Exercise sheet 14 – Drivers

Goals:

- Build a driver
- Load/unload a driver

Exercise 14.1: Using Drivers

Change into the directory `OS_exercises/sheet_14_drivers`
You may need admin privileges for some of the exercises.

(a) Check if `scull` the scull module/driver is already loaded. There are two possible methods for this: a command or the file `/proc/devices`

(b) Build the `scull` driver with the included `Makefile`.

(c) Open an additional shell and open (+follow) the kernel log `/var/log/kern.log` with `tail -f /var/log/kern.log`

(d) Load the module `scull.ko` into the system.

(e) Did something happen in the kernel log? Read the `major device number` which was printed there.

(f) Check if the module is now loaded inside the system.

(g) Create the two device files `scull0` and `scull1`.

(h) Change the owner of both device files to `dev` with

```
sudo chown dev scull0 scull1
```

(i) Look at the source code of `mycat.c`. What does this program?

(j) Compile `mycat.c` into `mycat`

(k) Write a string to `scull0`, use `mycat` to copy the string from `scull0` to `scull1`. Then read the content of `scull1`.

```
echo "This is a test!" > scull0
./mycat scull0 > scull1
./mycat scull1
```

(l) Was anything written to the log file?

(m) Unload the driver module! Check if it is now unloaded!

(n) Remove both device files!

Exercise 14.2: Look into the module sources

To answer these questions, look inside `scull.c`

(a) Which user functions are supported by the driver?

(b) Where does the system get to know about the supported functions?

(c) How and where is the memory for the device created?

(d) How and where is the memory for the device released?

(e) Which function is used to print messages into the logfile?

(f) Why is it not possible to read the same data twice from the device?

(g) How is this type of reading called?

(h) How is it avoided that the module is being unloaded while it is in use?

(i) In which mode is the module running? In kernel (system mode) or user (user mode) space?