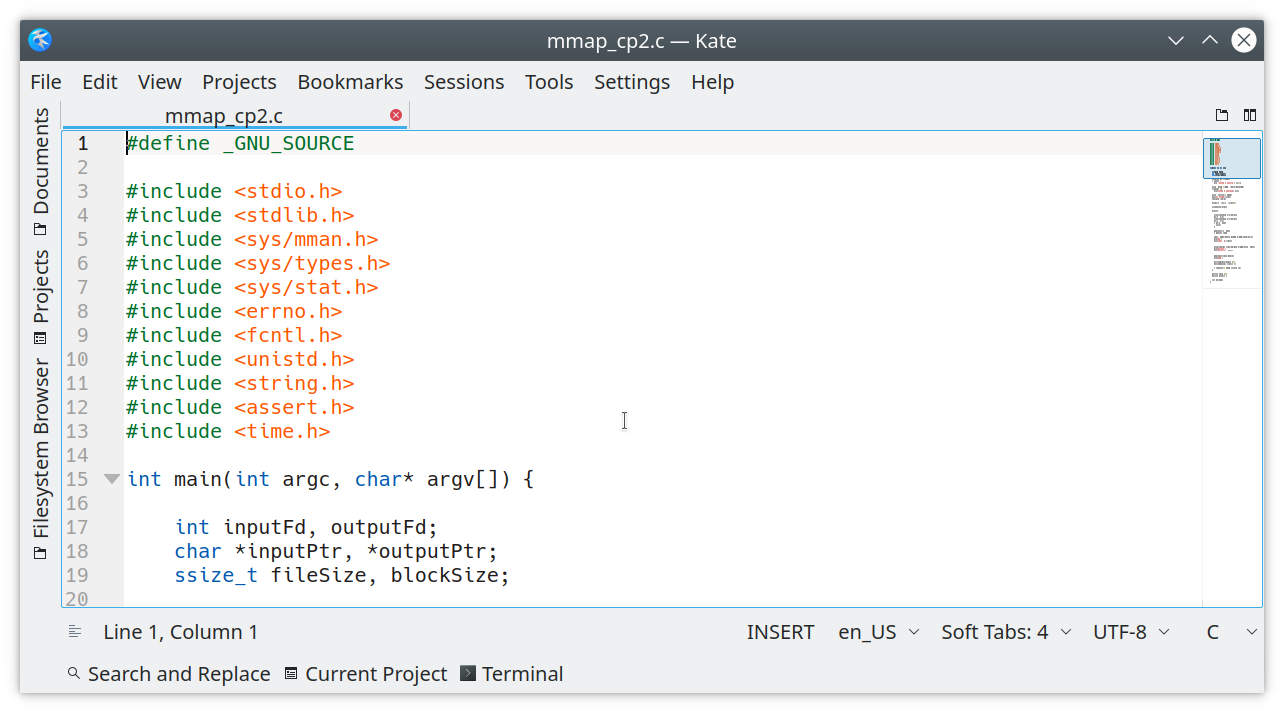
System Programming Homework #2

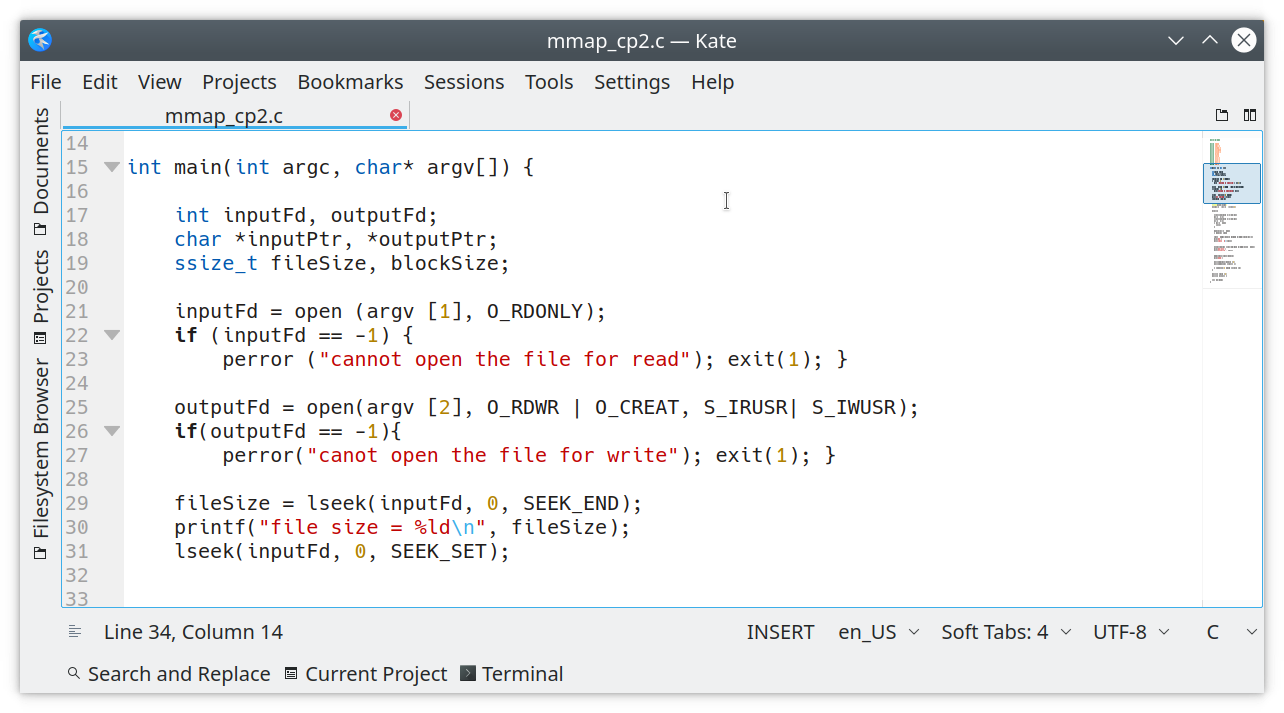
王興彥 408410113

目標：修改mmap\_cp.c為mmap\_cp2.c使得程式在複製檔案時能夠跳過file hole，只複製檔案中真正的內容。

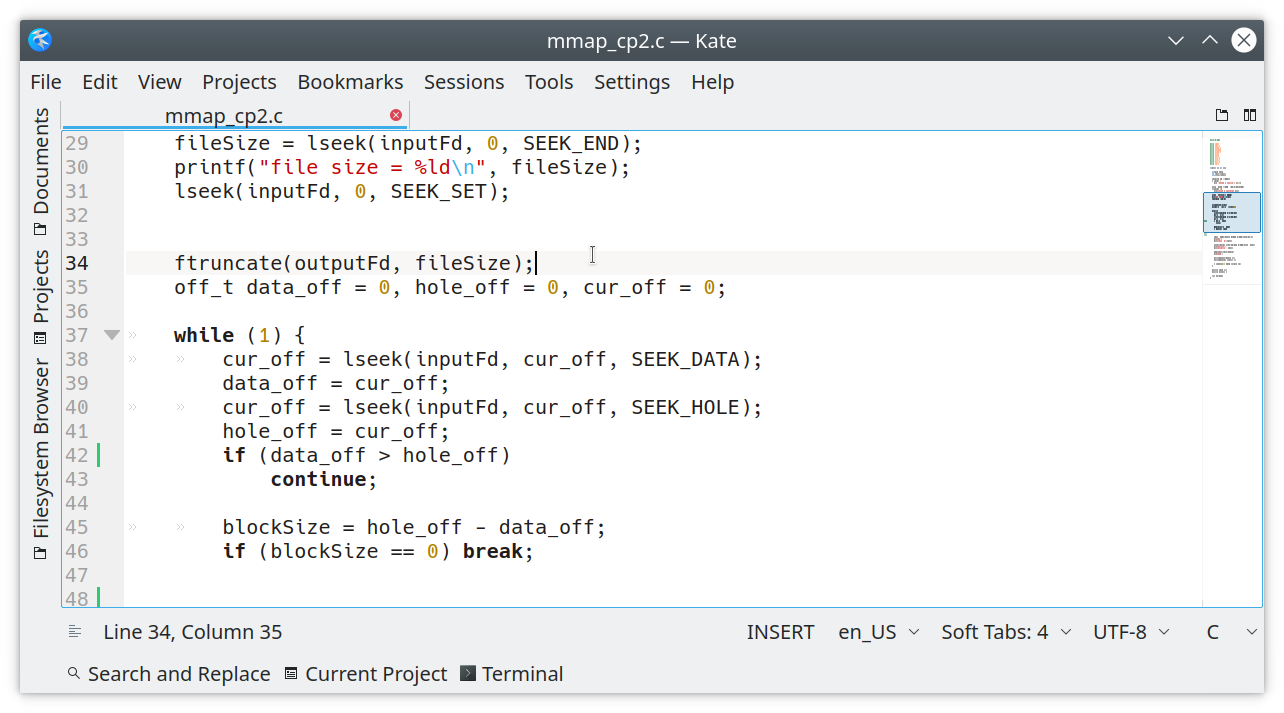
1. 為使用lseek function，必須在程式開頭#define \_GNU\_SOURCE(Line:1)，並include程式所需header(Line:3~13)。



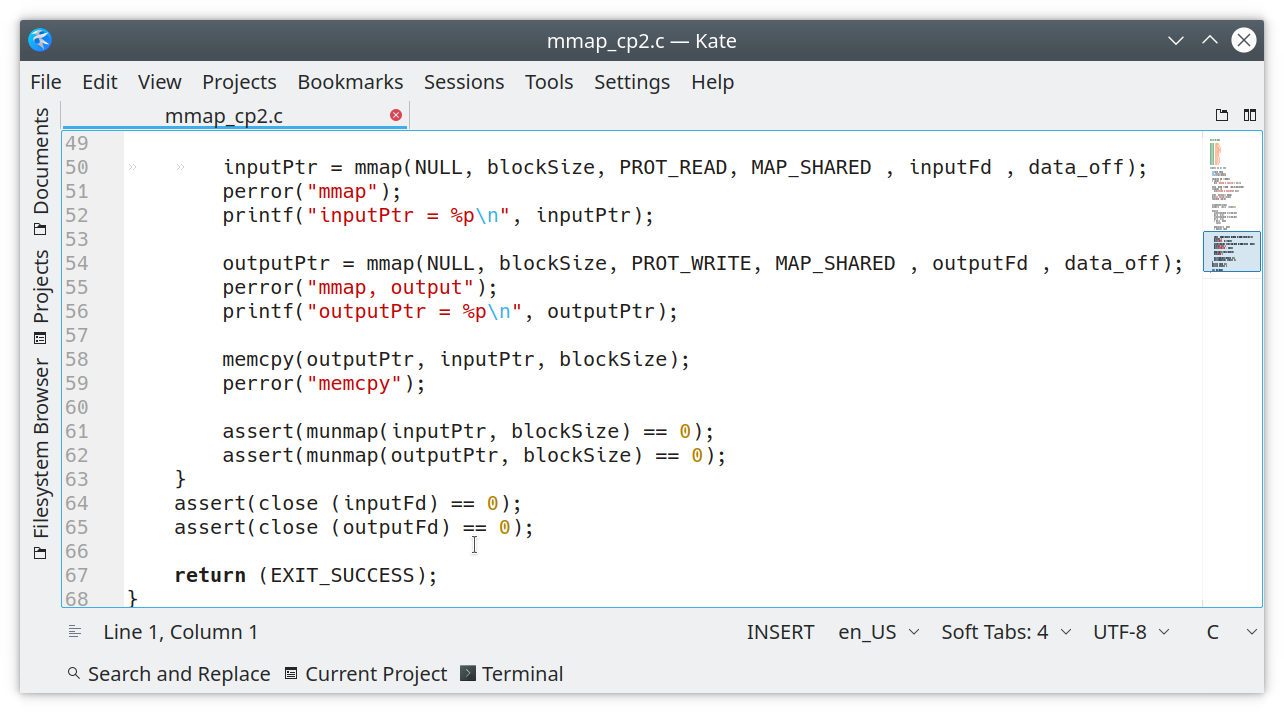
1. 宣告所需變數和指標(Line:17~19)，並從command line讀取被複製檔案及目標檔案名稱(Line:21~27)，用lseek從檔案的頭(offset:0)讀取到檔案的結尾，判斷文件大小，再使用lseek將讀檔位置設回到文件頭(Line:29~31)。



1. 使用ftruncate一次將輸出檔案的大小設置好，避免後續複製時覆蓋到不屬於程式的記憶體造成程式出錯(Line:34)，宣告offset變數，並開始搜尋data和hole的offset，同時使用技巧以確保hole offset > data offset，相減offset取得真正內容的blocksize，若blocksize = 0(data offset = hole offset，到文件結尾)break跳出迴圈(Line:35~46)。

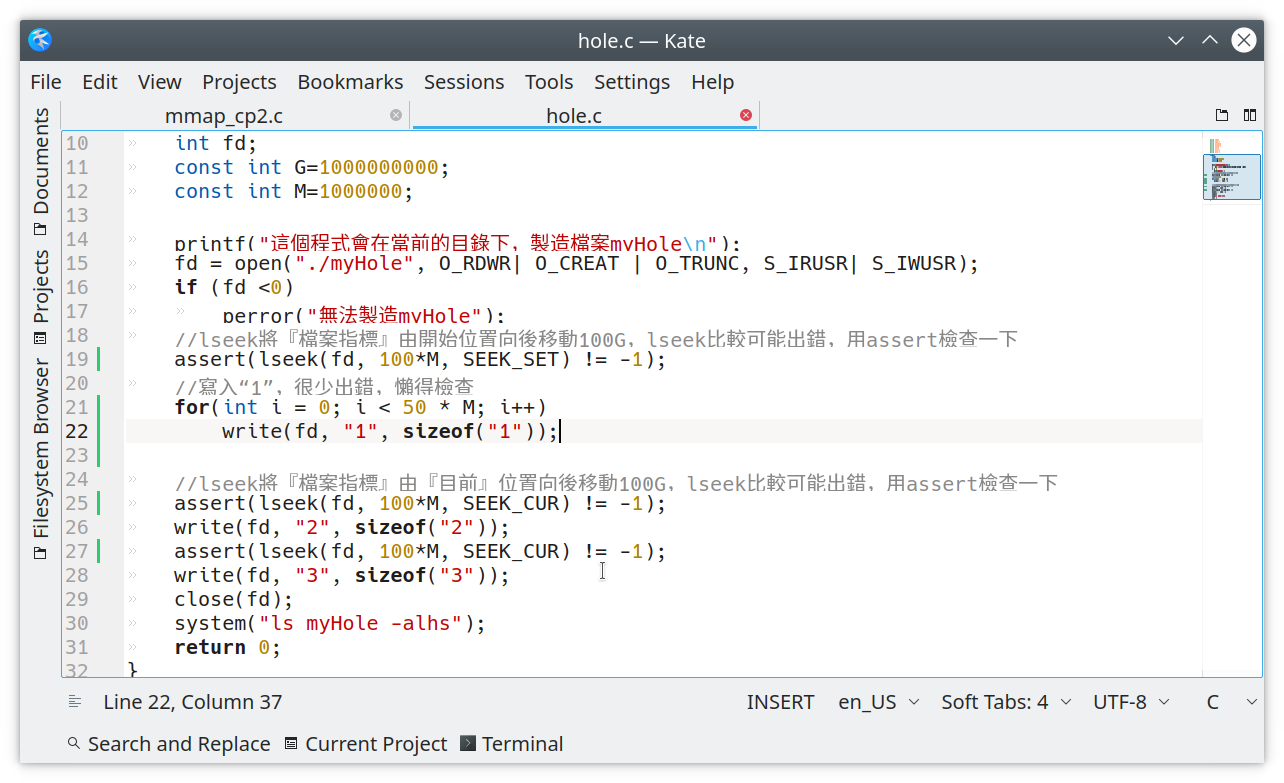


1. 使用mmap將指定的映射記憶體區塊assign給pointer(從data offset開始算起blocksize大小的區塊)，並使用memcpy複製記憶體內容(Line:50~59)，使用assert判斷munmap和關閉檔案時是否有出錯(Line:61~65)。

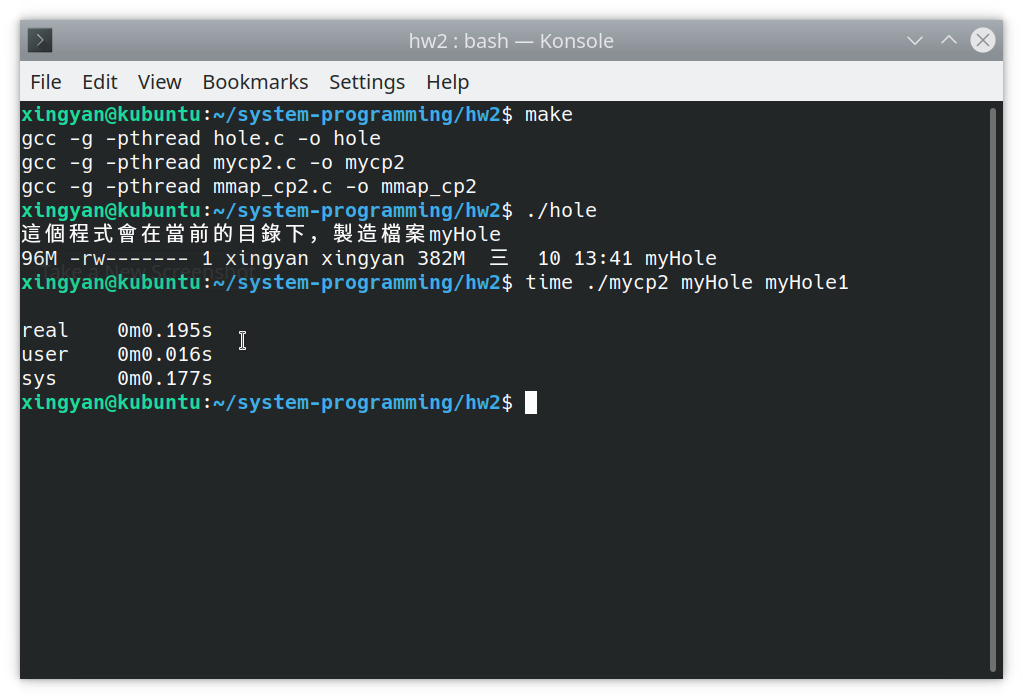


實驗：使用time來比較mmap\_cp2和mycp2兩程式的執行時間

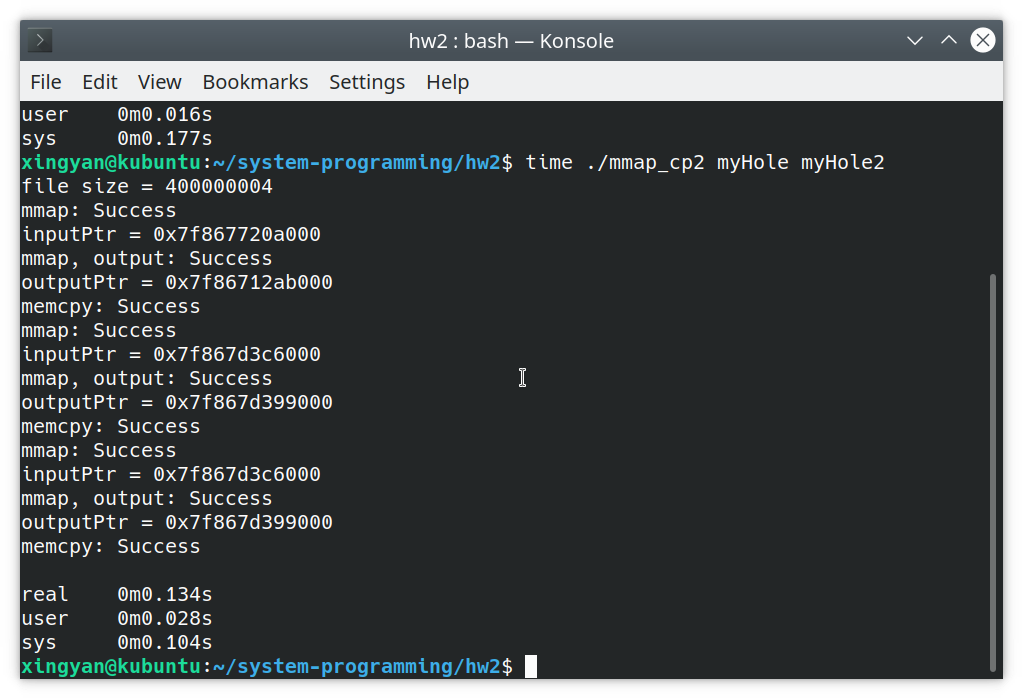
1. 為了方便觀察出差異，將實際data增加(Line:21~22)。

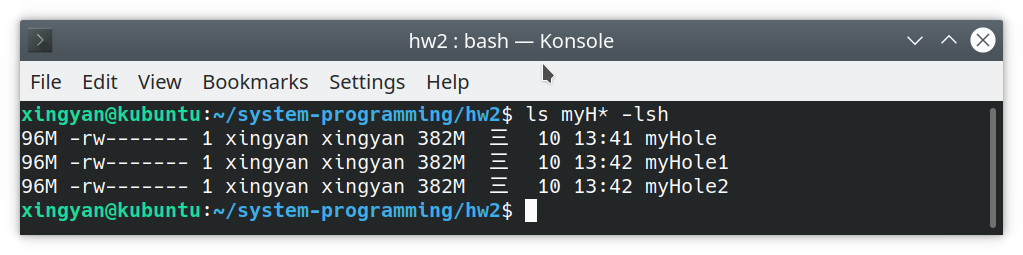


1. make後執行hole產生原始檔案，利用mycp2複製myHole至myHole1，real time為0.195s。



1. 接著使用mmap\_cp2複製myHole至myHole2，real time為0.134s。



1. 使用ls確認檔案複製成功且只複製真正的檔案內容。

結論：直接使用mmap和memcpy對記憶體進行複製，比將資料讀入再寫入的效率更高，此現象在越大的檔案越明顯。