

# Computer Architecture

Richard Röttger – SDU, Fall 2017

## Lab 2

### Lab 2 – Input/Output & Syscalls

*Please read the short x86-64 guide from the website beforehand!*

In this lab we will learn how to get some input into our programs instead of just using hard coded constants. One method is by reading from the stdin (standard input) using the syscall command.

Another method is using command line parameters. This is a very convenient method, as they are provided by the operating system ready to use on the stack. At the start of your program, the %rsp points to the top element of your program's stack:

%rsp	→ #args
%rsp+8	→ addr. of arg1 (name of program)
%rsp+16	→ addr. of arg2 (command line arg1)
%rsp+24	→ addr. of arg3 (command line arg2)
...	→ ...

*Keep in mind, if you use push or pop at some point, %rsp will increase and decrease accordingly.*

The addresses refer to a location in memory containing a zero terminated ASCII string. The first argument is always the name of your program.

### 1. Hello World

Objectives:

- Define a string constant "Hello World!\n".
- Output that string to stdout.

Hint:

- Use the *.string* directive in the section *.data* to define a string.
- Use the syscall *write* to output a string.

## 2. Use command line arguments

Objectives:

- Pass a command line argument to your program.
- Output that string to stdout.

Hint:

- See above for the locations of the command line argument strings.

## 3. Read & Output

Objectives:

- Read a string from stdin.
- Output that string to stdout.

Hint:

- Use the syscalls *read* and *write*.

## 4. Read from a file

Objectives:

- Create an input file next to your program with at least  $n$  characters.
- Read the name of the file as a command line argument.
- Open the file and load the first  $n$  bytes into a fixed size buffer of size  $n$ .
- Output the buffered string.

Hint:

- Use the file descriptor from an *open* syscall to read from the file.
- Use the *.space* directive in the section *.data* to create a fixed buffer.