x64 assembly programming with C library

Introduction

In this assignment, you will write programs using Intel x64 assembly(nasm syntax) with help of C library to solve some simple questions. We will be using the netwide assembler(nasm) and gcc to compile the assembly code into an executable for testing. You have to write the entire solution in x64 assembly: you cannot invoke any other code except the standard C library and code you write. All these are doable with help of the C library.

Question 1: Environment variable query tool

In this task, you will implement a tool that prints the value of an environment variable(if exists), whose name is passed to the program as a command line argument, to stdout. If no arguments are passed, the tool prints all the environment variables and their values to stdout. See sample session for more.

Sample session

```
$ ./get-env.out HOME
/home/blah
$ ./get-env.out USER
blah
$ ./get-env.out BLAH
Not found
$ ./get-env.out
...
HOME=/home/blah
USER=blah
....
```

How we will test your program

We will compile your assembly program using nasm and gcc and run the resulting executable. We may or may not pass arguments to your program, as stated in the description above.

Question 2: Fibonacci calculator

In this task, you will implement a program that computes the Nth fibonacci number and prints its value to stdout. If no fibonacci number exists, print -1. N will be passed as a command line argument to your tool. See sample session for more.

Sample session

```
$ ./get-fib.out 1 2 3 4 5
0
1
1
2
3
$ ./get-fib.out 10
34
$ ./get-fib.out -12
-1
$ ./get-fib.out 53
32951280099
```

How we will test your program

We will compile your assembly program using nasm and gcc and run the resulting executable. We will pass at least one N to your program as a command line argument, as shown in the sample session.

Question 3: File encryption

In this task, you will implement a simple tool that performs ROT-N encryption(eg: Caesar cipher is a ROT-3 cipher). We will pass two values to the program: a filename F and a shift N. The tool will read the contents of F, shift them by N and print the result to stdout. The file F will contain only uppercase alphabets(A-Z).

Sample session

```
$ cat file.txt
ABCDEFGHIJKLMNOPQRSTUVWXYZ
$ wc -c file.txt
26 file.txt
$ ./rotn-encrypt.out file.txt 5
FGHIJKLMNOPQRSTUVWXYZABCDE
$ ./rotn-encrypt.out file.txt -2
YZABCDEFGHIJKLMNOPQRSTUVWX
$ ./rotn-encrypt.out file.txt 1000
MNOPQRSTUVWXYZABCDEFGHIJKL
$ ./rotn-encrypt.out file.txt -1000
OPQRSTUVWXYZABCDEFGHIJKLMN
```

How we will test your program

We will compile your assembly code using nasm and gcc and run the resulting executable. We will pass the filename and shift value as command line arguments, as shown in the sample session.