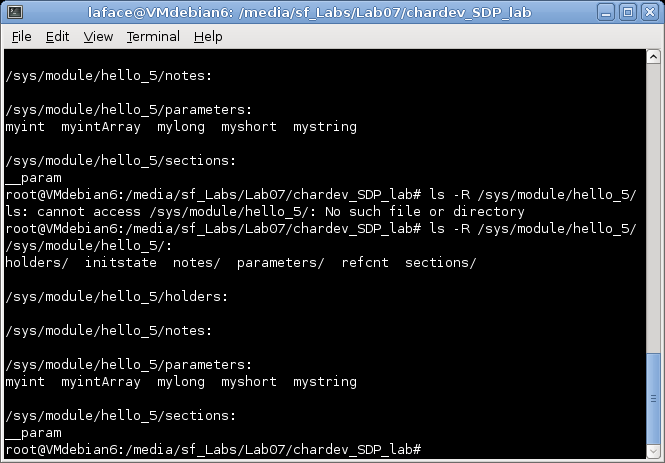
Lab 07 – Martino Mensio

# Exercise 1

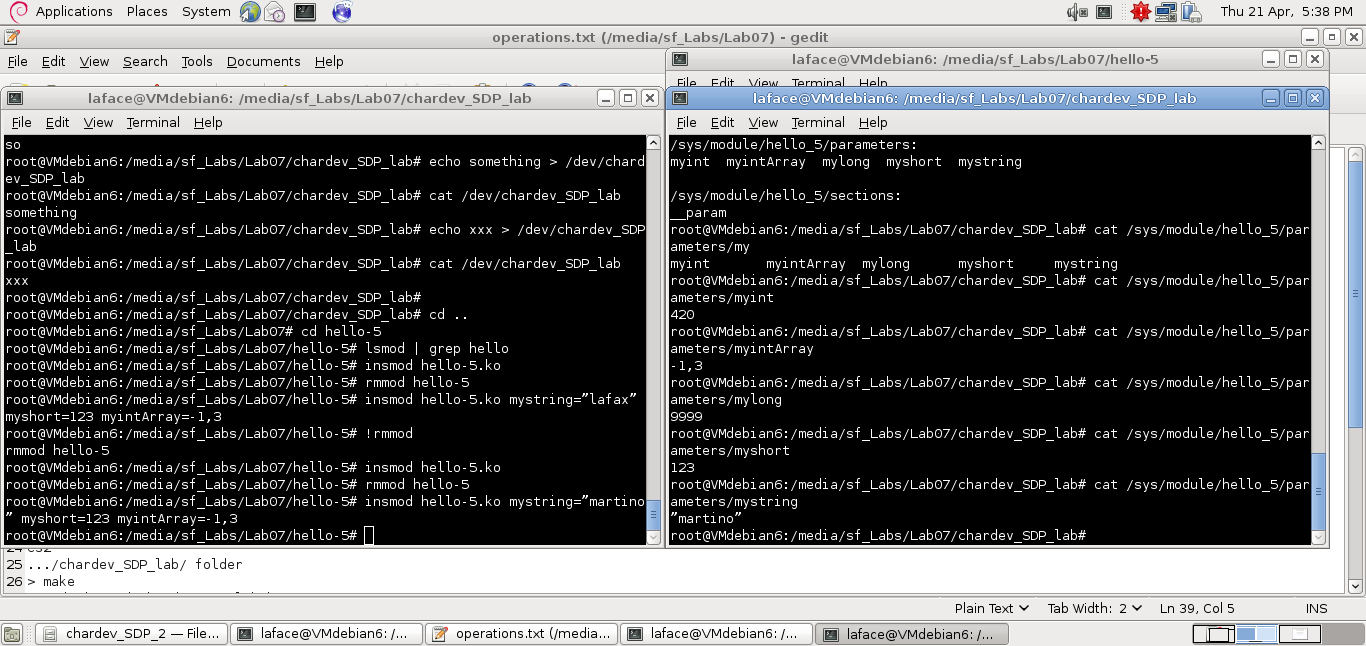
The first step is to compile the module from source code. In order to compile, I used the Makefile inside the folder of the module hello-5. To use it, I simply executed make in the current directory. The compilation generates some files, including a .ko one, that is the one we need for next steps.

Having the file hello-5.ko, in order to install the module on the system, I used the insmod command (insmod hello-5.ko). This command must be executed from super user for security reasons. To verify that the module is installed, I verified using the command lsmod | grep hello that displayed hello\_5 925 0. It is possible also to specify some parameters to the module when installed. If the module is installed without parameters, there exist some default values declared inside the module.

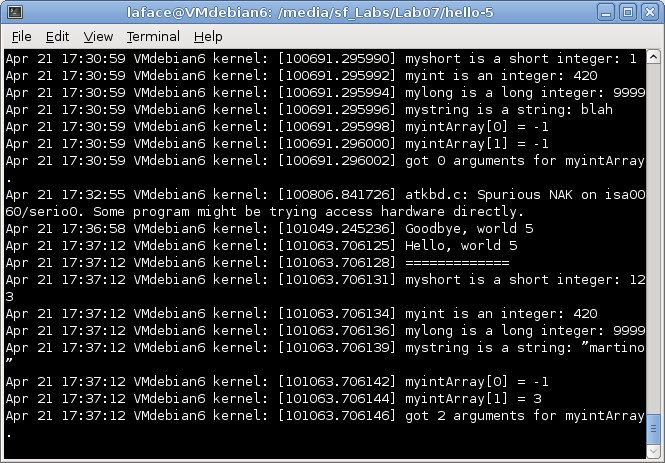
After the installation of the module, the folder /sys/module/hello-5/ contains the following:



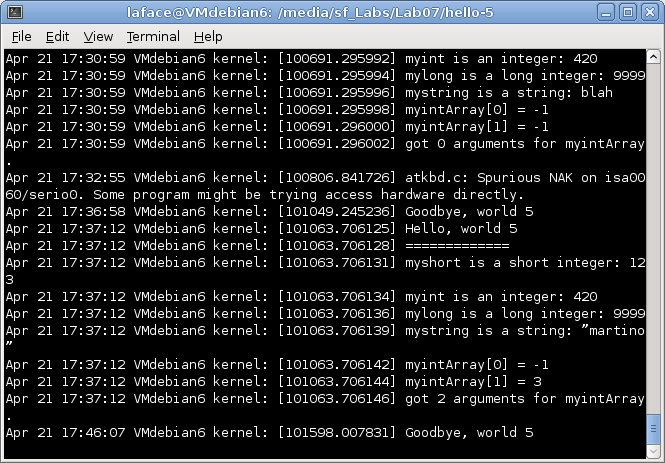
In the subfolder parameters there is a file for each parameter, that contains the value set on installation or the default values:



To see the messages that the module emitted on installation, there are different ways, since the messages are written in /var/log/messages. For example, by using the command dmesg | tail or some other text file visualizer. A more convenient way is by using the command tail -f /var/log/messages into another terminal window, so that new messages are shown immediately.

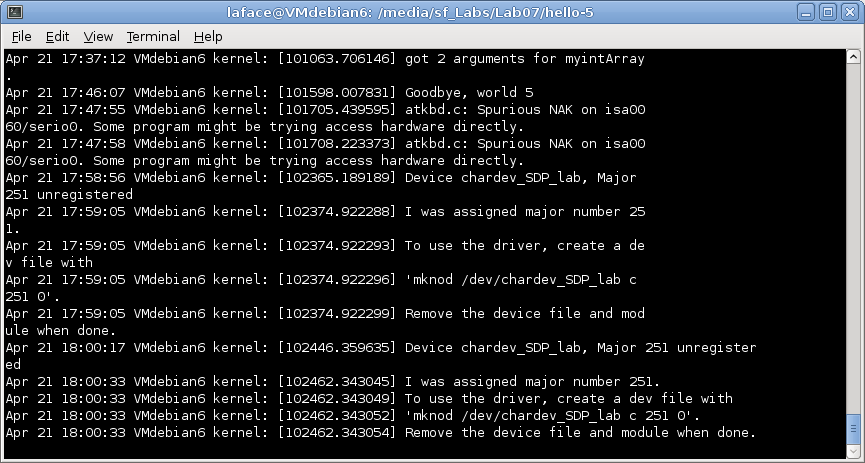


Then I proceeded with the removal of the module, by giving the command rmmod hello-5. A new line appears on the monitor window:



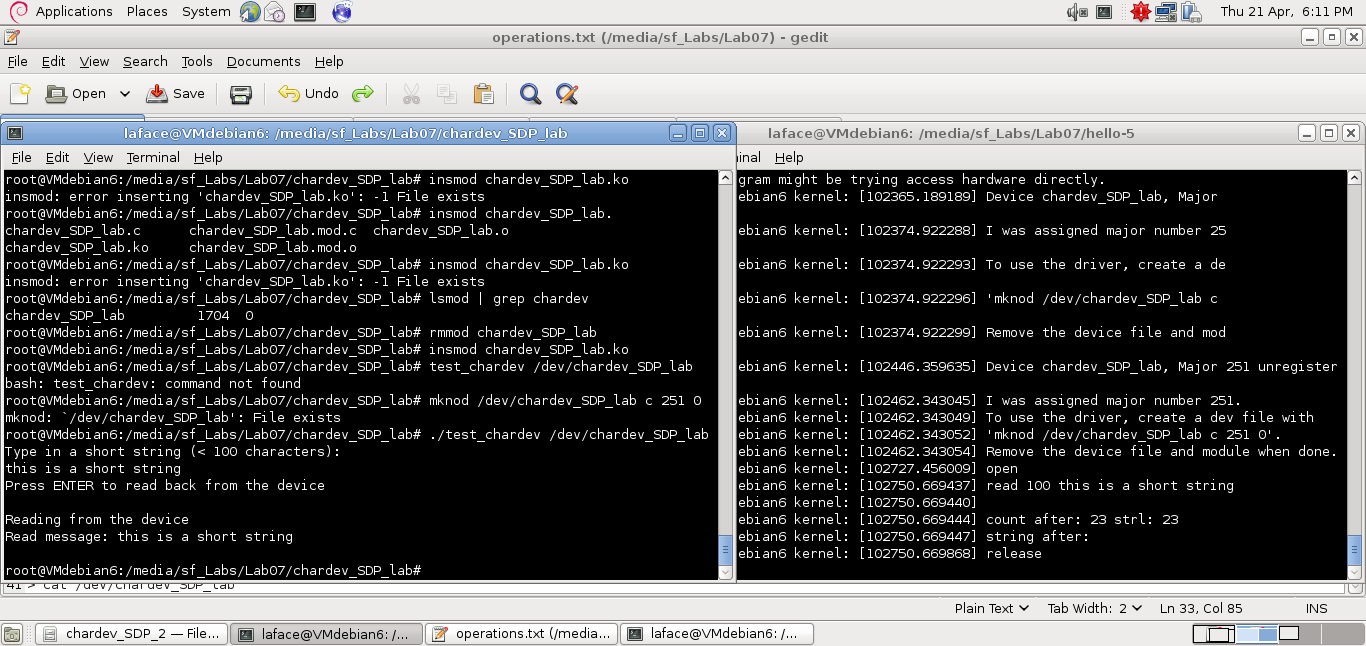
# Exercise 2

For the second exercise the module is chardev\_SDP\_lab. The steps for compilation, installation and uninstallation are the same. The difference is that in order to use this module, we need to create a special file that represents this character device. The special file is created in the /dev folder with a specific major number and minor number. Those values are determined during the installation of the module, using the system call alloc\_chrdev\_region and other defines included in some system libraries. The installation procedure writes some messages in /var/log/messages that provide to us the major and minor number to create the special file:

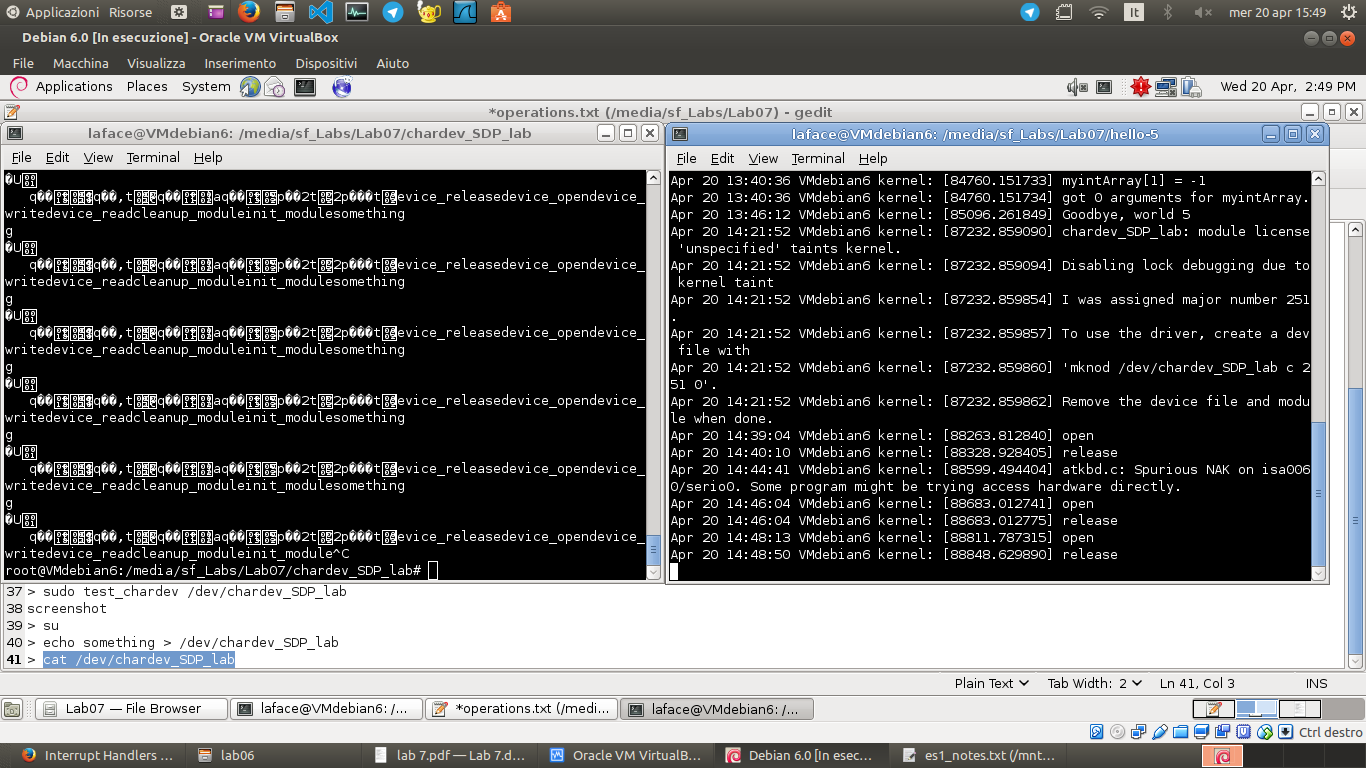


To create this special file I used the command given in the message.

After the compilation of the test program (gcc -g -o test\_chardev test\_chardev.c -Wall), if I execute test\_chardev /dev/chardev\_SDP\_lab I can see that the device is working:



If instead I use another way of reading from the device (echo something > /dev/chardev\_SDP\_lab ; cat /dev/chardev\_SDP\_lab), something strange happens:



The purpose of the exercise is to correct the device\_read function of the module.

The problem is that, when cat reads from the device, it asks for 32768 bytes, and the device replies with all of them (reading also after the buffer) and always returns a value greater than 0.

The modifications I did in the code of this function do the following: compute the string length, copy to the user buffer only this number of bytes and then clear the buffer. The buffer cleaning is necessary, so that next read execution computes a string length equal to 0 and returns 0. In this way the reading process (receiving 0 when performs the second read) stops its read loop.

