

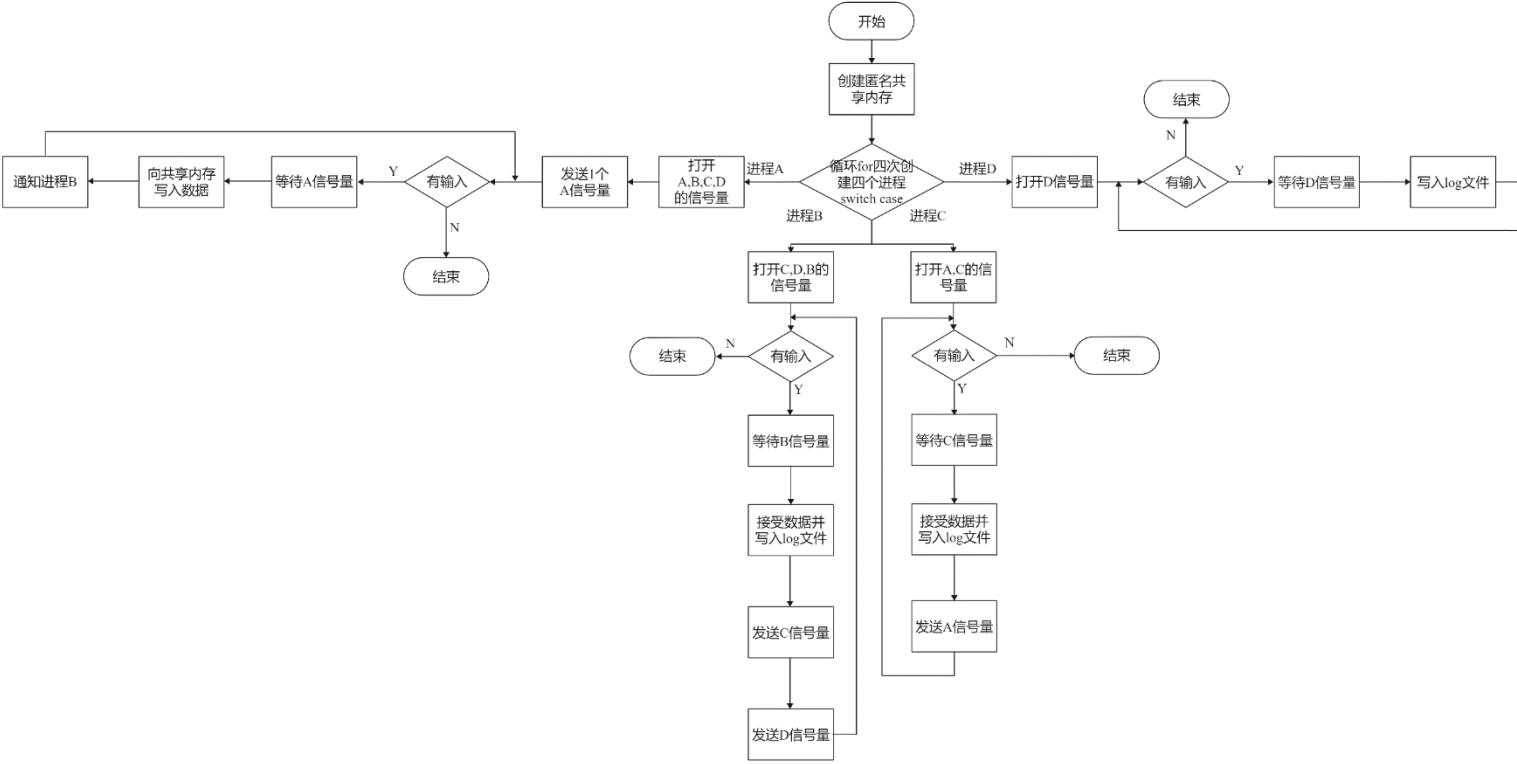
大作业 2 设计文档

需求说明

创建四个用户进程 A、B、C、D，当进程 A 启动后通过一个缓冲区不断向 B、C 进程发送消息，A 每次向缓冲区送入消息以后，必须等待 B、C 都取走后，才可以发送下一个消息。要求 B 取完消息后 C 才能取。

进程 B 收到进程 A 的消息以后，通过共享内存将消息通知到进程 D，进程 A、B、C、D 在处理完各自的任务以后，依次输出 logs 到同一个本地日志文件中。

代码流程图



API 接口及参数

```
class LogFile
{
public:
    LogFile(const char *path);
    bool writeLog(const char *log);
}
```

LogFile 类功能：方便各进程向同一日志文件写入信息。

构造函数 LogFile(const char *path);

参数：日志文件的路径

作用：打开日志文件

bool writeLog(const char *log);

返回值：写入日志成功返回 true，否则返回 false

参数：写入日志的内容

作用：向日志文件内写入当前时间以及内容

代码运行展示

```
Fri Mar 12 22:14:04 2021  
B start.  
Fri Mar 12 22:14:04 2021  
C start.  
Fri Mar 12 22:14:04 2021  
D start.  
Fri Mar 12 22:14:04 2021  
A start.  
Fri Mar 12 22:14:04 2021  
A Write 1  
Fri Mar 12 22:14:04 2021  
C Recv 1  
Fri Mar 12 22:14:04 2021  
B Recive 1  
Fri Mar 12 22:14:04 2021  
D processing...  
Fri Mar 12 22:14:05 2021  
A Write 2  
Fri Mar 12 22:14:05 2021  
B Recive 1  
Fri Mar 12 22:14:05 2021  
C Recv 1  
Fri Mar 12 22:14:05 2021  
D processing...  
Fri Mar 12 22:14:06 2021  
A Write 3  
Fri Mar 12 22:14:06 2021  
A:Exit  
Fri Mar 12 22:14:06 2021  
B Recive 4  
Fri Mar 12 22:14:06 2021  
D processing...  
Fri Mar 12 22:14:06 2021  
C Recv 4
```

上图为日志文件的内容。

通常一个过程为这样：

A 向共享内存写入数据

B 取出数据，并通知 C 取数据，同时通知 D，让 D 打印 processing

C 取出数据，并通知 A 继续写入