[PART I | EDF Scheduler Implementation.

1.The screenshot results (with the given format) of two task sets. (Tick 0 to tick 40 or the tick when a task missing the deadline) (10%)

(1)Task set $1 = \{\tau 1 (0, 2, 6), \tau 2 (0, 5, 9)\}$

```
🔟 D:\學校\台科\10901\嵌入式作業系統實作 Embedded OS Implementation\-OS-_code\μOSⅡ_code\Micrium_Win32_Kernel\Microsoft\
                                  task2 set(0, 3, 6)

CurrentTask ID

task(1)(0)

task(3)(0)

task(2)(0)

task(3)(1)
               (0, 1, 4)
Event
Completion
Completion
                                                                      task3 set(1, 1, 3)
MextTask ID ResponseTime
ask1 set(0,
                                                                    NextTask ID
task(3)(0)
task(2)(0)
                                                                                                                           # of ContextSwitch
               Completion
Completion
                                                                     task(
                                                                     task(
               Completion
Completion
                                         task(1)(1)
task(3)(2)
task(1)(2)
               Completion
Completion
                                                                     task(2
                                         task(2)
task(3)
task(3)
                                                                     task(3
               Completion
Completion
                                                                     task(
                                                                     task()
                                         task(
               Completion
Completion
                                         task(1)
                                                                     task(2
                                         task(2)
task(3)
                                                                     task(3
               Completion
Completion
                                                                     task(1
                                         task(
                                                                     task(3
               Completion
                                         task(3
                                                                     task(
               Completion
                                         task(
               Completion
MissDeadline
```

(2) Task set $2 = \{\tau 1 (0, 1, 4), \tau 2 (0, 3, 6), \tau 3 (1, 1, 3)\}$

```
配 D:\10901\嵌入式作業系統實作\µOSⅡ_code\Micrium_Win32_Kernel\Microsoft\Windows\Kernel\OS2\VS\Debug\OS2.exe
                      ed,
6)
                               task(2)(1)
task(2)(0)

CurrentTask ID
task(1)(0)
task(2)(0)
task(2)(1)
task(2)(1)
                  2, 6)
Event
                                                               NextTask ID
task(2)(0)
task(1)(1)
                                                                                          ResponseTime
                                                                                                                    # of ContextSwitch
             Completion
Completion
                                                               task(2)(
task(1)(
              Completion
              Preemption
              Completion
Completion
                                      task(1)(2)
                                                               task(2)
                                      task(2)(1)
                                                               task(6
             Preemption
Completion
                                      task(63
                                                               task(1)
                                      task(1)(3)
             Completion
Completion
Completion
                                                               task(1)
                                      task(2)(2
                                      task(1)(4)
                                                               task(2)
             Preemption
Completion
Completion
Preemption
Completion
                                      task(2)
                                                               task(1
                                      task(11
                                      task(2)
                                                               task(6
                                      task(63)
             Completion
```

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

3. A report that describes your implementation, including scheduling results of two task sets, modified functions, data structure, etc. (please **ATTACH** the screenshot of the code and **MARK** the modified part) (40%)

```
(1) Task set 1 = \{\tau 1 (0, 2, 6), \tau 2 (0, 5, 9)\}
```

```
#define TASK_STACKSIZE
                            2048
#define TASK1_PRIORITY
                           1
#define TASK2_PRIORITY
                           2
#define TASK1_ID
                           1
#define TASK2_ID
                           2
static OS_STK StartupTaskStk[APP_CFG_STARTUP_TASK_STK_SIZE];
static OS_STK Task1_STK[TASK_STACKSIZE];
static OS_STK Task2_STK[TASK_STACKSIZE];
static void StartupTask(void* p_arg);
static void taskl(void* p_arg);
static void task2(void* p_arg);
                                      //set : {start time, work time, period time}
TimeTask task1 set = \{0, 2, 6\};
TimeTask task2\_set = \{ 0, 5, 9 \};
                                     //set : {start time, work time, period time}
```

(2) Task set $2 = \{\tau 1 (0, 1, 4), \tau 2 (0, 3, 6), \tau 3 (1, 1, 3)\}$

2048

3

#define TASK_STACKSIZE

#define TASK1_PRIORITY #define TASK2_PRIORITY

```
#define TASK3_PRIORITY
                             1
#define TASK1_ID
#define TASK2_ID
#define TASK3_ID
static OS_STK StartupTaskStk[APP_CFG_STARTUP_TASK_STK_SIZE]; static OS_STK Task1_STK[TASK_STACKSIZE];
static OS_STK Task2_STK[TASK_STACKSIZE];
static OS_STK Task3_STK[TASK_STACKSIZE];
static void StartupTask(void* p_arg);
static void task1(void* p_arg);
static void task2(void* p_arg);
static void task3(void* p_arg);
TimeTask task1\_set = \{ 0, 1, 4 \};
                                      //set : {start time, work time, period time}
                                        //set : {start time, work time, period time}
TimeTask task2\_set = \{ 0, 3, 6 \};
TimeTask task3\_set = \{ 1, 1, 3 \};
                                        //set : {start time, work time, period time}
```

```
□void task1(void *p arg) {
                                           OSTaskCreateExt(
        (void)p arg;
                                                task1,
                                                &task1 set,
        while (1) {
                                                &Task1_STK[TASK_STACKSIZE - 1],
                                                TASK1_PRIORITY,
                                                TASK1 ID.
        }
                                                &Task1_STK[0],
                                                TASK_STACKSIZE,
                                                &task1_set,
                                                (OS_TASK_OPT_STK_CHK | OS_TASK_OPT_STK_CLR));
 □void task2(void* p_arg) {
        (void)p_arg;
                                            OSTaskCreateExt(
                                                task2,
        while (1) {
                                                &task2 set,
                                                &Task2_STK[TASK_STACKSIZE - 1],
                                                TASK2_PRIORITY,
                                                TASK2 ID.
                                                &Task2_STK[0],
                                                TASK_STACKSIZE,
 □void task3(void* p_arg) {
                                                &task2_set,
                                                (OS_TASK_OPT_STK_CHK | OS_TASK_OPT_STK_CLR));
        (void)p_arg;
        while (1) {
                                            OSTaskCreateExt(
                                                task3.
        }
                                                &task3_set,
                                                &Task3_STK[TASK_STACKSIZE - 1],
                                                TASK3_PRIORITY,
                                                TASK3_ID,
                                                &Task3_STK[0],
                                                TASK_STACKSIZE,
                                                &task3_set,
                                                (OS_TASK_OPT_STK_CHK | OS_TASK_OPT_STK_CLR));
/*
* project2 *
int response time = 0;
int context_switch = 0;
OS_TCB* ptcb;
if (OSPrioHighRdy != 0) {
  ptcb = OSTCBPrioTbl[OSPrioHighRdy];
TimeTask* point_t = ptcb->OSTCBExtPtr;
                                            //指向自定義數據以進行TCB擴展的指針
  if (OSTimeGet() % point_t->period_time == 0) {
    responsetime = point_t->period_time;
                                            //計算respondtime
         responsetime = OSTimeGet()% point_t->period_time - point_t->start_time; //計算respondtime
       printf("%-5s%d%-2s%d%-9s", "task(", ptcb->OSTCBId, ")(", point_t->executive_count, ")"); //print 現在的 task ID
       if (OSRdyTbl[y] == Ou) {
    OSRdyGrp &= (OS_PRIO)~ptcb->OSTCBBitY;
```

```
if (OSRdyTb1[y] == Ou) {
   OSRdyGrp &= (OS_PRIO)~ptcb->OSTCBBitY;
                 }
ptcb->OSTCBDly = point_t->period_time * (point_t->executive_count + 1) - OSTimeGet() + point_t->start_time;
OS_TRACE_TASE_DIT(point_t->period_time * (point_t->executive_count + 1) - OSTimeGet() + point_t->start_time);
OS_EXIT_CRITICAL();
                                                                                                                                                                                                          //Load ticks in TCB
                 y = OSUnMapTbl[OSRdyGrp];
           point_t->executive_count+;
if (CSFridHighRdy != (IJTSU)((y << 3u) + OSUnMapTb1[OSRdyTb1[y]]))
point_t->context_switch+; //計算conextswitch
context_switch = point_t->context_switch;//計算conextswitch
point_t->context_switch = 0; //contextswitchBeg
point_t->precaptive_time = 0; //precaptivetimeBeg
      //OSPrioHighRdy = (INT8U)((y << 3u) + OSUnMapTbl[OSRdyTbl[y]])
     }
else {
printf("%-5d", OSTimeGet());
printf("%-14s", "Premaption");
a = 1;
printf("%-25%d%-25%d%-9s", "task(", ptcb->OSTCBId, ")(", point_t->executive_count, ")");//print 現在的 task ID
           c = 0;
           point_t->context_switch++; //計算conextswitch
if (point_t->preemptive_time > 0) {point_t->preemptive_time = OSTimeGet() - point_t->preemptive_time;}/庚新preemptivetime
else {point_t->preemptive_time = OSTimeGet(); }//更新preemptivetime
//如果是idle task
else if (point_t == 0) {
    printf("s-6d', OSTimeGet());
    printf("s-14s", "Premption");
    printf("s-5s%d%-11s", "task(", OSPrioMighRdy, ")");//print idle task
 if (OSPrioRighRdy != (INTSU)((y << 3u) + OSUmMapTb1[OSRdyTb1[y]]) || (point t->context_switch == 0 && point t->executive_count != 0)) {
    OSPrioRighRdy = (INTSU)((y << 3u) + OSUmMapTb1[OSRdyTb1[y]]);
    ptcb = OSTCBPrioTb1[OSPrioRighRdy];
       //如果下一個task是idel task
else if (point_t == 0) {
    printf("%-5s%d%-17s", "task(", OSPrioNighRdy, ")");//print task(63)
         /
//print responsetime and context switch
       if (responsetime > 0) {
    printf("%-20d%d\n", responsetime, context_switch);
               responsetime = 0;
        //應理missdeadline
        ptcb = OSTCBList;
while (ptcb != (OS_TCB*)0) {
              le (ptob != (US_TUB*)0) (
InmeTask* point_t = ptob->OSTCBExtPtr;
if (point_t != 0 && OSTraeGet() > point_t->period_time * (point_t->executive_count + 1)) {
    printf("%-04s", "Missbeadline");
    printf("%-14s", "Missbeadline");
    printf("%-5:%d%-2:%d%-9:", "task(", ptob->OSTCBId, ")(", point_t->executive_count, ")");
    printf("%-5:%d%-2:%d%-9:", "task(", ptob->OSTCBId, ")(", point_t->executive_count, ")");
                     printf("----
                     system("pause");
              ptcb = ptcb->OSTCBNext;
```

[PART II] CUS Scheduler Implementation [40%]

1. The screenshot results (with the given format) of two task sets. (Tick 0 to tick 40 or the tick when a task missing the deadline). (10%)

Periodic Task Set1 = $\{\tau 1 \ (0, 1, 4), \tau 2 \ (0, 4, 10), \tau 3 \text{ServerSize} \ (0.3)\}$ Aperiodic Jobs Set1 = $\{j0 \ (4, 3, 16), j1 \ (17, 3, 30)\}$

III 選取 D:\10901\嵌入式作業系統實作\µOSⅡ_code\Micrium_Win32_Kernel\Microsoft\Windows\Kernel\OS2\VS\Debug\OS2.exe					
OSTick	created, Thr	ead ID 5068			
task1 set	(0, 1, 4)	task2 set (0, 4	4, 10) jO set (4, 3, 16) j1	set (17, 3, 30)
Tick	Event		ID NextTask ID	ResponseTime	# of ContextSwitch
1	Completion	task(1)(0)	task(2)(0)	1	İ
4 Aperiodic job(0) arrives and sets CUS server's deadline as 14.					
4	Preemption	task(2)(0)	task(1)(1)		
5	Completion	task(1)(1)	task(2)(0)	1	2
4 5 6 8 9	Completion	task(2)(0)	task(3)(0)	5	4
j 8	Preemption	task(3)(0)	task(1)(2)		
9.	Completion	task(1)(2)	task(3)(0)	1	2
10		b(0) is finish.			
10	Completion	task(3)(0)	task(2)(1)	4	4
12	Preemption	task(2)(1)	task(1)(3)		
13	Completion	task(1)(3)	task(2)(1)	Ī	2
15	Completion	task(2)(1)	task(63)	5	4
I b	Preemption	task(63)	task(1)(4)	1 111 00	
17		D(I) arrives and	sets CUS server's	s deadline as 27	
17	Completion	task(1)(4)	task(3)(1)	1	2
20		b(1) is finish.	+l-(1)(E)	4	2
20	Completion	task(3)(1)	task(1)(5)	4	2 2
21	Completion	task(1)(5)	task(2)(2)	1	2
24	Preemption Completion	task(2)(2) task(1)(6)	task(1)(6) task(2)(2)	1	2
25	Completion	task(1)(0) task(2)(2)	task(2)(2) task(63)	5	4
20	Preemption	task(2)(2) task(63)	task(1)(7)	,	4
20 20	Completion	task(1)(7)	task(63)	1	2
30	Preemption	task(63)	task(2)(3)	1	2
<u> </u>	Preemption Preemption	task(2)(3)	task(1)(8)		
33	Completion	task(1)(8)	task(2)(3)	1	2
35	Completion	task(2)(3)	task(63)	Ŝ	4
36	Preemption	task(63)	task(1)(9)		
10 12 13 15 17 20 21 24 22 28 29 33 33 33 33 37	Completion	task(1)(9)	task(63)	1	2
40	Preemption	task(63)	task(1)(10)		
41	Completion	task(1)(10)	task(2)(4)	1	2

Periodic Task Set2 = $\{\tau 1 \ (0, 2, 8), \tau 2 \ (0, 3, 10), \tau 3 \ (0, 5, 20), \tau 4_ServerSize \ (0.2)\}$ Aperiodic Jobs Set2 = $\{j0 \ (12, 3, 28), j1 \ (14, 2, 39)\}$

```
■ D\學校\会科\10901\嵌入式作業系統實作 Embedded OS Implementation\-OS-_code\µOSII_code\Micrium_Win32_Kernel\Microsoft\Windows\... —

OSTick created, Thread ID 4716
task1 set ( 0, 2, 8) task2 set ( 0, 3, 10) task3 set ( 0, 5, 20) job0 set ( 12, 3, 28) job1 set ( 14, 2, 39)

Tick Event CurrentTask NextTask ID ResponseTime # of ContextSwitch
2 Completion task(1)(0) task(2)(0) 2 1
Completion task(3)(0) task(1)(1) 2 2 2
Aperiodic job(0) arrives and sets CUS server's deadline as 27
Completion task(3)(0) task(4)(0) 8 4

14 Aperiodic job(1) arrives and sets CUS server's deadline as 24
14 Preemption task(4)(0) task(4)(1)
16 Aperiodic job(1) is finish.
16 Completion task(4)(1) task(1)(2) 2 2 2
20 Aperiodic job(0) is finish.
20 Completion task(4)(0) task(1)(1) 7 5
23 Completion task(4)(0) task(1)(1) 7 5
24 Preemption task(3)(1) task(1)(3) 2
25 Completion task(3)(1) task(1)(3) 2
26 Completion task(3)(1) task(1)(3) 3 2
27 Completion task(3)(1) task(1)(3) 3 2
28 Completion task(3)(1) task(1)(3) 42
29 Completion task(1)(2) task(4)(0) 2 2
30 Completion task(1)(3) task(3)(1) 2 2
31 Completion task(3)(1) task(1)(3) 42
32 Completion task(3)(1) task(1)(3) 42
33 Completion task(3)(1) task(3)(1) 2 2
34 Preemption task(3)(1) task(3)(1) 2 2
35 Completion task(1)(4) task(63) 2 2
36 Completion task(1)(5) task(2)(3) 2
```

2. A report that describes your implementation, including scheduling results of two task sets, modified functions, data structure, etc. (please **ATTACH** the screenshot of the code and **MARK** the modified part). (30%)

Aperiodic Jobs Set1 = $\{j0 (4, 3, 16), j1 (17, 3, 30)\}$

```
#define TASK_STACKSIZE
                           2048
#define TASK1_PRIORITY
                           1
#define TASK2_PRIORITY
                           2
#define TASK1_ID
#define TASK2_ID
static OS_STK StartupTaskStk[APP_CFG_STARTUP_TASK_STK_SIZE];
static OS_STK Task1_STK[TASK_STACKSIZE];
static OS_STK Task2_STK[TASK_STACKSIZE];
static void StartupTask(void* p_arg);
static void task1(void* p_arg);
static void task2(void* p_arg);
                                   //set : {start time, work time, period time}
TimeTask task1\_set = \{ 0, 1, 4 \};
TimeTask task2\_set = \{ 0, 4, 10 \};
                                    //set : {start time, work time, period time}
```

Periodic Task Set2 = $\{\tau 1 \ (0, 2, 8), \tau 2 \ (0, 3, 10), \tau 3 \ (0, 5, 20), \tau 4_ServerSize \ (0.2)\}$ Aperiodic Jobs Set2 = $\{j0 \ (12, 3, 28), j1 \ (14, 2, 39)\}$

```
#define TASK_STACKSIZE
                            2048
#define TASK1 PRIORITY
                            1
#define TASK2_PRIORITY
                            2
                            3
#define TASK3_PRIORITY
#define TASK1 ID
                            1
                            2
#define TASK2_ID
                            3
#define TASK3_ID
static OS_STK StartupTaskStk[APP_CFG_STARTUP_TASK_STK_SIZE];
static OS STK Task1 STK[TASK STACKSIZE];
static OS_STK Task2_STK[TASK_STACKSIZE];
static OS_STK Task3_STK[TASK_STACKSIZE];
static void StartupTask(void* p_arg);
static void task1(void* p_arg);
static void task2(void* p_arg);
static void task3(void* p_arg);
□void task1(void *p_arg) {
                                       OSTaskCreateExt(
       (void)p_arg;
                                           task1,
                                           &task1 set.
       while (1) {
                                           &Task1_STK[TASK_STACKSIZE - 1],
                                           TASK1_PRIORITY,
                                           TASK1_ID,
       }
                                           &Task1_STK[0],
                                           TASK_STACKSIZE,
                                           &task1_set,
                                           (OS_TASK_OPT_STK_CHK | OS_TASK_OPT_STK_CLR));
□void task2(void* p_arg) {
       (void)p_arg;
                                        OSTaskCreateExt(
                                           task2,
       while (1) {
                                           &task2 set.
                                           &Task2_STK[TASK_STACKSIZE - 1],
                                           TASK2_PRIORITY,
                                           TASK2_ID,
                                           &Task2_STK[0],
                                           TASK_STACKSIZE,
                                           &task2_set,
□void task3(void* p_arg) {
                                           (OS_TASK_OPT_STK_CHK | OS_TASK_OPT_STK_CLR));
      (void)p_arg;
       while (1) {
                                        OSTaskCreateExt(
                                           task3,
                                           &task3_set,
                                           &Task3_STK[TASK_STACKSIZE - 1],
                                           TASK3_PRIORITY,
                                           TASK3_ID,
                                           &Task3_STK[0],
                                           TASK STACKSIZE.
                                           &task3_set,
                                           (OS_TASK_OPT_STK_CHK | OS_TASK_OPT_STK_CLR));
```

```
* project2 *
int jo_1[3] = { 12,3,28 };
int jl_1[3] = { 14,2,39 };
int t4 = 2; int d1 = 0; int d0 = 0;
d0 = jo_1[0] + (jo_1[1]*10 / t4);
d1 = jl_1[0] + (jl_1[1]*10 / t4);
int j0_in = 0; int jl_in = 0;
if (OSTimeGet() == jo_1[0]) {
  printf("%-5d Aperiodic job(0) arrives and sets CUS server's deadline as %d\n",OSTimeGet(),d0);
'else if (OSTimeGet() = j1_1[0]) {
| printf("%-5d Aperiodic job(1) arrives and sets CUS server's deadline as %d\n", OSTimeGet(), dl);
*/
  int jo_1[3] = { 12,3,28 };
int j1_1[3] = { 14,2,39 };
   int t4 = 2; int d1 = 0; int d0 = 0;
  d0 = j_0_1[0] + (j_0_1[1]*10 / t4);

d1 = j_1_1[0] + (j_1_1[1]*10 / t4);
  int j0_in = 0; int j1_in = 0;
  if (OSTimeGet() == jo_1[0]) {
      printf("%-5d Aperiodic job(0) arrives and sets CUS server's deadline as %d\n",OSTimeGet(),d0);
  else if (OSTimeGet() == j1_1[0]) {
     printf("%-5d Aperiodic job(1) arrives and sets CUS server's deadline as %d\n", OSTimeGet(), d1);
  int responsetime = 0:
  int context_switch = 0;
  OS_TCB* ptcb;
  if (OSPrioHighRdy != 0) {
                                                                        //程式開始執行
      ptcb = OSTCBPrioTb1[OSPrioHighRdy];
      TimeTask* point_t = ptcb->OSTCBExtPtr;
                                                                        //指向自定義數據以進行TCB擴展的指針
      if (point_t != 0) {
                                                                        //如果不是idle task
          //如果是completion_task
          if (OSTimeGet() - point t->current start time == point t->work time + point t->preemptive time) {
```

```
a = 0;
if (OSTimeGet() % point_t->period_time == 0) {
   responsetime = point_t->period_time;
                                                           responsetime = OSTimeGet() % point_t->period_time - point_t->start_time; //計算respondtime
                                      | responsetume = Unimode(t) * point(-*period_name - point(**period_name - point(**perio
                                                   //Delay current task
                                                           }
ptcb>OSTCEOly = point_t-period_time * (point_t->executive_count + 1) - OSTimeGet() + point_t->start_time;
OS_TRACE_TARK_BLY(point_t->period_time * (point_t->executive_count + 1) - OSTimeGet() + point_t->start_time);
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              //Load ticks in TCB
                                    }
point_t->xxccutive_count++;
if (OSF-indigiaRe) = (NISSD)((y << Su) + OSI/alea/Th[(OSR/Thl[y]]))
point_t->constit_switch+-; / 片澤constiswitch
constit_switch = point_t->constit_switch+) / 片字 constitution
point_t->constitution=0; //constitution##
point_t->constitution=0; //constitution##
point_t->constitution=0; //constitution###
                            |/OST-codipiddy = (NTSD)((y << So) + OSIndapTol(OSRdyTol(y)))
| (元果是presention_task
| (OSInde(t) - point_t > point_
                                    printf("%-be%-144%-386%-286%-286%-286%-286%-138%-286%); USIMBOOK() + 1, "Umpjetion", "take(", 2, ")
else {
printf("%-b6", ONImBOOK());
printf("%-14*, "Premption");
printf("%-14*, "Premption");
printf("%-586%-2860%-98", "take(", ptob~205TCBId, ")(", point_t->executive_count, ")");//print 現在的 take ID
c = 0;
                                        }
point_t->context_switch++; //計算concutswitch
if (point_t->prementive_time > 0) (point_t->prementive_time = GSTimeGet() - point_t->prementive_time)//更新prementivetime
else (point_t->prementive_time = OSTimeGet() ) //更新prementivetime
                                            //如果是idle task
                                            //MuxEndle task else if (point t == 0) {
    if (OSTimeGet() |= 14 && OSTimeGet() != 16 && OSTimeGet() != 20) {
        printf("%-6d", OSTimeGet());
        printf("%-14s", "Preemption");
        printf("%-14s", "Preemption");
        printf("%-5s%d%-11s", "task(", OSPrioMighRdy, ")");//print idle task
                                                                   | relation 
                                            if (OSPrioRighRdy != (INTSU)((y << 3u) + OSUnMapTb1[OSRdyTb1[y]]) !! (point t->context_switch == 0 && point t->executive_count != 0)) {
    OSPrioRighRdy = (INTSU)((y << 3u) + OSUnMapTb1[OSRdyTb1[y]]);
    ptcb = OSTCBPrioTb1[OSPrioRighRdy];
                                                                     ptob = OSTCBFrioTbl[GSFrioNitghRdy];
point_t = ptob-OSTCBExtPtr;
if (point_t != 0) {
    if (a == 0) { printf("%-5%M%-2%M%-15%", "task(", ptob-OSTCBId, ")(", point_t->executive_count, ")"); c = 1; } }
    else { printf("%-5%M%-2%M%-15%"n, "task(", ptob-OSTCBId, ")(", point_t->executive_count, ")"); }
    if (OSFrioNitghRdy! = OSFrioCur) { point_t->context_switch+;}/fi\bar{E}_comext_switch
    if (point_t->preemptive_time == 0) { point_t->context_switch == OSTimeOst();}//iki\bar{E}_task start time
    if (point_t->preemptive_time >= 0) { point_t->preemptive_time = OSTimeOst() - point_t->preemptive_time;}//更新preemptive_time
                                                                       //如果下一個task是idel task
                                                                       else if (point_t == 0) {
| printf("%-5s%d%-17s", "task(", OSPrioHighRdy, ")");//print task(63)
                                                                       if (responsetime > 0) {
   printf("%-20d%d\n", responsetime, context_switch);
                                                                                                    responsetime = 0:
```

```
//print response time and context switch
        if (responsetime > 0) {
           printf("%-20d%d\n", responsetime, context_switch);
           responsetime = 0;
        else {
           printf("\n");
         //處理missdeadline
        ptcb = OSTCBList;
while (ptcb != (OS_TCB*)0) {
           TimeTask point_t = ptcb->OSTCBExtPtr;
if (point_t != 0 && OSTimeGet() > point_t->period_time * (point_t->executive_count + 1)) {
             (point_t != U aw UslimeSet() > point_t->period_time * (point_t->executive_count + 1)) {
  printf("%-0d", OSTimeSet());
  printf("%-14s", "MissDeadline");
  printf("%-15x8d%-25x8d%-9s", "task(", ptcb->OSTCBId, ")(", point_t->executive_count, ")");
  printf("-------\n");
              system("pause");
           ptcb = ptcb->OSTCBNext;
   //initizal setting
   else if (OSPrioHighRdy == 0) {
      OSPrioHighRdy = (INTSU)((y << 3u) + OSUnMapTbl[OSRdyTbl[y]]);
      ptcb = OSTCBList;
      while (ptcb != (OS_TCB*)0) {
         TimeTask* point_t = ptcb->OSTCBExtPtr;
         if (point_t != 0 && point_t->start_time > Ou) {
                                                                    // O means no delay!
            OS_ENTER_CRITICAL();
             y = ptcb->OSTCBY;
                                                     //Delay current task
            OSRdyTb1[y] &= (OS_PRIO)~ptcb->OSTCBBitX;
OS_TRACE_TASK_SUSPENDED(ptcb);
            if (OSRdyTbl[y] == Ou) {
               OSRdyGrp &= (OS_PRIO)~ptcb->OSTCBBitY;
            ptcb->OSTCBDly = point_t->start_time;
                                                    //Load ticks in TCB
            OS_TRACE_TASK_DLY(point_t->start_time);
OS_EXIT_CRITICAL();
         ptcb = ptcb->OSTCBNext;
       y = OSUnMapTb1[OSRdyGrp];
      OSPrioHighRdy = (INT8V)((y << 3u) + OSVnMapTbl[OSRdyTbl[y]]);
   |
|-
|
* project2 *
*/
≒typedef struct TimeTask {
                         //作業開始時間
   int start time:
                         //工作工作時間
    int work_time;
                         //任務期
    int period_time;
    int context_switch;
                          //任務完成後,清除上下文開關
    int executive_count;
                          //任務執行時間
    int current_start_time;
                          //作業的實際開始時間
                          //搶占任務時間
    int preemptive_time;
 }TimeTask;
 project2
*/
```

