CS Basics - Exercises
Procedures in Assembler

E. Benoist

Fall Term 2017-18

1 Recursion

Exercise 1 *Write an assembler procedure, that takes as input* `RAX` *and* `RCX` *and changes the value of* `RAX` *into* RAX^{RCX} .

- *Implement this procedure with a loop.*
- *Implement this procedure with a recursion*

Take extra care to protect the registers, such that the calling piece of code is not influanced by the manipulation of the registers you do inside the procedure.

2 Write a program for computing logarithm, using procedures

Exercise 2 *Compile the hexdump1 example.*
Write a file containing the following procedures:

- `printstring` *Prints on standard output a string, the address of the string is written in RAX, the length of the string is written in RBX. Use of the procedure:*

```
mov RAX, String   ; Copy the address in RAX
mov RBX, STRLEN   ; Copy the length of the string in RBX
call printstring  ; Make the call to the procedure you need to write
```

- `Read Buff` *Reads a string from standard input into a buffer. It reads from the standard input and writes the data into the memory. ¡br¿ Inputs:*
  - *The address where to put the information (i.e. the buffer) is in RAX*
  - *The maximal number of characters to read is in RBX*

  Outputs
  - *The number of read characters (can be different from input) is in RAX.*

- `readint` *Transforms a string in memory (like "123") into a number in a register. ¡br¿ Input:*
  - *Address of the string in RAX*
  - *Length of the string in RBX*

  Output:
  - *The number in RAX*
- `printint` *Prints out on the standard output a number. Input is a number in RAX.*
- `logarithm` *Computes the logarithm of a number. The number is written as input in RAX, the output is written also in RAX.*