CS Basics - Exercises
Procedures in Assembler

E. Benoist

Fall Term 2022-23

1 Recursion

Write two assembler procedures, which take as input RAX and RCX and return the value RAX^{RCX} in RAX:

1. Write an implementation using a loop
2. Write an implementation using recursion

Take care of properly saving and restoring caller- and callee-saved registers.

2 Write a program to compute logarithms, using procedures

Assemble, read and understand the `hexdump2.asm` example. Then, write the following procedures:

- `print_string`: Writes a string to standard output. It has the following arguments:
  - RAX: Address of the string to write
  - RBX: Length of the string

  There is no return value.
- `read_string`: Reads a string from standard input to a buffer. It has the following arguments:
  - RAX: Address of the buffer, where the received input should be written
  - RBX: The maximal number of characters to read

  The return value of the procedure is the number of characters actually read. It is returned in RAX.

- `parse_int`: Transforms an ASCII string in memory (e.g. "123") into a number. It has the following arguments:
  - RAX: Address of the string with the number
  - RBX: Length of the string

  The parsed number is returned in register RAX.
- `print_int`: Prints a given number in ASCII to standard output. It has the following arguments:
  - RAX: The number to print

  There is no return value.
- `logarithm`: Computes the *binary* logarithm of a given number. It has the following arguments:
  - RAX: The number for which the logarithm should be calculated

  The logarithm is returned in register RAX.

3 Library

Write a small math library containing the following procedures:

- Binary logarithm
- Exponent

Rewrite the program from the previous exercise to use your new library.