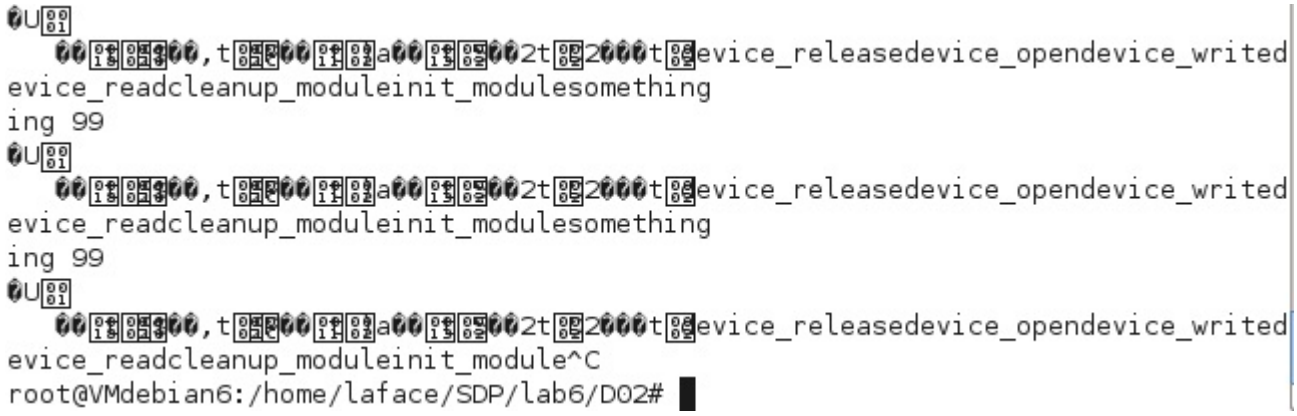
Lab 6 Ex 2 - SDP - Davide Gallitelli S241521

The code of the original *chardev_SDP_Lab* driver, provided among the sources for this lab, generates an unexpected result from the command *cat*, due to a problem in the reading of the special file */dev/chardev_SDP_lab*.



However, the file can be correctly read by means of the provided *test_chardev.c* file:

```
root@VMdebian6:/home/laface/SDP/lab6/D02# mknod /dev/chardev_SDP_lab c 251 0
root@VMdebian6:/home/laface/SDP/lab6/D02# ./test_chardev /dev/chardev_SDP_lab
Type in a short string (< 100 characters):
myShortString 99
Press ENTER to read back from the device

Reading from the device
Read message: myShortString 99

root@VMdebian6:/home/laface/SDP/lab6/D02#
```

This is caused by *cat* behaviour, which tries to read 32K worth of data from the file specified as its argument, and the driver does not manage in any way the file dimension and EOF.

In order to solve this, a global variable has been introduced in the driver, *bytes_written*, which is updated with the number of bytes whenever the device is opened in order to be written by a program. The *device_write* method now contains:

```
if (bytes_written != 0) bytes_written--;
ret = copy_from_user(char_dev_buf+bytes_written, buff, count);
bytes_written += count-ret;
return count;
```

The decrement operator of the first line is there to allow for multiple data insert: the "offset" represented by the global variable is moved one step backward so that the CR character is overwritten. Then, it is incremented, after the *copy_from_user* function by the *count* value, which represents the actual bytes written, minus *ret*, which are the bytes that could not be written, if any.

Then, this value is used in the *device_read* method:

```
unsigned long ret;
count = (size_t) bytes_written;
printk(KERN_INFO "My count is: %d\n", count);
ret = copy_to_user(buff, char_dev_buf, count);
printk(KERN_INFO "Number of bytes that could not be copied: %lu \n", ret);
bytes_written = 0;
return count;
```

The variable *bytes_written* is used to update the *count* variable, which is used to know how many bytes are to be read from the file. After the read is done, it is then reset to 0, so that no more bytes will be read at the next iteration.

A proof that the program is working correctly is in the following screenshot:

```
root@VMdebian6:/home/laface/SDP/lab6/D02# !ech
echo " hello" > /dev/chardev_SDP_lab
root@VMdebian6:/home/laface/SDP/lab6/D02# echo " hello" > /dev/chardev_SDP_lab
root@VMdebian6:/home/laface/SDP/lab6/D02# echo " hello" > /dev/chardev_SDP_lab
root@VMdebian6:/home/laface/SDP/lab6/D02# echo " hello" > /dev/chardev_SDP_lab
root@VMdebian6:/home/laface/SDP/lab6/D02# cat /dev/chardev_SDP_lab
 hello hello hello hello
```