

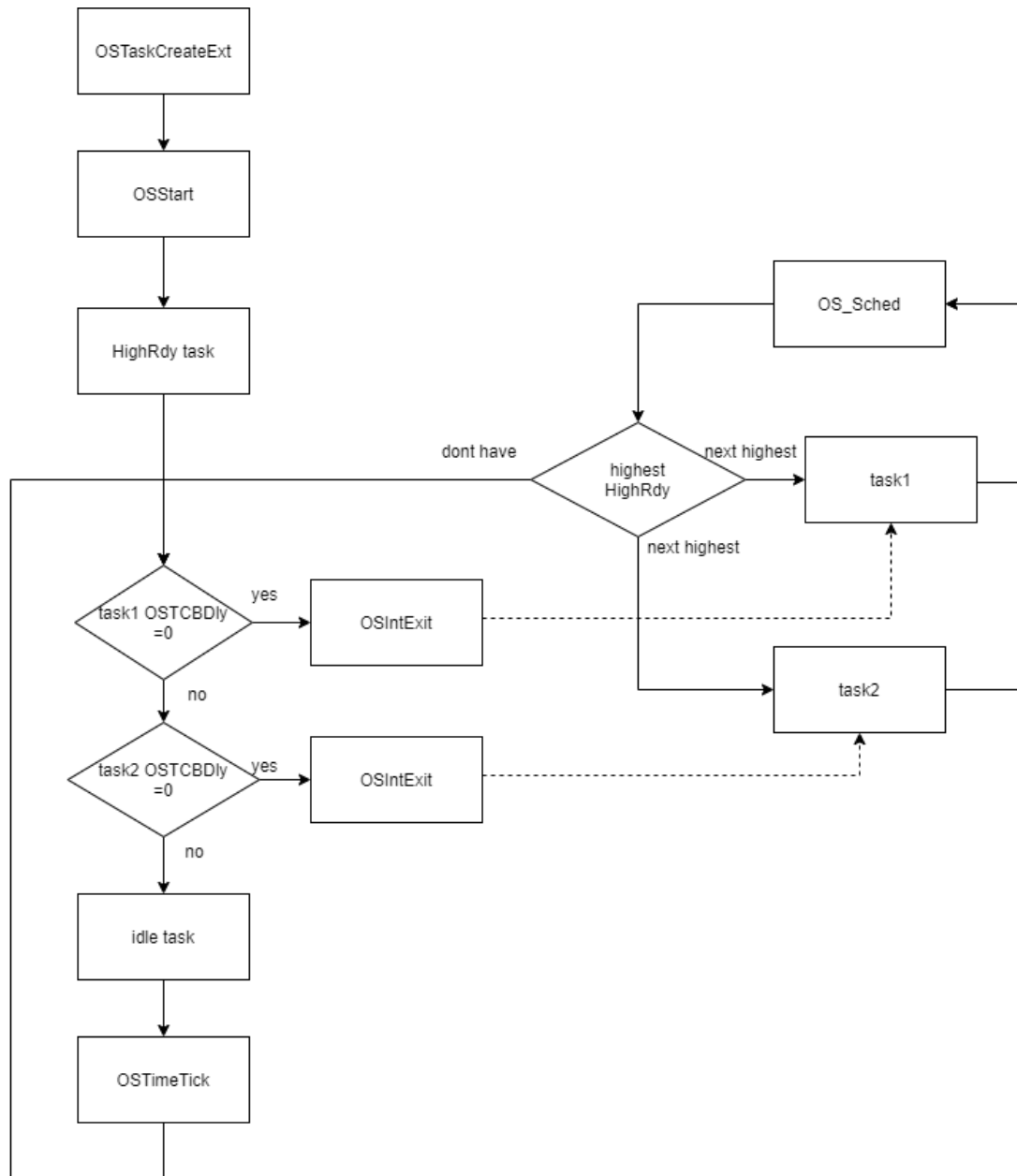
國立台灣科技大學
National Taiwan University of Science and Technology

Embedded OS Implementation, Fall 2020
Homework #1 (due October 21, 2020
(Wednesday) 13:00)

老師：陳雅淑

學生：邱曄凱

— 、 flow and process



名稱	解釋
OSTaskCreateExt	創建 task
OSStart	程式開始時執行
idle task	沒事時做的，維持程式運作
OSTimeTick	當 tick 一次， 把所有的 OSTCBDly 減一
OS_Sched	從 task 出來，找下一個要做的
OSIntExit	從中斷中出來，找下一個要做的

二、result

```

C:\Users\user\Documents\Git\嵌入式作業系統實作HW\20201014_HW1\M
OSTick    created, Thread ID 14720
Task[ 63] created, Thread ID 15056
Task[  1] created, Thread ID  8828
Task[  2] created, Thread ID  8504

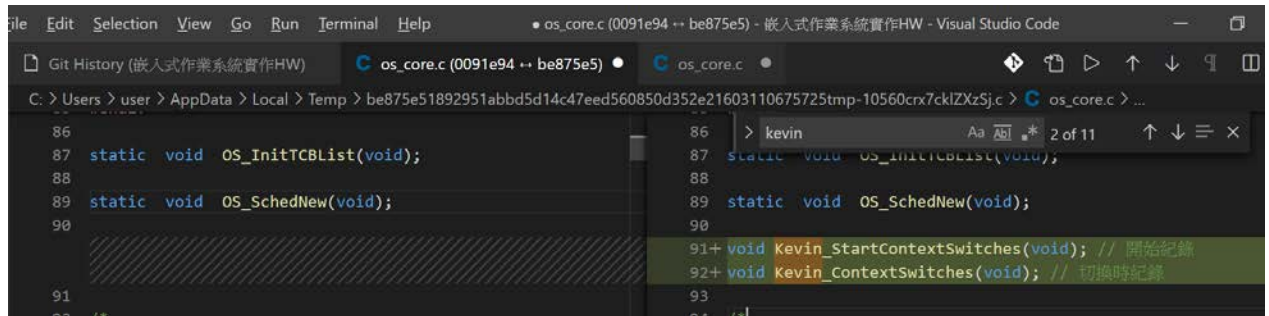
Tick      From Task      To Task
0         *****
0         task(1)         task(1)
0         task(1)         task(2)
0         task(2)         task(63)
3         task(63)        task(1)
3         task(1)         task(63)
6         task(63)        task(1)
6         task(1)         task(2)
6         task(2)         task(63)
9         task(63)        task(1)
9         task(1)         task(63)
12        task(63)        task(1)
12        task(1)         task(2)
12        task(2)         task(63)
15        task(63)        task(1)
15        task(1)         task(63)
18        task(63)        task(1)
18        task(1)         task(2)
18        task(2)         task(63)
21        task(63)        task(1)
21        task(1)         task(63)
24        task(63)        task(1)
24        task(1)         task(2)

```

三、code and MARK

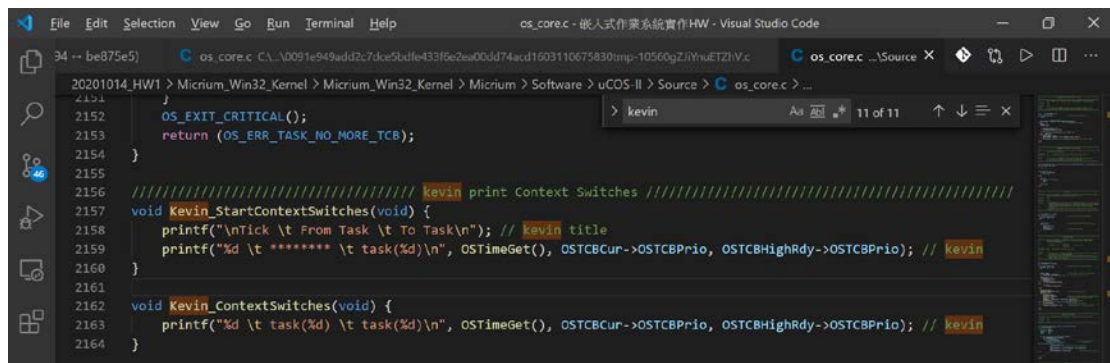
In File -> os.core.c

1. 宣告 funsion



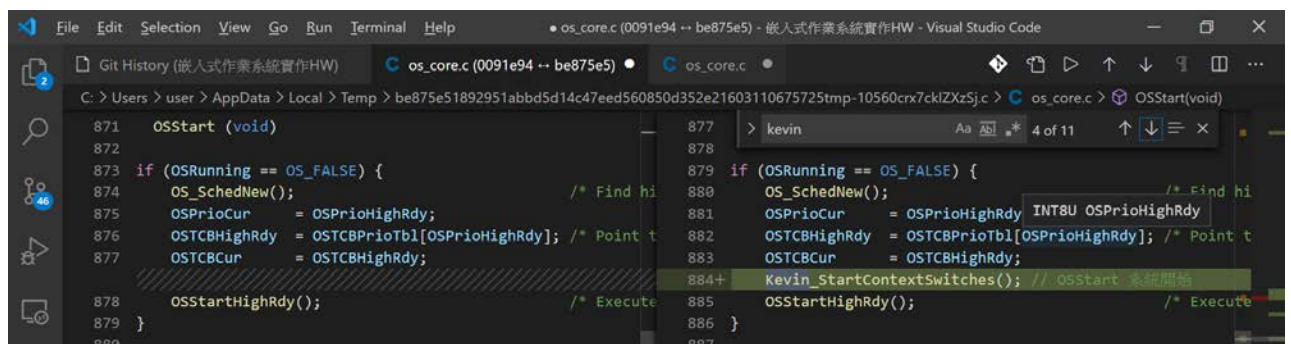
```
86
87 static void OS_InitTCBList(void);
88
89 static void OS_SchedNew(void);
90
91
92
93
```

2. Funsion 程式



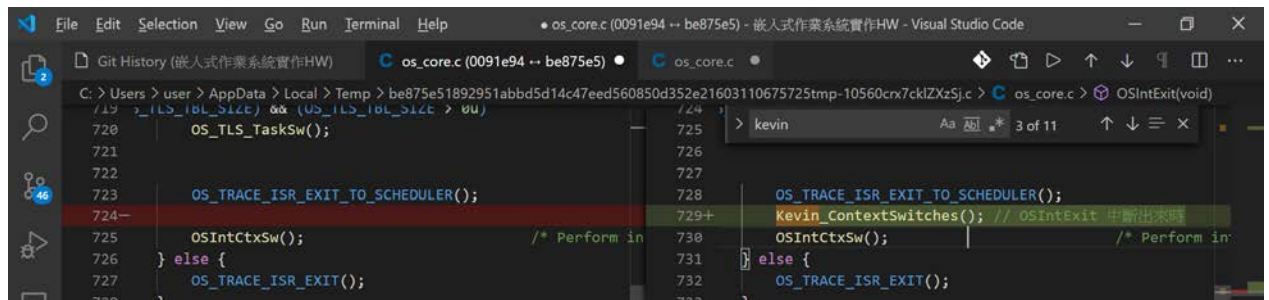
```
2152 OS_EXIT_CRITICAL();
2153 return (OS_ERR_TASK_NO_MORE_TCB);
2154 }
2155
2156 /////////////////////////////////////////////////// kevin print Context Switches ///////////////////////////////////
2157 void Kevin_StartContextSwitches(void) {
2158     printf("\nTick \t From Task \t To Task\n"); // kevin title
2159     printf("%d \t ***** \t task(%d)\n", OSTimeGet(), OSTCBCur->OSTCBPrio, OSTCBHighRdy->OSTCBPrio); // kevin
2160 }
2161
2162 void Kevin_ContextSwitches(void) {
2163     printf("%d \t task(%d) \t task(%d)\n", OSTimeGet(), OSTCBCur->OSTCBPrio, OSTCBHighRdy->OSTCBPrio); // kevin
2164 }
```

3. 在 OSStart 列印第一行



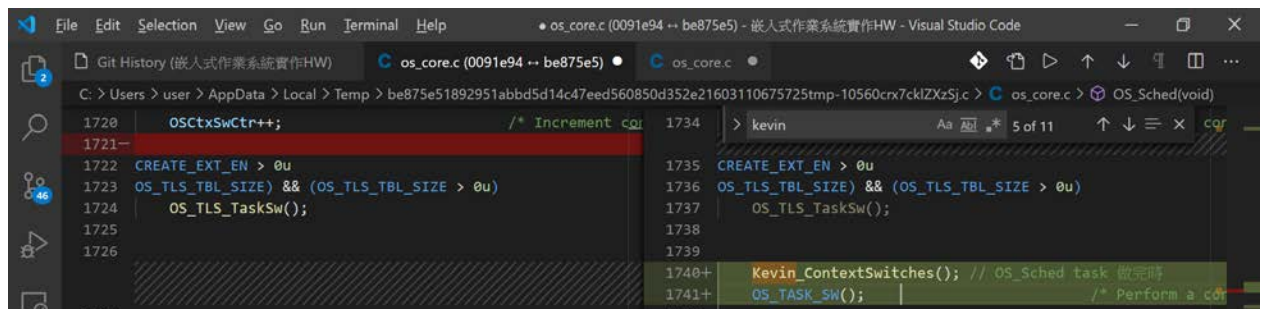
```
871 OSStart (void)
872
873 if (OSRunning == OS_FALSE) {
874     OS_SchedNew(); // Find hi
875     OSPrioCur = OSPrioHighRdy;
876     OSTCBHighRdy = OSTCBPrioTbl[OSPrioHighRdy]; // Point t
877     OSTCBCur = OSTCBHighRdy;
878     OSStartHighRdy(); // Execute
879 }
880
881
882 if (OSRunning == OS_FALSE) {
883     OS_SchedNew(); // Find hi
884     OSPrioCur = OSPrioHighRdy;
885     OSTCBHighRdy = OSTCBPrioTbl[OSPrioHighRdy]; // Point t
886     OSTCBCur = OSTCBHighRdy;
887     Kevin_StartContextSwitches(); // OSStart 系統開始
888     OSStartHighRdy(); // Execute
889 }
```

4. 在 OSIntExit 列印，紀錄中斷結束後，準備 context switch 的過程



```
File Edit Selection View Go Run Terminal Help
os_core.c (0091e94 ↔ be875e5) - 嵌入式作業系統實作HW - Visual Studio Code
Git History (嵌入式作業系統實作HW)
os_core.c
C:\Users> user > AppData > Local > Temp > be875e51892951abbd5d14c47eed560850d352e21603110675725tmp-10560crx7cklZXzSj.c > os_core.c > OSIntExit(void)
720 OS_TLS_TaskSw();
721
722
723 OS_TRACE_ISR_EXIT_TO_SCHEDULER();
724 OSIntCtxSw(); /* Perform in
725 } else {
726 OS_TRACE_ISR_EXIT();
727
728 OS_TRACE_ISR_EXIT_TO_SCHEDULER();
729+ Kevin_ContextSwitches(); // OSIntExit 中斷出來時
730 OSIntCtxSw(); /* Perform in
731 } else {
732 OS_TRACE_ISR_EXIT();
733
```

5. 在 OS_Sched 列印第一行，紀錄 task 結束後，準備 context switch 的過程



```
File Edit Selection View Go Run Terminal Help
os_core.c (0091e94 ↔ be875e5) - 嵌入式作業系統實作HW - Visual Studio Code
Git History (嵌入式作業系統實作HW)
os_core.c
C:\Users> user > AppData > Local > Temp > be875e51892951abbd5d14c47eed560850d352e21603110675725tmp-10560crx7cklZXzSj.c > os_core.c > OS_Sched(void)
1720 OS_CtxSwCtr++; /* Increment co
1721 CREATE_EXT_EN > 0u
1722 OS_TLS_TBL_SIZE && (OS_TLS_TBL_SIZE > 0u)
1723 OS_TLS_TaskSw();
1724
1725
1726
1727
1728
1729
1730
1731
1732
1733
1734 Kevin
1735 CREATE_EXT_EN > 0u
1736 OS_TLS_TBL_SIZE && (OS_TLS_TBL_SIZE > 0u)
1737 OS_TLS_TaskSw();
1738
1739
1740+ Kevin_ContextSwitches(); // OS_Sched task 做完時
1741+ OS_TASK_SW(); /* Perform a co
1742
```