Embedded OS Implementation, Fall 2023

Homework #1 (due October 4th, 2023 (Wednesday) 12:00)

Hello uCOS-II

Problem Definition:

(a) Please draw the system flow of "Hello μC/OS-II (the modified main.c in Lab1)" and explain the process (functions). **Note: Please start from the function "OSTaskCreateExt".**

(b) Consider two periodic tasks (τ_1 ,τ_2) and their delay time are 3 ticks and 5 ticks, respectively. Task priority of two tasks (τ_1 ,τ_2) are 1 and 2, respectively. Please add some code to the uCOS-II scheduler **in the kernel level** to observe how CPU is switched among tasks by means of context switches.

Tick	CurrentTask ID	NextTask ID	Number of ctx switch
##	**********	task(ID)(job number)	##
##	task(ID)(job number)	task(ID)(job number)	##

※ If the task is Idle Task, print "task(priority)".

This project is executed on "Visual Studio". Please show the results by using it.

```
Tick    CurrentTask ID          NextTask ID             Number of ctx switches
0       **********              task( 1)( 0)             0
0       task( 1)( 0)            task( 2)( 0)             0
0       task( 2)( 0)            task(63)                 1
3       task(63)                task( 1)( 1)             2
3       task( 1)( 1)            task(63)                 3
5       task(63)                task( 2)( 1)             4
5       task( 2)( 1)            task(63)                 5
6       task(63)                task( 1)( 2)             6
6       task( 1)( 2)            task(63)                 7
9       task(63)                task( 1)( 3)             8
9       task( 1)( 3)            task(63)                 9
10      task(63)                task( 2)( 2)            10
10      task( 2)( 2)            task(63)                11
12      task(63)                task( 1)( 4)            12
12      task( 1)( 4)            task(63)                13
15      task(63)                task( 1)( 5)            14
15      task( 1)( 5)            task( 2)( 3)            15
15      task( 2)( 3)            task(63)                16
18      task(63)                task( 1)( 6)            17
18      task( 1)( 6)            task(63)                18
20      task(63)                task( 2)( 4)            19
20      task( 2)( 4)            task(63)                20
21      task(63)                task( 1)( 7)            21
21      task( 1)( 7)            task(63)                22
24      task(63)                task( 1)( 8)            23
24      task( 1)( 8)            task(63)                24
25      task(63)                task( 2)( 5)            25
25      task( 2)( 5)            task(63)                26
27      task(63)                task( 1)( 9)            27
27      task( 1)( 9)            task(63)                28
30      task(63)                task( 1)(10)            29
30      task( 1)(10)            task( 2)( 6)            30
30      task( 2)( 6)            task(63)                31
```

The output result is below:

Output.txt

```
0       ***********             task( 1)( 0)            0
0       task( 1)( 0)            task( 2)( 0)            0
0       task( 2)( 0)            task(63)                1
3       task(63)                task( 1)( 1)            2
3       task( 1)( 1)            task(63)                3
5       task(63)                task( 2)( 1)            4
5       task( 2)( 1)            task(63)                5
6       task(63)                task( 1)( 2)            6
6       task( 1)( 2)            task(63)                7
9       task(63)                task( 1)( 3)            8
9       task( 1)( 3)            task(63)                9
10      task(63)                task( 2)( 2)            10
10      task( 2)( 2)            task(63)                11
12      task(63)                task( 1)( 4)            12
12      task( 1)( 4)            task(63)                13
15      task(63)                task( 1)( 5)            14
15      task( 1)( 5)            task( 2)( 3)            15
15      task( 2)( 3)            task(63)                16
18      task(63)                task( 1)( 6)            17
18      task( 1)( 6)            task(63)                18
20      task(63)                task( 2)( 4)            19
20      task( 2)( 4)            task(63)                20
21      task(63)                task( 1)( 7)            21
21      task( 1)( 7)            task(63)                22
24      task(63)                task( 1)( 8)            23
24      task( 1)( 8)            task(63)                24
25      task(63)                task( 2)( 5)            25
25      task( 2)( 5)            task(63)                26
27      task(63)                task( 1)( 9)            27
27      task( 1)( 9)            task(63)                28
30      task(63)                task( 1)(10)            29
30      task( 1)(10)            task( 2)( 6)            30
30      task( 2)( 6)            task(63)                31
```

Crediting :

Your homework need to show the following information.

- The system flow and the explanation of the process(functions). **(45%)**
- The screenshot of the result. **(10%)**
- A report that describes your implementation (please attach the screenshot of the code and **Mark** the modified part). **(45%)**

Hints:

1. Call the function **OSTimeSet(0)** before the OS starts to initialize the start time.
2. Use **OSTimeGet()** to get the current tick in the system.
3. Use '**/t**' to format your code.
4. If your project size is too large for uploading, you can try to delete the ".vs" or the "Debug" folders

Homework submit:

Submit to Moodle

Submit deadline : October 4th, 2023 (Wednesday) 12:00

File name format : RTOS_ your student ID_HW1.zip

RTOS_ Student ID_HW1.zip includes :

※ The report (RTOS_ your student ID_HW1.**pdf**).

※ Folder with executable µC/OS-II project (**Myyyddxxx_RTOS_HW1**).

※ Standard input and output filenames in the project are necessary for the checker, please check before submitting.

```
#define INPUT_FILE_NAME "./TaskSet.txt"
#define OUTPUT_FILE_NAME "./Output.txt"
```

※ Plagiarizing is strictly prohibited.

※ RTOS_Myyyddxxx_HW1.zip must be including files as follow: