

NOTE: This Technical Note has been [retired](#). Please see the [Technical Notes](#) page for current documentation.

Technical Note PT18

IOP Based Serial Differences Under A/UX

CONTENTS

[Bad Character Bits Are Not Passed to an Application](#)

[Break Always Returns a NUL to an Application](#)

[Multiple Errors in a "Chunk" of Characters Are Not Reported](#)

[The Current "DMA Hang" IOP Code Patch is Incomplete](#)

[IIfx Serial Switch cdev](#)

[References](#)

[Downloadables](#)

This Technical Note discusses use of the Macintosh IIfx IOP-based serial driver under A/UX, especially under certain error conditions which cause it to perform differently than documented in `termio(7)`. (The SCC driver, used on non-IOP machines, conforms to `termio(7)` in all cases.) References to "the driver" herein refer to the IOP-based serial driver, `seriop.c`.

[Aug 01 1990]

Bad Character Bits Are Not Passed to an Application

Because the IOP does not return the bits of a character which is received with a parity error, the A/UX driver always returns NUL to an application. Applications which use `PARMRK` mode to recover a character with a parity error always think the character was a NUL.

[Back to top](#)

Break Always Returns a NUL to an Application

Because the IOP always returns a NUL character when a break occurs, the A/UX driver passes it along to an application. There is no way for the driver to determine that the NUL is superfluous. Other break processing is unaffected. Programs which use `IGNBRK` mode receive the unexpected NUL in the data stream.

[Back to top](#)

Multiple Errors in a "Chunk" of Characters Are Not Reported

The IOP only reports parity and framing errors on the first such occurrence in its internal buffer. The A/UX driver always reads characters in "chunks" from the IOP; therefore, only the first error is reported and subsequent errors go unnoticed. For example, assume an application sets `PARMRK` mode, expects to read a 10-byte packet (call it "ABCDEFGHJ"), and four parity errors occur during transmission of the packet. (The received data is "ABxDExxHxJ," where the x characters are parity errors). The IOP returns six valid data characters, marking only the first bad character ("ABxDEHJ"). The subsequent errors go unreported, simply causing missing characters. The A/UX driver then marks the bad character (marking it as NUL as previously described), and returns "ABmmxDEHJ" (where mm are the 0xff 0x00 marker bytes) to the application.

This situation causes two problems:

- Since only nine bytes, and not 10, are returned to the application, it is possible for the application's read to block permanently. This would occur if the application simply issued a 10-byte read request.
- The application has no way of knowing that several characters were dropped.

Since the timing of the A/UX driver's "chunk reads" and the arrival of data can vary, there is no way to predict or prevent the occurrence of this problem.

[Back to top](#)

The Current "DMA Hang" IOP Code Patch is Incomplete

The existing IOP code has been patched, both under the Macintosh OS and A/UX, to code around a bug in its DMA logic. Should any errant characters be received during servicing of a DMA operation, the IOP silently discards them, never reporting any error. This can cause unreported dropping of characters. Since this situation is timing-dependent, there is no way to predict or prevent the occurrence of this problem.

[Back to top](#)

IIfx Serial Switch cdev

The IIfx Serial Switch cdev does not, itself, work under A/UX. If users need to enable "Compatibility" mode, they should do so first under the Macintosh OS. A/UX, upon booting, honors the switch setting in parameter RAM. Refer to M.HW.MacIIfx for more details on the IIfx Serial Switch cdev.

References

A/UX System Administrator's Reference

[M.HW.MacIIfx](#)

[Back to top](#)

Downloadables



Acrobat version of this Note (K).

[Download](#)

Technical Notes by [Date](#) | [Number](#) | [Technology](#) | [Title](#)
[Developer Documentation](#) | [Technical Q&As](#) | [Development Kits](#) | [Sample Code](#)