

Hw4

使用 strace 測量程式碼的 printf 呼叫的 system call 的數量

系級：資工二 學號：409410114 姓名：周述君

一、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）

- buffersize 為 0 (unbuffered)

```
sujean@sujean:~/sp/sphw4$ time ./fileperf sourceFile.txt destFile.txt 0
real    0m0.009s
user    0m0.000s
sys     0m0.005s
```

- buffersize 為 -1 (linebuffered)

```
sujean@sujean:~/sp/sphw4$ time ./fileperf sourceFile.txt destFile.txt -1
real    0m0.005s
user    0m0.005s
sys     0m0.000s
```

- buffersize 為 4KB ($1024*4=4096$)

```
sujean@sujean:~/sp/sphw4$ time ./fileperf sourceFile.txt destFile.txt 4096
real    0m0.003s
user    0m0.003s
sys     0m0.000s
```

- buffersize 為 16KB ($1024*16=16384$)

```
sujean@sujean:~/sp/sphw4$ time ./fileperf sourceFile.txt destFile.txt 16384
real    0m0.004s
user    0m0.000s
sys     0m0.003s
```

- buffersize 為 64KB ($1024*64=65536$)

```
sujean@sujean:~/sp/sphw4$ time ./fileperf sourceFile.txt destFile.txt 65536
real    0m0.005s
user    0m0.003s
sys     0m0.003s
```

- buffersize 為 1MB ($1024*1024=1048576$)

```
sujean@sujean:~/sp/sphw4$ time ./fileperf sourceFile.txt destFile.txt 1048576
real    0m0.006s
user    0m0.003s
sys     0m0.003s
```

- buffersize 為 8MB ($1024*1024*8=83388608$)

```
sujean@sujean:~/sp/sphw4$ time ./fileperf sourceFile.txt destFile.txt 83388608
real    0m0.004s
user    0m0.003s
sys     0m0.000s
```

可以看出在 `unbuffered` 的時間較其他慢，此外 `fullybuffered` 的情況下，`buffer` 超過某個大小效能就不再增加。

二、使用 `ltrace` 觀察你的應用程式呼叫「函數庫的情況」

甲、對，`ltrace` 我沒教過

乙、嗯，也沒人教過我啊，我是自己上網學的

丙、會用 `ltrace`，看到別人寫出很厲害的程式，可以偷看一下他呼叫了什麼函數庫。

丁、使用方法跟 `strace` 很像

戊、`call(x86, asm)`

- `buffer` 為 0 時 (`unbuffered`)：731 個 `fputs`、1292 個 `fputc`、660 個 `getc`，共呼叫 2025 次函式庫(輸出)和 660 次函式庫(輸入)。

```
sujean@sujean:~/sp/sphw4$ ltrace -c ./fileperf sourceFile.txt destFile.txt 0
% time      seconds    usecs/call   calls      function
-----
21.65      0.001292      1292         1 fputc
18.97      0.001132      1132         1 __monstartup
12.25      0.000731      731          1 fputs
11.06      0.000660      660          1 getc
 8.35      0.000498      498          1 __fprintf_chk
 7.12      0.000425      425          1 strtol
 6.44      0.000384      384          1 __cyg_profile_func_exit
 4.54      0.000271      271          1 fopen
 4.19      0.000250      250          1 setvbuf
 2.77      0.000165      165          1 __cxa_atexit
 2.66      0.000159      159          1 __cyg_profile_func_enter
-----
100.00     0.005967                      11 total
```

- `buffer` 為 -1 時 (`linebuffered`)：191 個 `fputs`、221 個 `fputc`、118 個 `getc`，共呼叫 412 次函式庫(輸出)和 118 次函式庫(輸入)。

```
sujean@sujean:~/sp/sphw4$ ltrace -c ./fileperf sourceFile.txt destFile.txt -1
% time      seconds    usecs/call   calls      function
-----
24.57      0.000499      499          1 __monstartup
11.13      0.000226      226          1 __cyg_profile_func_exit
10.88      0.000221      221          1 fputc
 9.45      0.000192      192          1 __fprintf_chk
 9.40      0.000191      191          1 fputs
 6.60      0.000134      134          1 fopen
 6.15      0.000125      125          1 __cxa_atexit
 5.86      0.000119      119          1 strtol
 5.81      0.000118      118          1 getc
 5.37      0.000109      109          1 setvbuf
 4.78      0.000097      97          1 __cyg_profile_func_enter
-----
100.00     0.002031                      11 total
```

- buffersize 為 4096 時 (fullybuffered)：402 個 fputs、378 個 fputc、859 個 getc，共呼叫 780 次函式庫(輸出)和 859 次函式庫(輸入)。

```

sujean@sujean:~/sp/sphw4$ ltrace -c ./fileperf sourceFile.txt destFile.txt 4096
% time      seconds      usecs/call      calls      function
-----
23.55      0.001561           1561           1  __monstartup
12.96      0.000859            859           1  getc
11.10      0.000736            736           1  setvbuf
11.04      0.000732            732           1  fopen
 8.39      0.000556            556           1  __cxa_atexit
 6.28      0.000416            416           1  strtol
 6.07      0.000402            402           1  fputs
 5.99      0.000397            397           1  __cyg_profile_func_enter
 5.70      0.000378            378           1  fputc
 4.89      0.000324            324           1  __fprintf_chk
 4.03      0.000267            267           1  __cyg_profile_func_exit
-----
100.00      0.006628                    11 total

```

三、使用 strace 觀察你的應用程式呼叫「作業系統的情況」

甲、syscall(x86, asm, trap, software interrupt,大概是 call(x86, asm)的速度的 1/500)

- buffersize 為 0 時 (unbuffered)：共呼叫 816 次 read、120 次 write。

```

sujean@sujean:~/sp/sphw4$ strace -c ./fileperf sourceFile.txt destFile.txt 0
% time      seconds      usecs/call      calls      errors  syscall
-----
80.45      0.015648           19           816           read
17.52      0.003408           28           120           write
 1.02      0.000198           66           3           close
 0.44      0.000085           17           5           openat
 0.27      0.000053           17           3           rt_sigreturn
 0.12      0.000024           24           1           writev
 0.09      0.000018           9           2           setitimer
 0.09      0.000017           8           2           rt_sigaction
 0.00      0.000000           0           2           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           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
-----
100.00      0.019451                    978           2 total

```

- buffersize 為 -1 時 (linebuffered)：共呼叫 816 次 read、120 次 write。

```

sujean@sujean:~/sp/sphw4$ strace -c ./fileperf sourceFile.txt destFile.txt -1
% time      seconds    usecs/call   calls   errors syscall
-----
67.55      0.014681        17         816         read
15.20      0.003303         27         120         write
4.26       0.000926        185          5         openat
3.34       0.000725        725          1         execve
2.22       0.000482         68          7         mmap
1.20       0.000260         86          3         close
1.17       0.000255         42          6         pread64
1.05       0.000228         76          3         brk
0.92       0.000199         66          3         mprotect
0.80       0.000173         86          2         fstat
0.69       0.000149         74          2         1 arch_prctl
0.42       0.000091         45          2         setitimer
0.40       0.000087         87          1         munmap
0.31       0.000067         67          1         1 access
0.23       0.000051         25          2         rt_sigaction
0.17       0.000036         18          2         rt_sigreturn
0.09       0.000020         20          1         writev
-----
100.00     0.021733                977         2 total

```

- buffersize 為 4096 時 (fullybuffered)：共呼叫 13 次 read、82 次 write。

```

sujean@sujean:~/sp/sphw4$ strace -c ./fileperf sourceFile.txt destFile.txt 4096
% time      seconds    usecs/call   calls   errors syscall
-----
34.12      0.000363        121          3         close
23.59      0.000251         50          5         openat
15.41      0.000164         82          2         write
5.36       0.000057         28          2         setitimer
5.26       0.000056         28          2         rt_sigaction
3.85       0.000041         13          3         read
3.48       0.000037         37          1         writev
3.38       0.000036          9          4         fstat
3.01       0.000032         32          1         munmap
2.54       0.000027          9          3         brk
0.00       0.000000          0          7         mmap
0.00       0.000000          0          3         mprotect
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
-----
100.00     0.001064                46         2 total

```

四、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？

甲、clock_gettime(), 在 x86 與 ARM 上的理論精準度為 1nanosecond

乙、clock_gettime(&t1);f();clock_gettime(&t2);return t2-t1;

Ans: 由 2 和 3 的結果可知，若 buffersize 為 0/-1 時，system call 的時間大於函式庫呼叫，但若 buffersize 為 4096 時(fullybuffered)，system call 的時間小於函式庫呼叫，整體來說 ltrace 所花費的時間較 strace 多，因此函數庫呼叫的成本大於 system call。

linebuffered、unbuffered、fullybuffered 的意義

When a stream is *unbuffered*, characters are intended to appear from the source or at the destination as soon as possible. Otherwise characters may be accumulated and transmitted to or from the host environment as a block. When a stream is *fully buffered*, characters are intended to be transmitted to or from the host environment as a block when a buffer is filled. When a stream is *line buffered*, characters are intended to be transmitted to or from the host environment as a block when a new-line character is encountered. Furthermore, characters are intended to be transmitted as a block to the host environment when a buffer is filled, when input is requested on an unbuffered stream, or when input is requested on a line buffered stream that requires the transmission of characters from the host environment. Support for these characteristics is implementation-defined, and may be affected via the `setbuf` and `setvbuf` functions.

`strace` 會列出 run time 所用到的 system call

`ltrace` 會列出 run time 用到的 library function

- f: 同時跟蹤當前進程創建的子進程(fork 或者 clone)
- l: 指定特定的庫文件名
- o :指定輸出文件名
- p: 指定進程 PID
- r: 打印相對時間
- S: 打印系統調用
- T: 打印每次函數調用耗時
- c: 統計每個系統指令次數

`wchar` 能處理 unicode 字元(例：中文字)，而 `char` 只能處理 ascii 字元