CS Basics - Exercises
Test your development environment

E. Benoist

Fall Term 2018-19

1 Compilation

1.1 Update and upgrade your system

Each time you start Linux, you shoud update your system:

```
> sudo apt-get update
> sudo apt-get upgrade
```

1.2 Install software

Using the package manager in a shell (see the syntax exercise of the chapter2), install the following packages: `nasm` and `ddd`. You should allready have installed the packages: `bless` and `emacs24`

1.3 Find source file, and compile

The source files are in the directory helloWorld. You can edit the file `eatsyscall64.asm` using `emacs`, `gedit` or any linux editor you like (`vi`, or `vim` if you like them).

In the comments of the program: change add your name as "modified by" You should not change the program itself, since we did not learn the syntax.

Execute manually `nasm` and `ld` to build an executable (see slides for the syntax).

Compile the file using `make`. You may need to `touch` the source to force the compilation, otherwise nothing happens.

1.4 Start the debugger

The documentation for DDD is to be found at: `://www.gnu.org/software/ddd/manual/`.

```
> ddd myprogram &
```

- Place a breakpoint (rightclick on the line and select breakpoint) on the line "move eax 4".

- Click on the run button. The system should stop at your breakpoint.
- Show the register values
- Go on step by step, util you have in a register the address in memory of the string. and also its length (read the comments to know where).
- Print the memory at the found address.

1.5 Edit the file

Change the text of the file to "Hello World", and recompile the file, debug the executable and print out the memory.

Store the value 35 in the register R8 (using the instruction `mov r8,35`). Examine in the debugger the value in the register.

Store the value 35 in the register RAX. Examine in the debugger the value in the register.

Store the value -1 into the register EAX (32-bit register)

Store the value 10 into the register AX (16-bit register)

Do not forget to notice the value "really" stored inside the register (in the debugger) and compare with what you expect.