SubKilo: Test Matrix

Eingaeph

November 10, 2018

1 Introduction

The worth of the SubKilo library of course depends on the achievement of error free functionality. Our method for achieving that goal is to construct a test matrix, with entries demonstrating a complete list of expected functionalities. This paper describes the SubKilo test matrix.

A note. We acknowledge of the contribution of Antirez, whose work shows that a short code suffices. This fact was not known to us before Antirez, as the essentials of editing were hidden in a cloud of competitor editors, complete with at least one entire book on editing. We follow Antirez's coding when we can.

Another note. Our coding is optimized for readability by its authors, so it contains lots of whitespace and comments, and so it is not optimized for a small number of lines of code. This style, like the use of many assert statements, is for the convenience of the authors.

2 Ruten is Our Guide to Constructing a Test Matrix

A previous version of this project failed because the complexity of testing got ahead of the available coding resource. The problem originated in part from an unorganized testing protocol.

To systemize the work we now use the testing protocol used by Ruten, who successfully recoded Kilo while documenting the functionality he was aiming for. His stepwise testing path is very helpful to us, since our coding is very similar to Ruten's, who achieves the same economy in lines of code as does Antirez.

The next section owes a great deal to Ruten.

3 The Test Protocol in a Numbered List

The following numbered list is performed in directory TESS by entering ***testnumb=1 make tester*** at the command line. Here the value of testnumb determines the particular test performed.

1. Query the screen for size. If there is no input file throw up a welcome screen and terminate execution.

   Expect a screen with tilde's along the side. Confirm that uvset returned the number of rows and columns available on the display screen.

2. Read a file into the globally available struct text. Display the first page.

   Expect the screen to be filled with the first lines of the file. Expect the cursor to be placed at the home position, that is at the top left.

3. Page Down twice; Page Up Once: