowed from the remote host.

Network Queueing System (NQS)

1. **Devices are not supported in this release**
   Device support has not been implemented for this release of the NQS software. The Network Queueing System Manual refers to devices, device queues, and print queues in numerous areas. While the underlying NQS software may support devices and the queue structures associated with devices, Intel has not evaluated this area of the software and does not support this feature at this time.

2. **qpr command does not work**
   Since device queues are not supported in this release, the **qpr** command does not work.

3. **Times displayed for qsub -a may be inaccurate during Daylight Savings Time**
   The times displayed by the command **qsub -a** may be an hour off if Daylight Savings Time is currently in effect. After submitting a job, verify the time to run by using the **qmgr** command **show long queue**.

4. **qsub -s should always be used with -x**
   Always use the **-x** option (export environment variables) with the **-s** option (use alternate shell) to the **qsub** command. Omitting the **-x** option will cause the default environment on the remote machine to be used, which will probably not give the desired results with the new shell.

5. **qmgr command "set mail *userid*" does not work**
   The **qmgr** command **set mail**, which is supposed to change the user to whom mail is sent when a job is executed, does not work. For example, if user *chris* uses the command **set mail scott**, user *scott* does not receive the mail. Instead, *chris* receives it.
   [1273]

6. **Invalid cubetype for remote job submission on pipe queue fails silently**
   If a job in a pipe queue asks for a cubetype that is not valid or not installed, and the user who submitted the job is on a different system from the intended batch queue, the job is deleted but the user doesn't get any indication of why it was deleted.

   Workaround: Always submit pipe queue jobs on the same system as the intended batch queue.
   [1278]

iPSC®/2 Simulator

1. **Process creation**
   The simulator creates one UNIX/XENIX process for every simulated node or host process. Therefore, the size of a simulation is limited by the maximum number of processes allowed by the operating system being used. XENIX software allows 14 processes to be created; and UNIX 4.2 BSD, UNIX 5.2 ATT, and UNIX 5.3 ATT allow 23 processes to be created. Limit the cube's dimension or the number of processes per node to conform to these limitations.

2. **CFS calls and three node calls not supported**
   The CFS calls (such as **cread()**, **cwrite()**, **iowrite()**, **iodone()**, and **iomode()**) are not supported by the simulator. The NX **handler()**, **dclock()**, and **hwclock()** calls are not supported either.

3. **Clock calls and timing**
   Note that all timing uses a common time base. Because the simulated processes are executed in sequential timeslices by UNIX software, the values from the clock are different from those obtained on the iPSC/860 system.

4. **Remote host and cube sharing**
   Remote host and cube sharing functions are not supported by the simulator. Related calls (such as **getcube()**) should be commented out before you compile.

5. **System Resource Manager (SRM) programs**
   System Resource Manager programs are started within the simulator. Thus, the command line cannot be used to supply arguments to the SRM programs or redirect their standard input or output.

6. **File descriptors**
   File descriptors 9, 10, 11, and 12 are reserved for the simulator.

7. **Signals**
   Signal number 28 in the UNIX 4.2 BSD environment; and signal number 16 on XENIX, UNIX 5.2 ATT, and UNIX 5.3 ATT are reserved for the simulator.

8. **Interchanging calls**
   The simulator accepts all host and node calls in both host and node programs, whereas the iPSC system does not. Therefore, it is possible to write simulator programs that will not run on the iPSC system. Refer to the *iPSC®/2 and iPSC®/860 Programmer's Reference Manual* for more information on the calls supported on the host and nodes.

UNIX Software

1. **Using the right shell for a script under csh**
   If you use the C-shell (**csh**):

   - To make a script execute under **csh**, make the first two characters of the script a pound sign and an exclamation mark (#!).
   - To make a script execute under **sh**, make the first character of the script a colon (:).

   If the first character is neither a pound sign nor a colon, the script executes under **sh**.

2. **Scripts in sh always run under sh**
   If you use the Bourne shell (**sh**), all scripts execute under **sh**. To execute a script under **csh**, use the command **csh -c script**.

3. **unsetenv and ulimit not supported**
   The **unsetenv** command is not supported in UNIX System V version 3.2. The **unsetenv** and **ulimit** built-in commands are not included in the **csh**.

4. **Duplicate symbol defined in include files**
   The symbol `SYMESZ` is defined in both */usr/include/syms.h* and */usr/include/ldfcn.h*. Including both of these files in your application causes a compiler warning.

5. **c-shell will core dump with set long PATH variables**
   The c-shell will core dump when long PATH variables are set. The current limit is 128 characters.

TCP/IP on the SRM

1. **telnet problem**
   telnet will not work from a shell layer (**shl**(1)).

Cube Diagnostic Program (CDP)

1. **SAT help descriptions scroll off screen**
   Some help descriptions in the System Acceptance Test scroll off the top of the screen without pausing.

   Workaround: Use `<Ctrl-S>/<Ctrl-Q>` to control scrolling.
   [1202]

2. **SAT cannot test mixed cubes**
   The System Acceptance Test cannot currently test a cube that contains a mixture of CX and RX nodes.
   [729]

3. **CDP loops on 387 tests when running optional tests forever**
   CDP will run the first test from the optional hardware test menu forever and will not circulate to the other tests in the menu.

   Workaround: Run the cdp from the main menu.
   [2142]

4. **Interrupting disk test kills CDP**
   Interrupting the hard disk optional hardware tests usually kills CDP and returns you to the shell prompt.

   Workaround: Let the test finish to completion.
   [2144]

5. **Hard disk drive test does not ignore mapped-out disks**
   If some disk drives are commented out in devconf, the optional hardware hard drive test for Drive ID fails on the mapped-out disks identifying that cdp failed to get a null ID string from the disk.

   Workaround: Set the options to continue on error and ignore the Drive ID errors for the mapped-out drives.
   [2171]

6. **Advanced disk utilities format causes reduced disk space**
The format routine under the Advanced Utilities Menu with cdp's Optional Hardware / Hard Drive Menu uses different parameters other than those set at the factory.

   Workaround: Use the Disk Scrubber to format disks.
   [2188]

7. **The optional board test for I/O fails on the Panther Drive**
If there are more than two panther drives on one I/O node, the optional board test fails with the following errors:

```
slot xxx:  I/O:  Media Test ERROR
Interrupt timeout

slot xxx:  I/O:  Media  Write/Read Test ERROR
Interrupt timeout

slot xxx:  I/O:  Worst Case Seek Test ERROR
wait-int-status xxxx
```

   Workaround: You may ignore the error messages unless you have problems with the panther drives. In this case, please report any problems you encounter to SSD Customer Support Response Center at:

   1-800-421-2823 (Customer Support Response Center)
   Your Local Intel Sales Office (in Europe)
   support@ssd.intel.com (Internet address)

Performance Analysis Tool (PAT)

1. **Graphical glitches in xtool and ctool**
Clicking on button borders causes graphics glitches in **xtool** and **ctool** and **etool** under Sunview.
[1857]

2. **xtool, ctool show wild (large) counts for hsend, hrecv handlers**
Data collected on code called from interrupt handlers is often incorrect. These tools do not support monitoring of interrupt driven code.
[1859]

3. **xtool shows nonsense graphics for source code histogram**
The source code histogram graph contains loops under some conditions. These loops should be ignored.
[1860]

4. **ctool shows inconsistent time for flick()**
   The percentage of time in **flick()** as given in **ctool** does not agree with the percentage of time in **flick()** given in **xtool**. This occurs because there are two different methods of collecting the data. In most cases the discrepancy should be within reasonable bounds.
   [1958]

5. **etool displays events out of order**
   In some cases etool displays events on separate nodes slightly out of order. For example: A message is sent from node 1 to node 2, but **etool** shows that node 2 receives before the node 1 send has completed. This is caused by the differences in time basis on each node and the accuracy of node synchronization.
   [2026]

6. **xtool omits some source lines from assembly view**
   xtool omits some of the source lines from its assembly view. One type of source line that is omitted is a closing brace other than at the end of a function.
   [2064]

7. **xtool histogram usage display traces not aligned with instructions**
   The histogram display in **xtool** for source and assembly lines is slightly out of alignment with the instruction.
   [2079]

8. **ctool hardcopy takes user out of "Usage vs. Node" display**
   If a hardcopy is requested from inside the "Usage vs. Node" display of **ctool**, the "Usage vs. Node" display is exited.
   [2089]

9. **PAT tools report undefined symbol**
   PAT tools report an undefined symbol when they start. This message should be ignored.
   [2140]

10. **Node selection menus wrong for cubes with more than 16 nodes.**
    In the node selection menus, the "Next Page" and Prev. Page" buttons falsely checkmark all the node buttons and fail to relabel the buttons for the newly selected page. The checkmarks should be ignored, and the proper node labels appear when individual node buttons are selected.

Software Enhancements 5

Introduction

This chapter describes all bugs fixed for the following Release 3.3.1 system software components:

NX Node Executive
Node Shell
Concurrent File System™
Concurrent File System™ Tape
Remote Host
PC 586 Driver
TCP/IP on the SRM
Cube Diagnostic Program (CDP)
Performance Analysis Tools (PAT)
Interactive Parallel Debugger (IPD)
RX Profiler

NOTE

Read the following sections carefully. Report any problems you encounter while using your iPSC system to SSD Customer Support Response Center at:

1-800-421-2823 (Customer Support Response Center)
Your Local Intel Sales Office (in Europe)
support@ssd.intel.com (Internet address)

NOTE

Most of the following Software Enhancements contain a number in brackets at the end of each Enhancement. That number represents Intel's internal tracking system. Please refer to that number when refering to a Enhancement documented in this chapter.

NX Node Executive

1. **DCM driver loss of message headers reduced**
   DCM driver may lose a message header which causes the system to hang. The DCM driver has been improved to reduce this problem.

2. **bootcube now checks for another bootcube process**
   bootcube has been fixed to check whether another **bootcube** is in progress. To continue with **bootcube** regardless, use **bootcube -f**.
   [1535]

3. **bootcube checks on a FTP session**
   bootcube has been fixed to determine whether a FTP to the cube is active. **bootcube -f** may be used if **bootcube** should proceed with no checking.

4. **getcube no longer returns "Not available" error message inappropriately**
   commser has been fixed to properly clean up the internal tables.

5. **Race condition between handler used in I/O mode 1 and user messages**
   CFS I/O mode is fixed to bypass user messages.

6. **Can now use performance port**
   NX operating system has been fixed to allow writing to the performance port.
   [1972]

7. **plogon does not detach from the current cube**
   plogon now preserves the current cube.
   [2017]

8. ***iocube* no longer attached to the window where the plogon command executed**
   plogon command detaches from *iocube* when it is done.
   [2018]

9. **Account_hook routine's return value now has meaning**
   If the account_hook routine returns a non-zero value, it is interpreted as an error number (*errno*).
   [1902]

10. ***inetd* no longer dies soon after bootcube, due to spawning on wrong node**
    inetd is no longer spawned on a customer-specific VME service node.
    [2043]

11. **relcube no longer releases more cubes than intended with -c switch**
    relcube -c on the SRM no longer releases remote host cube. To release a remote host cube from the SRM, use **relcube -f**.
    [2067]

12. **getcube and newserver retry to fork fserver**
    getcube and newserver will retry the failed fork until they are successful.
    [1983]

13. **Incorrect "PID in use" message no longer occurs.**
    This message occurred when a process running on the cube was not properly terminated. The NX operating system has been updated to prevent this problem.

14. **killcube no longer hangs when messages do not flush properly**
    Host messages get flushed properly to prevent **killcube** from hanging.

15. **DCM driver caused panics eliminated**
    Changes have been made to the DCM driver to prevent corruption of the message header queues which resulted in SRM panics.

16. **System console no longer freezes**
    Under some circumstances the DCM driver could go into an infinite loop or it could cause the system to go into an idle state. This problem has been fixed.

17. **Communication server hangs reduced**
    The *commser* process no longer hangs trying to send a message to another host process when the DCM driver is out of buffers.

18. **bootcube panics eliminated**
    Changes have been made to the DCM driver which have eliminated panics during the bootcube process.

19. **`Commser not responding` messages reduced**
    This message occurred on the SRM as a result of *commser* sending cube information to the cube for process logging. The communications server has been updated to prevent this problem. This fix may cause process logging information to be out of order in the process log file.

20. **Multiple file servers no longer created on the SRM**
    Multiple file servers attached to the same cube no longer occur. This problem may have caused output to go to the wrong window.

21. **Losing a cube partition problem reduced**
    The NX operating system has been updated to minimize this problem.

22. **Loading process improved**
The *loader* functionality is now supported in the library, therefore, the SRM can now directly load a cube without going through the *loader*. This reduces the number of *loader* hangs.

23. **killcube no longer hangs when deleting the loading of a large file**
Messages from *loader* are processed properly to prevent **killcube** from hanging.

24. **getcube returns "no more buffers" but allocates a cube**
If a lot of messages are being sent to the host and have not yet been received, the host's buffers can fill up. If you attempt to get a cube using **getcube** while the buffers are full, it gives the error message "no more buffers," but it will no longer allocate a usable cube.
[1317]

Node Shell

1. **nsh core dumps if a long path name is used**
CFS **nsh** now supports long pathnames.
[684]

2. **Commands under nsh on ENET service node fails with the message "No more processes"**
The **adminproc** daemon is fixed to clean up the process table entry when the ftp to the ENET node is completed.

Concurrent File System™

1. **New CFS *nameproc***
This *nameproc* allows the creation of more than 55K files. It also increased CFS file system performance.

2. **New file checkers**
The file checkers now run on all the compute nodes in parallel producing increased performance for CFS file systems containing a large number of files.

3. **CFS corrupted header problem fixed**
Under some circumstances, file header information in the CFS could be corrupted or destroyed. This problem prevented file access and caused errors from the CFS file checkers. This problem has been fixed.

Concurrent File System™ Tape

1. **Cube killed when accessing a tape drive**
   Cube would be killed when accessing a tape drive. Subsequent requests for the tape device would be rejected because the open connection to the tape device was not properly closed by the **killcube**. This has been fixed.

2. **CFS tape problems fixed**
   The following CFS tape problems have been fixed:

   - Improved **<Ctrl-C>** handling during **tar** operations.
   - The end of device is now detected properly.
   - The tape device cannot be opened more than once at the same time.
   - The tape open connection errors generated during a **tar** have been cured.

Remote Host

1. **The cusrid() call hangs under remote host**
   Use **getenv**("RUSERNAME") instead of **cusrid()**, since the **load()** command is changed to copy the user name into the environment.
   [1803]

2. **If getcube() fails from the remote host, the cube is now removed**
   The remote host software has improved to reliably clean up the remote host *commser*.
   [1786]

3. **A getcube() call after a getcube() and load() call is successful from the remote host**
   Remote host **getcube** is fixed to handle multiple **getcubes**.
   [2016]

4. **getcube() with long host names succeeds**
   Remote host now supports SRM network host names longer than 16 characters in length.

5. **UID between remote host and CFS is now correct**
   UID for the remote host user is now obtained from the SRM */etc/passwd* file. Now all the remote host users are required to have an entry in the SRM */etc/passwd* file.

6. **Cube cannot be allocated from a remote system before bootcube is finished**
   The *rcam* daemon now checks whether a **bootcube** is in progress before it allows any remote host commands.
   [1732]

7. **Remote host load command can load programs from CFS**
   Remote host commser and load command are changed to support the loading of CFS files.

8. **Interrupting remote host load no longer causes *rcam* to hang**
   The *rcam* daemon has been fixed to handle a broken socket.

9. **Remote host *commser.log* and *boot_lck* files now unique on system with common NFS directory**
   These file names are made unique by utilizing remote host names.

10. **Improved remote host across gateways**
    Remote host can now get and use a cube through gateway and other network domains.

11. **Reduced `Commser not responding` message on remote host**
    The **getcube** command has been improved for the remote host *commser* to reduce occurrences of the message `Commser not responding` on the remote host.

12. **Remote host no longer creates duplicate cubes**
    The *commser* has been fixed so it no longer creates duplicate cubes on the SRM using remote host.

13. **Triplicate *commsers* eliminated on remote host**
    The remote host used to create triplicate *commsers*. This problem has been fixed.

14. **Remote host application no longer hangs on small message**
    Remote host applications used to hang while receiving a message if the message was smaller than the receiving buffer. This problem has been fixed.

PC586 Driver

1. **Changes in the PC586 driver for the new PC586 board.**
   One of the visible differences in the new PC586 driver is in the driver sign-on message at boot. The new sign-on message appears as follows:

```
PC586 v2.9 Copyright (c) 1987, 1988, 1989, 1990 Intel Corp., All Rights Reserved
PC586 board 0 was found --- Ethernet Address: 00:aa:00:03:6e:cf
```

The old one was:

```
PC586 v2.4 Copyright (c) 1987, 1988, 1989 Intel Corp., All Rights Reserved
PC586 board 0 was found, ethernet id = 00:AA:00:03:6E:CF
```

Another difference is that the **netstat(1)** command will now display internet statistics kept in the driver. When the **-i** or **-I** option to the **netstat(1)** command was used with the old driver, the user would get the following message:

```
Name    Mtu     Network     Address      Ipkts   Ierrs  Opkts   Oerrs  Collis
en0     1500    mfg         aslan        No Statistics Available
lo0     2048    loopback    localhost    1229    0      1229     0        0
```

The user will get full statistics with the new driver:

```
Name    Mtu     Network     Address      Ipkts   Ierrs  Opkts   Oerrs  Collis
en0     1500    mfg         aslan        144373  0      52175   0      493
lo0     2048    loopback    localhost    1229    0      1229    0      0
```

There is one more visible difference with the new PC586 driver. It now warns the user when the Ethernet cable is not attached. These warnings are displayed on the console and look like this:

```
WARNING: PC586 transmission link error - check cable.
```

This may be annoying if the user does not want the SRM attached to the Ethernet, although it can be disabled.

To stop the route daemon from sending, edit */etc/tcp* and change the startup of routed to include the **-q** (for quit) option. This is at line 130 of the file.

TCP/IP on the SRM

1. **TCP/IP panics eliminated**
   The TCP/IP driver has been fixed to eliminate panics.

Cube Diagnostic Program (CDP)

1. **Cube diagnostic program now checks whether a bootcube is in progress**
   The cdp command checks whether a bootcube is running and creates a lockfile to prevent bootcube from killing it.
   [1736]

2. **The cube diagnostic program can now load optional hardware tests on D7**
   The cdp command now uses different methods for loading the optional hardware tests onto the I/O nodes.

Performance Analysis Tools (PAT)

1. **Mis-labeled axis in xtool corrected**
   The "calls v. node" graph in **xtool** had both axes labeled "number of calls". The bottom axis is now labeled "node number".
   [1839]

2. **xtool graphics now match text (-p) output for call counts**
   Under some circumstances **xtool** would report incorrect call counts. This has been fixed.
   [1842]

3. **ctool now shows "xxx vs. node" graphs properly**
   Some graphs in **ctool** were blank even though there was data to display. The graphs are now displayed.
   [1845]

4. **xtool no longer uses background color for 7th plot**
   Some of the lines and fill colors for graphed data were in the background color so they did not show up on the graph. This has been corrected.
   [1853]

5. **xtool and ctool no longer show three nodes for d1 cube for "xxx vs. node" graphs**
   In some cases **xtool** and **ctool** displayed multiple 0 nodes in the "xxx vs. node" graphs. The correct number of nodes is now displayed.
   [1855]

6. **etool now properly tracks objects added or removed**
   etool would sometimes loose track of the data being analyzed. This has been corrected.
   [1858]

7. **The "Show Detail" button in etool now removed**
   The "Show Detail" button which had no effect, has been removed from **etool.**
   [1865]

8. **etool graphics mode can display more than seven labels**
   etool would not display more than seven labels. A scrolling label window has been added to allow more labels to be viewed.
   [1874]

9. **ctool "Erase" button removed**
   ctool's "Erase" button, which functioned incorrectly, has been removed.
   [1905]

10. **ctool "usage vs. node" graph total is corrected**
    Under some circumstances **ctool**'s "usage vs. node" graph displayed incorrect data for the total. This problem has been corrected. **flick()** time is now measured more consistently by **ctool.**
    [1907]

Interactive Parallel Debugger

1. **Missing .file directives in assembly language routines are now handled by IPD**
   Previously, a "multiple text records" error message would be displayed and further reading of symbols would be stopped if a .file directive was missing. Now a .file directive is simulated using a file name of the form "unknown#.s" (where # is a number beginning with 1) and the reading of symbols continues. This dummy file name can be used just as any other file name. Also, its routines and variables can be accessed symbolically (provided that the file was compiled with debug information.
   [1802]

2. **Display's -alpha switch working correctly**
   The **-alpha** switch for the **display** command now converts the contents of Fortran variables in RX node programs to an alphanumeric character rather than a boolean.
   [1797]

3. **Breakpoint display of non-executable lines corrected**
   Previously, setting a breakpoint on a non-executable line prior to the first executable line of a routine would be accepted and displayed as such but the actual break would not occur until the first executable line was reached. Now the display shows where the breakpoint will actually occur (i.e. at the first executable line). Attempting to set a breakpoint on any other non-executable line will fail with an error message.
   [1172]

4. Problems with set and alias resolved

   - A set variable can be used multiple times in a command line without causing the debugger to abort.
     [1227]

   - A recursive alias definition displays an error message rather than aborting the debugger.
     [1361]

   - A set variable can be used within an alias definition without causing an internal debugger error.
     [1390]

5. **Break -after switch now recognized**
   When setting a breakpoint on a procedure name, the **-after** switch is no longer ignored.
   [1258]

6. **Step and breakpoint conflicts resolved**
   Several instances where a step would fail if a breakpoint existed inside a routine or on returning from a routine have been fixed.
   [1460]

7. **Data breakpoint on heap is non-fatal**
   Running a program that has had a data breakpoint on the heap of a previous run results in an informational message rather than a fatal debugger error.
   [1559]

8. **Internal assertion failures resolved**
   Executing some debugger commands (e.g., **list**, **break**, and **display**) in a multi-process context when the processes are stopped in different routines used to result in an internal assertion failure. This has been fixed.
   [1737, 1139]

9. **Application output now flushed to *stdout***
   ipd is now flushing output to stdout. This fixes a problem where an application's read call would appear to be executed prior to the printing of the prompt which preceded it.
   [1953]

10. **Adjustable size array arguments information now correct after a break**
    Accessing adjustable size array arguments (e.g. *real a(n)*) immediately after breaking on the Fortran routine name now works correctly.
    [2035]

11. **Address assignment now defaults to 4 bytes**
    Assigning a value to an address used to result in 16 bytes of data being written starting at the given address. Now a default length of 4 bytes is written unless a length switch is specified.
    [2086]

Enhancements to IPD

12. **load command has default context**
    The **load** command no longer requires that a context be specified by the user. It will default to a context of **(all:0)** if the context is omitted.

13. **Command line arguments supported**
    The **run** command now supports command line arguments. A new command, **rerun**, was added to make the functionality complete.

14. **Stepping one instruction now available**
    Machine instruction single stepping is now supported. This facility is invoked by the **-instruction** switch on the **step** command.

15. **More information now displayed by the status command**
    The IPD version number is now included in the **status** command's display.

16. **Easy access to Fortran variable addresses**
    The address of a Fortran variable can now be displayed by prefixing the variable name with an & (ampersand), just as the C language allows.

RX Profiler

1. **Large programs can now be profiled**
   prof860 can now be used to profile programs larger than 128K bytes.

2. **prof860 profiling information is dumped properly**
   Under some circumstances the **prof860** profiling information was deleted. This problem has been fixed.

REMOVE TCP : CAN'T FIND /usr/include/sys/socket.h
/if.h