CS Basics - Exercises
Procedures in Assembler

E. Benoist

Fall Term 2017-18

1 Shifts and boolean functions

Exercise 1 *Write an assembler file, where you put 20h in the RAX. Then you divide this number each time by two. Each time you increase the counter RCX (originally initialised to 0).*

Loop until RAX contains zero.

RCX should contain the logarithm of the value in RAX at the beginning.

2 Compute Binary-Logarithms

Exercise 2 *To solve this exercise, you will need to use some pieces of code, that you will pick in the example seen in course, hexdump.*

Compile the hexdump1 example, to take as input one number in hexadecimal on the standard input. The output should be the logarithm of this number (on the standard output), also in hexadecimal.

- *In a first version, you will read the input from the data and write the result in a register (it will work inside a debugger)*
- *Write a code that takes RBX chars from memory as a string of chars and transforms them in a number in RAX.*
- *In a second version, you will use the code provided in the hexdump1 example to read the number from the standard input. For easy programming, we will only look at numbers with 4 decimal digits (some of them can be 0). The program should work like:*

  ```
  > myprogram
  1234
  ```

  The logarithm is written in a register (see it inside the debugger).

- *write a division for transforming a number in a register in a string. You will need a three bytes string in the data section. You will use the addition (+30h transforms a digit 9 for instance into a character '9') to create characters. You need to use integer division DIV (RAX is divided by the operand and result goes into RAX (result) and RDX (remainder)).*
- *Finally, you will output the content of the result register out to the standard output.*
- *Modify your program to accept any number.*