RTOS-HW1

學號：M11007326 姓名：黃鈞臨

[PART I] EDF Scheduler Implementation [70%]

● **The correctness of schedule results of examples. Note the testing task set might not be the same as the given example task set. (20%)**

Example Task Set 1 = {τ_1 (1, 0, 4, 11), τ_2 (2, 0, 3, 9)}:

Tick	Event	CurrentTask ID	NextTask ID	ResponseTime	#of ContextSwitch	PreemptionTime	OSTimeDly
3	Completion	task(2)(0)	task(1)(0)	3	1	0	6
7	Completion	task(1)(0)	task(63)	7	2	0	4
9	Preemption	task(63)	task(2)(1)				
12	Completion	task(2)(1)	task(1)(1)	3	2	0	6
16	Completion	task(1)(1)	task(63)	5	2	0	6
18	Preemption	task(63)	task(2)(2)				
21	Completion	task(2)(2)	task(63)	3	2	0	6
22	Preemption	task(63)	task(1)(2)				
26	Completion	task(1)(2)	task(63)	4	2	0	7
27	Preemption	task(63)	task(2)(3)				
30	Completion	task(2)(3)	task(63)	3	2	0	6
33	Preemption	task(63)	task(1)(3)				
37	Completion	task(1)(3)	task(2)(4)	4	2	0	7
40	Completion	task(2)(4)	task(63)	4	2	0	5

Example Task Set 2 = {τ_1 (1, 0, 2, 6), τ_2 (2, 0, 3, 8)}:

Tick	Event	CurrentTask ID	NextTask ID	ResponseTime	#of ContextSwitch	PreemptionTime	OSTimeDly
2	Completion	task(1)(0)	task(2)(0)	2	1	0	4
5	Completion	task(2)(0)	task(63)	5	2	0	3
6	Preemption	task(63)	task(1)(1)				
8	Completion	task(1)(1)	task(2)(1)	2	2	0	4
11	Completion	task(2)(1)	task(63)	3	2	0	5
12	Preemption	task(63)	task(1)(2)				
14	Completion	task(1)(2)	task(63)	2	2	0	4
16	Preemption	task(63)	task(2)(2)				
18	Preemption	task(2)(2)	task(1)(3)				
20	Completion	task(1)(3)	task(2)(2)	2	2	0	4
21	Completion	task(2)(2)	task(63)	5	4	2	3
24	Preemption	task(63)	task(1)(4)				
26	Completion	task(1)(4)	task(2)(3)	2	2	0	4
29	Completion	task(2)(3)	task(63)	5	2	0	3
30	Preemption	task(63)	task(1)(5)				
32	Completion	task(1)(5)	task(2)(4)	2	2	0	4
35	Completion	task(2)(4)	task(63)	3	2	0	5
36	Preemption	task(63)	task(1)(6)				
38	Completion	task(1)(6)	task(63)	2	2	0	4
40	Preemption	task(63)	task(2)(5)				

Example Task Set 3 = {τ_1 (1, 0, 2, 5), τ_2 (2, 0, 4, 8), τ_3 (3, 1, 2, 6)}:

Tick	Event	CurrentTask ID	NextTask ID	ResponseTime	#of ContextSwitch	PreemptionTime	OSTimeDly
2	Completion	task(1)(0)	task(3)(0)	2	1	0	3
4	Completion	task(3)(0)	task(2)(0)	3	2	0	3
8	Completion	task(2)(0)	task(1)(1)	8	2	0	
10	Completion	task(1)(1)	task(3)(1)	5	2	0	
12	Completion	task(3)(1)	task(1)(2)	5	2	0	1
14	Completion	task(1)(2)	task(2)(1)	4	2	0	1
16	MissDeadline	task(2)(1)	----------				

● **Implement and describe how to handle the missing deadline situation under EDF. (10%)**

此表格內容在下一頁。

```
if (OSTCBCur != NULL && OSTCBCur->OSTCBPrio != 63) {
    if (OSTimeGet() >= OSTCBCur->OSTCBDeadLine) {
        if (OSTCBCur->OSTCBWorkTime == OSTCBCur->OSTCBExecutionTime) {
            exitflag = 1;
        }
        else {
            printf("%2d     MissDeadline     ", OSTimeGet());
            printf("task(%2d)(%2d)   ------------\n", OSTCBCur->OSTCBId, OSTCBCur->OSTCBJobNum);
            if ((Output_err = fopen_s(&Output_fp, "./Output.txt", "a")) == 0)
            {
                fprintf(Output_fp, "%2d     MissDeadline     ", OSTimeGet());
                fprintf(Output_fp, "task(%2d)(%2d)   ------------\n", OSTCBCur->OSTCBId, OSTCBCur->OSTCB
                fclose(Output_fp);
            }
            exit(0);
        }
    }
}
```

在 OS_SchedNew() 裡面加入判斷式，若系統時間大於等於 deadline 的話，將會被分成兩個情況，一種是現在的任務工作剛好做完，那麼就讓系統在 OSIntExit() 裡暫時不要更新最高優先權，避免任務被其他人搶佔，另外一種情況則是任務還沒做完的情況，那麼就直接宣告 MissDeadline，將會直接關閉系統。

● **A report that describes your implementation (please attach the screenshot of the code and MARK the modified part). (40%)**

```
ptcb = OSTCBList;                                   /* Point at first TCB in TCB list               */
hptcb = OSTCBPrioTbl[0];

while (ptcb != 0) {     /* Go through all TCBs in TCB list              */
    OS_ENTER_CRITICAL();
    int x = ptcb->OSTCBId;
    if (ptcb->OSTCBPrio == 63) {
        ptcb->OSTCBDeadLine = 10000;
    }
    if ((ptcb->OSTCBStat & OS_STAT_SUSPEND) == OS_STAT_RDY && ptcb->OSTCBDly == 0) {
        if (ptcb->OSTCBDly == 0 && hptcb->OSTCBDly != 0 || ptcb->OSTCBStat != OS_STAT_RDY) {

            hptcb = ptcb;
        }
        else if (ptcb->OSTCBDeadLine <= hptcb->OSTCBDeadLine && ptcb->OSTCBWorkTime != ptcb->OSTCBExecutionTime) {
            if (ptcb->OSTCBDeadLine == hptcb->OSTCBDeadLine && ptcb->OSTCBId < hptcb->OSTCBId)
                hptcb = ptcb;
            else if (ptcb->OSTCBDeadLine < hptcb->OSTCBDeadLine)
                hptcb = ptcb;
        }
        else if (OSTimeGet() == hptcb->OSTCBDeadLine && ptcb->OSTCBDeadLine < hptcb->OSTCBDeadLine + hptcb->OSTCBP
            hptcb = ptcb;
        }

    }
    ptcb = ptcb->OSTCBNext;                         /* Point at next TCB in TCB list                */
    OS_EXIT_CRITICAL();
}
```

EDF 的概念是越接近 Deadline 的 Task 優先權越大，因此在 OS_SchedNew() 裡面更改演算法，並將 Deadline 最小的設置為最高優先權。

首先進入 TCBList 中，比較當前 task 與目前最高優先權的 task，要注意的是，必須將 idle task 的 Deadline 設為最長(綠色部分)，接下來是設定最高優先權的部分(橘色方框)，比較的順序為:

1.若當前 task 的 delay = 0 而當前最高 delay != 0 ，則把當前最高更新為

當前 task 。
2.若當前的任務 Deadline 短於目前最高的 deadline，且當前任務尚未完成，則更新最高優先權任務，需注意的是，當任務的 Deadline 相同時，ID 較靠前的 Task 較高。
3. 若最高優先權的任務已經到 Deadline Time，若當前的任務 Deadline < 最高優先權的下一個 Deadline ，則更新最高優先權。

[PART II] CUS Scheduler Implementation [30%]

- **The correctness of schedule results of examples. Note the testing task set might not be the same as the given example task set. (15%)**

Periodic Task Set = {τ_1 (1, 0, 2, 8), τ_2 (2, 0, 3, 10), τ_3 (3, 0, 4, 15), $\tau_{4\_ServerSize}$ (4, 25%)}

Aperiodic Jobs Set = {j_0 (0, 12, 3, 26), j_1 (1, 14, 2, 34)}

Tick	Event	CurrentTask ID	NextTask ID	ResponseTime	#of ContextSwitch	PreemptionTime	OSTimeDly
2	Completion	task(1)(0)	task(2)(0)	2	1	0	6
5	Completion	task(2)(0)	task(3)(0)	5	2	0	5
9	Completion	task(3)(0)	task(1)(1)	9	2	0	6
11	Completion	task(1)(1)	task(2)(1)	3	2	0	5
12	Aperiodic job(0) arrives and sets CUS server's deadline as 24.						
14	Aperiodic job(1) arrives . Do nothing.						
14	Completion	task(2)(1)	task(4)(0)	4	2	0	6
16	Preemption	task(4)(0)	task(1)(2)				
18	Completion	task(1)(2)	task(4)(0)	2	2	0	6
19	Aperiodic job(0) is finished.						
19	Completion	task(4)(0)	task(3)(1)	7	4	2	N/A
20	Preemption	task(3)(1)	task(2)(2)				
23	Completion	task(2)(2)	task(3)(1)	3	2	0	7
24	Aperiodic job(1) sets CUS server's deadline as 32.						
26	Completion	task(3)(1)	task(1)(3)	11	4	3	4
28	Completion	task(1)(3)	task(4)(1)	4	2	0	4
30	Aperiodic job(1) is finished.						
30	Completion	task(4)(1)	task(2)(3)	16	2	0	N/A
32	Preemption	task(2)(3)	task(1)(4)				
34	Completion	task(1)(4)	task(2)(3)	2	2	0	6
35	Completion	task(2)(3)	task(3)(2)	5	4	2	5
39	Completion	task(3)(2)	task(63)	9	2	0	6
40	Preemption	task(63)	task(1)(5)				

- **A report that describes your implementation (please attach the screenshot of the code and MARK the modified part). (15%)**

```
if (TaskParameter[j].TaskExecutionTime == 0) {
    TaskParameter[j].CusUti = TaskParameter[j].TaskArriveTime;
    TaskParameter[j].TaskArriveTime = 0;
    cus++;
}
```

為了讀取 CUS 的資料，在 InputFile() 內有特別加上判斷，若當前的任務執行時間是 0 的話，並記錄 CUS 的 seversize 到 CusUti。

```
void AInputFile() {

    errno_t err;
    if ((err = fopen_s(&fp, AINPUT_FILE_NAME, "r")) == 0) { ... }
    else { ... }

    char* ptr;
    char* pTmp = NULL;
    int TaskInfo[INFO], i = 0, j = 0;
    ATASK_NUMBER = 0;
    while (!feof(fp)) { ... }

    fclose(fp);
}
```

為了讀取非週期任務的資料，有另外寫一個 AInputFile，內容只是讀的檔案不同，並把原本的 TaskParameter 改為 ATaskParameter，以區分一般任務和非週期任務。

```
OSTaskCreateExt(CUS,
    &TaskParameter[TASK_NUMBER-1],
    &Task_STK[TASK_NUMBER-1][TASK_STACKSIZE - 1],
    TASK_NUMBER-1,
    TaskParameter[TASK_NUMBER-1].TaskID,
    &Task_STK[TASK_NUMBER-1][0],
    TASK_STACKSIZE,
    &TaskParameter[TASK_NUMBER-1],
    (OS_TASK_OPT_STK_CHK | OS_TASK_OPT_STK_CLR),
    TaskParameter[TASK_NUMBER-1].TaskExecutionTime,
    TaskParameter[TASK_NUMBER-1].TaskArriveTime,
    TaskParameter[TASK_NUMBER-1].TaskPeriodic,
    TaskParameter[TASK_NUMBER-1].CusUti);
```

```
void CUS(void* p_arg) {

    while (1)
    {
        while (1) {
            if (OSTCBCur->OSTCBStartTime == 0 && OSTCBCur->OSTCBWorkTime == 0) {
                OSTCBCur->OSTCBStartTime = OSTimeGet();
            }
            if (OSTimeGet() == OSTCBCur->OSTCBDeadLine) {
                OS_Sched();
            }
            if (OSTCBCur->OSTCBWorkTime == OSTCBCur->OSTCBExecutionTime) {

                OSTCBCur->OSTCBStat = OS_STAT_SUSPEND;
                printf("%2d    Aperiodic job(%d) is finished.\n",OSTimeGet(), completeNumber);
                if ((Output_err = fopen_s(&Output_fp, "./Output.txt", "a")) == 0)
                {
                    fprintf(Output_fp, "%2d    Aperiodic job(%d) is finished.\n", OSTimeGet(), completeNumber);
                    fclose(Output_fp);
                }
                completeNumber++;
                OS_Sched();
                break;
            }
        }
    }
}
```

在 main() 裡面 CUS task 的創立是分開的，目的只是要將 task 裡面做的事情分開，主要是 CUS 做完後要把 Task 的狀態設為 SUSPEND，而且要 printf 其他訊息比較方便，當任務完成時，紀錄 cus 完成件數的 completeNumber 也會 +1。

```
2281        if(CusUti!=0)
2282            ptcb->OSTCBStat = OS_STAT_SUSPEND;
```

在 OS_TCBInit() 裡面，若當前任務是 Cus，那麼先將任務狀態設為暫停以免被排程。

此表格內容在下一頁。

```
void OS_Cusevent (void) {
    OS_ENTER_CRITICAL();
    OS_TCB* ptcb;
    ptcb = OSTCBPrioTbl[TASK_NUMBER-1];
    for (int n = completeNumber; n < ATASK_NUMBER; n++) {
        if (OSTimeGet() == ATaskParameter[n].TaskArriveTime) {
            printf("%2d    Aperiodic job(%d) arrives",OSTimeGet(),n);
            if ((Output_err = fopen_s(&Output_fp, "./Output.txt", "a")) == 0)
            {
                fprintf(Output_fp, "%2d    Aperiodic job(%d) arrives", OSTimeGet(
                fclose(Output_fp);
            }
            if (ptcb->OSTCBStat != OS_STAT_SUSPEND) {
                printf(" . Do nothing.\n");
                if ((Output_err = fopen_s(&Output_fp, "./Output.txt", "a")) == 0)
                {
                    fprintf(Output_fp, " . Do nothing.\n");
                    fclose(Output_fp);
                }
            }
        }
```

OS_Cusevent()是自己創的 function，它的功能是在檢查是否有非週期任務抵達，並且 printf 訊息，這邊把 n 設為 completeNumber，因此做完任務的不會被考慮，以下兩個表格都是 OS_Cusevent()的內容。

```
if (OSTimeGet() >= ATaskParameter[n].TaskArriveTime && ptcb->OSTCBStat == OS_STAT_SUSPEND) {
    ptcb->OSTCBExecutionTime = ATaskParameter[n].TaskExecutionTime;
    if (ptcb->OSTCBDeadLine == 0 ){
        ptcb->OSTCBDeadLine = OSTimeGet() + ptcb->OSTCBExecutionTime / (ptcb->OSTCBCusUti /
        printf(" and sets CUS server's deadline as %2d.\n", ptcb->OSTCBDeadLine);
        if ((Output_err = fopen_s(&Output_fp, "./Output.txt", "a")) == 0)
        {
            fprintf(Output_fp, " and sets CUS server's deadline as %2d.\n", ptcb->OSTCBDeadL
            fclose(Output_fp);
        }
        ptcb->OSTCBStat = OS_STAT_RDY;
    }
```

接著判斷非週期任務是否抵達，且現在 cus 的狀態是否為暫停，這可以避免其他非週期的任務覆蓋當前任務，接著依照講義上的公式設判斷式:
1.若當前 deadline 為 0，則把 cus 的 deadline 設為當前時間加上執行時間除於利用率，且將 cus 的狀態改為 ready。

```
else if(OSTimeGet() >= ptcb->OSTCBDeadLine && ptcb->OSTCBStat == OS_STAT_SUSPEND){
    if(ATaskParameter[n].TaskArriveTime < OSTimeGet()){
        ptcb->OSTCBDeadLine = ptcb->OSTCBDeadLine + ptcb->OSTCBExecutionTime / (ptcb->OSTCBCusUti / 100.0);
        printf("%2d    Aperiodic job(%d) sets CUS server's deadline as %2d.\n",OSTimeGet(),n, ptcb->OSTCBDeadLine);
        if ((Output_err = fopen_s(&Output_fp, "./Output.txt", "a")) == 0)
        {
            fprintf(Output_fp, "%2d    Aperiodic job(%d) sets CUS server's deadline as %2d.\n", OSTimeGet(), n, ptcb->OSTCBD
            fclose(Output_fp);
        }
        ptcb->OSTCBStat = OS_STAT_RDY;
    }
    else if(ATaskParameter[n].TaskArriveTime == OSTimeGet()){
        ptcb->OSTCBDeadLine = OSTimeGet() + ptcb->OSTCBExecutionTime / (ptcb->OSTCBCusUti / 100.0);
        printf(" and sets CUS server's deadline as %2d.\n", ptcb->OSTCBDeadLine);
        if ((Output_err = fopen_s(&Output_fp, "./Output.txt", "a")) == 0)
        {
            fprintf(Output_fp, " and sets CUS server's deadline as %2d.\n", ptcb->OSTCBDeadLine);
            fclose(Output_fp);
        }
        ptcb->OSTCBStat = OS_STAT_RDY;
    }
}
```

2.若當前的時間大於目前 cus 的 deadline，而且 cus 狀態為 suspend 時
a.先判斷新的非週期任務在這個時間點是否已經抵達，若已經抵達的話 cus 的 deadline 設為原本的 deadline 加上執行時間除於利用率。
b.若不符合上述條件，代表新的非週期任務尚未抵達，則只需要在抵達時更新 deadline，更新時 cus 的 deadline 設為當前時間加上執行時間除於利用率。
OS_Cusevent()的內容到這邊結束。

```
static  void  OS_SchedNew (void)
{
#if OS_LOWEST_PRIO <= 63u                        /* See if we support up to 64 tasks
    INT8U   y;

    OS_Cusevent();


if (OSTCBCur->OSTCBCusUti != 0) {
    if (OSTimeGet() >= ATaskParameter[completeNumber].TaskPeriodic) {
        if (OSTCBCur->OSTCBWorkTime == OSTCBCur->OSTCBExecutionTime) {
            exitflag = 1;
        }
        else {
            printf("%2d     MissDeadline     ", OSTimeGet());
            printf("task(%2d)(%2d)   ------------\n", OSTCBCur->OSTCBId, OSTCBCur->OSTCBJobNum);
            if ((Output_err = fopen_s(&Output_fp, "./Output.txt", "a")) == 0)
            {
                fprintf(Output_fp, "%2d     MissDeadline     ", OSTimeGet());
                fprintf(Output_fp, "task(%2d)(%2d)   ------------\n", OSTCBCur->OSTCBId, OSTCBCur->OSTCB
                fclose(Output_fp);
            }
            exit(0);
        }
    }
}
```

最後，把 OS_Cusevent()放入 OS_SchedNew()裡面，讓每次要更新優先權時都先去 check 非週期任務的狀態，然後，因為非週期任務除了系統 Deadline 以外還有自己的 Deadline，所以在判斷 Deadline 這裡有特別加上去。(非週期任務的 deadline 是存在 TaskPeriodic 裡面)