# SP\_Hw4

### 410410043 林秉燁

1. 不同buffersize的執行速度 自己設計了一個大小大約為137MB的檔案來進行複製,藉此延長執行時間

```
bine0619@bine0619-virtual-machine:~/Desktop/SP/hw4$ time ./fileperf input.txt output.txt 0
real 1m30.771s
user 0m5.851s
sys 1m24.743s
```

```
bine0619@bine0619-virtual-machine:~/Desktop/SP/hw4$ time ./fileperf input.txt output.txt -1
real    0m4.733s
user    0m1.323s
sys    0m3.400s
```

```
bine0619@bine0619-virtual-machine:~/Desktop/SP/hw4$ time ./fileperf input.txt output.txt 4
real    0m1.177s
user    0m0.200s
sys    0m0.972s
```

```
bine0619@bine0619-virtual-machine:~/Desktop/SP/hw4$ time ./fileperf input.txt output.txt 16
real    0m1.222s
user    0m0.372s
sys    0m0.843s
```

```
bine0619@bine0619-virtual-machine:~/Desktop/SP/hw4$ time ./fileperf input.txt output.txt 64

real 0m1.154s
user 0m0.514s
sys 0m0.634s
```

BufferSize	real (s)	user (s)	sys (s)
0 (unbuffered)	90.771	5.851	84.743
-1 (linebuffered)	4.733	1.323	3.4

BufferSize	real (s)	user (s)	sys (s)
4KB	1.177	0.2	0.972
16KB	1.122	0.372	0.843
64KB	1.154	0.514	0.634
1MB	1.233	0.380	0.845
8MB	1.165	0.196	0.949

#### 2. Itrace 觀察呼叫函式庫的情況

```
bine0619@bine0619-virtual-machine:-/Desktop/SP/hw4$ ltrace -T -a 80 ./fileperf input2.txt output.txt 1000

__monstartup(0x5578b6926200, 0x5578b69267b1, 0x7ffe880e0ea0, 0x5578b6926710) = 0 <0.000579>
__cxa_atexit(0x7f54db681b10, 0, 0x5578b6929090, 0) = 0 <0.000212>
__cyg_profile_func_enter(0x5578b6926349, 0x7f54db584083, 0x7ffe880e0ea0, 0) = 0x7f54db584083 <0.000187>
fopen("input2.txt", "r") = 0x5578b7404d20 <0.000363>
atoi(0x7ffe880e1319, 44, 0, 8) = 1000 <0.000191>
setvbuf(0x5578b7404d20, 0, 0, 1024000) = 0 <0.000246>
getc(0x5578b7404d20, 0x5578b74060f0, 0, 0) = 97 <0.000313>
fputs("abcdefghijklmnopqrstuvwxyz ", 0x5578b7404f00) = 1 <0.000444>
memset(0x7ffe880e0d10, '\0', 82) = 0x7ffe880e0d10 <0.000177>
fputc('\n', 0x5578b7404d20) = 0 <0.000275>
__cyg_profile_func_exit(0x5578b6926349, 0x7f54db584083, 0x5578b7404d20, 2) = 0x7f54db584083 <0.000194>
+++ exited (status 0) +++
```

### 3. strace 觀察呼叫作業系統的情況

```
read(), "stunwnyy abcdefghtjklnnopgrstuw", 4898) = 4896
write(4, "bodefghtjklnnopgrstuwszyzhnabcde"..., 4898) = 4896
read(3, "klnnopgrstuwszyzabcdefghtjklnno"... 4898) = 4896
write(4, "hobdefghtjklnnopgrstuwszyzabcd"., 4898) = 4896
read(3, "cdefghtjklnnopgrstuwszyzabcde", 4898) = 4898
write(4, "sabcefghtjklnnopgrstuwszyz"... 4898) = 4898
read(3, "waxyznabcdefghtjklnnopgrstuwszyz"... 4898) = 4898
write(4, "sayzhabcdefghtjklnnopgrstuwszyz"... 4898) = 4898
write(4, "wayzhabcdefghtjklnnopgrstuwszyz"... 4898) = 4898
write(4, "wayzhabcdefghtjklnnopgrstuwszyz"... 4898) = 4898
write(4, "wayzabcdefghtjklnnopgrstuwszyz"... 4898) = 4898
write(4, "wayzabcdefghtjklnnopgrstuwszyzhab"... 4898) = 4898
write(4, "wayzabcdefghtjklnnopgrstuwszyzhab"... 4898) = 4898
write(4, "wayzabcdefghtjklnnopgrstuwszyzhab"... 4898) = 4898
write(4, "tunwszyzabcdefghtjklnnopgrstuwszynab"... 4898) = 4898
write(4, "tunwszyzabcdefghtjklnnopgrstuwszynab"... 4898) = 4898
write(4, "stunwszyzabcdefghtjklnnopgrstuwszynab"... 4898) = 4898
write(4, "stunwszyzabcdefghtjklnnopgrstuwszynabcdefghtjklnnopgrstuwszynabcdefghtjklnnopgrstuwszynabcdefghtjklnnopgrstuwszynabcdefghtjklnnopgrstuwszynabcdefghtjklnnopgrstuwszynabcdefghtjklnnopgrstuwszynabcdefghtjklnnopgrstuwszynabcdefghtjklnnopgrstuwszynabcdefghtjklnnopgrstuwszynabcdefghtjklnnopgrstuwszynabcdefghtjklnnopgrstuwszynabcdefghtjklnnopgrstuwszynabcdefghtjklnnopgrstuwszynabcdefghtjklnnopgrstuwszynabcdefghtjklnnopgrstuwszynabcdefghtjklnnopgrstuwszynabcdefghtjklnnopgrstuwszynabcdefghtjklnnopgrstuwszynabcdefghtjklnnopgrstuwszynabcdefghtjklnnopgrstuwszynabcdefghtjklnnopgrstuwszynabcdefghtjklnnopgrstuwszynabcdefghtjklnnopgrstuwszynabcdefghtjklnnopgrstuwszynabcdefghtjklnnopgrstuwszynabcdefghtjklnnopgrstuwszynabcdefghtjklnnopgrstuwszynabcdefghtjklnnopgrstuwszynabcdefghtjklnnopgrstuwszynabcdefghtjklnnopgrstuwszynabcdefghtjklnnopgrstuwszynabcdefghtjklnnopgrstuwszynabcdefghtjklnnopgrstuwszynabcdefghtjklnnopgrstuwszynabcdefghtjklnnopgrstuwszynabcdefghtjklnnopgrstuwszynabcdefghtjklnnopgrstuwszynabcdefghtjklnnopgrstuwszynabc
```

SP\_Hw4 2

其中有非常多的write 跟 read 所以就省略只擷取快結束的部份

### 4. 分析效能

由兩張圖可以明顯比較出來 system call的次數較多,且當我們用unbuffered測試時,會發現read跟write會更頻繁,因此時間成本就會增加,而當有buffer時,可以有效的減少寫入的數量,進而減少system call的數量,讓時間成本降低。

## 處理中文問題

```
嘎嘎嘎嘎嘎嘎嘎嘎嘎嘎嘎嘎嘎嘎嘎嘎
我卻沒有 嘎嘎嘎嘎嘎嘎嘎嘎嘎嘎嘎嘎嘎嘎嘎嘎嘎 嘎嘎嘎嘎嘎嘎嘎嘎嘎嘎嘎嘎
汪汪汪汪汪汪汪 汪汪汪汪汪汪汪 汪汪汪汪汪汪汪汪汪汪汪 汪汪汪汪汪汪汪
诺诺诺诺诺诺诺诺诺诺
```

在我的另一支程式fileperf\_wchar.c裡,有利用到<wchar.h> 以及 <locale.h> 來處理中文字,先利用setlocale來設定文字編碼,處理資料時用wint\_t, wchar\_t, WOEF 來取代原本的int, char, EOF,讓系統可以正常處理中文的寬字元,最後實現了原本要求的效果。

最後上面的數據都是用複製英文字的程式跑出來的,因為發現可以複製中文的那個程式效率差距很大,會跑很久,所以只讓他進行比較短的中文複製,不過測試過仍然可以正常做英文轉換。

SP\_Hw4