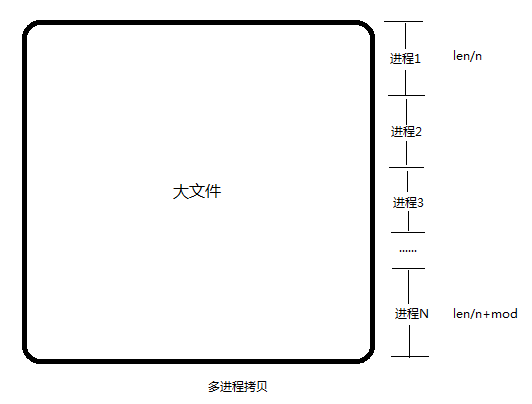
## 阶段综合练习一

实现文件多进程拷贝。

假设有一个超大文件，需对其完成拷贝工作。为提高效率，可采用多进程并行拷贝的方法来实现。假设文件大小为len，共有n个进程对该文件进行拷贝。那每个进程拷贝的字节数应为len/n。但未必一定能整除，我们可以选择让最后一个进程负责剩余部分拷贝工作。可使用len % (len/n)将剩余部分大小求出。

为降低实现复杂度，可选用mmap来实现源、目标文件的映射，通过指针操作内存地址，设置每个进程拷贝的起始、结束位置。借助MAP\_SHARED选项将内存中所做的修改反映到物理磁盘上。



## 阶段综合练习二

实现简单的交互式shell。

使用已学习的各种C函数实现一个简单的交互式Shell，要求：

1、给出提示符，让用户输入一行命令，识别程序名和参数并调用适当的exec函数执行程序，待执行完成后再次给出提示符。

2、该程序可识别和处理以下符号：

1) 简单的标准输入输出重定向：仿照例 “父子进程ls | wc -l”，先dup2然后exec。

2) 管道（|）：Shell进程先调用pipe创建管道，然后fork出两个子进程。一个子进程关闭读端，调用dup2将写端赋给标准输出，另一个子进程关闭写端，调用dup2把读端赋给标准输入，两个子进程分别调用exec执行程序，而Shell进程把管道的两端都关闭，调用wait等待两个子进程终止。类似于“兄弟进程间ls | wc –l”练习的实现。

你的程序应该可以处理以下命令：

○ls△-l△-R○>○file1○

○cat○<○file1○|○wc△-c○>○file1○

注：○表示零个或多个空格，△表示一个或多个空格

实现步骤：

1. 接收用户输入命令字符串，拆分命令及参数存储。（自行设计数据存储结构）

2. 实现普通命令加载功能

3. 实现输入、输出重定向的功能

4. 实现管道

5. 支持多重管道

## 阶段综合练习三

简易本地聊天室

借助IPC完成一个简易的本地聊天功能。设有服务器端和客户端两方。服务启动监听客户端请求，并负责记录处理客户端登录、聊天、退出等相关数据。客户端完成登录、发起聊天等操作。可以借助服务器转发向某个指定客户端完成数据包发送(聊天)。

客户端向服务发送数据包，可采用如下协议格式来存储客户端数据，使用“协议号”区分客户端请求的各种状况。服务器依据包号处理客户端对应请求。

