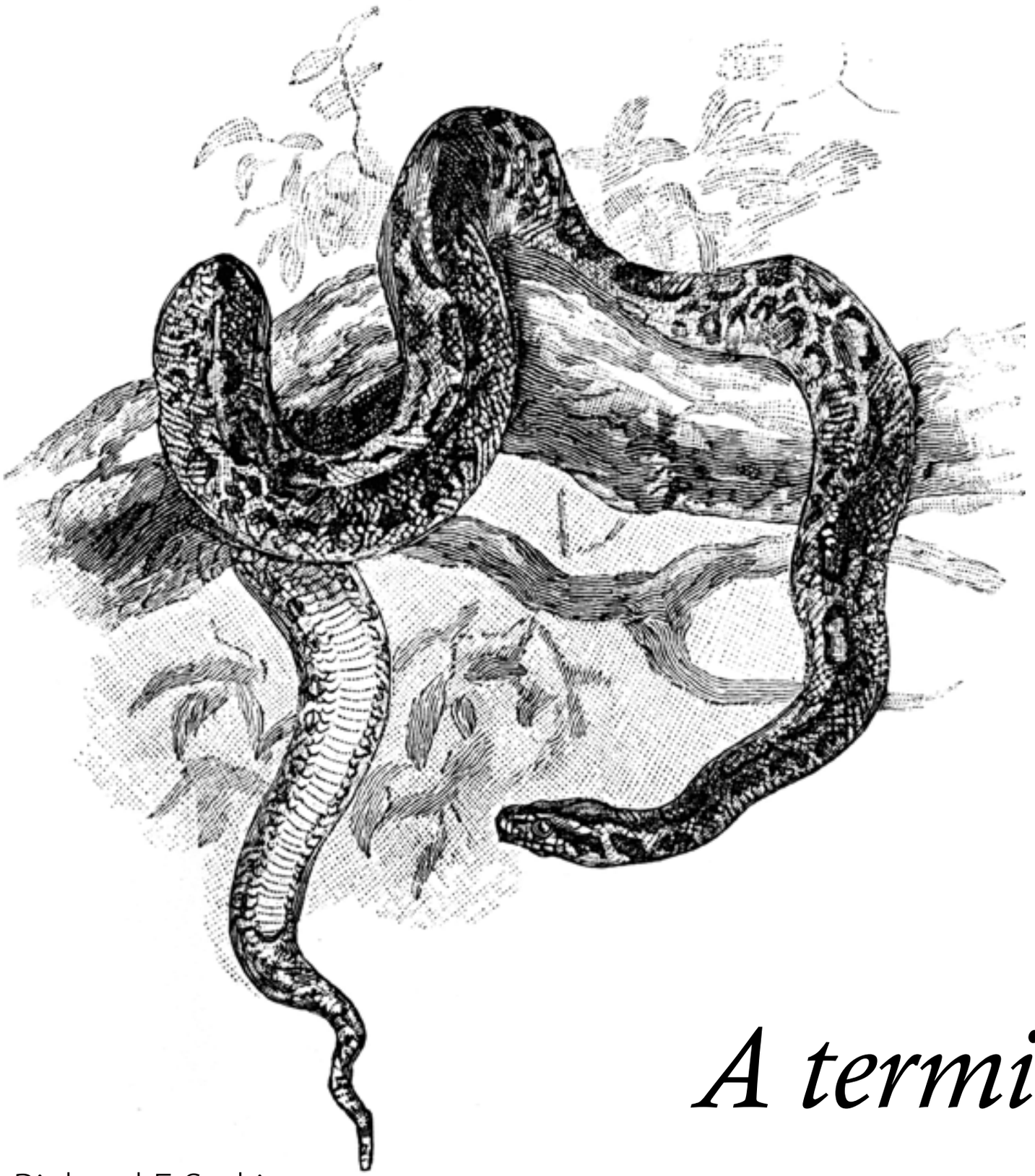

tmux

A terminal multiplexer

Richard E Sarkis
Rochester's Python User Group
June 17th, 2014 Meeting

rich -- puddle-jum... ⌘1

```
6
7 """
8 Create and convert python callback function
9 """
10 rl_command_func_t = CFUNCTYPE(c_int, c_int, c_int)
11 rl_hook_func_t = CFUNCTYPE(c_int)
12
13
14 def set_pre_input_hook(func):
15     readline.set_pre_input_hook(func)
16
17
18 def set_event_hook(func):
19     # From: http://osdir.com/ml/python.ctypes/2006-12/msg00045.html
20     ptr = c_void_p.in_dll(pythonapi, "PyOS_InputHook")
21     ptr.value = cast(rl_hook_func_t(func), c_void_p).value
22
23
24 def fix_command(count, c):
25     from subprocess import call
26     from tempfile import NamedTemporaryFile
27
28     with NamedTemporaryFile(delete=False) as tf:
29         tfName = tf.name
30         tf.write("\"\"\"\n")
31         for item in xrange(0, readline.get_current_history_length() - 1):
32             hist_item = readline.get_history_item(item)
33             tf.write("## History item {}\n".format(item))
34             tf.write((hist_item if item is not None else "") + "\n")
35             tf.write("\n")
36
37         current_line = readline.get_line_buffer()
38         tf.write("\n## Current input line below:")
39         tf.write(current_line + "\n")
40         tf.write("\"\"\"\n")
41
42     if call(['/usr/local/bin/vim', tfName]) != 0:
43         return None  # Editor died or was killed
44
45     # Get modified content
46     with open(tfName).readlines() as result:
47         os.remove(tfName)
48         print result
49     return 0
50
51 prev_state = []
52
53
54 def event_hook():
55     global prev_state
```

~/Desktop/readline-ext.py [unix] [PYTHON] [116] [0033,0050][25%] [127 lines]

```
Traceback (most recent call last):
  File "_ctypes/callbacks.c", line 314, in 'calling callback function'
  File "readline-ext.py", line 46, in fix_command
    with open(tfName).readlines() as result:
AttributeError: __exit__
Illegal instruction: 4

[14][ rich@puddle-jumper ][ Thu Jun 12 : 07:57 PM ][ ttys005 ]
[~/rich/Desktop]$ python -i readline-ext.py
>>> set(['RL_STATE_MACRODEF', 'RL_STATE_METANEXT', 'RL_STATE_INPUTPENDING', 'RL_STATE_SIGHANDLER', 'RL_STATE
_VICMDONCE', 'RL_STATE_MULTIKEY', 'RL_STATE_DISPATCHING', 'RL_STATE_SEARCH', 'RL_STATE_DONE', 'RL_STATE_MACR
OINPUT', 'RL_STATE_MOREINPUT', 'RL_STATE_NUMERICARG', 'RL_STATE_ISEARCH', 'RL_STATE_COMPLETING', 'RL_STATE_O
VERWRITE', 'RL_STATE_NSEARCH', 'RL_STATE_REDISPLAYING', 'RL_STATE_READCMD', 'RL_STATE_INITIALIZING', 'RL_STA
TE_VIMOTION', 'RL_STATE_UNDOING'])

>>> set(['RL_STATE_VICMDONCE'])
Traceback (most recent call last):
  File "_ctypes/callbacks.c", line 314, in 'calling callback function'
  File "readline-ext.py", line 46, in fix_command
    with open(tfName).readlines() as result:
AttributeError: __exit__
Illegal instruction: 4

[15][ rich@puddle-jumper ][ Thu Jun 12 : 09:28 PM ][ ttys005 ]
[~/rich/Desktop]$
```

```
[3][ rich@puddle-jumper ][ Tue Jun 17 : 04:40 PM ][ ttys007 ]
[/Users/rich]$
```

[1] 0:bash* "rich -- puddle-jumpe" 16:40 17-Jun-14

`tmux`, why?

- Useful to different people in different ways:
  - Create efficient terminal workspaces, locally or remotely
    - Create persistent, re-attachable remote terminal sessions
- Considered the successor to **`screen`**
- Multi-platform, doesn't rely in application-specific features (split-panes, etc)

tmux vs. screen

- menu driven for session, window or client selection
- vi or emacs style command modes, with auto-complete
- easier configuration
- not a drop-in replacement for screen, w.r.t. keybindings

`tmux` vs. `screen`

- well-defined client-server model
- a consistent, well-documented command interface
- easily scriptable from the shell
- multiple paste buffers
- clean codebase, modern and BSD licensed
- `screen` supports serial and telnet, wider-platform support (IRIX and HP-UX)

Functional Structure