How to create and import a custom C library in Python (Linux).

This is something I picked up recently. I am unsure what the best practices are-- but the method I picked up appears to work. What you will need is a compiler (we will be using GCC), and of course, Python.

We begin by writing some C code we want to compile into a library. For demonstration, I wrote some simple code. Note I didn't divide this code into a header file and a source file. That would only complicate the tutorial, but in practice you would most likely divide your code by standard C practices such as using a header file and a source file. The code I wrote is similar to Python's randint function-- we will make our own random function.

test.c

#include <stdlib.h>

#include <time.h>

#define TRUE 1

#define FALSE 0

static int randomized = FALSE

void randomize(void)

{

srand(time(NULL));

randomized = TRUE;

}

/\* returns a random number between base and max, inclusive \*/

int get\_random(int base, int max)

{

if (randomized == FALSE)

randomize();

}

return rand() % (max - base + 1) + base;

}

Well-- there we have it. We have the C code, but we need to create a library (in Linux, a .so file). We can do so by compiling our code and outputting a shared library. For example, here is how you can do it with GCC:

gcc -Wall -O3 -shared ~/test.c -o test.so

This turns on warning, optimizes the code and exports the test code as a file called, 'test.so' into our home directory. The shared flag is what outputs the shared library.

Finally, we need to use this in our Python script. The first thing we want to do is import the ctypes module.

import ctypes

Next we need to load our library, in Linux we will do it like so (no pun intended):

my\_test\_lib = ctypes.cdll.LoadLibrary('/home/username/test.so')

Then you can call functions inside your library by calling them through the library object, like this:

my\_test\_lib.get\_random(1, 10)

Here's the complete source for the test Python code I wrote.

view sourceprint?

#!/usr/bin/env python

import ctypes

def main():

my\_test\_lib = ctypes.cdll.LoadLibrary('/home/usrname/test.so')

for i in range(10):

# Note, this uses the Python 2 print

print "Random = %d" % my\_test\_lib.get\_random(1, 10)

if \_\_name\_\_ == '\_\_main\_\_':

main()

Ending notes:

The Windows version of this is slightly different-- you need to compile the library in a slightly different way and the way you import the library will be a little different.