

# System Programming – Project 1

You should implement a character device driver module that will play the board game "Master Mind". In this two-player game, one of the players (A) chooses a secret 4-digit number and the other player (B) tries to guess what it is. For each guess, the first player (player A) reports the number of digits correctly in-place (represented by a +) and the number of digits which are out-of-place (represented by a -). For example, if the secret number is "4283" and the guess is "1234" the report is "1+ 2-". The device will play the role of player A, the player who chooses the secret number.

Each write operation on the device should be interpreted as a guess by player B. It will be evaluated by the driver and the report will be stored in an internal buffer in the form "xxxx m+ n- abcd\n" (total length: 16 bytes), where "xxxx" is the guess, "m" is the number of in-place digits, "n" is the number of out-of-place digits and "abcd" is the number of guesses so far. The total size of the buffer will be 4096 bytes (therefore maximum 4096/16=256 lines). Each read operation on the device will be a regular read from the internal buffer.

The device node will be named "/dev/mastermind". The major number will be a module parameter with the default value 0 (dynamic selection). The minor number will always be 0. The secret number will be a module parameter to be specified as in:

```
$ insmod ./mastermind.ko mmind_number="4283"
```

Another module parameter will be the allowed number of guesses (mmind\_max\_guesses, default 10). If the maximum guess count is reached writing to the device will not be allowed.

After that, the game can be tested as follows:

```
$ echo "1234" > /dev/mastermind
$ cat /dev/mastermind
1234 1+ 2- 0001
$ echo "4513" > /dev/mastermind
$ cat /dev/mastermind
1234 1+ 2- 0001
4531 1+ 1- 0002
```

Also, three ioctl commands will be implemented:

1. MMIND\_REMAINING: returns number of remaining guesses
2. MMIND\_ENDGAME: clears the guess history and guess count
3. MMIND\_NEWGAME: starts a new game with a new secret value as parameter

You are expected to submit a ZIP archive containing the code for the driver module through Ninova.

---

## Important notes!

- Pay attention to the general guidelines for projects.
- You are required to submit the following files through the Ninova system as a zip file containing source codes of your test programs and information on how to use the program (if required).
- Make sure to return a meaningful error code when an error occurs.

- Don't blindly copy any code from other sources. (e.g. quantum related methods and declarations in scull. )
- This is a **group project**. However, each member of the team **must make an individual submission**, even though the submitted files are the same for all members.
- Team members will be graded individually based on their performance in the lab/demo session as well as based on the project submitted. Students who are not present during the lab session will not receive a grade for the project, even though they may have made a submission through the Ninova system.

Any form of cheating or plagiarism will not be tolerated. The submitted work should be the work of the team; collaboration or code sharing between different teams will be regarded as cheating. Cheating also includes actions such as, but not limited to, submitting the work of others as one's own (even if in part and even with modifications) and copy/pasting from other resources, including Internet resources, (even when attributed). Serious offenses will be reported to the administration for disciplinary measures.