**嵌入式作業系統實作**

**Embedded OS Implementation**

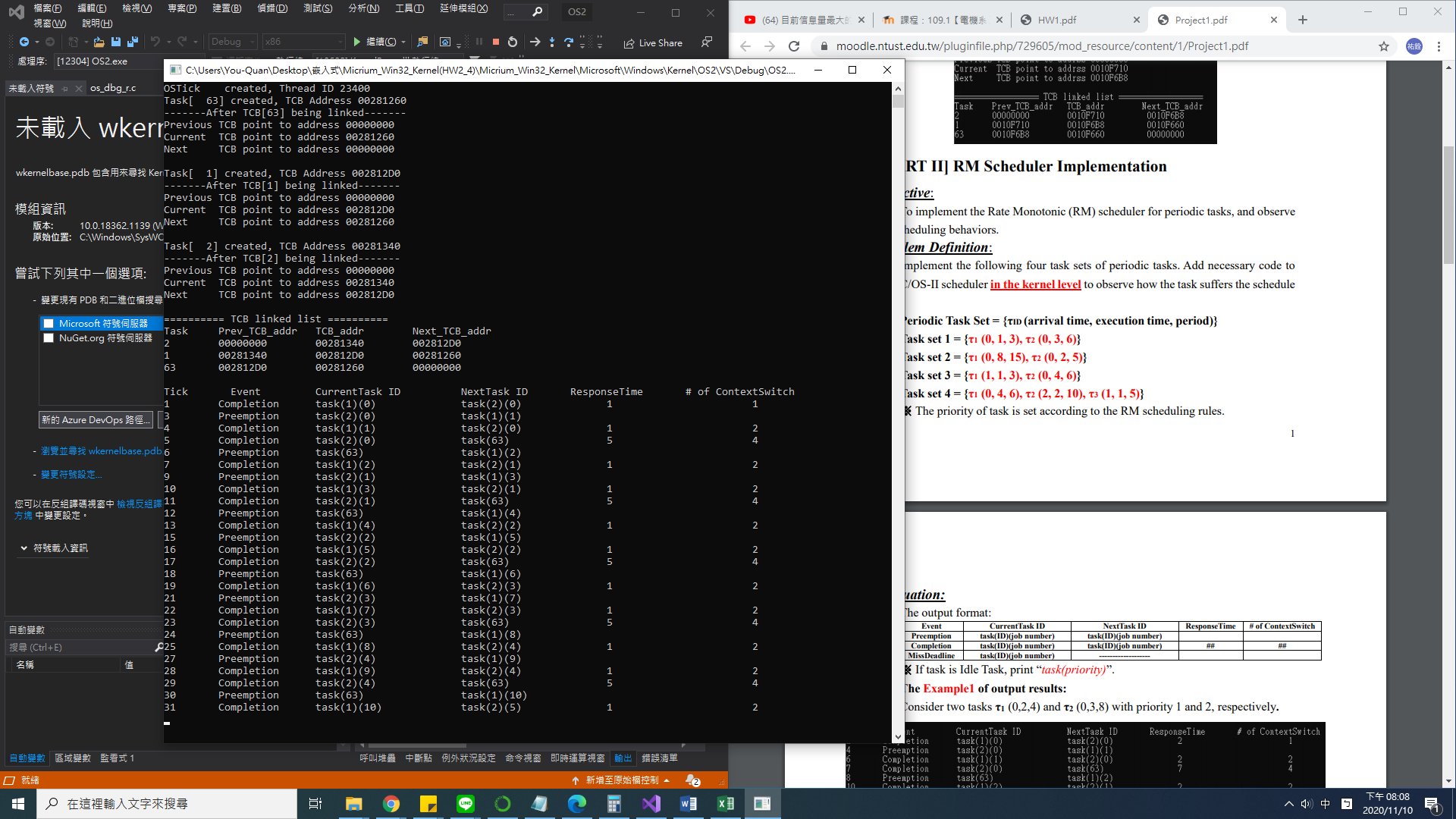
**PA\_1**

指導教授: 陳雅淑 教授

課程學生: M10907314張祐銓

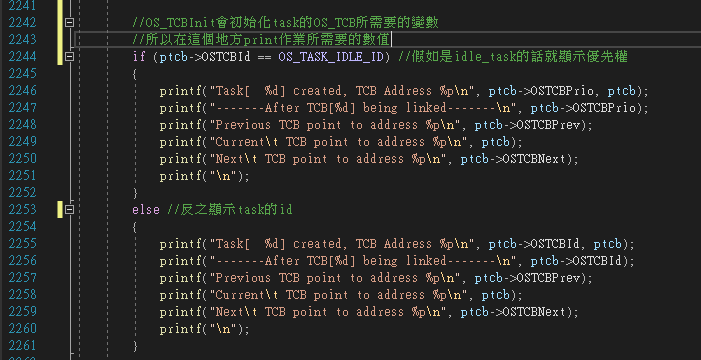
**[ PART I] Task Control Block Linked List [20%]**

* The screenshot results. (10%)

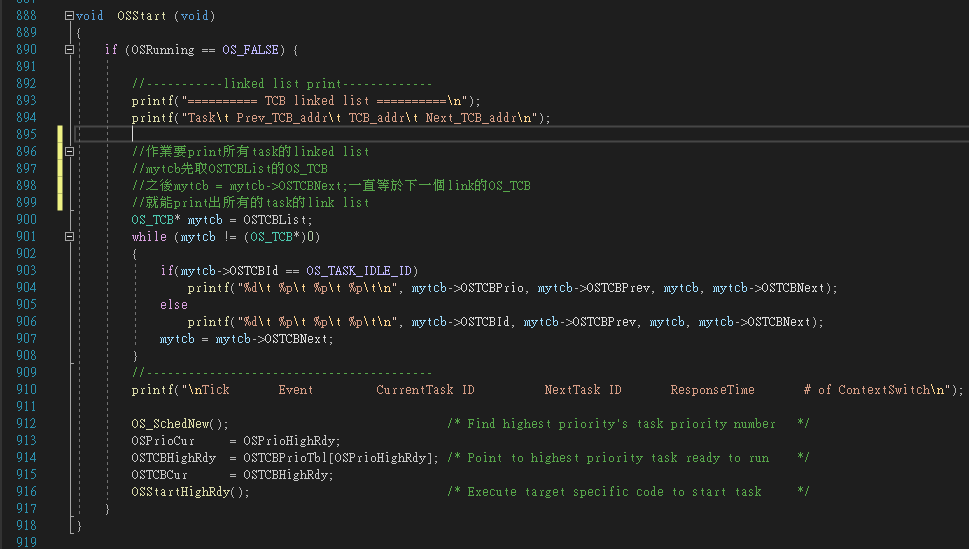


* A report that describes your implementation (please attach the screenshot of the code and MARK the modified part). (10%)

1. 上半部顯示task被創建時的link，這段程式碼加在OS\_TCBInit下，因為在執行OSTaskCreateExt建立Task內會執行到OS\_TCBInit這個function，OS\_TCBInit會初始化task的OS\_TCB的數值，所以可以在這裡抓到task被建立時的link數值。



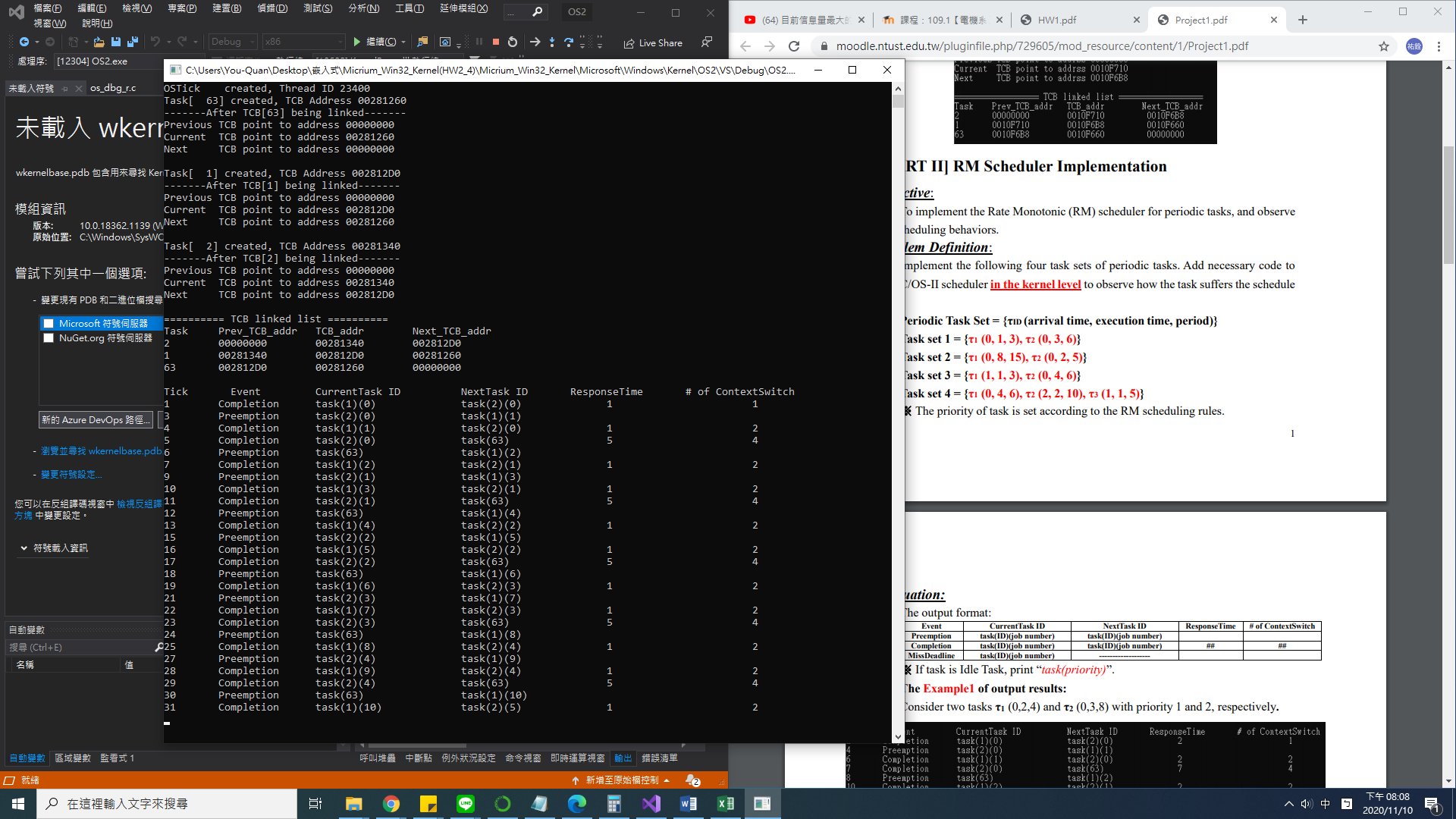
1. 下半部顯示linked list的表，取OSTCBList做開頭，使用一個while一直指向下一個link的OS\_TCB，然後顯示出所有task的link情況。



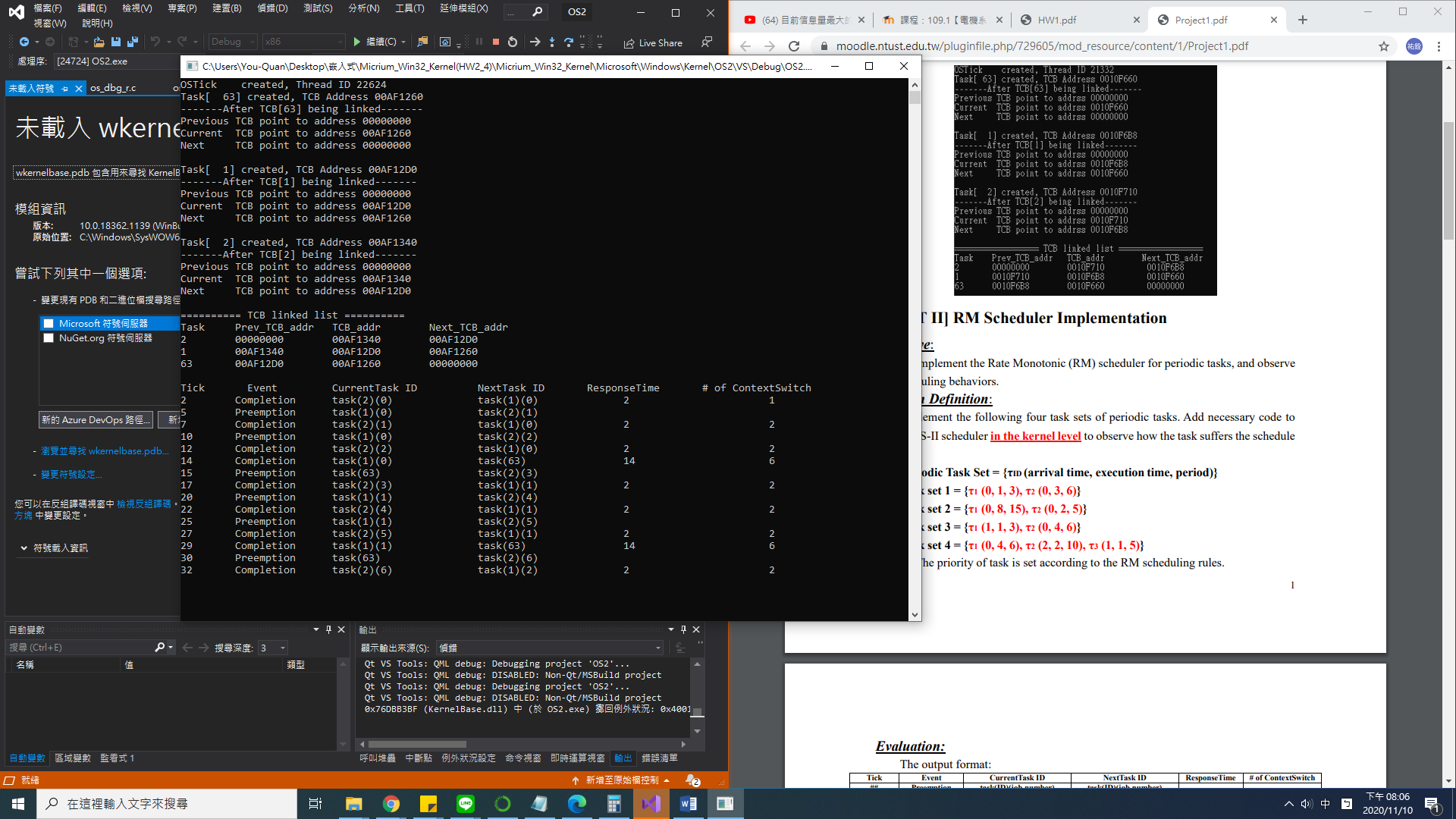
**[ PART II] RM Scheduler Implementation [80%]**

* The screenshot results (with the given format) of four task sets. (Time ticks 0-30 or miss deadline). (40%)

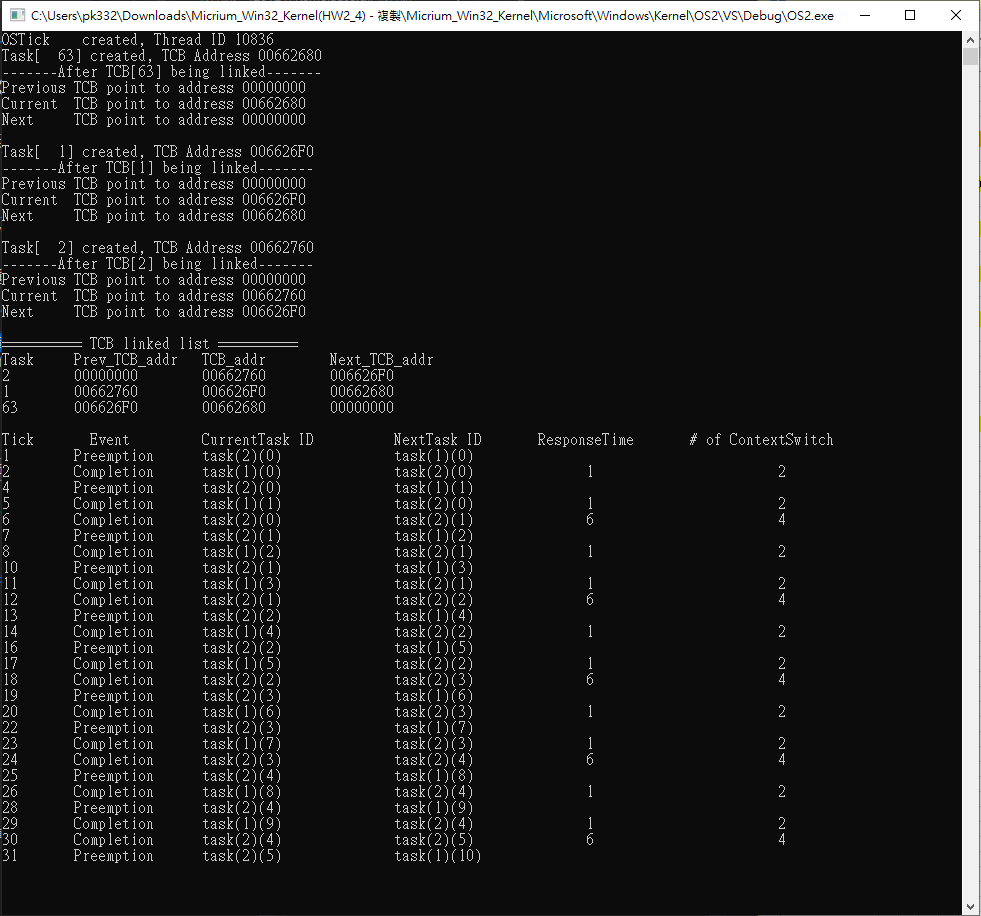
**Task set 1 = {τ1 (0, 1, 3), τ2 (0, 3, 6)}**



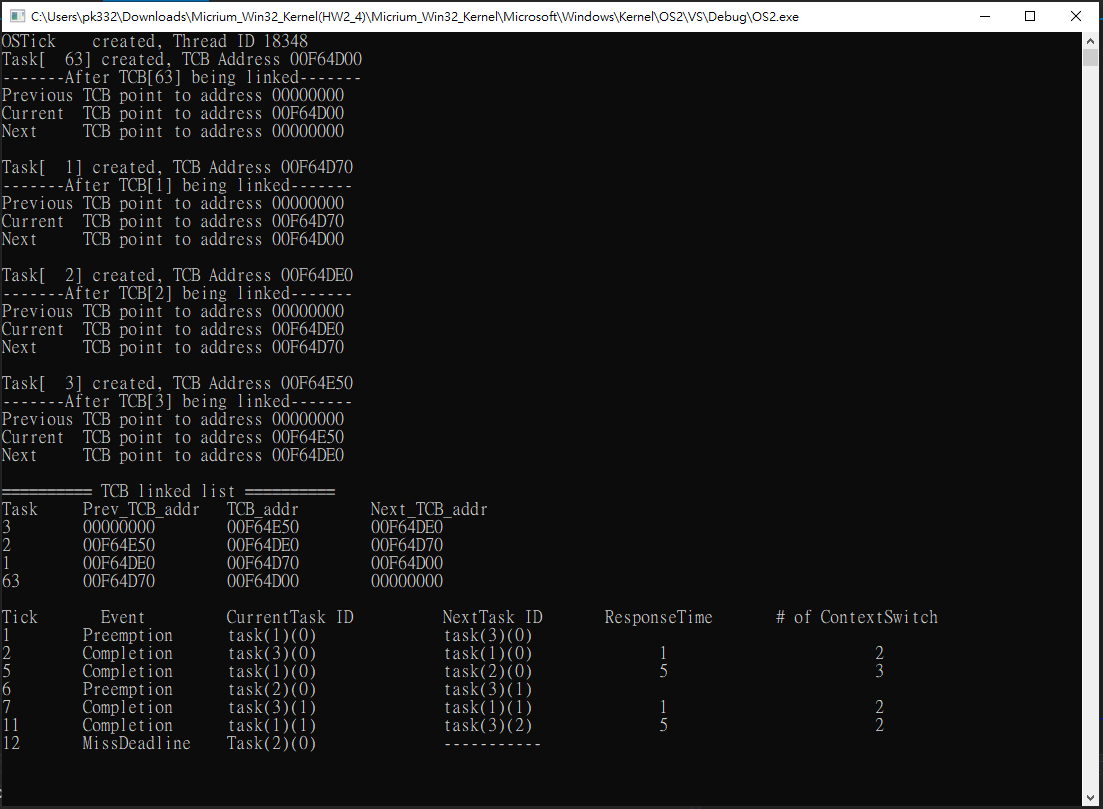
**Task set 2 = {τ1 (0, 8, 15), τ2 (0, 2, 5)}**



**Task set 3 = {τ1 (1, 1, 3), τ2 (0, 4, 6)}**



**Task set 4 = {τ1 (0, 4, 6), τ2 (2, 2, 10), τ3 (1, 1, 5)}**



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

實現方式條列簡述說明:

1. 在OS\_TCB中新增變數

INT32U begin\_ready\_time; //紀錄task變ready的時間點

INT32U response; //反應時間

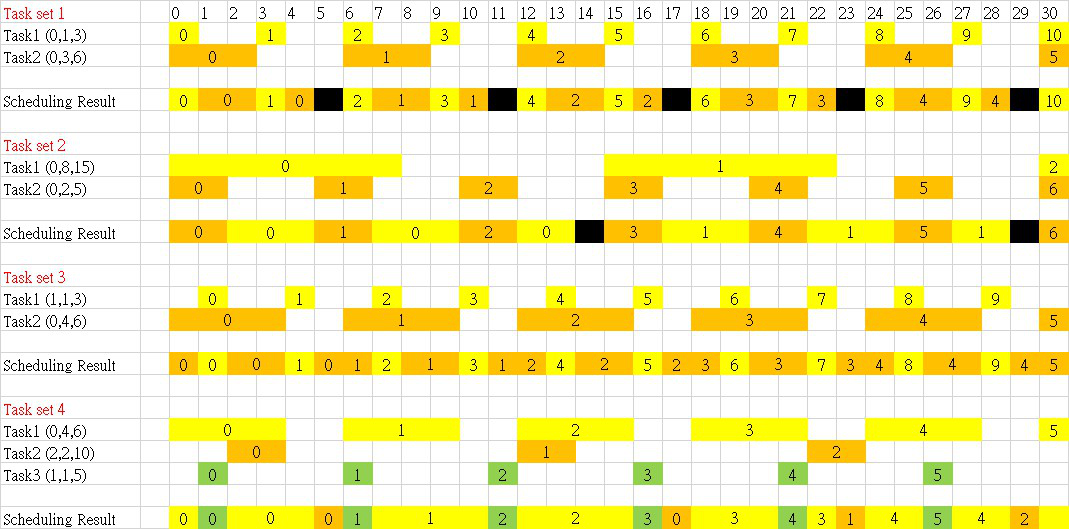
INT32U arrival; //到達時間

INT32U execution; //執行時間

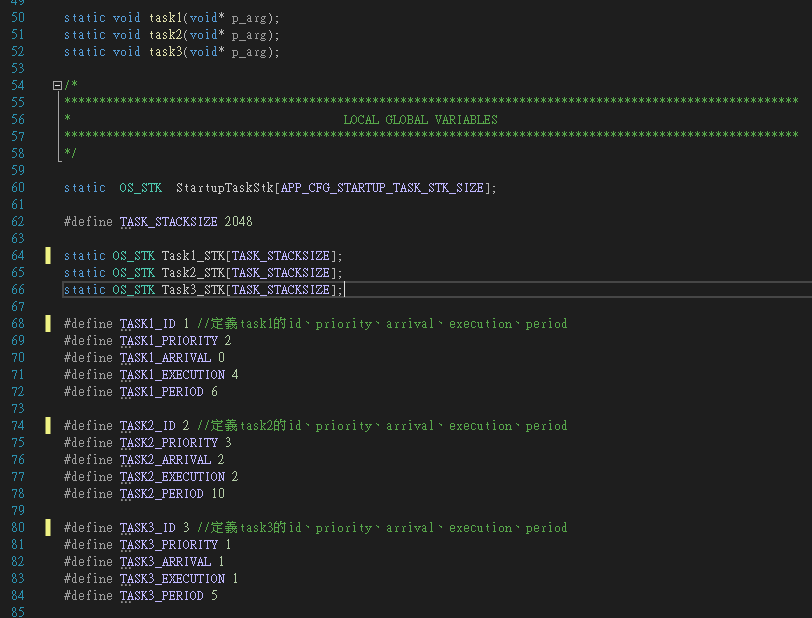
INT32U period; //週期

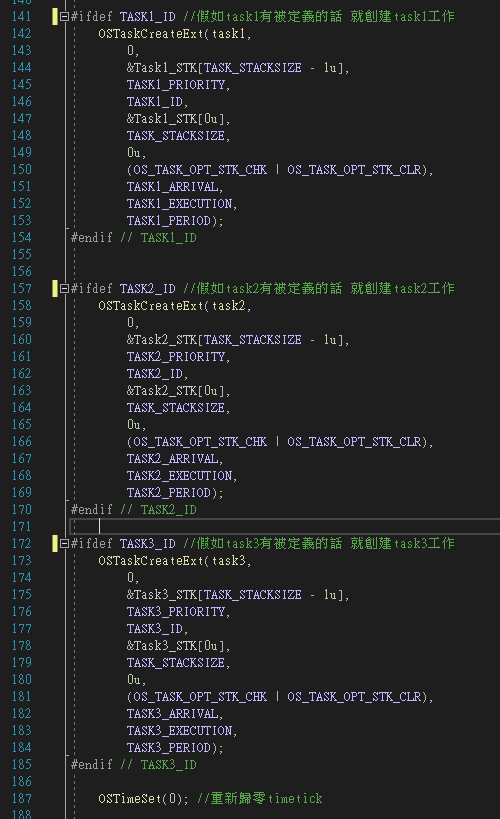
INT32U job\_id; //工作次數

1. 在OSTimeTick中去察看目前的ReadyTable有哪些Task正在Ready狀態，把所有正在Ready且不是目前正在執行的task的所有task的response都加一，因為task正在Ready卻沒有被執行表示被延後了一個tick，response一開始初始值是週期內的執行時間，所以task工作執行只要沒有執行滿response的時間就會一直卡在while迴圈內。
2. Task執行完會執行OS\_Sched()去切換給下個Task，所以在OS\_Sched()中顯示完成的字串。
3. 當TimeTick中斷產生之後會進OSIntExit()把低優先權的task中斷給高優先權的task，所以在這裡顯示中斷狀態的字串。
4. 當有task已經快要完成時有可能會被其他優先權高的task給搶占，所以會在OSIntExit()增加判斷程式迴避掉那次的context switch，讓快要做完的task先完成它的工作。
5. 當有一個task完成了一個周期內的工作後要做下個週期的工作時因為不會進入OS\_Sched()做context switch，所以在OSTimeDly(0)時表示task的反應時間等於週期，在OSTimeDly內去print完成的字串。
6. 判斷是否有task已經miss deadline，只要在OSTimeTick中去判斷reponse是否大於period，表示說反應時間已經大於週期就是miss deadline。
7. 以下是程式的截圖以及內部程式碼中較詳細的註解說明



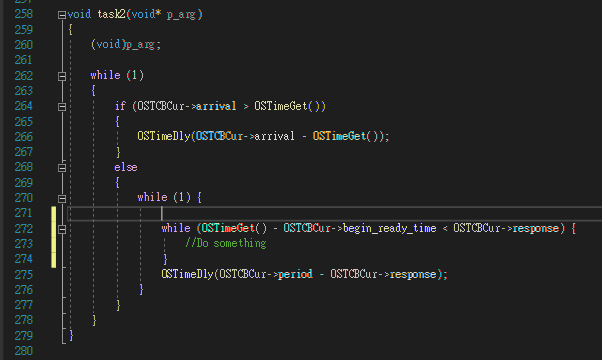
Main.c

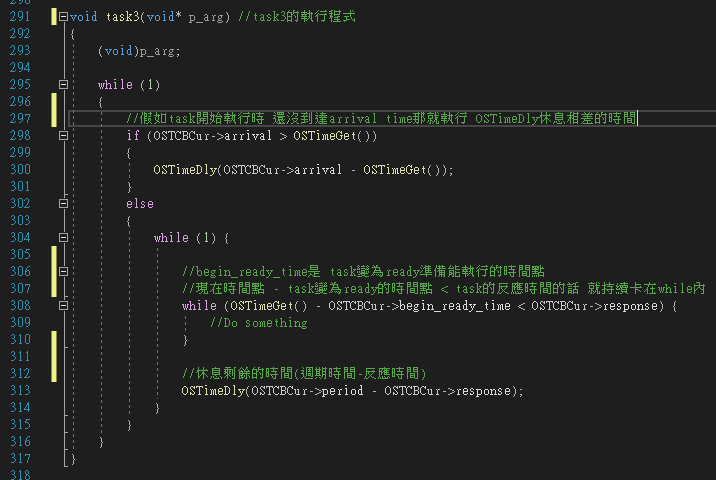




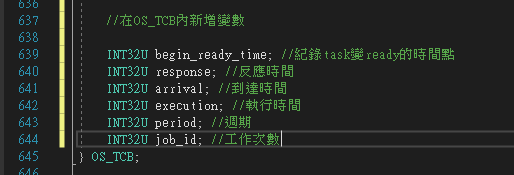




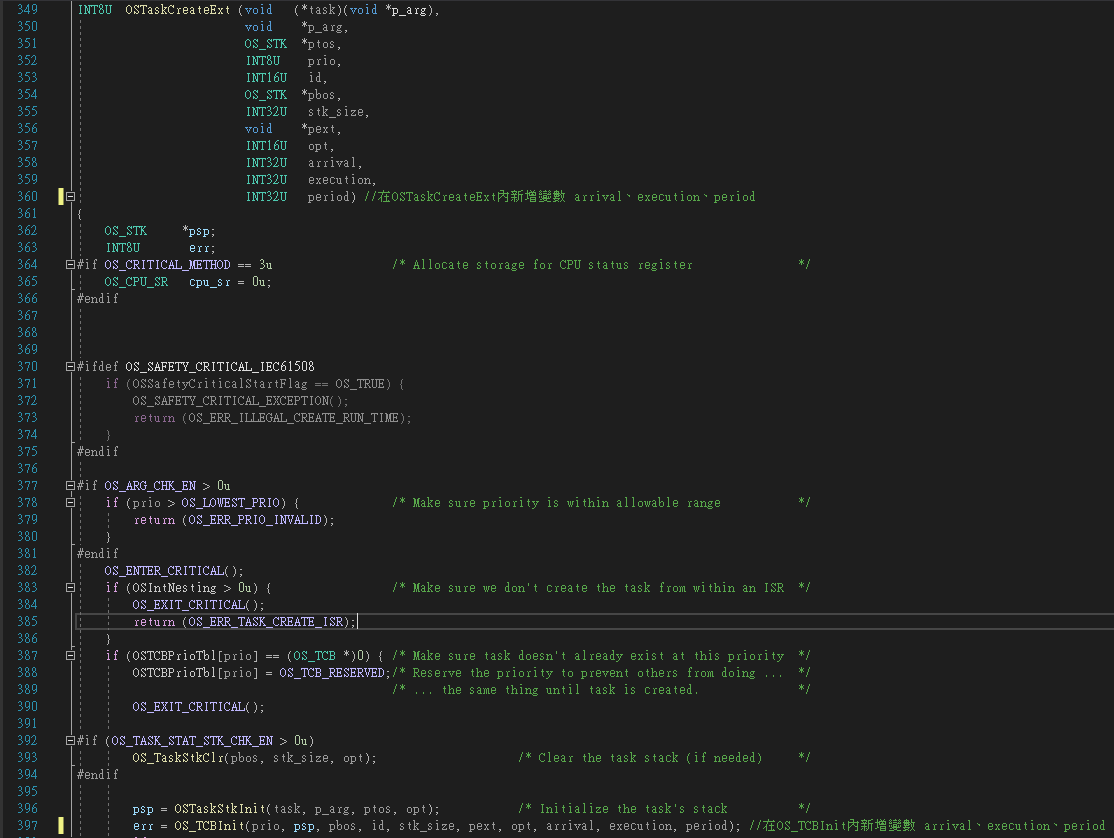




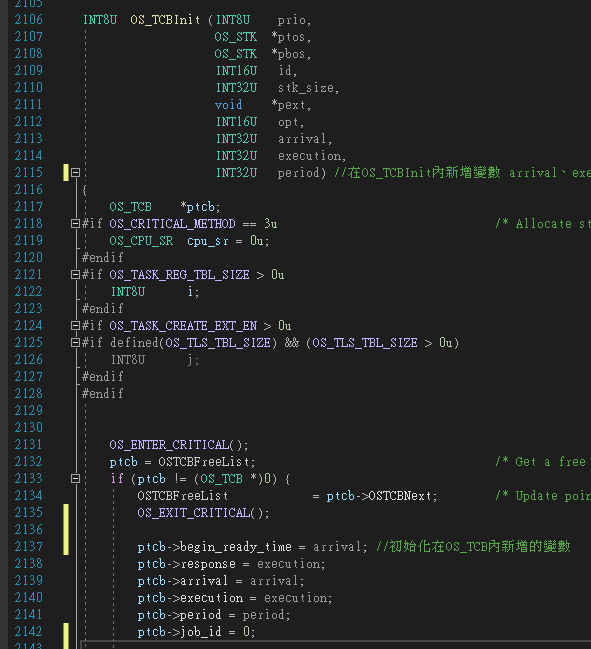
ucos\_ii.h

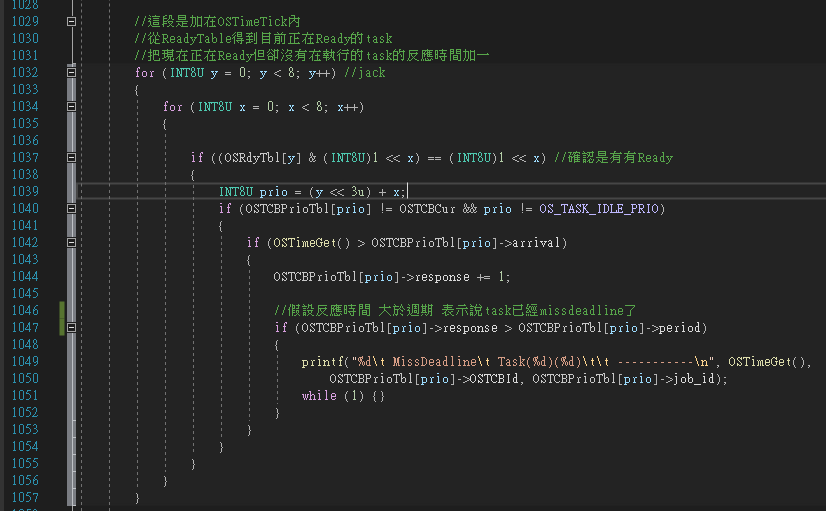


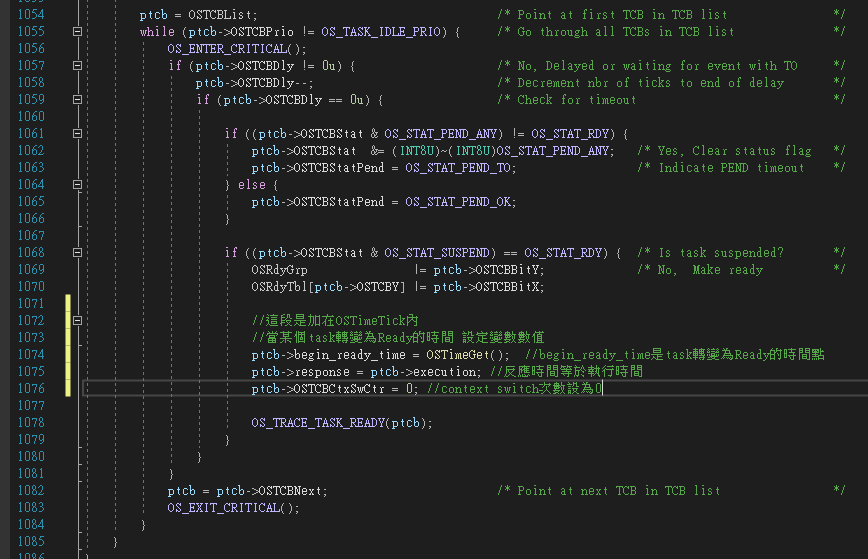
os\_task.c

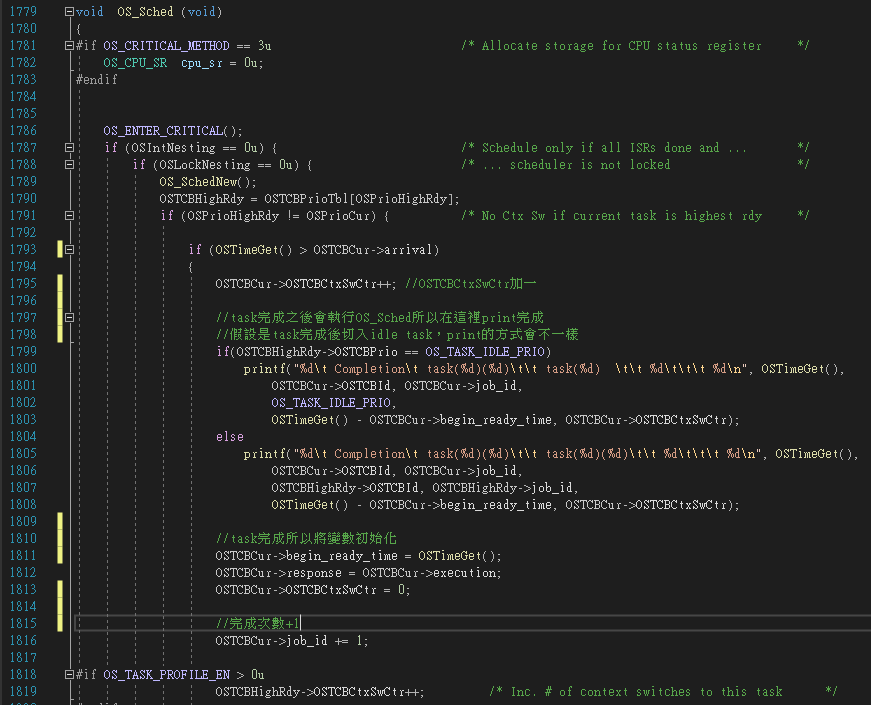


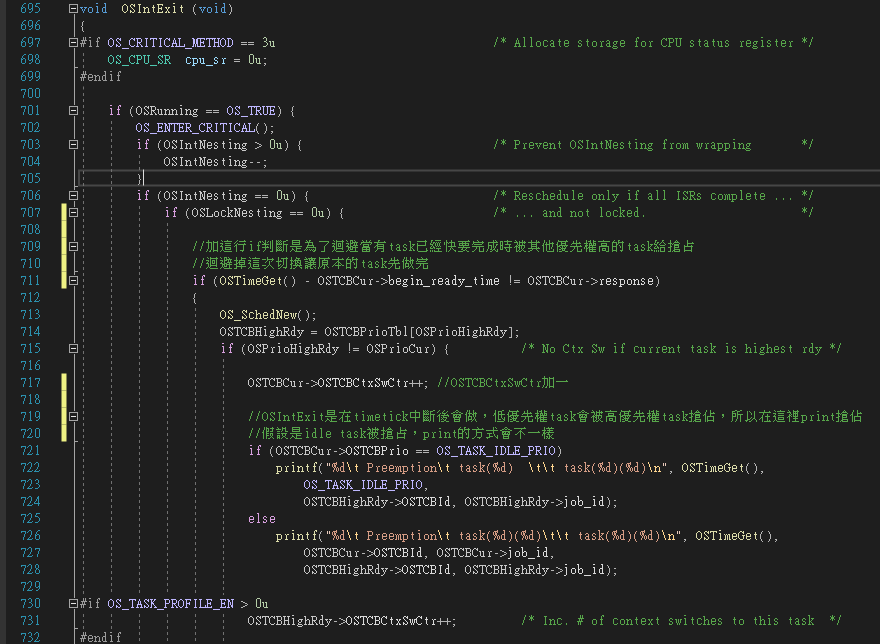
os\_core.c







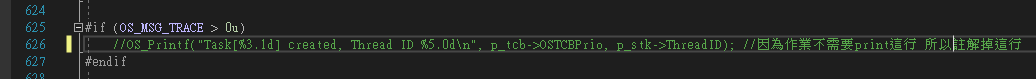




os\_time.c



os\_cpu\_c.c



os\_cpu\_c.c、main.c、os\_time.c、ucos\_ii.h、os\_task.c、os\_core.c