# CS2610
# Lab 5 : Privilege Switching and Timer Interrupt in RISC-V

## Problem 1: Switching between Machine mode, Supervisor mode, and User mode (100 points)

Write an assembly program continuously switching between Machine mode, Supervisor mode, and User mode ($M \rightarrow S \rightarrow U \rightarrow S \rightarrow M \rightarrow S \rightarrow U$). The flow of the program should be as follows:

1. The program should start in Machine mode, modify the required *csrs* (Control and Status Registers), and returns to Supervisor mode using the *mret* instruction.
2. In Supervisor mode, set up the necessary CSRs to return to User mode using the *sret* instruction.
3. Perform some arbitrary operations in User mode and then call *ecall*. This should raise a trap in Supervisor mode.
4. Write a trap handler in Supervisor mode that reads the values of registers *scause* and *sepc* and then makes an *ecall*. This ecall should be delegated to Machine mode. After return from *ecall*, the trap handler should return back to User mode.
5. Write a trap handler in Machine mode that reads the values of registers *mcause* and *mepc* and then return to the updated value of *mepc*. This should return to the next instruction in Supervisor Trap Handler which returns back to User mode.
6. The User mode will again call *ecall* and the program will continue doing this in a loop.

**Note 1:** Make sure the *medeleg* register is correctly set so that exceptions from User space are delegated to Supervisor and exceptions from Supervisor space are delegated to Machine.

**Note 2:** Read the values in mepc, mcause, mtvec, mstatus, sepc, scause, and stvec registers while in different modes and submit the screenshots.

**Note 3:** When you run your code with the proxy kernel (pk), it is executed in User mode but for the code to execute in Machine mode, you will have to run your program without pk. The commands to compile and run your code are given below.

```
$riscv64-unknown-elf-gcc -nostartfiles -T link.ld <your_program>.s
$spike -d a.out
```

The linker file *link.ld* is provided.

For simplicity, assume that the user code, supervisor code and machine code are all parts of the same program (implemented using different functions). A template for the same program is given below:

```
.section .text
.global main

main:
  #modify necessary csrs : mstatus, mepc, mtvec, medeleg
  mret

 mtrap_handler:
    #read registers like mepc, mcause
    #update the value of mepc
    #return

  scode:

    #modify necessary csrs : sepc, stvec
    #ensure control returns to user code
     sret


 strap_handler:
      #read registers like sepc, scause
       ecall
      #return

 ucode:
   #do some user space operations
    ecall
```

## What you need to submit:

1. Code files
2. Screenshots of the output

**Note:**

1. All the files should be submitted in a zipped folder through Microsoft Teams.
2. The zipped folder should be named $< Roll_No >$_Lab5.zip.

## Resources

https://people.eecs.berkeley.edu/~krste/papers/riscv-privileged-v1.9.pdf