# Convert this docx to PDF before submitting it

*Last updated: 2/1/2024*

# Exception levels, out and back

1. (40) Trigger an exception at EL1, handle it at EL2, and return to EL1. Modify the kernel source (all the exp3 code given to the student) to implement the behavior.

**Note:** make sure you understand how stack pointers (SP) work for EL1 and for EL2. Many of the bugs are related to not having proper EL2 stack pointer.

|  |
| --- |
| File name: **[computingid]-2.tar.gz.** e.g. lg8sp-2.tar.gz. All lower case. Use the command 'tar -czvf' to generate.  (Optional) A file README-cs4414.txt, which contains anything TA should know (e.g., any caveats of your code that TA shall be aware of, extra build instructions)  Absolutely no any binaries (e.g. \*.elf, \*.o, \*.bin) or .git/ subdirectory in the tarball. |

1. (20) In a paragraph, describe the exception you plan to trigger, what EL2 exception handlers you will implement, and how to demonstrate your implementation correctness.

I plan to trigger a synchronized exception by calling the “hvc” command which will automatically call an exception and in el1 and take it up to el2. I will implement the sync\_invalid\_el1 in the vectors table for el2 which will simply call eret. Once eret is called, it will cause the exception to go back to el1 because it will have looked at the spsr which stores the el1 spsr value. Additional details include setting the sp in boot.s to 0x40000000 and setting vbar\_el2 register.

I will demonstrate correctness using qemu.log where I will check that an exception is called in el1, that the exception transfers to el2, and goes back to el1.

Changelog:

2/1/24: removed the timer question