MASSACHUSETTS INSTITUTE OF TECHNOLOGY
CAMBRIDGE, MASSACHUSETTS 02139

Here it is. Feel free to make any corrections you like.

CompSys has $ 1, 342 remaining as of whenever the accounting runs were done this weekend.

Code_converter is in>udd>ml>u>ssb>codes I don't think that anyone is using it. Dave Clark may want to put it on tape for the summer unless someone wants to SUBMIT ~~instal~~l it. Ken Pogran has documentation.

I will be available at KI7-3233 until tomorrow and at
30 Woodland Road
Longmeadow
Massachusetts 01106
therafter.

Scott S. Brown

Scott S. Brown

Errors and Inadequacies in the Teletypewriter Device Interface Module

Scott S. Brown

When input includes escapes:

1. It is impossible to edit an escape sequence. That is, Ø139#3 results in input of 3 not Ø133.

That is the correct way - not on ours!

2. To the canonicalizer, escapes appear to occupy only one character position. Thus,

Typist: abcdefØ133(BS)^{5}_
Printed line: abcdefØ133
Input: abcdef[

That can be documented rather than fixed.

3. Backspaces and new line chatacters which are entered as escapes are canonicalized exactly if a regular backspace or new line character also appears. Thus,

Typist: a(SP)(BS)Ø010_
and Typist: aØ010_

both having

Printed line: aØ010_

?

cause different strings to be input. Backspaces and new line characters entered as escapes should never be canonicalized.

why not?

WHEN input includes horizontal tabs:

4. It is impossible to backspace before a horizontal tab. Thus,

Typist: abcde(HT)f(BS)^{11}#
Printed line: #bcde f
Input: abcdef

4' TABS ALWAYS PUT AT END

SOMETIMES TABS ARE PUT IN ILLOGICAL PLACES

When the nelem argument is not large:

5. No more characters than are needed to fill the input buffer are processed. Subsiquent backspacing into the range read is not seen (see also 6 and 8). Thus

```
call ios_$read ("user_input", readp, 0, 10, nelemt, status);
```

Typist: abcdefghij(BS)^{10}_
Printed line: abcdefghij
Input: abcdefghij — where did the _ go?

Next read should input ~~[illegible]~~ (BS)^{10}_

This is the correct operation.

6. Fewer than nelem characters may be read:

```
call ios_$read ("user_input", readp, 0, 10, nelemt, status);
```

Typist: (SP)(BS)a#bcdefg##ijklm
Printed line: a#bcdefg##ijklm
Input: bcdefg

Should have a mode which expands tabs to blanks.

The rule that only the first nelem characters typed are considered, however, holds only when the canonicalizer is invoked, when a backspace or a new line character appears, and if escapes are counted as one character. Thus

```
call ios_$read ("user_input", readp, 0, 10, nelemt, status);
Typist:         a#bcdefg##ijklm
Printed line:   a#bcdefg##ijklm
Input:          bcdeijklm
```

NOT A PROBLEM IN MY DIM

7. NO INPUT MESSAGE CAN CONSIST ENTIRELY of BACKSPACES AND NEW LINE CHARA ~~A backspace or a new line character cannot be read alone.~~ In:

```
do i = 1 to 4;
call ios_$read ("user_input", readp, 0, 1, nelemt, status);
end;
Typist:         a(BS)_
Printed line:   a
```

the first read call will sucessfully read the a. On the second read call ttydim_ will call tty_read which will read the backspace and call the canonicalizer which will canonicalize the backspace to a null string.. Tty_read will then return to ttydim_ with a zero nelemt. Ttydim_ now thinks that tty_read could not find any input so it causes your process to go blocked until another line is entered. When your process receives a wakeup ttydim_ will call tty_read again and the underscore will be read, the backspace having been lost.

FIX IN TTY-READ

8. After reading the required characters tty_read scans the remainder of the line for a kill character. If it locates a kill character it discards all characters typed before the kill character without regard to their position in the printed line (i.e. before canonicalization), without verifying that the kill character is not escaped, and without verifying that the dim is in error-kill mode. Thus even in iomode not erkl:

```
call ios_$read ("user_input", readp, 0, 5, nelemt, status);
Typist:         abcdefg(NL)¢@_
Printed line:   ¢@cdefg
Input:          _
```

When in iomode erkl:

9. The convention that an erase character which is not the only graphic in a character position erases that character position in inconvenient. An erase character should erase only characters which were typed before it was typed. Characters typed after the erase character are surely desired. Changing this convention would allow a user to backspace and change a character which he has mistyped.

But you can't tell from the printed image what went in.

errors in ttydim_ page three

When printing:

DONE 10. Tty_write will output backspace characters even if the printing element is at the first position of the line. Bad!

MAYBE 11. TRAILING WHITE SPACE IS REMOVED BEFORE CANONICALIZATION?

It should be removed as part of canonicalization, if followed by a CR/LF or New Line.

I ONLY PLAN TO FIX THE STARRED ITEMS

ADDENDUM TO ERRORS IN TTYDIM—

SCOTT BROWN

12. IF A NEW LINE CHARACTERS APPEARS IN A ~~ct~~ STRING FOR WHICH THE CANONICALIZER IS INVOKED THEN THE NEW LINE IS TREATED AS A CARRIAGE RETURN AND A NEW LINE IS APENDED TO THE MESSAGE

EXAMPLE

TYPIST: (SP)(BS)abc¢012---

PRINTED LINE: abc¢012---

INPUT: $\underline{a}\,\underline{b}\,\underline{c}$

ALTERNITIVELY

TYPIST: abc¢012---

PRINTED LINE: abc¢012---

INPUT: abc

Should produce consistent results, but it doesn't matter which.

It would be a valuable service if you could document the entire canonicalizer, including the tradeoffs & design decisions in it.