SubKilo: The Test Matrix

Eingaeph

November 18, 2018

1 The Test Matrix Entries Are Numbered

At the command line in diretory TESS, entering an instruction similar to ***testnumb=1 make tester*** runs an automated test. The value of testnumb determines the particular test performed. The appropriate test number is that used in the following list.

2 Display Welcome Screen Or the First Page of Text, testnumb=1

Display a welcome screen if no input file is present in the argument list, or display the first lines of the input file when there is an input file. Confirm that uvset returned the proper number of rows and columns available on the display screen.
When an input file is specified as a command line argument, read the file into the globally available struct with name ***text.*** Display the first page contained in ***text.*** Expect the screen to be filled with the first lines of the data previously read in. Expect the cursor to be placed at the home position, that is at the top left.

```
1 // local variable store contains command test sequence
2             store[j] = CTRL_Q;      j++;
```

3 A Simple Pager, testnumb=2

Read an input file, and display its first page. Page Down twice then Page Up Once:
Expect lines 24 to 47 to be displayed to the screen. Expect the cursor to be at line 24, column 1.

```
1             store[j] = PAGE_DOWN;    j++;
2             store[j] = PAGE_DOWN;    j++;
3             store[j] = PAGE_UP;      j++;
4
5             store[j] = CTRL_Q;       j++;
```

4 Move the Cursor, Enter Characters, testnumb=3

Read an input file, etc. Arrow Down twice, Page Down, Arrow Right three times, Arrow Down, enter two printable characters, then Arrow Down.

```
1              store[j] = ARROW_DOWN;    j++;
2              store[j] = ARROW_DOWN;    j++;
3              store[j] = ARROW_RIGHT;   j++;
4              store[j] = ARROW_RIGHT;   j++;
5              store[j] = ARROW_RIGHT;   j++;
6 c = 'a';     store[j] = c;             j++;
7 c = 'b';     store[j] = c;             j++;
8              store[j] = ARROW_DOWN;    j++;
9
10             store[j] = CTRL_Q;        j++;
```

Expect the cursor on line 24, then move three times to the end of the line, then insert three characters at the end of the line, then cursor moves down to the end of the next line to position line 25, column 3, that is just past the last character on line 25.

5 Move the Cursor Outside the Window, testnumb=4

move the insertion point above the visible window move the insertion point below the visible window move the insertion point to the right of the visible window

```
1              store[j] = ARROW_DOWN;    j++;       // cursor at col 01 row 02
2              store[j] = PAGE_DOWN;     j++;       // cursor at col 01 row 24
3              store[j] = ARROW_UP;      j++;
4              store[j] = ARROW_RIGHT;   j++;
5              store[j] = ARROW_RIGHT;   j++;
6              store[j] = ARROW_RIGHT;   j++;
7              store[j] = CTRL_Q;      j++;
```

6 Move to the End of Long Line, Arrow Up etc., testnumb=5

Test handling an input file with a very long line. Arrow Up and Arrow Down to shorter lines.

```
1              store[j] = PAGE_DOWN;     j++;
2              store[j] = END_KEY;       j++;
3              store[j] = ARROW_UP;      j++;
4              store[j] = CTRL_Q;        j++;
```

7 Conclusion

Its not much of a paper but its a start.

Appendix A The Makefile

Here is a makefile for testing editAline.c and associated functions.

Appendix B An Example replay.c File

Here is one of the replay.c files stored in TESS.