3.3 Release Notes:
Operating System

This file contains release notes for the 3.2, 3.1, and 3.0 releases of the operating system. There were no changes for Release 3.3. Items specific to or introduced in Release 3.2 are listed first, followed by the 3.1 and 3.0 notes.   Some information that would be useful to users or administrators has been moved to **/NextLibrary/Documentation/NextAdmin/ReleaseNotes/OperatingSystem.rtf**.

Notes Specific to Release 3.2

New Features

The following new features have been added to the operating system since Release 3.1.

- Building POSIX applications is now supported.   To compile POSIX 1003.1 conforming source, specify the **-posix** flag to **cc**. Note that POSIX kernel interfaces are not supported for NEXTSTEP applications.

Known Problems

The following bugs have appeared since Release 3.1.

Reference: 37706

Problem: **siglongjmp** is not POSIX conforming under NEXTSTEP for Intel Processors

Description: **setsigjmp/siglongjmp** offers the option of not restoring the signal mask when the long jump occurs. On Intel systems, the signal mask is always restored.

Workaround: None.

Reference: 37707

Problem: On NEXTSTEP for Intel, the serial driver doesn't notify the tty driver of parity errors

Description: It needs to do this for POSIX compatibility.

Workaround: None.

Bugs Fixed in Release 3.2

These bugs have been fixed in Release 3.2:

Reference: 36695, 36696

Problem: The **zs** driver didn't return proper error codes with incorrect ioctls or certain line disciplines

Description: Improper return of OK status codes by the **zs** driver caused various problems with programs attempting to use it. This has been fixed.

Notes Specific to Release 3.1

New Features

The following new features have been added to the operating system since Release 3.0.

- Support for Intel hardware. See the Driver Kit release note for information on writing drivers for Intel hardware.

- Two new Mach virtual memory functions, **vm_deactivate()** and **vm_set_policy()**. See the

Mach operating system documentation for information about these functions.

- The tty driver has been rewritten to support all features required for POSIX 1003.1. While the termios interface defined by POSIX is not available to non-POSIX applications, all features of the driver can be used through the ioctl interface. See the header files **sys/ioctl.h** and **sys/termios.h** (under **/usr/include/bsd**) and the **termios**(4p) man page for information about these features. Also, the tty driver now handles special characters properly, allowing 8-bit input characters and 8-bit special characters. A special character is disabled by setting it to 0xff.

Known Problems

The following bug has appeared since Release 3.0.

Reference: 34051

Problem: On Gateway PCs, leaving the NMI mini-monitor can cause the keyboard to hang.

Description: Entering the mini-monitor with Alt-Num Lock and hitting 'n' to continue will hang the keyboard on the second or third try.

In Release 3.1, building POSIX applications isn't supported. POSIX will be supported in a future release.

In Release 3.1, the behavior of the new tty driver may differ from what is expected in a number of ways:

- The tty driver now uses some of the higher bits (above the low byte) to encode framing and parity error conditions. In the event that a device driver passes an 8-bit character that is then sign extended, this might be a problem.

- In pty packet mode, only the state associated with the old interface is passed in a TIOCPKT_IOCTL packet. Additional state associated with the new driver is ignored.

- Using both old and new ioctls from the same application can lead to unexpected behavior. In general, old ioctls should override the behavior of new ones, though not all cases have been tested.

- Output delays are disabled when character size is 8 bits.

- The original tty driver recognized certain input special characters (e.g., ^Z, ^W) only in the new-tty line discipline. This is no longer the case.

- Output delays are disabled when character size is 8 bits ((c_flag & CSIZE) == CS8).

- The break condition used to be translated (by the device driver) into an interrupt character. It is now implemented differently, with the effect that the action does not depend on the existence of an interrupt character (which can be undefined), or on the flag NOFLSH. It also no longer echoes the interrupt character.

Bugs Fixed in Release 3.1

These bugs have been fixed in Release 3.1:

Reference: 26111

Problem: Tty doesn't always open and close properly

Description: A bug in the tty driver sometimes prevents a program from closing and then and re-opening a device.

Reference 28909

Problem Mach threads are not recovered under certain circumstances.

Description Mach threads are not recovered under certain circumstances (such as using telnet in a particular way). The threads appear as <mach-task>s in a ps listing apparently owned by **root** and have a process id of -1. Letting these task proliferate can eventually lock the Workspace. Killing these tasks can hang the machine. The bug appears to be induced by the use of select on asynchronous media.

Reference: 31114

Problem: The 3.0 serial driver doesn't handle the new baud rates properly

Description: When some new baud rates were added for 3.0, all the related tables were not updated which causes problems at the new baud rates.

Reference: 31270

Problem: Incorrect handling of writeback data on memory exceptions

Description: On the 040 the kernel needs to retry writes after a user level exception handler fixes a page fault (due to the 040 exception continuation model). The kernel also needs to properly handle page faults that occur during handling of the 040 "writeback" registers.

Notes Specific to Release 3.0

These notes were included with the Release 3.0 version of the operating system. Sections that are no longer relevant have been marked with an italicized comment.

This file contains information about the following topics:

- Additional features in the Mach Interface Generator (MiG)
- A new location for **errno**
- The obsolescence of a Bootstrap Server function
- Enhancements to the sg (SCSI generic) driver

Mach Interface Generator (MiG)

New Features

The following are new features that have been added to MiG since Release 2.0.

- MiG now supports the **-sheader** option to make a header file suitable for inclusion in servers is generated. This includes server-side names for routines as well as the *_server()* procedure prototype. The header file is ifdefed using the format *("_%s%s", ServerPrefix, SubsystemName);* this means that server-header files and user-header files are mutually exclusive if *ServerPrefix == UserPrefix* or there is neither *ServerPrefix* nor *UserPrefix*.

- MiG now supports send timeouts. Send timeouts are simmilar to wait timeouts, except they pertain to the msg_send side of an RPC. They can be used in the same way as the

WaitTime keyword to specify a *SND_TIMEOUT* option to a *msg_send* or *msg_rpc* either from an argument to the function or as a global for the interface.

- MiG will optionally generate a handler style interface in place of a server style interface. This is the mechanism prefered by the kernel loader. Specifying the **-handler** argument in place of the **-server** argument will cause a handler interface to be generated into the specified file name. This will also modify the behavior of the **-sheader** file to reflect the handler interface.

- MiG outputs maximum Request/Reply message sizes in the server header file. This can be used in servers to preallocate the appropriate input or output message buffers.

New Location of Errno

The extern declaration of **errno** has moved from **<stddef.h>** to **<errno.h>** (under the **ansi** directory of **/NextDeveloper/Headers**). The change was made to conform to the ANSI definition.

Bootstrap Server Functions

The **bootstrap_get_unpriv_port()** function is obsolete. You should use **bootstrap_subset()**

instead.

Generic SCSI Driver

New Features

The sg driver has been enhanced to control up to 4 devices, at /dev/sg[0123]. Previously, it could handle only one open device at a time. See the sg(4) man page for more information.

Known Problems

- In the 2.0/2.2/3.0 system the setuid/setruid/seteuid, and setgid/setrgid/setegid system calls are not thread safe. If one thread is performing file system activity such as a stat(), open(), link(), access()... which use the credentials structure, and one of the setuid() calls is made in a second thread, there is the potential to corrupt the system and cause a panic. To avoid this, implement thread-locking so that calls to procedures which change credentials (setuid(), setruid(), etc.) cannot overlap calls which reference the credentials structure (e.g. file system access routines).

- In the case where a program is executed on a client machine and then rebuilt on a server, if the program is immediately executed on the client after the build, the changes may not be seen due to caching of data on the client side. To avoid this, re-exec the program, as there is a timeout for cached NFS file attributes.

- If a file is being written on client A, and read on client B, changes appended to the file from client A after client B opens the file will not be seen by client B. This is only true for the read() system call. Changes can be seen if the file is remapped via the map_fd() system call. Use map_fd(), and check for changes to the file via fstat(). If the file has changed, remap.