

Lab1-Report

21302010042

侯斌洋

实验环境：ubuntu22.04

1：实验数据

实验数据源文件为 lab1_test_data.xlsx。

以下为截图：

(1) 测试 read()

	A	B	C	D
1	bufsize/Byte	real/s	user/s	sys/s
2	1	10.123	1.271	8.802
3	4	2.541	0.345	2.184
4	16	0.651	0.060	0.585
5	64	0.170	0.005	0.162
6	256	0.050	0.012	0.034
7	1024	0.016	0.001	0.015
8	4096	0.010	0.001	0.009
9	8192	0.009	0.001	0.008
10	16384	0.009	0.001	0.007
11	32768	0.009	0.001	0.008
12	65536	0.008	0.001	0.007
13	131072	0.008	0.002	0.006
14	262144	0.008	0.001	0.007
15				
16	read size is 40960000			
17				

(2) 测试 fread()

	A	B	C	D
1	bufsize/Byte	real/s	user/s	sys/s
2	1	0.642	0.582	0.057
3	4	0.180	0.162	0.013
4	16	0.068	0.043	0.009
5	64	0.020	0.010	0.010
6	256	0.013	0.000	0.013
7	1024	0.011	0.001	0.009
8	4096	0.008	0.001	0.008
9	8192	0.009	0.001	0.008
10	16384	0.008	0.001	0.008
11	32768	0.008	0.001	0.008
12	65536	0.008	0.001	0.007
13	131072	0.007	0.001	0.006
14	262144	0.007	0.001	0.007
15				
16	read size is 40960000			
17				

(3) 测试带 O_SYNC 的 write()

	A	B	C	D
	bufsize/Byte	real/s	user/s	sys/s
1	1	90.404	0.002	23.187
2	4	25.085	0.004	6.119
3	16	5.870	0.001	1.366
4	64	1.475	0.000	0.362
5	256	0.310	0.001	0.088
6	1024	0.085	0.001	0.023
7	4096	0.033	0.002	0.007
8	8192	0.018	0.006	0.001
9	16384	0.010	0.004	0.001
10	32768	0.007	0.001	0.003
11	65536	0.006	0.003	0.001
12	131072	0.006	0.000	0.003
13	262144	0.005	0.002	0.001
14	409600	0.006	0.003	0.000
15				
16				
17	write size is 409600			

(4) 测试不带 O_SYNC 的 write()

	A	B	C	D
	bufsize/Byte	real/s	user/s	sys/s
1	1	1.106	0.049	0.956
2	4	0.290	0.004	0.234
3	16	0.075	0.001	0.058
4	64	0.017	0.001	0.016
5	256	0.006	0.001	0.005
6	1024	0.003	0.001	0.002
7	4096	0.002	0.001	0.001
8	8192	0.003	0.001	0.002
9	16384	0.003	0.002	0.001
10	32768	0.003	0.002	0.001
11	65536	0.003	0.001	0.002
12	131072	0.003	0.001	0.001
13	262144	0.004	0.002	0.000
14	409600	0.003	0.001	0.002
15				
16				
17	write size is 409600			

数据特点:

- (1) 系统 CPU 要比用户 CPU 调用的时间更长, 时钟时间也主要由系统 CPU 组成。
在 bufsize 从 1 增大到 4096 的过程中, 时钟时间明显减少, 从 4096 增大到 262144 的过程中时钟时间变化很小, 考虑到 time 测试程序时间的误差, 可以认为 bufsize 超过 4096 后时钟时间基本保持稳定。
- (2) bufsize 小于 64 时时钟时间主要由用户 CPU 组成, 大于 64 时主要由系统 CPU 组成。
在 bufsize 从 1 增大到 4096 的过程中, 时钟时间明显减少, 从 4096 增大到 262144 的过程中时钟时间变化很小, 考虑到 time 测试程序时间的误差, 可以认为 bufsize 超过 4096 后时钟时间基本保持稳定。
###fread 与 read 数据进行对比的话, 可以看出在 bufsize 小于 4096 时 fread 的时钟时间明显更短, 而 bufsize 大于 4096 后差距不大。
- (3) bufsize 小于 1024 时系统 CPU 要比用户 CPU 时间明显更长, 大于 1024 时差距不明显。
系统 CPU 和用户 CPU 之和明显小于时钟时间, 此特点为 (3) 所独有。
在 bufsize 从 1 增大到 32768 的过程中, 时钟时间明显减少, 从 32768 增大到 409600 的过程中时钟时间变化很小,

- (4) bufsize 小于 64 时系统 CPU 要比用户 CPU 时间明显更长，大于 64 时差距不明显。在 bufsize 从 1 增大到 1024 的过程中，时钟时间明显减少，从 1024 增大到 262144 的过程中时钟时间变化很小
- ###带 O_SYNC 和不带 O_SYNC 的 write 相比，不带 O_SYNC 的 write 时钟时间明显###更短。

2：分析与思考

在 bufsize 较小时，循环调用 read，write 的次数会很多，而 read 等系统调用内部存在大量的代码，故系统调用的时间会明显大于用户 CPU。而当 bufsize 较大时，循环次数减少，系统调用的次数也减少，故系统 CPU 的时间会明显减少，到达一定范围时，调用次数减少不那么明显了，由于单次系统调用本身开销也较大，故会保持稳定。

对于 (3) 中系统 CPU 和用户 CPU 之和明显小于时钟时间的特点，这是由于 read，fread，不带 O_SYNC 的 write 在进行 IO 时 CPU 不会一直等待，而是切换到其他进程，而带 O_SYNC 的 write 则会一直等待 IO 慢慢完成。此时 time 算在了时钟时间里但由于并未执行任何代码而只是等待，故会存在 real 与 sys+user 的差距。

read 和 fread 的 bufsize 均在 4096 左右出现临界值，这是由于磁盘上的一页大概就是 4096 的大小，由于磁盘上的文件是以页来区分的，每一页都需要单独寻找，故如果磁盘每次找到的一页能被完全读取，则就可以达到最大 IO 效率。所以会在 4096 处出现临界值。而之所以 fread 比 read 在 bufsize 较小时效率明显高，是因为 fread 是包装函数，函数内部自动设立了较大的缓冲区，故效率相对较高。

对于 (3) (4) 中 bufsize 效率临界出现在 32768 和 256，则可能与 write 的内部实现，文件系统空间分配以及磁盘的 IO 方式等多种因素有关，不过可以看到的是 (3) (4) 都存在一个临界值可以使 IO 效率更高，说明 write 对处理某个大小的整块的效率也更高。

3：提高 IO 效率

- (1) 一般情况下可以尽量避免使用 read 等系统调用，转而使用 fread 等包装函数。因为使用系统调用时需要仔细考虑 bufsize 的设置，否则会严重影响 IO 效率，而使用 fread 等包装函数则不需要考虑 bufsize 的问题，包装函数内部会自动设置较为合理的缓冲区。但由于 fread 进行了包装，内部存在一些冗余代码，可能也会影响程序效率，要根据具体场景进行取舍。
- (2) 如果对所使用的设备有一定了解的话，比如知道磁盘一页的大小，文件系统的运作方式等，则可以在编写程序时使用合适的 bufsize 来提高 IO 效率。
- (3) 系统调用会占用很多 CPU 时间，在使用 read,write 等系统调用循环读取时要注意减少循环的次数以减少系统调用的开销。