

MCR10191
Document the FNP I/O address space
(FIRST_TTY_BUF_PAGEX, tty_buf.absorig)
Revision 1.0

Eric Swenson

October 13, 2026

1 Problem Description

[Ticket #88](#): while debugging FNP (IBM 3270) support in the emulator, much time was lost working out what `FIRST_TTY_BUF_PAGEX` and `tty_buf.absorig` mean, and how the addresses `dn355` puts in mailboxes (including the “DCWs” in accept-input mailboxes) are interpreted.

- `dn355_data.incl.pl1` already has a table of the FNP’s I/O page table layout, but does not explain how the constants relate to it.
- `tty_buf.incl.pl1` describes `absorig` as the “abs address of this seg”, which it is not: `fnp_init` sets it to `1024 * FIRST_TTY_BUF_PAGEX` (octal 200000), the address of `tty_buf` in the FNP’s I/O address space.

2 Proposed Fix

Comments only; no code changes.

- `dn355_data.incl.pl1`: after the layout table, a paragraph on `FIRST_BOOTLOAD_PAGEX` and `FIRST_TTY_BUF_PAGEX`. They are page numbers in the table, filled in by `fnp_util`. Word `W` of `tty_buf` is at I/O address `1024 * FIRST_TTY_BUF_PAGEX + W`, which is what `tty_buf.absorig` holds the base of. Every `tty_buf` address given to the FNP (`sub_mbx.data_addr`, output DCW lists, accept-input buffer addresses) is computed this way and mapped by the IOM through this page table. Addresses from the FNP are turned back into offsets by subtracting `absorig`.
- `tty_buf.incl.pl1`: the `absorig` comment now says it is the address of the segment as the FNP sees it, not an absolute address.
- `dn355.pl1` (`process_accept_input`): a comment that the “DCWs” filled into the accept-input mailbox are read by the FNP, and that their addresses are in the FNP’s I/O address space.

The changes can be seen below:

```
cpa >ldd>incl>dn355_data.incl.pl1 ==
```

Inserted in B:

```
B166      /**** FIRST_BOOTLOAD_PAGEX and FIRST_TTY_BUF_PAGEX are page numbers in
B167          the table above; fnp_util fills in those pages of the FNP's page
B168          table (get_page_table, fill_bootload_page_table). The FNP sees
B169          word W of tty_buf at I/O address FIRST_TTY_BUF_PAGEX * 1024 + W
B170          (200000 octal + W). fnp_init stores FIRST_TTY_BUF_PAGEX * 1024 in
B171          tty_buf.absorig, and every tty_buf address given to the FNP is
B172          tty_buf.absorig + the word offset in tty_buf: sub_mbx.data_addr,
B173          the DCWs of an output DCW list (dn355 process_send_output), and
B174          the buffer addresses filled into an accept-input mailbox
B175          (dn355 process_accept_input). These are not absolute memory
B176          addresses. The FNP uses them in the transfers it asks the IOM to
B177          do on its direct channel, and the IOM maps them through this
B178          page table. Addresses coming back from the FNP are converted to
B179          tty_buf offsets by subtracting tty_buf.absorig. */
B180
```

Preceding:

```
A166      declare FIRST_BOOTLOAD_PAGEX fixed bin init (4) int static options (constant);
```

Comparison finished: 1 difference, 15 lines.

and in:

```
cpa >ldd>incl>tty_buf.incl.pl1 ==
```

```
A37          2 absorig fixed bin (24),                                /* abs address of this seg
/*
Changed by B to:
B37          2 absorig fixed bin (24),                                /* address of this seg as
the FNP sees it, in its
B38                                                  I/O page table: not an
absolute address
B39                                                  (see dn355_data.incl.pl1)
*/
```

Comparison finished: 1 difference, 4 lines.

and in:

```
cpa [lpn dn355.pl1] ==
```

Inserted in B:

```
B982                                                  /* These "DCWs" are read by
the FNP, not the IOM;
B983                                                  the address is in the
FNP's I/O address space
B984                                                  (see dn355_data.incl.pl1)
*/
```

Preceding:

```
A982          prev_blockp = blockp;
```

Comparison finished: 1 difference, 3 lines.

3 Documentation

The changes are documentation in the source.

4 Test

None needed beyond checking that `dn355` still compiles (only a comment changed); no object changes. The include file changes do not require recompiling anything.

5 Files Changed

<code>>ldd>include</code>	<code>dn355_data.incl.pl1</code> , <code>tty_buf.incl.pl1</code>
<code>bound_355_wired.s.archive</code>	<code>dn355.pl1</code> (comment only; rebinding not needed)

6 Version History

2026-10-10	1.0	Initial version of the MCR.
------------	-----	-----------------------------