Embedded OS Implementation, Fall 2021
Project #1 (due Nov. 10, 2021 (Wednesday) 13:00)

[PART I] Task Control Block Linked List

Objective:

Following the previous homework (HW1), please add some code to the μC/OS-II scheduler **in the kernel level** to observe the operations of the task control block (TCB) and TCB linked list.

※ The TCB address is dynamic.

The output results are shown below:

```
OSTick    created, Thread ID 22576
Task[ 63] created, TCB Address 009BF640
------After TCB[63] being linked------
Previous TCB point to address 00000000
Current  TCB point to address 009BF640
Next     TCB point to address 00000000

Task[  1] created, TCB Address 009BF698
------After TCB[1] being linked------
Previous TCB point to address 00000000
Current  TCB point to address 009BF698
Next     TCB point to address 009BF640

Task[  2] created, TCB Address 009BF6F0
------After TCB[2] being linked------
Previous TCB point to address 00000000
Current  TCB point to address 009BF6F0
Next     TCB point to address 009BF698

==================== TCB linked list ====================
Task    Prev_TCB_addr   TCB_addr        Next_TCB_addr
2       00000000        009BF6F0        009BF698
1       009BF6F0        009BF698        009BF640
63      009BF698        009BF640        00000000
```

[PART II] RM Scheduler Implementation

Objective:

To implement the Rate Monotonic (RM) scheduler for periodic tasks and observe the scheduling behaviors.

Problem Definition:

Implement the following three task sets of periodic tasks. Add necessary code to the μC/OS-II scheduler **in the kernel level** to observe how the task suffers from the scheduler. We give the files for the parameter of the task.

Periodic Task Set = {τ_{ID} (ID, arrival time, execution time, period)}

Example Task Set 1 = {τ_1 (1, 0, 2, 4), τ_2 (2, 0, 3, 8)}

Example Task Set 2 = {τ_1 (1, 0, 6, 15), τ_2 (2, 0, 3, 6)}

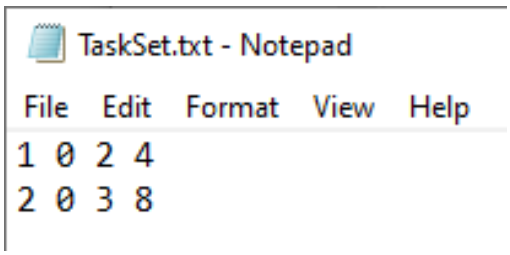
Example Task Set 3 = {τ_1 (1, 1, 1, 5), τ_2 (2, 0, 8, 15), τ_3 (3, 2, 2, 10)}

※ The priority of the task is set according to the RM scheduling rules.

The input file format:

Task ID	Arrive Time	Execution Time	Task Periodic
##	##	##	##

Example of task set file 1:



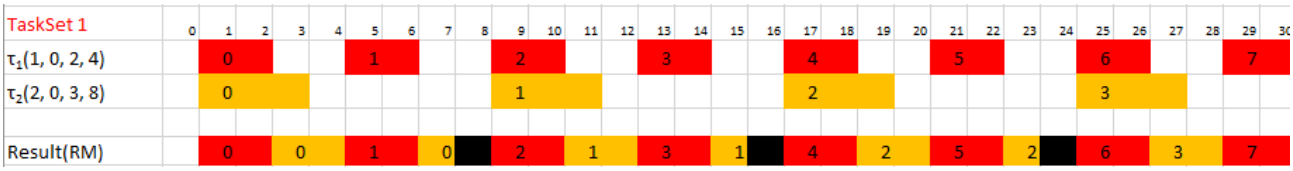
Evaluation:

The output format:

Tick	Event	CurrentTask ID	NextTask ID	Response Time	# of ContextSwitch	Preemption Time
##	**Preemption**	**task(ID)(job number)**	**task(ID)(job number)**			
##	**Completion**	**task(ID)(job number)**	**task(ID)(job number)**	##	##	##
##	**MissDeadline**	**task(ID)(job number)**	----------			

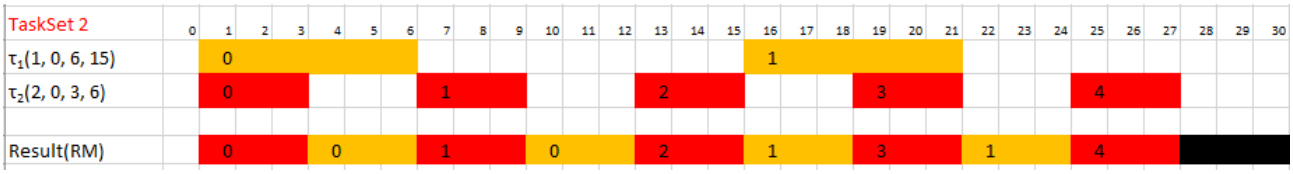
※ If the task is Idle Task, print "*task(priority)*".

The output results of Task Set 1:



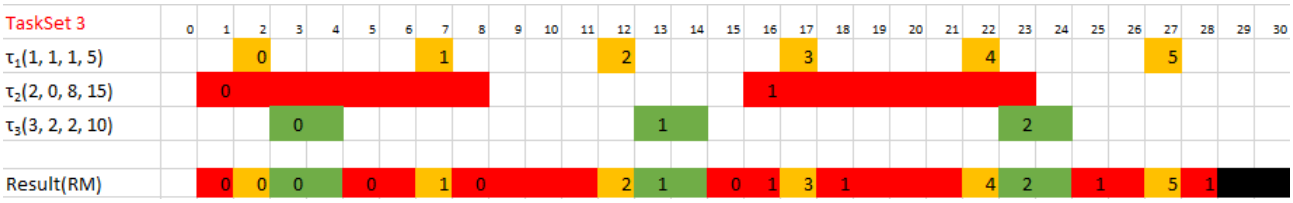
Tick	Event	CurrentTask ID	NextTask ID	ResponseTime	# of ContextSwitch	PreemptionTime
2	Completion	task(1)(0)	task(2)(0)	2	1	0
4	Preemption	task(2)(0)	task(1)(1)			
6	Completion	task(1)(1)	task(2)(0)	2	2	0
7	Completion	task(2)(0)	task(63)	7	4	2
8	Preemption	task(63)	task(1)(2)			
10	Completion	task(1)(2)	task(2)(1)	2	2	0
12	Preemption	task(2)(1)	task(1)(3)			
14	Completion	task(1)(3)	task(2)(1)	2	2	0
15	Completion	task(2)(1)	task(63)	7	4	2
16	Preemption	task(63)	task(1)(4)			
18	Completion	task(1)(4)	task(2)(2)	2	2	0
20	Preemption	task(2)(2)	task(1)(5)			
22	Completion	task(1)(5)	task(2)(2)	2	2	0
23	Completion	task(2)(2)	task(63)	7	4	2
24	Preemption	task(63)	task(1)(6)			
26	Completion	task(1)(6)	task(2)(3)	2	2	0
28	Preemption	task(2)(3)	task(1)(7)			
30	Completion	task(1)(7)	task(2)(3)	2	2	0

The output results of Task Set 2:



Tick	Event	CurrentTask ID	NextTask ID	ResponseTime	# of ContextSwitch	PreemptionTime
3	Completion	task(2)(0)	task(1)(0)	3	1	0
6	Preemption	task(1)(0)	task(2)(1)			
9	Completion	task(2)(1)	task(1)(0)	3	2	0
12	Completion	task(1)(0)	task(2)(2)	12	4	3
15	Completion	task(2)(2)	task(1)(1)	3	2	0
18	Preemption	task(1)(1)	task(2)(3)			
21	Completion	task(2)(3)	task(1)(1)	3	2	0
24	Completion	task(1)(1)	task(2)(4)	9	4	3
27	Completion	task(2)(4)	task(63)	3	2	0
30	Preemption	task(63)	task(2)(5)			

The output results of Task Set 3:



Tick	Event	CurrentTask ID	NextTask ID	ResponseTime	# of ContextSwitch	PreemptionTime
1	Preemption	task(2)(0)	task(1)(0)			
2	Completion	task(1)(0)	task(3)(0)	1	2	0
4	Completion	task(3)(0)	task(2)(0)	2	2	0
6	Preemption	task(2)(0)	task(1)(1)			
7	Completion	task(1)(1)	task(2)(0)	1	2	0
11	Preemption	task(2)(0)	task(1)(2)			
12	Completion	task(1)(2)	task(3)(1)	1	2	0
14	Completion	task(3)(1)	task(2)(0)	2	2	0
15	Completion	task(2)(0)	task(2)(1)	15	6	7
16	Preemption	task(2)(1)	task(1)(3)			
17	Completion	task(1)(3)	task(2)(1)	1	2	0
21	Preemption	task(2)(1)	task(1)(4)			
22	Completion	task(1)(4)	task(3)(2)	1	2	0
24	Completion	task(3)(2)	task(2)(1)	2	2	0
26	Preemption	task(2)(1)	task(1)(5)			
27	Completion	task(1)(5)	task(2)(1)	1	2	0
28	Completion	task(2)(1)	task(63)	13	7	5
30	Preemption	task(63)	task(2)(2)			

[Bonus] Miss Deadline Handling

Objective:

Implement the RM scheduler for periodic tasks, and handle the miss deadline behaviors.

Problem Definition:

Implement and describe how to handle the deadline missing situation under RM.

Example Task Set 4 = {τ_1 (1, 1, 2, 6), τ_2 (2, 0, 6, 15), τ_3 (3, 2, 3, 10)}

※ The priority of the task is set according to the RM scheduling rules.

Credit:

[PART I] Task Control Block Linked List [20%]

- The screenshot results. (10%)
- A report that describes your implementation (please attach the screenshot of the code and **MARK** the modified part). (10%)

[PART II] RM Scheduler Implementation [80%]

- The correctness of schedule results of examples. Note the testing task set might not be the same as the given example task set. (40%)
- A report that describes your implementation (please attach the screenshot of the code and **MARK** the modified part). (40%)

[Bonus I] Miss Deadline Handling [10%]

- The correctness of schedule results of examples. Note the testing task set might not be the same as the given example task set. (5%)
- Implement and describe how to handle the deadline missing situation under RM. (5%)

※ You must modify the source code!

※ Standard input and output filenames in the project are necessary for the checker. Please check the file names before submitting.

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

※ Please set the system end time as 30 seconds in this project.

```
#define SYSTEM_END_TIME 30
```

※ We will use different task sets to verification your code.

※ When the current task is completed, the completion information shall be printed even there is one task missing its deadline.

Project submit:

Submit to Moodle.

Submit deadline: Nov. 10, 2021 (Wednesday) 13:00

File name format: RTOS_Myyyddxxx_PA1.zip

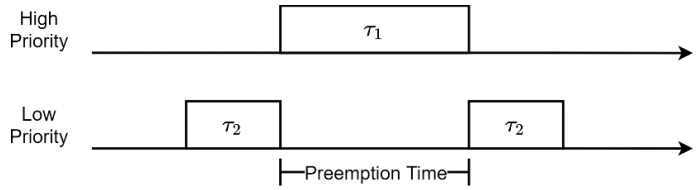
RTOS_Myyyddxxx_PA1.zip includes (The tree structure of files is shown as hints):

- The report (RTOS_Myyyddxxx_PA1.pdf).
- Folder with the executable µC/OS-II project (RTOS_Myyyddxxx_PA1).

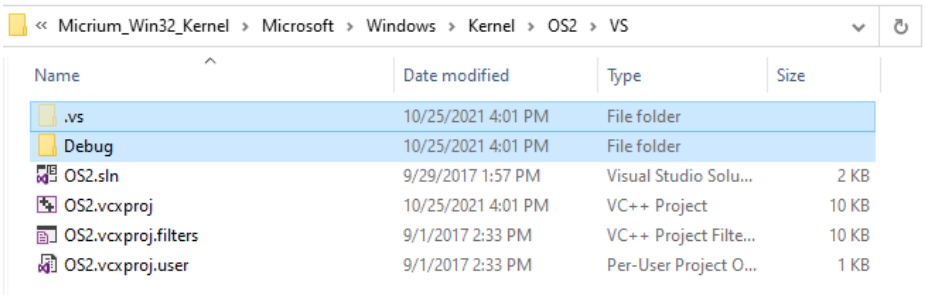
※ Plagiarizing is strictly prohibited.

Hints:

1. Preemption time is introduced in multiple tasking.



2. If your project size is too large for uploading, you can try to delete the ".vs" or the "Debug" folders.



3. RTOS_Myyyddxxx_PA1.zip include files as follow: