# Assignment12

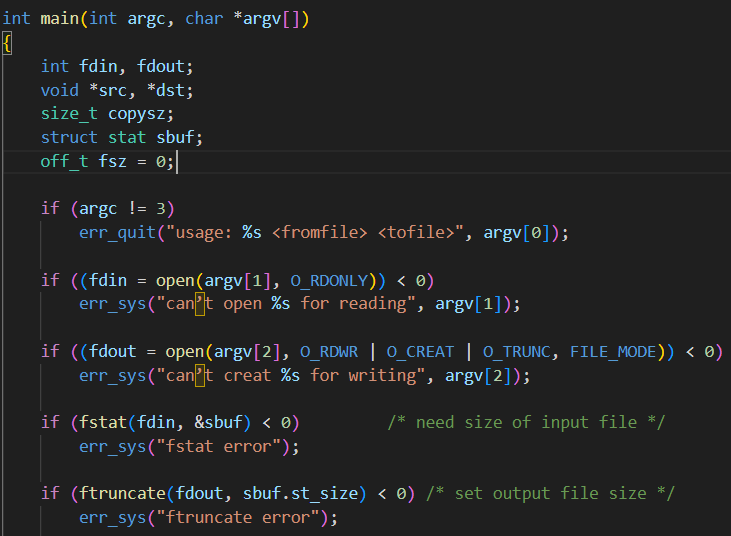
**Team18**

112062519廖思愷

112062636游竣量

111065547游述宇

* **Explanations of implemented code**

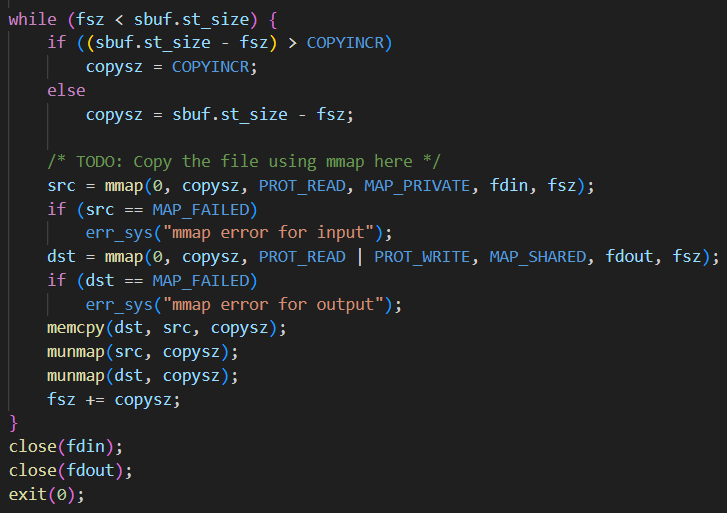
****

**變數解釋**

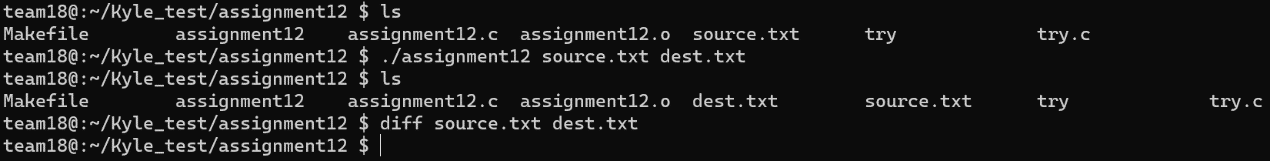
* **int fdin, fdout**：檔案描述符。**fdin** 用於來源檔案，**fdout** 用於目標檔案。
* **void \*src, \*dst**：這兩個指標變數將在後面用於指向映射到記憶體中的檔案區域。**src** 用於來源檔案，**dst** 用於目標檔案。
* **size\_t copysz**：這是一個表示大小的變數，用於確定每次映射和複製的數據量。
* **struct stat sbuf**：這是一個結構體，用於存儲檔案的狀態資訊，如檔案大小等。
* **off\_t fsz = 0**：**off\_t** 是表示檔案偏移量的數據類型。**fsz** 用於記錄已經複製的數據量，初始化為 0。

**檔案前處理**

* **if (argc != 3)**：檢查命令行參數的數量是否正確。
* **open(argv[1], O\_RDONLY)**：使用 **open** 系統呼叫打開來源檔案為唯讀模式。**argv[1]** 是命令行中的第二個參數，代表來源檔案路徑。
* **open(argv[2], O\_RDWR | O\_CREAT | O\_TRUNC, FILE\_MODE)**：同樣使用 **open** 打開目標檔案，但是模式為可讀寫，如果不存在則創建，如果已存在則截斷。
* **fstat(fdin, &sbuf)**：使用 **fstat** 獲取來源檔案的大小等資訊。
* **ftruncate(fdout, sbuf.st\_size)**：將目標檔案的大小設定為與來源檔案相同。



* **循環條件**：當 **fsz**（已複製的大小）小於來源檔案的總大小 (**sbuf.st\_size**) 時，繼續執行循環。
* **計算複製大小**：**copysz** 變數用於確定這一次從來源檔案中要映射和複製多少數據。如果剩餘的數據量大於 **COPYINCR**（定義的最大複製量），則使用 **COPYINCR** 作為這次複製的大小；否則，使用剩餘的數據量。
* **映射來源檔案**：使用 **mmap** 將來源檔案的一部分映射到記憶體中。這裡的 **PROT\_READ** 表示映射區域可被讀取，**MAP\_PRIVATE** 表示對映射區域的修改不會影響原始檔案。
* **映射目標檔案**：同樣使用 **mmap**，但此處的標誌是 **PROT\_READ | PROT\_WRITE**（可讀寫），並且使用 **MAP\_SHARED**，表示對映射區域的修改將會反映到檔案本身。
* **數據複製**：使用 **memcpy** 將數據從來源 (**src**) 複製到目標 (**dst**)，且對**dst** memory區塊的修改會反映到檔案本身。
* **釋放映射**：用 **munmap** 釋放之前映射的兩個記憶體區域。
* **更新已複製數據大小**：**fsz** 被更新，加上剛才複製的大小，為下一次循環做準備。
* **關閉檔案**：使用 **close** 函數關閉先前打開的來源和目標檔案的檔案描述符。
* **退出程式**：通過 **exit(0)** 終止程序。這裡的 **0** 表示程序正常結束。
* **screenshot of result**

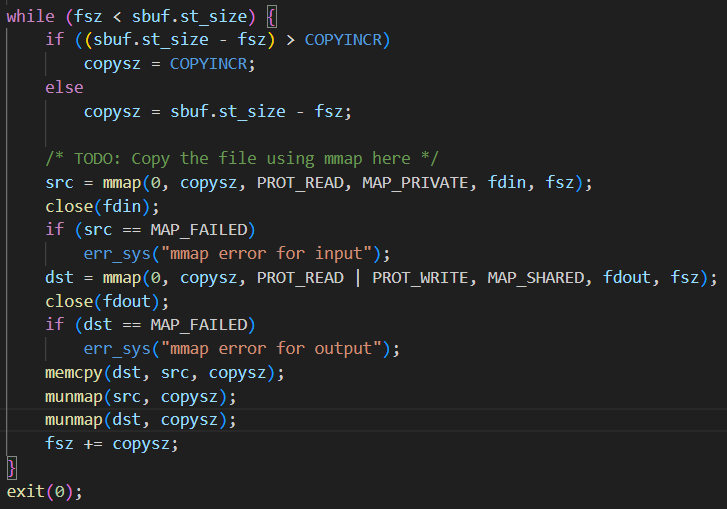
****

由diff的結果可知，source.txt與dest.txt內容一樣，複製成功。

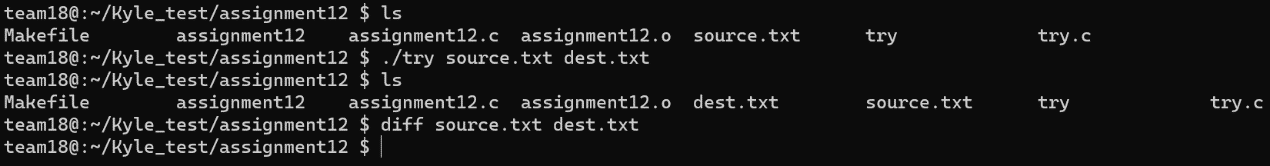
* **Will closing the file descriptor invalidate the memory-mapped I/O?**

不會，解釋如下：

在呼叫 mmap 後關閉 source file 及 dest file (修改後的程式碼片段)



結果



仍然可以讀取source file並寫入dest file。

所以，關閉file descriptor後memory-mapped I/O仍然有效，這表示仍然可以對mapping的記憶體區域進行存取，寫入mapping的記憶體區域也會反映到設有MAP\_SHARED flag的檔案。