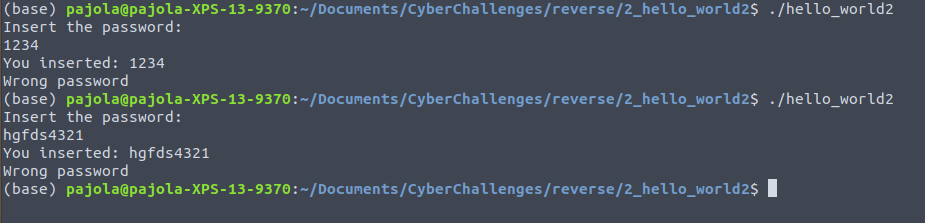
Hello World 0 - Alternative Solution

Let execute the program:



We need to figure out the password , or, more in general, reach our flag in someway.

We can have a look on the assembly of the main function:

*objdump -d hello\_world\_2*

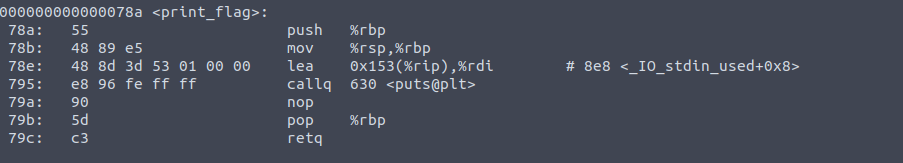
At first, let’s have a look on the functions that the *main* can call during its execution:

1. *7cb* - *puts* : it prints something -- likely “insert password [...]”;
2. *7df - gets* : get the user inserted value -- likely the password that we are guessing;
3. *7fa - printf* : print something -- likely “you inserted”;
4. *815 - strncmp* : comparison between two strings;
5. *823 - callq* : call to print\_flag -- likely it contains our flag ;
6. *831 - puts* : print something.

The first thing that we should notice is:

* The called functions are from stdio / string libraries and print\_flag;
* It seems that the program is not opening any file that contains the flag, i.e., the password is stored in the program.

It seems also that this comparison it is done on the strncmp, which is done between two registers, and thus, we do not have any info right now of the password. We need to be sure that this password comparison is done in the instructions *815*; the only other place that a password matching could be done is inside *print\_flag*, so we should check its assembly:



The program seems to print the flag and nothing more, i.e., our hypothesis of line *815* seems right.

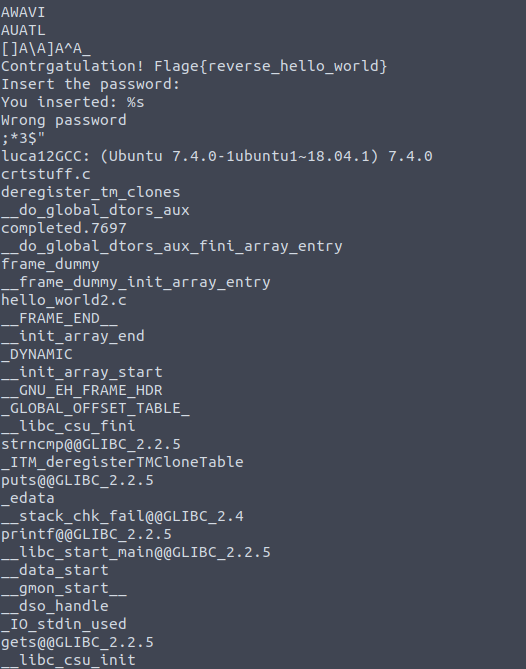
If the password is saved in a string inside the program, we can easily check with another bash function called *strings*; type the following:

*man strings*

The *man* command returns the manual of a given program. In this case, you can see that *strings* “print the strings of printable characters in files”.

Now we can see the effect on an object:

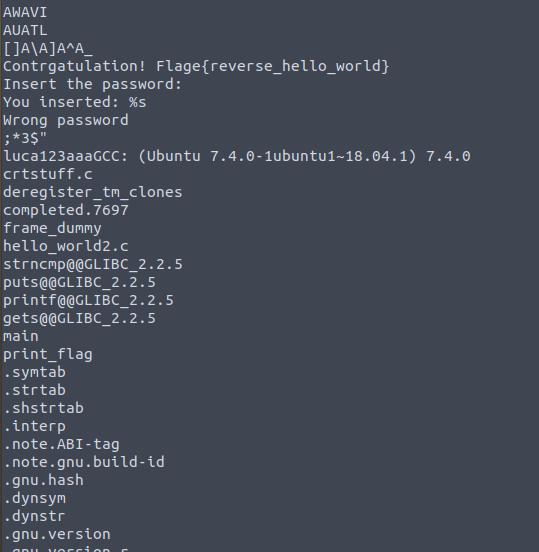
*strings hello\_world2*



There are a lot of things here, such as the possible flag (*{reverse\_hello\_world}*). However, let’s try to understand the password. Since there are a lot of strings, we can discard those that are related to the systems (i.e. starting with “\_”).

The *grep* command can help us:

strings hello\_world2 | grep -v ‘^\_’



Here we find something suspicious “luca123aaa” … is it the password?

We can try some combination (e.g., luca, luca123, aaa … ) and we can find that luca123 is the right one.

By inserting “luca123” as password, we can capture the flag.