

实验报告一

1. 实验名称：timewrite 程序编写
2. 实验内容描述：编写程序
timewrite <outfile> [sync]
不得变更程序的名字和使用方法。sync 参数为可选，若有，则输出文件用 O_SYNC 打开。
3. 设计思路：
 - a. 首先对输入的格式进行检测，非法输入则报错退出并提示正确的输入格式
 - b. 然后在保证输入格式正确的前提下检测是否有 sync，若有则在 open 函数使用时加上 O_SYNC 标识符
 - c. 成功打开后使用 read 函数进行读操作，并创建好 buf 数组储存数据，并开始计时
 - d. 开始循环操作，每次循环改变 BUFFERSIZE 大小，按题目要求从 256 到 128k，每次循环大小乘 2
 - e. 循环内嵌套 while 循环，用 RestOfLength 记录还剩多少数据没有写完，在其归零之前反复执行 write 函数，将 buf 中的数据写入目标文件，每次循环 loop 加 1
 - f. while 结束后获取结束时间，并打印结果
 - g. 在所有循环结束之后用 close 函数关闭文件，然后结束程序。
4. 实验结果：
源程序：timewrite.c erro.c apue.h Makefile
可执行程序：timewrite
执行过程：用 make 命令使用 Makefile 写好的规则来生成，具体如下：
timewrite.c 和 erro.c 分别编译成 o 文件
然后 timewrite.o 和 erro.o 一起编译成 timewrite 可执行文件
运行程序 在目标文件夹下使用命令 ./timewrite <outfile> [sync]
程序结果：

```
[cs214126@mc0re 1]$ ./timewrite f2.txt <f1.txt sync
```

BUFFSIZE	用户 CPU(s)	系统 CPU(s)	时钟时间(s)	循环次数
256	0.00	0.01	2.27	512
512	0.00	0.01	1.07	256
1024	0.00	0.01	0.52	128
2048	0.00	0.01	0.25	64
4096	0.00	0.01	0.14	32
8192	0.00	0.01	0.06	16
16384	0.00	0.01	0.05	8
32768	0.00	0.01	0.02	4
65536	0.00	0.01	0.01	2
131072	0.00	0.01	0.01	1

```
[cs214126@mc0re 1]$ ./timewrite <f1.txt f2.txt
```

BUFSIZE	用户 CPU(s)	系统 CPU(s)	时钟时间(s)	循环次数
256	0.00	0.00	0.01	512
512	0.00	0.00	0.00	256
1024	0.00	0.00	0.00	128
2048	0.00	0.00	0.00	64
4096	0.00	0.00	0.00	32
8192	0.00	0.00	0.00	16
16384	0.00	0.00	0.00	8
32768	0.00	0.00	0.00	4
65536	0.00	0.00	0.00	2
131072	0.00	0.00	0.00	1

5.体会和建议:

体会：现代计算机计算速度非常快，可见在此程序运行中用户 cpu 和系统 cpu 时间可忽略不计，此外使用 sync 来写文件可以看出速度明显慢于默认格式，让我们明白平常对普通文件的处理应当多用缓存写入方案，sync 用于重要且不允许丢失的文件写入

建议：提升 C 标准支持，本次实验过程中 for 循环内 int size=256 会报错，必须要提前定义好 size 然后在 for 循环使用，另外在当前标准下，已经 include<fcntl.h>了却还是会出现报错：‘S_IRUSR’ 未声明（在此函数内第一次使用），需要 include 更多头文件最终解决问题

6.完成人姓名：李国楷

完成时间：2023.10.14