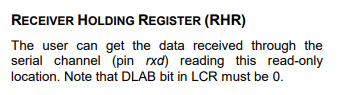
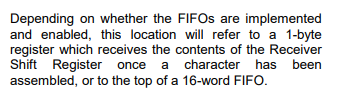
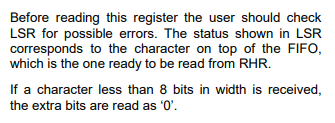
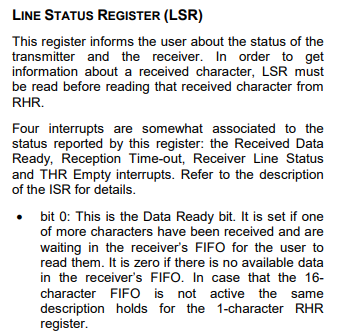


為了能夠讓作業系統從UART收到我們輸入的字，需要先知道要控制哪些UART控制暫存器。於是我開始查閱規格書。











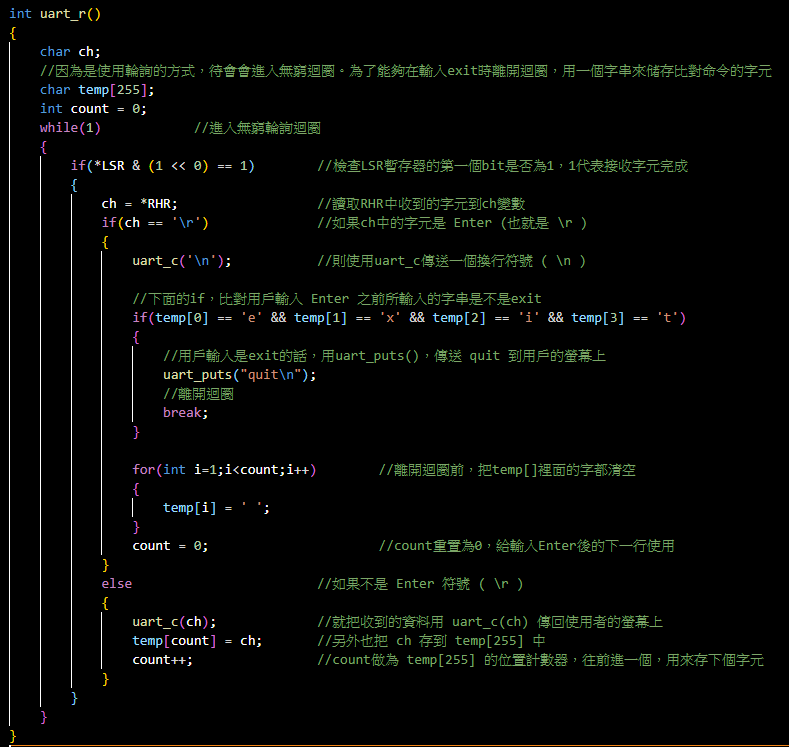
從上面這些敘述中瞭解到，如果要存取RHR中收到的訊號，要先檢查LSR的第0個bit是否為1

於是我開始寫用來收訊息的函數

首先來定義RHR的地址替代巨集。RHR和THR在規格上共用同一個地址

#define RHR  ((volatile unsigned char \*)(0x10000000L + 0))

接著造一個無窮迴圈來不斷檢查是否收到資料了，並判斷用戶輸入的是那些字



接著將新寫的 uart\_r( ) 加到OS中

extern void uart\_r(void);

在uart初始化後使用它

void main(void)

{

    uart\_init();                        //執行uart\_init函數初始化UART

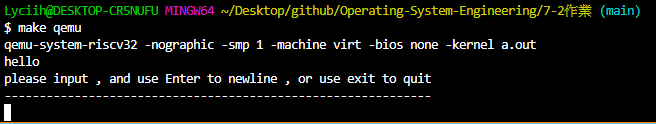
    uart\_puts("hello\n");               //使用uart\_puts函數傳輸 hello 字串

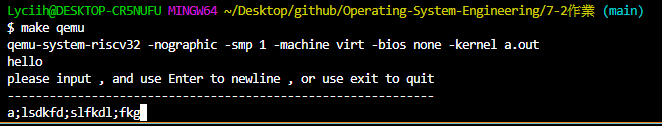
    uart\_puts("please input , and use Enter to newline , or use exit to quit\n");

    uart\_puts("-------------------------------------------------------------\n");

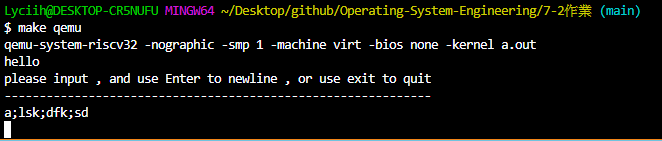
    uart\_r();

編譯執行看看，游標停在最下方等待我們輸入

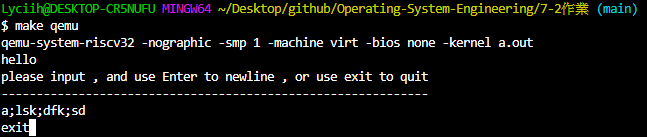


隨便亂打字，都會出現在螢幕上代表UART成功收到了字元並回傳給我們

按下Enter看看，成功換行



輸入exit指令看看



按下Enter，成功跳出迴圈，進入到OS的盡頭

