## 为什么做这个项目？

我从github上参考的，用来锻炼自己的Linux系统编程技能，弥补本科阶段对操作系统的学习

## 项目功能描述一下？

这是一个文件管理系统，实现了类似网盘的操作，项目的服务端和客户端间通信我采用的的是TCP协议，并且自己编写了**应用层的传输协议**(小火车协议，防止tcp协议传输的粘包和半包问题)，通信要对传输的数据包进行解包

考虑到会与多个客户端通信，在服务器端编写了**线程池**的框架，并注册了**epoll**实例来监听所连接的客户端就绪事件，当epoll监听到有事件就绪时，会将其放入到线程池的任务队列中，并将其交给子线程处理，这里也用到了**生产者消费者**模型，来进行同步和互斥操作，避免客户端链接的过程中线程间竞争资源可能会导致意外退出x

服务端接收到客户端发送过来的命令后解析命令，执行相应的操作。并将结果返回给客户端，客户端收到回送的消息再次解析消息执行相应的操作。

## 项目有啥亮点？

框架采用的是线程池结合epoll多路复用的模型处理并发事件的发生；上传以及下载文件基于自己使用了零拷贝技术重新设计；用Mysql作为数据库，实现一个虚拟文件系统；实现文件的断点续传和秒传等功能；用户登录注册也进行了加密操作

## 用到什么技术？

数据库，线程池，秒传，零Copy，加解密等。

水平触发的epoll I/O多路复用，零拷贝技术，客户端和用户端发送文件均采用的是sendfile，客户端接收文件采用的是splice，用户端接收文件是自己设计的recvn。

## 计算md5

在文件引用记数表中计算文件内容的MD5值，并用其作为文件名

用户上传和下载的时候是数据库会根据计算文件的MD5值判断与记数表的文件MD5是否一致

## 数据库是如何设计的

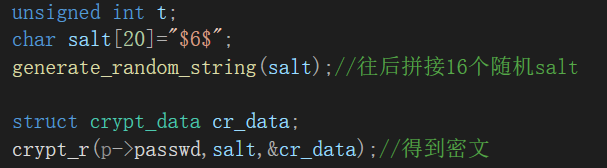
使用MySQL，创建了三张表，分别是用户表，文件目录表和文件引用记录表。用户注册时候会将密码进行加密存入数据库，登录的时候会对密码进行加密与数据库密文作对比验证。文件目录表是一个虚拟文件表，表示文件目录之间的关系，标识文件所属用户，用户之间目录隔离，而文件引用记录表是为了秒传的设计，表明服务器端现在有哪些用户，若不同用户上传同一文件就可以秒传。

客户端与MySQL数据库连接，因为只有一个服务器端，MySQL本身相当于是一个集群，没有使用中间件，它接收到客户端的请求就转发给服务器端进行通信

## 登录过程是如何的

注册：明文密码，用公钥加密，密文传递给服务器，服务器把明文密码进行sha512散列算法

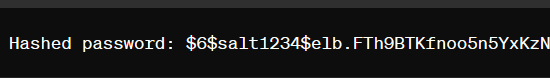
登录：服务器得到客户端的用户名密码，服务器对该密码进行sha512散列算法，并查询数据库用户名对应的盐值和密文，比对数据库内存储的是否一致



$6$ 是一个标识符，表示使用的是SHA-512算法进行加密，用于告知加密函数使用的哈希算法。

crypt\_r 返回的是一个指向加密结果的字符串，这个字符串就是经过哈希处理的密码，它包含了算法标识符、盐值以及哈希后的密码





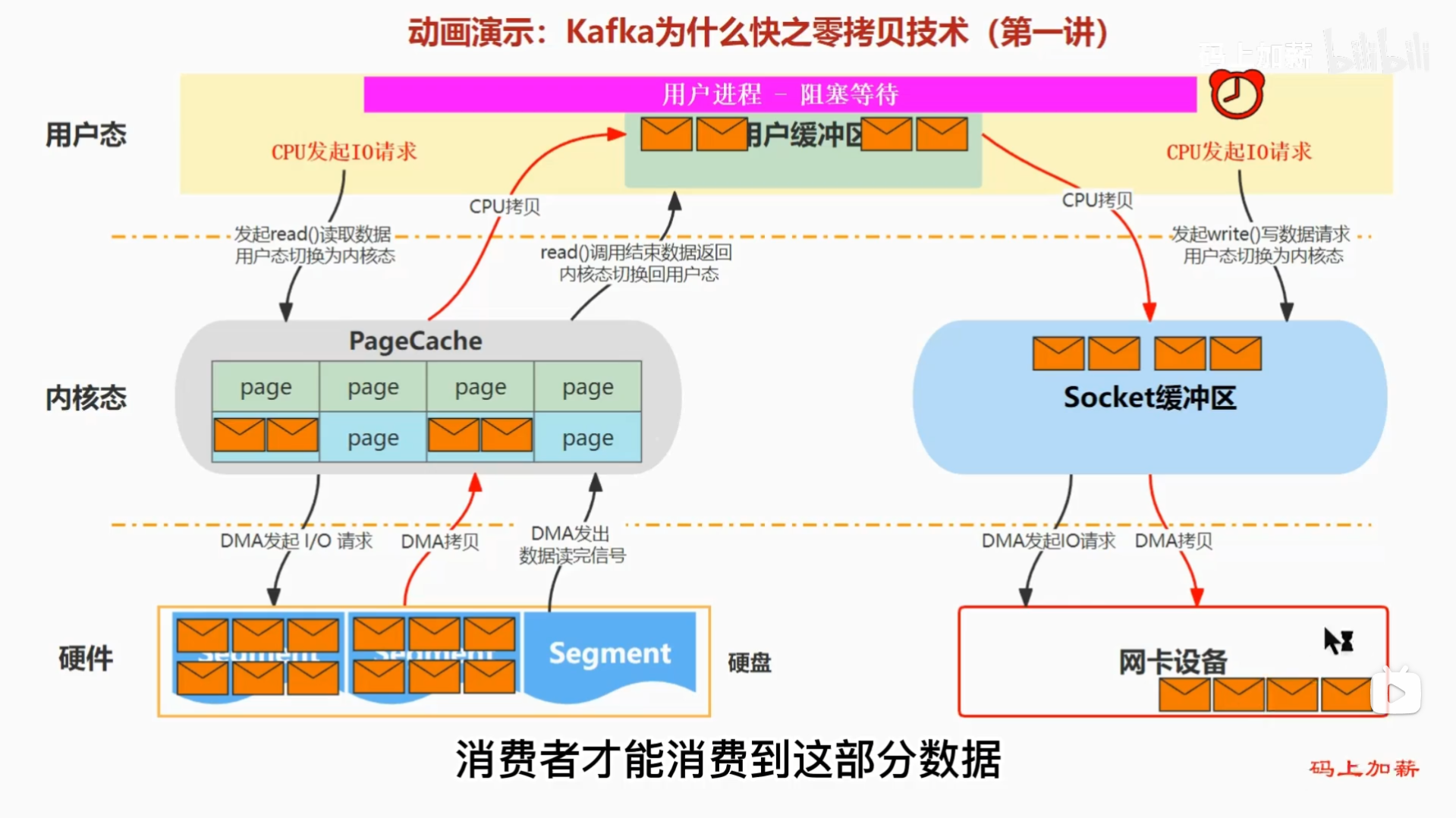
账户加密：crypt\_r() 函数用于加密用户密码的存储是很好的，但是它不适用于一般的数据加密。

传输文件：用md5验证数据完整性。

## 零copy的原理是什么

通过系统调用，可以在文件描述符之间进行数据的直接传输，而无需在用户态和内核态之间进行数据复制，sendfile可以直接在内核态进行一次CPU拷贝到socket缓冲区，再直接传输到目标套接字中，splice可以在内核态中通过管道的方式将数据拷贝到socket缓冲区，同样直接传输到目标套接字上。

传统的数据拷贝过程：



线程池实现的TCP协议网盘(c语言)

我的项目是一个基于TCP协议的文件管理系统，能实现用户的注册、登录、上传、下载等功能，搭建了一个线程池提高了服务器的并发处理能力。

在Linux 环境下用C语言编写的，

技术难点: socket接口编程，自己设计的应用层传输协议、

线程池项目描述:搭建一个服务端，多个客户端，客户端通过对服务器发送简单的命令来实现相应的功能。C/S 之间的通信有自己编写的应用层协议，服务端采用了线程池。

客户端的运行界面类Linux终端，具体命令如下:

gets + 文件名:从服务器下载相应的文件 ，并且实现断点续传

puts +文件名:上传相应的文件到服务器

项目实现细节:自主设计传输协议、

## 会哪些Linux命令？

Ls,cd,rmdir,cp,mv,grep,ps-aux,-l

## 进程间通信的方式有哪些（进程间同步方式）

管道；消息队列（生产者消费者模型里，几个线程处理生产数据，几个线程处理数据）；信号（sigpipe，sigchld）；socket（在网络通信进程与进程之间）

## 进程与线程有什么区别

首先，进程拥有独立的内存空间，进程之间相互独立，是程序执行的基本单位，而线程共享同一进程的地址空间，是操作系统调度的最小单位。一个进程可以包括多个线程。

其次，进程的创建和销毁开销大，需要分配释放独立的内存空间，而线程开销小，只需要分配释放一些堆栈空间

还有，进程上下文切换开销大，线程的话只需要保存和恢复寄存器状态

最后，进程间是通过IPC，管道以及消息队列等通信，而线程是通过共享相同的地址空间

## fork创建的子进程有什么特点

子进程拥有独立的地址空间，同时继承了父进程的大部分资源（文件描述符，用户标识等），在父进程中fork会返回子进程PID，在子进程fork会返回0（与原始线程具有相同的层级关系，都是父进程的线程）

## 什么是多路复用，应用场景是

首先，IO多路复用允许单个进程同时监视和等待多个文件描述符，并在其中任何一个事件通知就绪时执行相应的操作，通常用于实现高效的并发编程。

其次，多路复用可以应用在网络编程上，实现并发服务器，管道通信

## Mysql数据库如何进行关联查询

在表中定义外键，指定引用的表和列，建立起关联关系，查询可以使用JOIN语句，也可以像项目一样在程序中使用sprintf拼接语句，进行更灵活的查询（即使没有外键关联也可以）

## 数据库的索引有哪几种

单列索引（Email），唯一索引（Email/id），复合索引（type,price），哈希索引（id）

## 进程池，线程池是什么

进程池是一种并发编程模型，它预先创建了一组进程，并放入一个池中等待任务的到来。进程池使得进程可以重复利用，进程间相互独立，也可以并行执行，不会存在线程间的竞争和同步开销，能够更好地利用 CPU 资源，适合处理 CPU 密集型任务。

线程池的工作流程同进程池类似，但线程的创建和销毁开销小，有着更快的响应速度，会减少系统资源的消耗。而线程也存在着同步和互斥的问题，可以使用互斥锁解决。

进程池线程池都可以实现进程和线程的复用，即执行完一个任务后，并不会销毁进程，线程，而是继续保持在池中，等待下一个任务的到来，提高系统的效率。

## 什么是生产者，消费者模型

生产者：生产者线程负责生产任务并将其放入任务队列中。

消费者：消费者线程负责从任务队列中取出任务并进行处理。

缓冲区：任务队列充当了缓冲区的角色，用于存储生产者生产的任务。

在等待期间，生产者线程会进入阻塞状态，一旦有空闲位置可供放入新的任务时，其他线程会通过条件变量发送信号（或广播），唤醒生产者线程。被唤醒的生产者线程会重新尝试获取任务队列的互斥锁，然后继续向任务队列中添加任务。

## 虚拟地址和物理地址怎么转换的？

在网络编程中，程序通过网络套接字进行数据的接收和发送。接收到的数据通常存储在内存中的缓冲区中，而这个内存缓冲区的地址就是虚拟地址。程序生成的虚拟地址实际上是相对于进程的虚拟地址空间的偏移量。当应用程序需要处理这些数据时，操作系统会通过内存管理单元（MMU）将虚拟地址转换为物理地址，以便访问实际的物理内存。