Semi-Automated Testing: Directory TESS

Eingaeph

November 7, 2018

1 Motivation

SubKilo provides a couple dozen essential functions. As a consequence the SubKilo test matrix will be extensive and might be expensive to repeat. After most of the compilation has been successfuly done in other directories, and after the library has passed a few sanity checks then more extensive tests are done in directory TESS. In that way, the code modification and the testing don't clutter up the same directory.

2 Testing Function editAline.c

Many, even most, of the test matrix entries are checks on the actions of the functions called by editAline.c. Function editAline.c takes the encoded keystroke value, then calls functions to move the cursor around the screen, insert characters, delete characters, scroll the page, etc.

In testing editAline.c we lean heavily on replay.c which simulates encode keyvals from a number of keystrokes. As replay.c works we watch the screen. When the desired results make it to the screen the test is successful.

So many test matrix entries could be prohibitively expensive to repeat untl successful. Here the idea we use is to construct a makefile setting up each of the numerous tests using different versions of the text input, and different versions of replay.c. In this way tests can be easily repeated until, hopefully, the

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

```
1  stem     = /home/ubuntu/git/repo.1/libk/
2  new = REPL
3  old = EDAL
4  nstem = $(stem)$(new)/
5  ostem = $(stem)$(old)/
6  libn =  $(nstem)lib$(new).a
7  libo =  $(ostem)lib$(old).a
8
9  firs:
10         echo $(new)      > /dev/null
11         echo $(old)      > /dev/null
12         echo $(nstem)    > /dev/null
13         echo $(ostem)    > /dev/null
14         echo $(libn)     > /dev/null
15         echo $(libo)     > /dev/null
16
17 cleano:
18         rm -f *.o *.a
19
20 dryrun:
21         rm -f *.o *.a test.c
22         gcc -Wall -c $(nstem)*.c
23
24 linkall:
25         rm -f *.o *.a $(libn) test.c
26         ar x          $(libo)
27         ar rcs        $(libn) *.o
28         rm -f *.o
29         gcc -Wall -c $(nstem)*.c
30         ar rs         $(libn) *.o
```

```
31          ar t          $(libn)
32          rm -f *.o test
33          cp main.cold test.c
34          gcc -Wall test.c $(libn) -o test
35          rm -f test.c
36          ./test test.dat
37
38  count:
39          cat ../UVSE/*c >  counter
40          cat ../KEYP/*c >> counter
41          cat ../WARF/*c >> counter
42          cat ../EDAL/*c >> counter
43          cat ../REPL/*c >> counter
44          cat ../VT100.h >> counter
45          cat ../keyvals.h >> counter
46          cat ../libk.h  >> counter
47          wc counter
48          rm -f counter
49
50  compileall:
51          rm -f *.o *.a
52          gcc -Wall -c $(kdir)ReadKey.c
53          gcc -Wall -c $(kdir)die.c
54          gcc -Wall -c $(udir)disableRawMode.c
55          gcc -Wall -c $(udir)enableRawMode.c
56          gcc -Wall -c $(kdir)encode.c
57          gcc -Wall -c $(udir)getCursorPosition.c
58          gcc -Wall -c $(wdir)readAfile.c
59          gcc -Wall -c $(wdir)readAline.c
60          gcc -Wall -c $(udir)screenTest.c
61          gcc -Wall -c $(udir)uvset.c
62          gcc -Wall -c $(wdir)wind.c
63          ar  rcs       $(wdir)libWARF.a *.o
64          ar  t         $(wdir)libWARF.a
65          rm -f *.o
66          cp main.cold test.c
67          gcc -Wall test.c $(wdir)libWARF.a -o test
68          rm test.c
69          ./test test.dat
```