CSE411 Lab 3

Systick

- **Separate systick.c and systick.h files**
- Systick timer using interrupts, firing interrupt each 5 ms.
- Systick timer interrupt handler raises a flag and increases a counter by one (used for tasks periodicities and for the scheduler to determine the task that will be executed).

Scheduler

- **Separate scheduler.c and scheduler.h files**
- Scheduler having two functions.
- The first function: create_task, will return nothing but will take 2 arguments; pointer to function (tasks you will define in main.c), and its periodicity in ms, those 2 parameters will be updated in an array of struct (osThread). This structure has 2 parameters: pointer to function, and periodicity. Note that the structure is defined in scheduler.h file
- The second function: Tasks_Scheduler, will return nothing and will take no parameters. It is implemented inside a while(1) loop. It checks for the flag raised by the systick interrupt handler function.
- When flag raised (first do not forget to clear the flag), it means that 5 ms have passed, thus scanning for all tasks will occur to check which task will be executed. Tasks will be called using the array of structs previously defined and initialized.
- Note that this is a non-preemptive scheduler, so inside the while (1) loop, scanning the tasks will be considered as a critical section, so you must disable interrupts before, and enable interrupts after.
- NUM_OF_TASKS will be a macro defined in scheduler.h and will be taking the number of tasks your created. It will be used for tasks scanning in Tasks_Scheduler function.
- Note that your scheduler MUST BE GENERIC, NUM_OF_TASKS only can be modified, nothing else.

Main

- **Main.c file**
- Main function will call systick Initialization function,

create_task function (that will be called 3 times as you have 3 tasks), tasks_scheduler function, and while (1) loop that should never be reached.

- The file will also have the definitions of your 3 tasks.