Report on exercise #1

Matteo Corain S256654

Laboratory #5 – System and device programming – A.Y. 2018-19

The given `hello-5` module is a kernel module whose only purpose is to print a set of initialization messages on the kernel log, outputting the values of a set of internal variables which can be specified also by the user at installation time. The module also prints a goodbye message when it is uninstalled.

All of those messages can be reviewed by accessing the kernel log file, placed in `/var/log/kern.log`. This file may be analyzed via standard commands (e.g. `cat`, `tail`, `more`) or by means of the `dmesg` command (eventually coupled with `tail`, in order to show only the last lines of the file).

In order to test the `hello-5` module, the first step to take is to compile it to a kernel object. This can be done via the `make` command, whose associated `Makefile` contains all the needed commands to perform this action. In particular, two targets are defined in the `Makefile`:

- Target `all` is used to compile the `hello-5.c` file after having changed the working directory to the one where kernel headers are stored and restoring it after the compilation has been performed;
- Target `clean` is used to remove the compiled files from the current folder.

Moreover, at the beginning of the file, the name of the `hello-5.o` file is appended to the `obj-m` variable, to specify we want to compile the `hello-5.o` object to a kernel object. The first command to be used for the purpose of the exercise is therefore a `make all` in the folder where the `hello-5.c` is present. This action produces, among others, the `hello-5.ko` kernel object, ready to be installed.

In order to install the module, the `insmod` command is used. This command, which has to be called as superuser, can take as optional parameters a set of `key=value` pairs; each of these parameters is copied in a local variable `key` initialized to `value`. In this case, the `hello-5` module defines the following variables:

```
static short int myshort = 1;
static int myint = 420;
static long int mylong = 9999;
static char *mystring = "blah";
static int myintArray[2] = { -1, -1 };
static int arr_argc = 0;
```

As it can be seen, the module also declares a set of default values for each of them, so that if the user does not provide a value on the command line at the installation time, the provided values are used.

The first test has been executed by installing the kernel module without passing to `insmod` any parameter. As a result, the following lines are appended to the kernel log file:

```
[  158.153309] Hello, world 5
               =============
[  158.153312] myshort is a short integer: 1
[  158.153314] myint is an integer: 420
[  158.153316] mylong is a long integer: 9999
[  158.153318] mystring is a string: blah
[  158.153320] myintArray[0] = -1
[  158.153322] myintArray[1] = -1
```

```
[  158.153324] got 0 arguments for myintArray.
```

To remove the installed module, the `rmmod` command is used (again, to be called as superuser). When we execute `rmmod` on the `hello-5.ko` kernel object, the following lines are appended to the kernel log file:

```
[  158.208320] Goodbye, world 5
```

As a second test, the `hello-5` module has been installed by following the suggested example of usage: `insmod hello-5.ko mystring="sdplab" myshort=123 myintArray=-1,3`. As a result, the following lines are appended to the kernel log file:

```
[  238.272585] myshort is a short integer: 123
[  238.272586] myint is an integer: 420
[  238.272587] mylong is a long integer: 9999
[  238.272587] mystring is a string: sdplab
[  238.272588] myintArray[0] = -1
[  238.272589] myintArray[1] = 3
[  238.272589] got 2 arguments for myintArray.
```

It is possible to notice that the default values of the variables `mystring`, `myshort` and `myintArray` have been overwritten. When removing the module, the same goodbye message is printed.

Further tests have been conducted by passing "wrong" values for the parameters. As a first tentative, three values have been passed to the `myintArray` vector, which is only of length 2; in this case, `insmod` fails with the following error:

```
insmod: ERROR: could not insert module hello-5.ko: Invalid parameters
```

And in the kernel log the following line is added:

```
[  260.803460] myintArray: can only take 2 arguments
```

As a second tentative, an out-of-range value for `myshort` has been used; in this case, again `insmod` fails with the following error:

```
insmod: ERROR: could not insert module hello-5.ko: Numerical result out of range
```

And in the kernel log the following line is added:

```
[  292.729439] hello_5: `100000000000' invalid for parameter `myshort'
```