110.1 Embedded OS Implementation
PA#2 Report

四電機四乙 B10707119 陳俊宇

2021/12/14

[PART I] EDF Scheduler Implementation

Screenshot of modified code

(未以紅色括弧標示之截圖表示全部為新程式碼)

typedef struct OS_TCB

```
    /// PA#1
    //INT8U already_delay;
    INT8U remain_exe_time;
    //INT8U next_job_time;
    //INT8U self_continue;
    /// PA#1

    /// PA#2
    INT8U arrive_time;
    INT8U exe_time;
    INT8U period;
    INT8U deadline;
    /// PA#2


#if OS_TASK_DEL_EN > 0u
    INT8U           OSTCBDelReq;          /* Indicates whether a task needs to delete itself        */
#endif

#if OS_TASK_PROFILE_EN > 0u
    INT32U          OSTCBCtxSwCtr;        /* Number of time the task was switched in                */
    INT32U          OSTCBCyclesTot;       /* Total number of clock cycles the task has been running */
    INT32U          OSTCBCyclesStart;     /* Snapshot of cycle counter at start of task resumption  */
    OS_STK         *OSTCBStkBase;         /* Pointer to the beginning of the task stack             */
    INT32U          OSTCBStkUsed;         /* Number of bytes used from the stack                    */
#endif
```

OS_TCBInit()

```
INT8U  OS_TCBInit (INT8U    prio,
                   OS_STK  *ptos,
                   OS_STK  *pbos,
                   INT16U   id,
                   INT32U   stk_size,
                   void    *pext,
                   INT16U   opt,
                   INT8U arrive_time, INT8U exe_time, INT8U period)  /// PA#2
```

```
    /// PA#1 part1

    /// PA#2 初始化額外TCB變數
    if (id == OS_TASK_IDLE_ID)
    {
        ptcb->arrive_time = 0;
        ptcb->exe_time = 255;
        ptcb->period = 255;
        ptcb->deadline = 255;
        ptcb->remain_exe_time = 255;
    }
    else
    {
        ptcb->arrive_time = arrive_time;
        ptcb->exe_time = exe_time;
        ptcb->period = period;
        ptcb->deadline = arrive_time + period;
        ptcb->remain_exe_time = exe_time;
    }

    /// PA#2 如果arrive time == 0，設為ready；反之設定OSTCBDly = arrive_time
    if (arrive_time == 0)
    {
        OSRdyGrp |= ptcb->OSTCBBitY;          /* Make task ready to run                 */
        OSRdyTbl[ptcb->OSTCBY] |= ptcb->OSTCBBitX;
    }
    else
    {
        ptcb->OSTCBDly = arrive_time;
    }

    OSTaskCtr++;                                  /* Increment the #tasks counter           */
    OS_TRACE_TASK_READY(ptcb);
    OS_EXIT_CRITICAL();
    return (OS_ERR_NONE);
}
OS_EXIT_CRITICAL();
return (OS_ERR_TASK_NO_MORE_TCB);
}
```

OSTaskCreateExt()

```
#if OS_TASK_CREATE_EXT_EN > 0u
INT8U  OSTaskCreateExt (void   (*task)(void *p_arg),
                        void    *p_arg,
                        OS_STK  *ptos,
                        INT8U    prio,
                        INT16U   id,
                        OS_STK  *pbos,
                        INT32U   stk_size,
                        void    *pext,
                        INT16U   opt,
                        INT8U arrive_time, INT8U exe_time, INT8U period)
```

main.c (task creation and task function)

```
    /// PA#3，新增3個argument
    for (n = 0;n < TASK_NUMBER;n++)
    {
        /// printf("ID %d, prio %d\n", TaskParameter[n].TaskID, n+1);
        OSTaskCreateExt(task1,                                          /// task function
            &TaskParameter[n],                                          /// p_arg(給task function的參數)
            &Task_STK[n][TASK_STACKSIZE - 1],                           /// ptos
            n+1,                                                        /// prio, PA#1要求從1開始
            TaskParameter[n].TaskID,                                    /// id
            &Task_STK[n][0],                                            /// pbos
            TASK_STACKSIZE,                                             /// stack size
            &TaskParameter[n],                                          /// pext(TCB extentsion的pointer)
            (OS_TASK_OPT_STK_CHK | OS_TASK_OPT_STK_CLR),                /// opt
            TaskParameter[n].TaskArriveTime,                                /// arrive_time
            TaskParameter[n].TaskExecutionTime,                             /// exe_time
            TaskParameter[n].TaskPeriodic                                   /// period
        );
    }
```

```
void task1(void* p_arg) {
    task_para_set* task_data;
    task_data = p_arg;

    while (1)
    {
    }
}
```

OS_SchedNew()

```
static  void  OS_SchedNew (void)
{
#if OS_LOWEST_PRIO <= 63u                       /* See if we support up to 64 tasks                   */
    /// ==========================================RMS==========================================
    //INT8U   y;
    //y           = OSUnMapTbl[OSRdyGrp];
    //OSPrioHighRdy = (INT8U)((y << 3u) + OSUnMapTbl[OSRdyTbl[y]]);
    /// ==========================================RMS==========================================

    /// ==========================================EDF==========================================
    OSPrioHighRdy = find_ED_prio();
    // printf("%d : OSPrioHighRdy = %d \n", OSTimeGet(), OSPrioHighRdy);
    /// ==========================================EDF==========================================
```

```
  /// PA#2，For EDF，看過OSRdyTbl一遍，找出deadline最近的task的prio
int find_ED_prio()
{
    int ED_prio = OS_LOWEST_PRIO;
    int ED = 255;
    for (int prio = 0; prio <= OS_LOWEST_PRIO; prio++)
    {
        int row = prio / 8;
        int col_mask = 0b1 << (prio%8);

        if ((OSRdyTbl[row] & col_mask) != 0)  // 如果某個task是ready的
        {
            // printf("OSTCBPrioTbl[%d]->deadline  = %d \n", prio, OSTCBPrioTbl[prio]->deadline);
            if (OSTCBPrioTbl[prio]->deadline < ED)  // 如果這個task的deadline是目前看到最小的
            {
                ED_prio = prio;
                ED = OSTCBPrioTbl[prio]->deadline;
            }
        }
    }
    return ED_prio;
```

OSTimeTick() (包含 miss deadline handling)

```
                    /// PA#1
                    task_start_time[ptcb->OSTCBPrio] = OSTimeGet();  // 紀錄這個job ready時間，用於計算response time
                    /// PA#1

                    OS_TRACE_TASK_READY(ptcb);
                }
            }
        }

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

    /// PA#2：把remain_exe_time處理改到OSTimeTick()處理
    if (OSTCBCur->remain_exe_time > 0)
    {
        OSTCBCur->remain_exe_time--;
        // printf("tick %d choose %d to --\n", OSTimeGet(), OSTCBCur->OSTCBPrio);

        if (OSTCBCur->remain_exe_time == 0)
        {
            response_time = OSTimeGet() - task_start_time[OSTCBCur->OSTCBPrio];

            task_start_time[OSTCBCur->OSTCBPrio] = OSTimeGet();  // 紀錄這個job ready時間，用於計算response time

            OSTCBCur->arrive_time = OSTCBCur->arrive_time + OSTCBCur->period;  // 設下個job開始時間
            OSTCBCur->deadline = OSTCBCur->arrive_time + OSTCBCur->period;  // 設下個job的deadline
            OSTCBCur->remain_exe_time = OSTCBCur->exe_time;  // 重設剩餘執行時間

            INT8U delay_ticks = OSTCBCur->arrive_time - OSTimeGet();
            if (delay_ticks > 0u) {                /* 0 means no delay!                           */
                INT8U y = OSTCBCur->OSTCBY;        /* Delay current task                          */
                OSRdyTbl[y] &= (OS_PRIO)~OSTCBCur->OSTCBBitX;
                OS_TRACE_TASK_SUSPENDED(OSTCBCur);
                if (OSRdyTbl[y] == 0u) {
                    OSRdyGrp &= (OS_PRIO)~OSTCBCur->OSTCBBitY;
                }
                OSTCBCur->OSTCBDly = delay_ticks;  /* Load ticks in TCB                           */
```

```
        /// ============================================miss deadline============================================
        miss_deadline_prio = check_deadline();
        while (miss_deadline_prio != -1)
        {
            if ((Output_err = fopen_s(&Output_fp, "./Output.txt", "a")) != 0)
            {
                printf("can't open Output.txt");
                exit(0);
            }
            printf("%2d\t", OSTimeGet());
            fprintf(Output_fp, "%2d\t", OSTimeGet());

            printf("MissDeadline\t");
            fprintf(Output_fp, "MissDeadline\t");

            /// CurrentTask
            printf("Task(%2d)(%2d)\t", OSTCBPrioTbl[miss_deadline_prio]->OSTCBId, job_number[miss_deadline_prio]);
            fprintf(Output_fp, "Task(%2d)(%2d)\t", OSTCBPrioTbl[miss_deadline_prio]->OSTCBId, job_number[miss_deadline_prio]);

            /// NextTask
            printf("-----------\n");
            fprintf(Output_fp, "-----------\n");

            /// ============================================miss deadline handling============================================
            printf("========== Drop Task(%2d)(%2d) at tick %2d with remaining execution time %2d. ==========\n", \
                OSTCBPrioTbl[miss_deadline_prio]->OSTCBId, job_number[miss_deadline_prio], OSTimeGet(), OSTCBPrioTbl[miss_deadline_prio]->remain_exe_time);
            fprintf(Output_fp, "========== Drop Task(%2d)(%2d) at tick %2d with remaining execution time %2d. ==========\n", \
                OSTCBPrioTbl[miss_deadline_prio]->OSTCBId, job_number[miss_deadline_prio], OSTimeGet(), OSTCBPrioTbl[miss_deadline_prio]->remain_exe_time);
            fclose(Output_fp);

            task_start_time[miss_deadline_prio] = OSTimeGet();  // 紀錄這個job ready時間，用於計算response time

            OSTCBPrioTbl[miss_deadline_prio]->arrive_time = OSTimeGet();  // 重設job開始時間
            OSTCBPrioTbl[miss_deadline_prio]->deadline = OSTimeGet() + OSTCBPrioTbl[miss_deadline_prio]->period;  // 重設job的deadline
            OSTCBPrioTbl[miss_deadline_prio]->remain_exe_time = OSTCBPrioTbl[miss_deadline_prio]->exe_time;  // 重設剩餘執行時間

            missed_deadline = 1;
            miss_deadline_prio = check_deadline();
        }
```

```
/// PA#2：For EDF，看過OSRdyTbl一遍，檢查是否有miss deadline，有則retrun miss deadline的prio，無則return-1
int check_deadline()
{
    int miss_deadline_prio = -1;
    for (int prio = 0; prio <= OS_LOWEST_PRIO; prio++)
    {
        int row = prio / 8;
        int col_mask = 0b1 << (prio % 8);

        if ((OSRdyTbl[row] & col_mask) != 0)  // 如果某個task是ready的
        {
            if (OSTCBPrioTbl[prio]->deadline <= OSTimeGet())
            {
                miss_deadline_prio = prio;
                break;
            }
        }
    }
    return miss_deadline_prio;
}
```

Description of implementation

從 PA#1 進行修改，將排程方式由 RMS 改為 EDF，最主要的改變是 OS_SchedNew()中，原本 OSPrioHighRdy 會設為 priority 最高的 task，而 EDF 則要設為 deadline 最近的 task。

我的方法是在 TCB 中新增變數來紀錄各個 task 的 deadline，在每次排程時，遍歷所有 ready 的 task，從中選出 deadline 最小的 task，將其 priority assign 給 OSPrioHighRdy。

另外，與 PA#1 做法不同的地方還有：我將計算剩餘 execution time 的 code 從 task function 移至 OSTimeTick()，能夠更簡單、統一地處理各種類型的 context switch。

RMS 與 EDF 的共通點是必須要隨時更新、檢查各個 task 是否已 ready，因此保留了原本的 OSRdyTbl。其餘輸出資訊、response time 等等計算，亦與 PA#1 相同。

How to handle the missing deadline situation under EDF

我實作的 miss deadline 處理方式為：任何一個 task 發生 miss deadline 時，**立即中止執行中的 task**，顯示 miss deadline 的時間點、中止執行的 task ID、其剩餘的執行時間等。接下來，**將 miss deadline 的 job 視為不存在**，繼續依照 EDF 規則進行排程。

下圖為 TaskSet Example 2 的執行結果。

```
20      Completion      Task( 1)( 4)    Task( 3)( 6)        4               2                   0
21      Completion      Task( 3)( 6)    Task( 1)( 5)        2               2                   0
22      Completion      Task( 1)( 5)    Task( 2)( 3)        2               2                   0
24      MissDeadline    Task( 2)( 3)    ------------
========= Drop Task( 2)( 3) at tick 24 with remaining execution time  1. =========
25      Completion      Task( 3)( 7)    Task( 1)( 6)        3               1                   0
26      Completion      Task( 1)( 6)    Task( 3)( 8)        2               2                   0
27      Completion      Task( 3)( 8)    Task( 2)( 3)        2               2                   0
30      Completion      Task( 2)( 3)    Task( 3)( 9)        6               3                   0
31      Completion      Task( 3)( 9)    Task( 1)( 7)        3               2                   0
32      Completion      Task( 1)( 7)    Task( 3)(10)        4               2                   0
33      Completion      Task( 3)(10)    Task( 1)( 8)        2               2                   0
34      Completion      Task( 1)( 8)    Task( 2)( 4)        2               2                   0
36      MissDeadline    Task( 2)( 4)    ------------
========= Drop Task( 2)( 4) at tick 36 with remaining execution time  1. =========
37      Completion      Task( 3)(11)    Task( 1)( 9)        3               1                   0
38      Completion      Task( 1)( 9)    Task( 3)(12)        2               2                   0
39      Completion      Task( 3)(12)    Task( 2)( 4)        2               2                   0
```

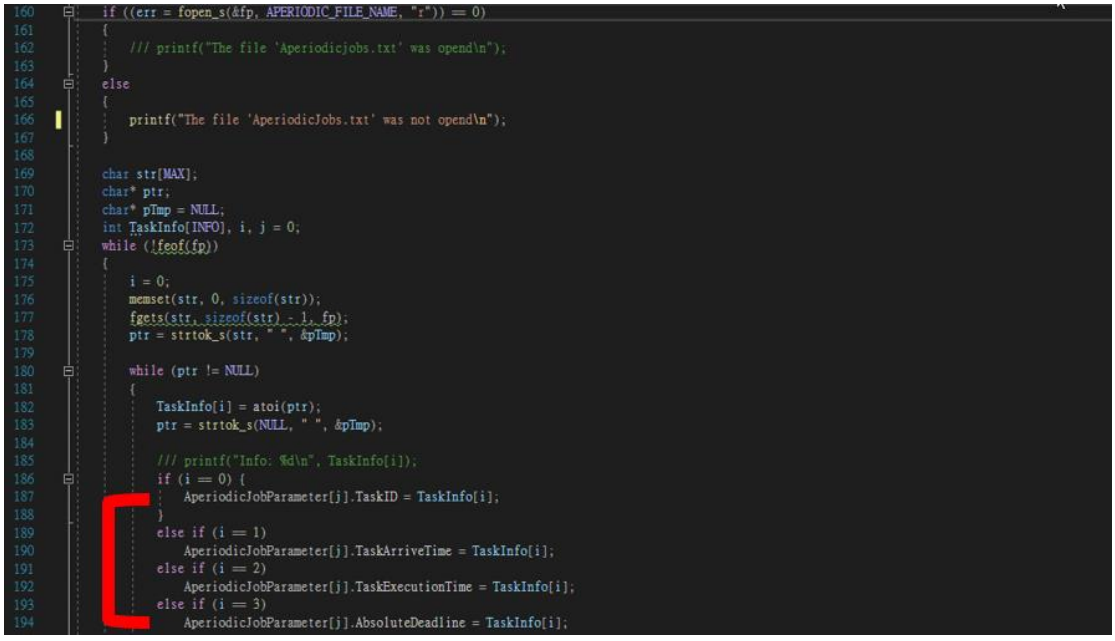
[PART II] CUS Scheduler Implementation

Screenshot of modified code

(未以紅色括弧標示之截圖表示全部為新程式碼)

Reading input files

```
126         while (ptr != NULL)
127         {
128             TaskInfo[i] = atoi(ptr);
129             ptr = strtok_s(NULL, " ", &pTmp);
130
131             /// printf("Info: %d\n", TaskInfo[i]);
132             if (i == 0) {
133                 TASK_NUMBER++;
134                 TaskParameter[j].TaskID = TaskInfo[i];
135             }
136             else if (i == 1)
137                 TaskParameter[j].TaskArriveTime = TaskInfo[i];
138             else if (i == 2)
139                 TaskParameter[j].TaskExecutionTime = TaskInfo[i];
140             else if (i == 3)
141                 TaskParameter[j].TaskPeriodic = TaskInfo[i];
142
143             i++;
144         }
145         j++;
146     }
147
148     SERVER_ID = TaskParameter[j - 1].TaskID;
149     SERVER_SIZE = TaskParameter[j - 1].TaskArriveTime;
150     fclose(fp);
```



```
160     if ((err = fopen_s(&fp, APERIODIC_FILE_NAME, "r")) == 0)
161     {
162         /// printf("The file 'Aperiodicjobs.txt' was opend\n");
163     }
164     else
165     {
166         printf("The file 'AperiodicJobs.txt' was not opend\n");
167     }
168
169     char str[MAX];
170     char* ptr;
171     char* pTmp = NULL;
172     int TaskInfo[INFO], i, j = 0;
173     while (!feof(fp))
174     {
175         i = 0;
176         memset(str, 0, sizeof(str));
177         fgets(str, sizeof(str) - 1, fp);
178         ptr = strtok_s(str, " ", &pTmp);
179
180         while (ptr != NULL)
181         {
182             TaskInfo[i] = atoi(ptr);
183             ptr = strtok_s(NULL, " ", &pTmp);
184
185             /// printf("Info: %d\n", TaskInfo[i]);
186             if (i == 0) {
187                 AperiodicJobParameter[j].TaskID = TaskInfo[i];
188             }
189             else if (i == 1)
190                 AperiodicJobParameter[j].TaskArriveTime = TaskInfo[i];
191             else if (i == 2)
192                 AperiodicJobParameter[j].TaskExecutionTime = TaskInfo[i];
193             else if (i == 3)
194                 AperiodicJobParameter[j].AbsoluteDeadline = TaskInfo[i];
```

```
94  typedef struct aperiodic_job_para_set {
95      INT16U TaskID;
96      INT16U TaskArriveTime;
97      INT16U TaskExecutionTime;
98      INT16U AbsoluteDeadline;
99  } aperiodic_job_para_set;
```

CUS task creation

```
    /// PA#3，OSTaskCreateExt新增3個argument
    for (n = 0;n < TASK_NUMBER-1;n++)
    {
        /// printf("ID %d, prio %d\n", TaskParameter[n].TaskID, n+1);
        OSTaskCreateExt(task1,                                          /// task function
            &TaskParameter[n],                                          /// p_arg(給task function的參數)
            &Task_STK[n][TASK_STACKSIZE - 1],                           /// ptos
            n+1,                                                        /// prio, PA#1要求從1開始
            TaskParameter[n].TaskID,                                    /// id
            &Task_STK[n][0],                                            /// pbos
            TASK_STACKSIZE,                                             /// stack size
            &TaskParameter[n],                                          /// pext(TCB extentsion的pointer)
            (OS_TASK_OPT_STK_CHK | OS_TASK_OPT_STK_CLR),                /// opt
            TaskParameter[n].TaskArriveTime,                                /// arrive_time
            TaskParameter[n].TaskExecutionTime,                             /// exe_time
            TaskParameter[n].TaskPeriodic                                   /// period
        );
    }

    /// server task
    OSTaskCreateExt(task1,                                          /// task function
        &TaskParameter[n],                                          /// p_arg(給task function的參數)
        &Task_STK[n][TASK_STACKSIZE - 1],                           /// ptos
        n + 1,                                                      /// prio, PA#1要求從1開始
        SERVER_ID,                                                  /// id
        &Task_STK[n][0],                                            /// pbos
        TASK_STACKSIZE,                                             /// stack size
        &TaskParameter[n],                                          /// pext(TCB extentsion的pointer)
        (OS_TASK_OPT_STK_CHK | OS_TASK_OPT_STK_CLR),                /// opt
        0,                                                          /// arrive_time  don't care
        0,                                                          /// exe_time  don't care
        255                                                         /// period  dont't care
    );
    SERVER_PRIO = n + 1;
```

OSTimeTick()

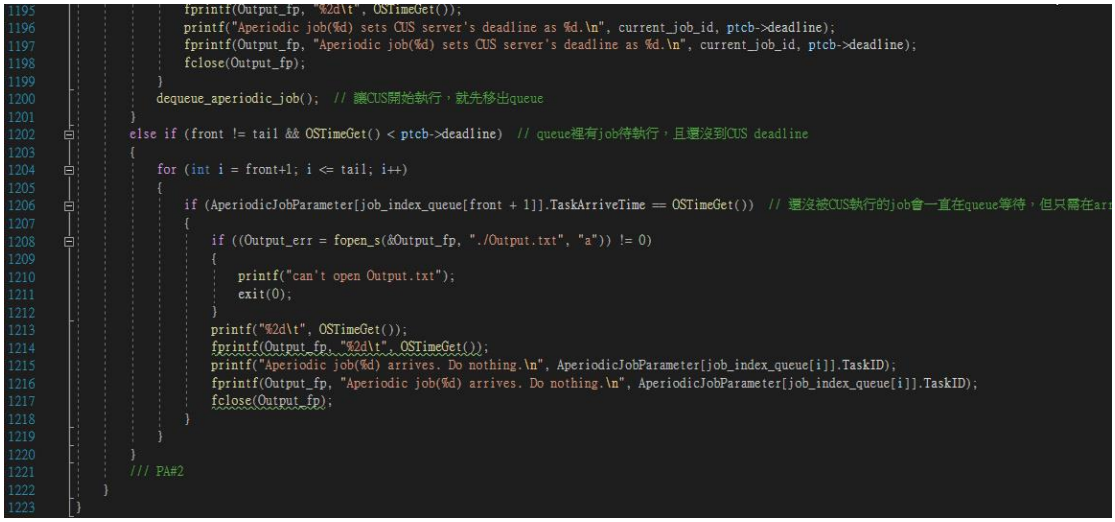
```
int job_index_queue[OS_MAX_TASKS] = { -1 };  // 紀錄server要去執行的job，其在AperiodicJobParameter的index
int front = -1;  // 指向queue的最後一個job -1的位置
int tail = -1;  // 指向queue的最後一個job
int current_job_id = -1;

/// PA#2，For CUS，看過AperiodicJobParameter一遍，檢查是否有aperiodic job要開始
void enqueue_aperiodic_job()
{
    for (int i = 0; i < OS_MAX_TASKS; i++)
    {
        if (AperiodicJobParameter[i].TaskArriveTime == OSTimeGet())
        {
            tail++;
            job_index_queue[tail] = i;
            aperiodic_job_start_time[AperiodicJobParameter[i].TaskID] = OSTimeGet();
        }
    }
}
void dequeue_aperiodic_job()
{
    front++;
}
```

```
        /// ================================== CUS ==================================
        enqueue_aperiodic_job();
        ptcb = OSTCBPrioTbl[SERVER_PRIO];
        if (front != tail && (OSTimeGet() >= ptcb->deadline || ptcb->arrive_time == 0))  // queue裡有job待執行，且目前CUS為空閒(OSTimeGet() > server dead
        {
            if ((ptcb->OSTCBStat & OS_STAT_SUSPEND) == OS_STAT_RDY) {  /* Is task suspended?       */
                OSRdyGrp |= ptcb->OSTCBBitY;                /* No,  Make ready          */
                OSRdyTbl[ptcb->OSTCBY] |= ptcb->OSTCBBitX;
            }

            ptcb->arrive_time = OSTimeGet();
            ptcb->deadline = OSTimeGet() + AperiodicJobParameter[job_index_queue[front + 1]].TaskExecutionTime * 100 / SERVER_SIZE;
            ptcb->remain_exe_time = AperiodicJobParameter[job_index_queue[front + 1]].TaskExecutionTime;
            current_job_id = AperiodicJobParameter[job_index_queue[front + 1]].TaskID;

            if (AperiodicJobParameter[job_index_queue[front + 1]].TaskArriveTime == OSTimeGet())  // arrive時，剛好CUS可以直接開始
            {
                if ((Output_err = fopen_s(&Output_fp, "./Output.txt", "a")) != 0)
                {
                    printf("can't open Output.txt");
                    exit(0);
                }
                printf("%2d\t", OSTimeGet());
                fprintf(Output_fp, "%2d\t", OSTimeGet());
                printf("Aperiodic job(%d) arrives and sets CUS server's deadline as %d.\n", current_job_id, ptcb->deadline);
                fprintf(Output_fp, "Aperiodic job(%d) arrives and sets CUS server's deadline as %d.\n", current_job_id, ptcb->deadline);
                fclose(Output_fp);
            }
            else  // job先等待了前一個CUS deadline，之後才開始
            {
                if ((Output_err = fopen_s(&Output_fp, "./Output.txt", "a")) != 0)
                {
                    printf("can't open Output.txt");
                    exit(0);
                }
                printf("%2d\t", OSTimeGet());
```



```
                fprintf(Output_fp, "%2d\t", OSTimeGet());
                printf("Aperiodic job(%d) sets CUS server's deadline as %d.\n", current_job_id, ptcb->deadline);
                fprintf(Output_fp, "Aperiodic job(%d) sets CUS server's deadline as %d.\n", current_job_id, ptcb->deadline);
                fclose(Output_fp);
            }
            dequeue_aperiodic_job();  // 讓CUS開始執行，就先移出queue
        }
        else if (front != tail && OSTimeGet() < ptcb->deadline)  // queue裡有job待執行，且還沒到CUS deadline
        {
            for (int i = front+1; i <= tail; i++)
            {
                if (AperiodicJobParameter[job_index_queue[front + 1]].TaskArriveTime == OSTimeGet())  // 還沒被CUS執行的job會一直在queue等待，但只需在ar
                {
                    if ((Output_err = fopen_s(&Output_fp, "./Output.txt", "a")) != 0)
                    {
                        printf("can't open Output.txt");
                        exit(0);
                    }
                    printf("%2d\t", OSTimeGet());
                    fprintf(Output_fp, "%2d\t", OSTimeGet());
                    printf("Aperiodic job(%d) arrives. Do nothing.\n", AperiodicJobParameter[job_index_queue[i]].TaskID);
                    fprintf(Output_fp, "Aperiodic job(%d) arrives. Do nothing.\n", AperiodicJobParameter[job_index_queue[i]].TaskID);
                    fclose(Output_fp);
                }
            }
        }
        /// PA#2
    }
}
```

Description of implementation

以part I的EDF排程為基礎，加上了aperiodic jobs、server task的處理。

首先，仿照了處理 TaskSet 的方式進行檔案讀取、並建立了新的 struct 來儲存 aperiodic jobs 的資訊。

Server 方面，在資料結構上，sever 與其他 periodic job 同為 task，但是 sever 需要進行特別的處理，不可混為一談，因此額外建立了全域變數 SERVER_ID, SERVER_PRIO 來紀錄 server 的 task ID, priority，方便後續處理、辨別。

排程的實作上，每次 OSTimeTick()時都會去檢查是否有 aperiodic jobs 的 arrival time == OSTimeGet()，並將所有符合條件者加入 queue。

若 queue 不為空，則再判斷目前 CUS 是否為空閒狀態(初始狀態或是已經過了前一個 deadline)，若為空閒，則根據 server size、queue 中第一個 aperiodic job 之 execution time 來計算、設定 server deadline，最後將此 job 移出 queue；若尚未經過前一個 CUS deadline，則需等待 deadline 到達時再進行上述動作。