

1. The system flow and the explanation of the process(functions).

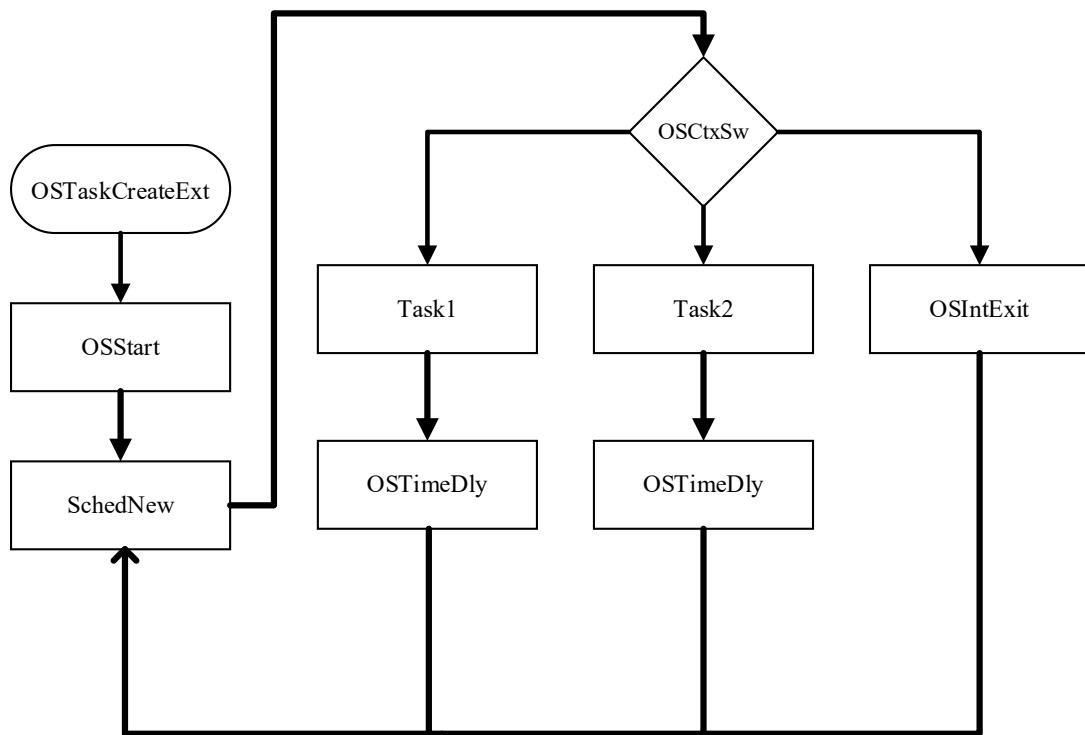


Fig 1. System flow

一開始，先從OSTaskCreateExt建立兩個task，進入到OSStart裡面的SchedNwe去設定參數，並將優先權更新，透過OSCtxSw去切換目前最高優先權，它會透過OStimeDly來決定優先度，當有空窗期時，像tick(3)時，它會切至Taskidle再到OSIntExit再回到task(1)繼續動作，當如果遇到像6、9、12等數字會先以，task(1)->task(2)->task(63)，其中只會進OSIntExit都會是task(63)，接下來就重複循環。

2. The screenshot of the result.

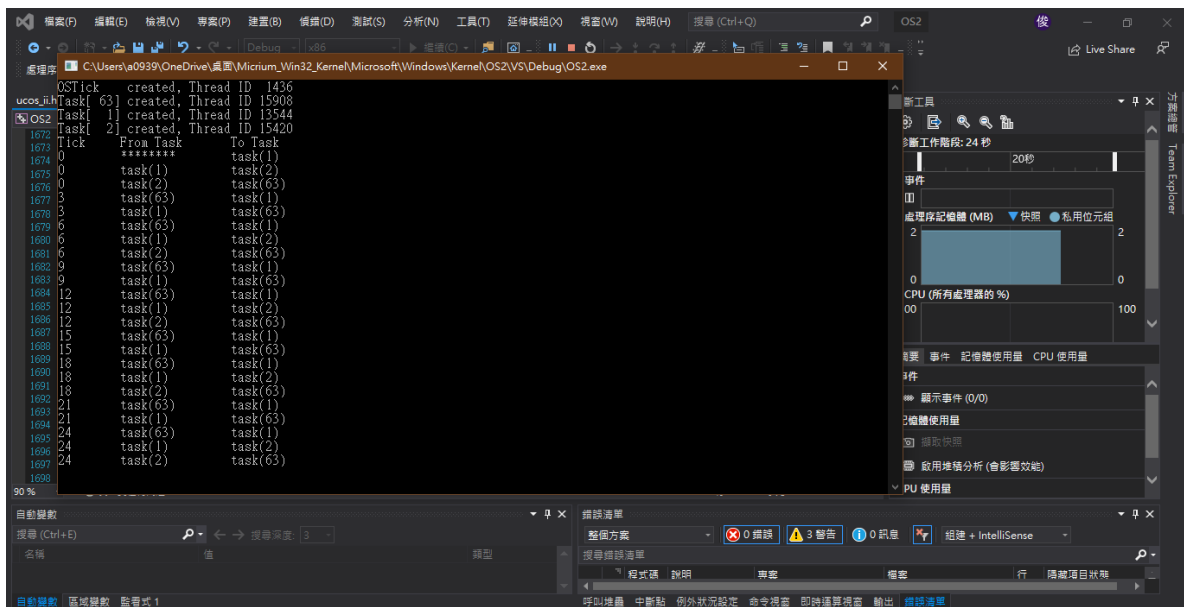


Fig 2. Result

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

在圖3. 中，我在OSStart將我增加的程式放在OS_SchedNew()下，第一次執行From Task會是task(0)我轉成”*****” ，而OSTimeGet()主要告訴目前的tick是多少，To Task我則是代入OSPrioHighRdy等下會執行的task。

```
void OSStart (void)
{
    if (OSRunning == OS_FALSE) {
        OS_SchedNew(); /* Find highest priority's task priority number */
        printf("\ntick \t From Task \t To Task\n"); //1019 add code
        if (OSPrioCur == 0) {
            printf("%d \t ***** \t task(%d)\n", OSTimeGet(), OSPrioHighRdy);
        }
        OSPrioCur = OSPrioHighRdy;
        OSTCBHighRdy = OSTCBPrioTbl[OSPrioHighRdy]; /* Point to highest priority task ready to run */
        OSTCBCur = OSTCBHighRdy;
        OSStartHighRdy(); /* Execute target specific code to start task */
    }
}
```

Fig 3. Code1

在圖4. 中，OS_Sched前面會先執行最高優先權的運算，因此我將printf放入其中，可以得知Tick、From Task 、TO Task目前的狀態。

```
if OS_TASK_CREATE_EXT_EN > 0u
if defined(OS_TLS_TBL_SIZE) && (OS_TLS_TBL_SIZE > 0u)
    OS_TLS_TaskSw();
endif
endif

printf("%d \t task(%d) \t task(%d)\n", OSTimeGet(), OSPrioCur, OSPrioHighRdy); //1019 add code
OS_TASK_SW(); /* Perform a context switch */
}
}
OS_EXIT_CRITICAL();
```

Fig 4. Code2

在圖5. 中，透過講義中的Embedded OS p6得知，OSIntExit()也會計算最高優先權，因此我也將printf放在OSIntCtxSw之前，已得知目前的狀態。

```
if OS_TASK_CREATE_EXT_EN > 0u
if defined(OS_TLS_TBL_SIZE) && (OS_TLS_TBL_SIZE > 0u)
    OS_TLS_TaskSw();
endif
endif

OS_TRACE_ISR_EXIT_TO_SCHEDULER();
printf("%d \t task(%d) \t task(%d)\n", OSTimeGet(), OSPrioCur, OSPrioHighRdy); //1019 add code
OSIntCtxSw(); /* Perform interrupt level ctx switch */
} else {
    OS_TRACE_ISR_EXIT();
}
} else {
    OS_TRACE_ISR_EXIT();
}
```

Fig 5. Code3