HW4 資工二 曹咏萱 409410082

(1)請問buffersize分別是：0、-1、4KB、16KB、64KB、1MB、8MB的執行速度分別為何？（使用time指令）

- sourceFile.txt

  0

```
reki@reki002:~/sp/hw04$ time ./fileperf sourceFile.txt result.txt 0

real    0m0.007s
user    0m0.001s
sys     0m0.006s
```

  -1

```
reki@reki002:~/sp/hw04$ time ./fileperf sourceFile.txt result.txt -1

real    0m0.003s
user    0m0.000s
sys     0m0.003s
```

  4KB

```
reki@reki002:~/sp/hw04$ time ./fileperf sourceFile.txt result.txt 4KB
4096

real    0m0.003s
user    0m0.000s
sys     0m0.003s
```

  16KB

```
reki@reki002:~/sp/hw04$ time ./fileperf sourceFile.txt result.txt 16KB
16384

real    0m0.002s
user    0m0.002s
sys     0m0.001s
```

  64KB

```
reki@reki002:~/sp/hw04$ time ./fileperf sourceFile.txt result.txt 64KB
65536

real    0m0.003s
user    0m0.002s
sys     0m0.001s
```

  1MB

```
reki@reki002:~/sp/hw04$ time ./fileperf sourceFile.txt result.txt 1MB
1048576

real    0m0.004s
user    0m0.000s
sys     0m0.004s
```

  8MB

```
reki@reki002:~/sp/hw04$ time ./fileperf sourceFile.txt result.txt 8MB
8388608

real    0m0.003s
user    0m0.002s
sys     0m0.001s
```

- input.txt

  0

```
reki@reki002:~/sp/hw04$ time ./fileperf input.txt result2.txt 0

real    0m17.032s
user    0m3.592s
sys     0m13.386s
```

-1

```
reki@reki002:~/sp/hw04$ time ./fileperf input.txt result2.txt -1

real    0m1.199s
user    0m0.511s
sys     0m0.687s
```

4KB

```
reki@reki002:~/sp/hw04$ time ./fileperf input.txt result2.txt 4KB
4096

real    0m0.370s
user    0m0.263s
sys     0m0.105s
```

16KB

```
reki@reki002:~/sp/hw04$ time ./fileperf input.txt result2.txt 16KB
16384

real    0m0.261s
user    0m0.232s
sys     0m0.029s
```

64KB

```
reki@reki002:~/sp/hw04$ time ./fileperf input.txt result2.txt 64KB
65536

real    0m0.326s
user    0m0.280s
sys     0m0.046s
```

1MB

```
reki@reki002:~/sp/hw04$ time ./fileperf input.txt result2.txt 1MB
1048576

real    0m0.310s
user    0m0.252s
sys     0m0.057s
```

8MB

```
reki@reki002:~/sp/hw04$ time ./fileperf input.txt result2.txt 8MB
8388608

real    0m0.338s
user    0m0.278s
sys     0m0.060s
```

總結下來unbuffered(0)所需要的時間遠遠多於其他選項，而line buffered(1)又稍微比fully buffered要慢一些。對於fully buffered來說，bufSize大的比bufSize小的效能要好一點，但其實超過16KB之後所測出來的時間就看不出甚麼太大的差距了。

(2)使用ltrace觀察你的應用程式呼叫「函數庫的情況」

試了很多次都沒有成功研究很久才發現要加上-z lazy去編譯

0

```
reki@reki002:~/sp/hw04$ ltrace -c ./fileperf input.txt result2.txt 0
% time     seconds  usecs/call     calls      function
------ ----------- ----------- --------- --------------------
 18.05    0.001332        1332         1 fopen
 17.44    0.001287        1287         1 fgetc
 13.76    0.001015        1015         1 fprintf
 10.10    0.000745         745         1 fputs
  8.65    0.000638         638         1 fputc
  7.69    0.000567         567         1 __ctype_b_loc
  7.08    0.000522         522         1 setvbuf
  6.93    0.000511         511         1 strcmp
  6.90    0.000509         509         1 malloc
  3.42    0.000252         252         1 fclose
------ ----------- ----------- --------- --------------------
100.00    0.007378                    10 total
```

-1

```
reki@reki002:~/sp/hw04$ ltrace -c ./fileperf input.txt result2.txt -1
% time     seconds  usecs/call     calls      function
------ ----------- ----------- --------- --------------------
 27.69    0.001924        1924         1 fopen
 11.77    0.000818         818         1 strcmp
 11.04    0.000767         767         1 fprintf
 10.62    0.000738         738         1 fgetc
  8.81    0.000612         612         1 malloc
  7.90    0.000549         549         1 fputc
  7.40    0.000514         514         1 fputs
  6.28    0.000436         436         1 setvbuf
  5.38    0.000374         374         1 __ctype_b_loc
  3.11    0.000216         216         1 fclose
------ ----------- ----------- --------- --------------------
100.00    0.006948                    10 total
```

4KB

```
reki@reki002:~/sp/hw04$ ltrace -c ./fileperf input.txt result2.txt 4KB
4096
% time     seconds  usecs/call     calls      function
------ ----------- ----------- --------- --------------------
 24.66    0.002313        2313         1 fopen
 11.89    0.001115        1115         1 fgetc
  7.61    0.000714         714         1 fprintf
  7.04    0.000660         660         1 fclose
  6.89    0.000646         646         1 malloc
  6.65    0.000624         624         1 __ctype_b_loc
  6.63    0.000622         622         1 printf
  5.80    0.000544         544         1 setvbuf
  5.24    0.000492         492         1 strtol
  5.06    0.000475         475         1 strlen
  4.57    0.000429         429         1 strcmp
  4.04    0.000379         379         1 fputs
  3.92    0.000368         368         1 fputc
------ ----------- ----------- --------- --------------------
100.00    0.009381                    13 total
```

8MB

```
reki@reki002:~/sp/hw04$ ltrace -c ./fileperf input.txt result2.txt 8MB
8388608
% time     seconds  usecs/call     calls      function
------ ----------- ----------- --------- --------------------
 52.22    0.010697       10697         1 fgetc
  9.45    0.001935        1935         1 fopen
  8.60    0.001761        1761         1 printf
  5.15    0.001054        1054         1 malloc
  4.72    0.000966         966         1 fprintf
  3.20    0.000656         656         1 setvbuf
  3.14    0.000643         643         1 strtol
  2.93    0.000601         601         1 strcmp
  2.71    0.000556         556         1 fputs
  2.19    0.000448         448         1 fclose
  2.10    0.000431         431         1 __ctype_b_loc
  2.10    0.000431         431         1 strlen
  1.49    0.000306         306         1 fputc
------ ----------- ----------- --------- --------------------
100.00    0.020485                    13 total
```

(3)使用strace觀察你的應用程式呼叫「作業系統的情況」

0

```
reki@reki002:~/sp/hw04$ strace -c ./fileperf input.txt result2.txt 0
% time     seconds  usecs/call     calls    errors syscall
------ ----------- ----------- --------- --------- ----------------
 68.46   96.184178           7  13699328           read
 31.54   44.316835           9   4641643           write
  0.00    0.001437         359         4           close
  0.00    0.001148         287         4           openat
  0.00    0.000354          50         7           mmap
  0.00    0.000308          51         6           pread64
  0.00    0.000279         279         1           execve
  0.00    0.000192          48         4           mprotect
  0.00    0.000109          36         3           brk
  0.00    0.000084          42         2         1 arch_prctl
  0.00    0.000081          40         2           fstat
  0.00    0.000081          81         1         1 access
  0.00    0.000035          35         1           munmap
------ ----------- ----------- --------- --------- ----------------
100.00  140.505121              18341006         2 total
```

-1

```
reki@reki002:~/sp/hw04$ strace -c ./fileperf input.txt result2.txt -1
% time     seconds  usecs/call     calls    errors syscall
------ ----------- ----------- --------- --------- ----------------
 95.40    1.213676           3    331255           write
  3.54    0.045048           3     13381           read
  1.06    0.013491        3372         4           close
  0.00    0.000000           0         2           fstat
  0.00    0.000000           0         7           mmap
  0.00    0.000000           0         4           mprotect
  0.00    0.000000           0         1           munmap
  0.00    0.000000           0         3           brk
  0.00    0.000000           0         6           pread64
  0.00    0.000000           0         1         1 access
  0.00    0.000000           0         1           execve
  0.00    0.000000           0         2         1 arch_prctl
  0.00    0.000000           0         4           openat
------ ----------- ----------- --------- --------- ----------------
100.00    1.272215                344671         2 total
```

4KB

```
reki@reki002:~/sp/hw04$ strace -c ./fileperf input.txt result2.txt 4KB
4096
% time     seconds  usecs/call     calls    errors syscall
------ ----------- ----------- --------- --------- ----------------
 52.84    0.129325          38      3346           write
 36.82    0.090124          26      3347           read
  7.94    0.019429        4857         4           close
  1.62    0.003971         992         4           openat
  0.20    0.000487         487         1           execve
  0.17    0.000415          59         7           mmap
  0.14    0.000349          58         6           pread64
  0.07    0.000173          43         4           mprotect
  0.07    0.000172          43         4           fstat
  0.05    0.000115          38         3           brk
  0.04    0.000103          51         2         1 arch_prctl
  0.02    0.000055          55         1         1 access
  0.02    0.000048          48         1           munmap
------ ----------- ----------- --------- --------- ----------------
100.00    0.244766                6730         2 total
```

8MB

```
reki@reki002:~/sp/hw04$ strace -c ./fileperf input.txt result2.txt 8MB
8388608
% time     seconds  usecs/call     calls    errors syscall
------ ----------- ----------- --------- --------- ----------------
 71.04    0.117292          35      3346           write
 19.53    0.032248        8062         4           close
  5.28    0.008716        2179         4           read
  2.76    0.004549        1137         4           openat
  0.41    0.000680         113         6           pread64
  0.31    0.000507          63         8           mmap
  0.25    0.000407         407         1           execve
  0.13    0.000213          53         4           mprotect
  0.12    0.000196          49         4           fstat
  0.08    0.000127          42         3           brk
  0.04    0.000065          65         1           munmap
  0.04    0.000059          29         2         1 arch_prctl
  0.03    0.000057          57         1         1 access
------ ----------- ----------- --------- --------- ----------------
100.00    0.165116                3388         2 total
```

(4)有辦法根據2和3分析一下「呼叫作業系統核心函數（system call）」和「函數庫呼叫」的「成本」差異嗎？

首先先整理一下對ltrace和strce下-c參數之後後出現的欄位

名稱	代表意義
seconds	呼叫此項目所花的時間(s)
usecs/call	代表每項的平均耗時(us)
calls	此項被呼叫的次數

從(2)(3)可以發現，作業系統的呼叫次數通常都很多，雖然seconds數字很驚人，但平均下來卻只有個位數微秒；而函數庫呼叫則相反，呼叫次數幾乎都只有一次，卻要耗費大量的時間，造成平均較高的情形。
總結來說就是，呼叫system call所需的時間較短，但程式中可能會被大量呼叫；而呼叫函數庫一次所需的時間較長，可通常只會被呼叫一次。
因此由單次來看的話，呼叫system call的成本較低，可如果放到整個程式來看，總體呼叫system call的時間就很可觀。

(5)關於字串處理的部分

作業4的內容描述好像跟hwx_example裡的檔案範例有些不一樣，所以我最後是以範例為準根據自己的理解來處理字串:

- 一行不超過80個字(包含空白和換行)
- 如果輸入檔案中包含換行會照常輸出
- 檔案最後一行有換行