

許家文 408530049

1.

原本我是用hw2\_example 中的sourceFile.txt(filesize:4K)當作input的file結果如下，發現由於檔案大小太小並不能從中透露出buffer size大的好處。

因此我也用input.txt(14MB)當作測試，可以發現到結果如同預期的buffer size愈高time越短，但到達一個上限後還是要花費每次I/O的時間。

(a.)

```
ulohg@ulohg-VirtualBox:~/system-programming/ch05$ time ./fileperf sourceFile.txt destFile.txt 0
unbuffered

real    0m0.004s
user    0m0.000s
sys     0m0.004s
ulohg@ulohg-VirtualBox:~/system-programming/ch05$ time ./fileperf sourceFile.txt destFile.txt -1
linebuffered

real    0m0.001s
user    0m0.001s
sys     0m0.000s
ulohg@ulohg-VirtualBox:~/system-programming/ch05$ time ./fileperf sourceFile.txt destFile.txt 4096
fullybuffered

real    0m0.001s
user    0m0.001s
sys     0m0.000s
ulohg@ulohg-VirtualBox:~/system-programming/ch05$ time ./fileperf sourceFile.txt destFile.txt 16384
fullybuffered

real    0m0.001s
user    0m0.001s
sys     0m0.000s
ulohg@ulohg-VirtualBox:~/system-programming/ch05$ time ./fileperf sourceFile.txt destFile.txt 65536
fullybuffered

real    0m0.001s
user    0m0.000s
sys     0m0.001s
ulohg@ulohg-VirtualBox:~/system-programming/ch05$ time ./fileperf sourceFile.txt destFile.txt 1048576
fullybuffered

real    0m0.001s
user    0m0.000s
sys     0m0.001s
ulohg@ulohg-VirtualBox:~/system-programming/ch05$ time ./fileperf sourceFile.txt destFile.txt 8388608
fullybuffered

real    0m0.001s
user    0m0.001s
sys     0m0.000s
```

(b)

```
ulohg@ulohg-VirtualBox:~/system-programming/ch05$ time ./fileperf input.txt destFile.txt 0
unbuffered

real    0m17.052s
user    0m6.023s
sys     0m11.004s
ulohg@ulohg-VirtualBox:~/system-programming/ch05$ time ./fileperf input.txt destFile.txt -1
linebuffered

real    0m1.566s
user    0m0.573s
sys     0m0.984s
ulohg@ulohg-VirtualBox:~/system-programming/ch05$ time ./fileperf input.txt destFile.txt 4096
fullybuffered

real    0m0.381s
user    0m0.328s
sys     0m0.040s
ulohg@ulohg-VirtualBox:~/system-programming/ch05$ time ./fileperf input.txt destFile.txt 16384
fullybuffered

real    0m0.395s
user    0m0.343s
sys     0m0.033s
ulohg@ulohg-VirtualBox:~/system-programming/ch05$ time ./fileperf input.txt destFile.txt 65536
fullybuffered

real    0m0.387s
user    0m0.353s
sys     0m0.025s
ulohg@ulohg-VirtualBox:~/system-programming/ch05$ time ./fileperf input.txt destFile.txt 1048576
fullybuffered

real    0m0.402s
user    0m0.329s
sys     0m0.040s
ulohg@ulohg-VirtualBox:~/system-programming/ch05$ time ./fileperf input.txt destFile.txt 8388608
fullybuffered

real    0m0.378s
user    0m0.341s
sys     0m0.028s
```

```

u1ohgg@u1ohg-VirtualBox:~/system-programming/ch05$ ltrace -c ./fileperf sourceFile.txt destFile.txt 1000
fullybuffered
% time      seconds      usecs/call      calls      function
-----
23.20      0.000794          794          1      getc
20.48      0.000701          701          1      fputc
12.71      0.000435          435          1      __cyg_profile_func_exit
9.82       0.000336          336          1      fputs
9.64       0.000330          330          1      __monstartup
6.02       0.000206          206          1      puts
4.62       0.000158          158          1      fopen
4.32       0.000148          148          1      setvbuf
3.89       0.000133          133          1      atoi
2.75       0.000094           94          1      __cyg_profile_func_enter
2.57       0.000088           88          1      __cxa_atexit
-----
100.00      0.003423          11      total

u1ohgg@u1ohg-VirtualBox:~/system-programming/ch05$ ltrace ./fileperf sourceFile.txt destFile.txt 1000
__monstartup(0x557ab4e8d1e0, 0x557ab4e8d7e1, 0x7fff276c6b90, 0x557ab4e8d740)      = 0
__cxa_atexit(0x7fdb05e77b10, 0, 0x557ab4e90088, 0)      = 0
__cyg_profile_func_enter(0x557ab4e8d329, 0x7fdb05d7a083, 0x7fff276c6b90, 0)      = 0x7fdb05d7a083
fopen("sourceFile.txt", "r")      = 0x557ab532ed70
atoi(0x7fff276c7335, 44, 0, 11)      = 1000
puts("fullybuffered"fullybuffered)      = 14
setvbuf(0x557ab532ed70, 0, 0, 1000)      = 0
getc(0x557ab532ed70, 0x557ab5330550, 0, 0)      = 73
fputs("Intel ", 0x557ab532ef50)      = 1
fputc('\n', 0x557ab532ef50)      = 10
__cyg_profile_func_exit(0x557ab4e8d329, 0x7fdb05d7a083, 0xfbad2498, 0xfbad2498)      = 0x7fdb05d7a083
+++ exited (status 0) +++

```

[illegible]

```

u1ohg@u1ohg-VirtualBox:~/system-programming/ch05$ strace -c ./fileperf sourceFile.txt destFile.txt 1000
fullybuffered
% time    seconds    usecs/call    calls    errors syscalls
-----
 0.00    0.000000         0         3        read
 0.00    0.000000         0         3        write
 0.00    0.000000         0         3        close
 0.00    0.000000         0         5        fstat
 0.00    0.000000         0         7        mmap
 0.00    0.000000         0         3        mprotect
 0.00    0.000000         0         1        munmap
 0.00    0.000000         0         3        brk
 0.00    0.000000         0         2        rt_sigaction
 0.00    0.000000         0         6        pread64
 0.00    0.000000         0         1        writev
 0.00    0.000000         0         1        1 access
 0.00    0.000000         0         2        setitimer
 0.00    0.000000         0         1        execve
 0.00    0.000000         0         2        1 arch_prctl
 0.00    0.000000         0         5        openat
-----
100.00    0.000000         0         48        2 total

```

strace是出run time所用到的system call, ltrace則是列出run time所用到的library function。

根據2.3.程式需要使用到的system call 遠比call c libray還要多, 但system call 所花的時間比c library所花的時間還要少許多。