# **hw2secws documentation - ori petel**

The submission consists 3 parts:

* /module: The kernel space code – "hw1secws.c", Makefile
* /user: The user space code – "user.c", Makefile
* Dry documentation (this file)

In the kernel space code, I followed the example Reuven shared with us, with 2 exeptions:

1. In the "sysfs show" implementation I passed the 2 unsigned integers more efficiently (concerning time and space) than the method in the example.

(Reuven mentioned that this is very important when kernel programing).

In Reuven's example, the data is delivered in human-readable format, while I deliver the data in binary format. (The integers are kind of hard-coding into string chars).

1. I also implemented the error handling more efficiently, using goto statements.

This cleaned many redundancy code lines and made the code more "kernel-like".

I chose to implement the user space program with **c language**.

In case of no input, I read the counters from /sys into local variables. And then print them in the required format.

In case of '0' input, I write into /sys the '\*' string and reset the counters to 0.  
Note: I implemented the device in a time efficient way, such that every string given as input to the device and starts with '\*' , will reset the counters.

If there is a file error, we exit and report.

If the arguments are invalid, we also exit and report.

I Hope You Will Enjoy My Code :)