# Semi-Automated Testing: Directory TESS

Eingaeph

November 7, 2018

## 1 Motivation

SubKilo provides a couple dozen distinct 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 testing is successful.

## 3 An Example Makefile

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

```
1  stem    = /home/ubuntu/git/repo.1/libk/
2  new = TESS
3  old = REPL
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 test.1:
25         rm -f *.o *.a $(libn) test.c inpu.dat
26         ar x          $(libo)
```

```
27          ar rcs        $(libn) *.o
28          rm -f *.o
29          #retrieve replay.1 to serve as replay.c during this test
30          cp replay.1 replay.c
31          gcc -Wall -c $(nstem)*.c
32          ar rs         $(libn) *.o
33          ar t          $(libn)
34          rm -f *.o *.c test
35          #retreive test.1 to serve as main during this test
36          cp main.1 test.c
37          gcc -Wall test.c $(libn) -o test
38          rm -f test.c $(libn)
39          cp inpu.1 inpu.dat
40          ./test inpu.dat
41
42  count:
43          cat ../UVSE/*c >  counter
44          cat ../KEYP/*c >> counter
45          cat ../WARF/*c >> counter
46          cat ../EDAL/*c >> counter
47          cat ../REPL/*c >> counter
48          cat ../VT100.h >> counter
49          cat ../keyvals.h >> counter
50          cat ../libk.h  >> counter
51          wc counter
52          rm -f counter
```