In this project, you need to understand the *read & witre* system call in Minix and add four new system calls to analysis the performance of *read & witre.*

Due: Oct 17, 2022

1. **Trace the *read* system call in Minix. You need to list all the functions and files involves when a program calls the read() system call step by step and explain what happens in each function.**
2. **Add four system calls,**
    1. **int trapCounter() : count and return the number of traps. When a trap occurs, the trap counter will increase by 1.**
    2. **void initTrapCounter() : reset the trap counter to 0**
    3. **int msgCounter() : count and return the number of messages the Minix system receives**
    4. **void initmsgCounter() : reset the the message counter to 0**
    5. **Use project 1's code to test these four system calls. Collect trap number and message number data from each measurement experiments.**
3. **You need to write a design document to describe how you implement these four system calls (no more than 2 pages, at least font size 11)**
4. **Draw one or more figures to present your data. Do you find new stuff from your data?**
5. **Propose and present a design to improve the "read()" & "write()" system call performance in Minix (at most 3 pages, minimum font size 11)**
6. **Submit the following files in the *Blackboard*:**
    1. Source code of your programs
    2. Any other executable and shell to test your programs in each OSs
    3. A readme on how to use and test your program
    4. A document include your design, data and explanation.
    5. A contribution list for each group member
    6. Only one assigment answer need to be uploaded in each team
    7. The format of the upload file is "project2-group#your_group_number.zip"
7. **You will be graded on the robustness of your program, and the grade for all the group members will be the same.**

- Working code (50%)
- Documentation

- Trace Of read() & write() (10%)
- Design of The Four New System Calls (10%)
- Present Data (10%)
- Proposed Design of The New read() & write() System Calls (20%)