

SP HW4

409410059 資工二 王秀晴

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

- 各參數執行時間

```
show@show-VirtualBox:~/system-programming/hw4$ time ./fileperf input.txt output.txt 0
real    0m7.465s
user    0m1.059s
sys     0m6.344s
show@show-VirtualBox:~/system-programming/hw4$ time ./fileperf input.txt output.txt -1
real    0m0.674s
user    0m0.164s
sys     0m0.462s
show@show-VirtualBox:~/system-programming/hw4$ time ./fileperf input.txt output.txt 4096
real    0m0.209s
user    0m0.139s
sys     0m0.025s
show@show-VirtualBox:~/system-programming/hw4$ time ./fileperf input.txt output.txt 16384
real    0m0.236s
user    0m0.145s
sys     0m0.044s
show@show-VirtualBox:~/system-programming/hw4$ time ./fileperf input.txt output.txt 65536
real    0m0.238s
user    0m0.155s
sys     0m0.033s
show@show-VirtualBox:~/system-programming/hw4$ time ./fileperf input.txt output.txt 1048576
real    0m0.244s
user    0m0.161s
sys     0m0.033s
show@show-VirtualBox:~/system-programming/hw4$ time ./fileperf input.txt output.txt 8388608
real    0m0.167s
user    0m0.102s
sys     0m0.017s
```

由測試時間可知，buffer size 越大，執行速度越快，但當 buffer size 大到一定程度後，差異並不明顯

2. 使用ltrace觀察你的應用程式呼叫「函數庫的情況」

- unbuffered (0)

```
show@show-VirtualBox:~/system-programming/hw4$ ltrace -c ./fileperf input.txt output.txt 0
% time      seconds  usecs/call   calls      function
-----
58.56      0.001978      1978         1  fopen
12.91      0.000436       436         1  fwrite
9.18       0.000310       310         1  fputc
6.60       0.000223       223         1  atoi
5.36       0.000181       181         1  getc
4.41       0.000149       149         1  setvbuf
2.99       0.000101       101         1  fclose
-----
100.00     0.003378              7  total
```

- linebuffered (-1)

```
show@show-VirtualBox:~/system-programming/hw4$ ltrace -c ./fileperf input.txt output.txt -1
```

% time	seconds	usecs/call	calls	function
26.32	0.000522	522	1	fopen
18.91	0.000375	375	1	fputc
16.19	0.000321	321	1	fwrite
14.52	0.000288	288	1	getc
12.00	0.000238	238	1	atoi
7.87	0.000156	156	1	setvbuf
4.19	0.000083	83	1	fclose
100.00	0.001983		7	total

- buffer size : 4KB

```
show@show-VirtualBox:~/system-programming/hw4$ ltrace -c ./fileperf input.txt output.txt 4096
```

% time	seconds	usecs/call	calls	function
45.95	0.000988	988	1	fopen
14.56	0.000313	313	1	atoi
10.88	0.000234	234	1	fputc
9.63	0.000207	207	1	setvbuf
8.42	0.000181	181	1	getc
6.51	0.000140	140	1	fwrite
4.05	0.000087	87	1	fclose
100.00	0.002150		7	total

在各參數下呼叫情況差不多，無較明顯的差異

3. 使用strace觀察你的應用程式呼叫「作業系統的情況」

- unbuffered (0)

```
show@show-VirtualBox:~/system-programming/hw4$ strace -c ./fileperf input.txt output.txt 0
```

% time	seconds	usecs/call	calls	errors	syscall
77.03	80.431669	5	13699328		read
22.97	23.980244	9	2524317		write
0.01	0.006083	1520	4		openat
0.00	0.000107	17	6		pread64
0.00	0.000105	15	7		mmap
0.00	0.000063	21	3		mprotect
0.00	0.000045	45	1		munmap
0.00	0.000022	7	3		brk
0.00	0.000008	4	2		fstat
0.00	0.000005	1	4		close
0.00	0.000005	2	2	1	arch_prctl
0.00	0.000000	0	1	1	access
0.00	0.000000	0	1		execve
100.00	104.418356		16223679	2	total

- linebuffered (-1)

```
show@show-VirtualBox:~/system-programming/hw4$ strace -c ./fileperf input.txt output.txt -1
```

% time	seconds	usecs/call	calls	errors	syscall
98.12	0.792306	2	325719		write
0.93	0.007530	2	3347		read
0.67	0.005377	1344	4		openat
0.28	0.002288	572	4		close
0.00	0.000014	3	4		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	0.807515		329102	2	total

- buffer size : 4KB

```
show@show-VirtualBox:~/system-programming/hw4$ strace -c ./fileperf input.txt output.txt 4096
% time      seconds    usecs/call     calls    errors syscall
-----
41.71      0.005803         1       3364         write
38.30      0.005328       1332         4       openat
16.83      0.002342         0       3347         read
 3.07      0.000427        106         4         close
 0.09      0.000013         3         4         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     0.013913                6747         2 total
```

buffer size 較小時需要一直呼叫 write 和 read ，會使執行速度變差

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？

- 若比較同類型的函數， system call 花的時間比 library call 少，但 system call 呼叫的次數較多

像 fwrite 和 write 在 linebuffered 所花時間為 321(usecs/call) 和 2(usecs/call)，呼叫次數為 1 和 325719 次

- 在各參數下所呼叫的 library call 差異不大，主要影響執行時間的為 system call 的次數