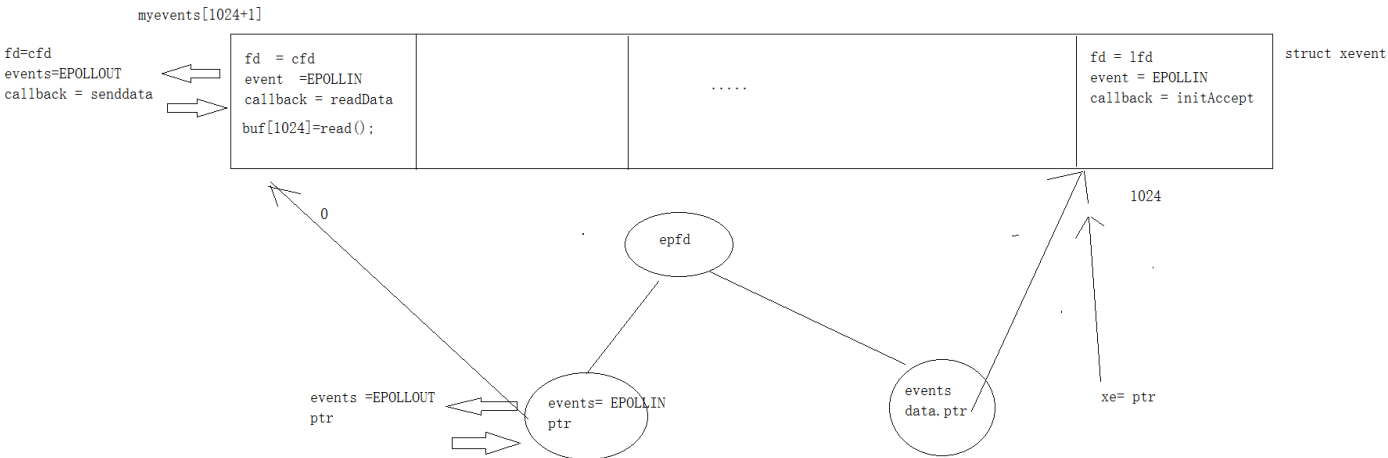


05 反应堆线程池

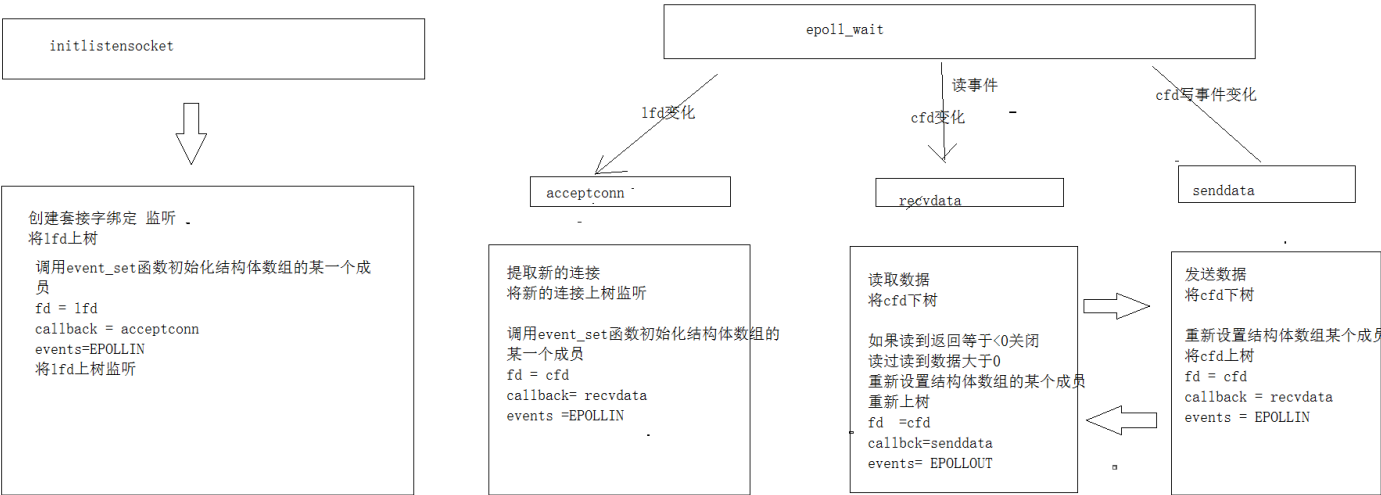
学习目标

- 1. 了解epoll反应堆设计思想
- 2. 了解线程池思想以及能看懂线程池实现源码
- 3. 掌握tcp和udp的优缺点和使用场景
- 4. 说出udp服务器通信流程
- 5. 说出udp客户端通信流程
- 6. 独立实现udp服务器代码
- 7. 独立实现udp客户端代码
- 8. 熟练掌握本地套接字进行本地进程通信

2 epoll反应堆
文件描述符 事件 回调函数 封装在一起
结构体来封装



反应堆复杂版代码



3 线程池
线程池,事先创建几个线程,,不停取任务,如果没有任务休眠,省去了不停的创建线程销毁销毁线程的事件和资源
注意: 线程池 处理的任务,所需要处理的时间必须很短
一个锁
两个条件变量
循环队列

事先创建几个线程

一个任务队列

线程池中的线程不停的去从任务队列取 任务, 有任务到来往任务队列添加 (生产者和消费者模型)

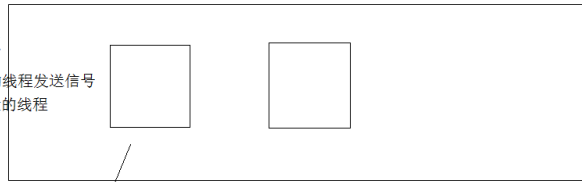
任务队列 - 锁

条件变量 2 个 一个阻塞取任务线程
一个阻塞添加者线程

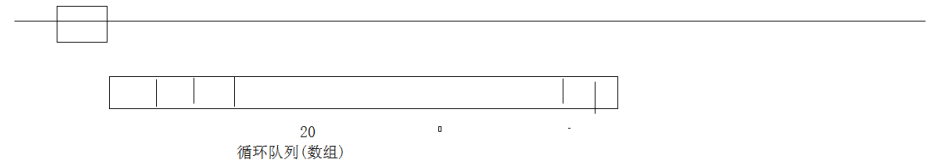
pthread_cond_wait 解锁 等待条件 条件满足抢锁

pthread_cond_signal 给某一个等待在条件变量的线程发送信号

pthread_cond_broadcast 给所有等待在等待条件变量的线程发送信号



抢锁
有任务, 直接取任务, 解锁
如果没有任务, 解锁 等待条件, 条件满足抢锁, 取任务 解锁



4 UDP和TCP

TCP: 传输控制协议 安全可靠 丢包重传 面向连接(电话模型)

UDP: 用户数据报协议 不安全不可靠 丢包不重传 快 不面向连接(邮件模型)

tcp通信流程:

服务器: 创建流式套接字 绑定 监听 提取 读写 关闭

客户端: 创建流式套接字 连接 读写 关闭

收发数据:

read recv

ssize_t recv(int sockfd, void *buf, size_t len, int flags); // flags==MSG_PEEK 读数据不会删除缓冲区的数据

write send

ssize_t send(int sockfd, const void *buf, size_t len, int flags); // flags=1 紧急数据

udp通信流程

服务器: 创建报式套接字 绑定 读写 关闭

客户端: 创建报式套接字 读写 关闭

发数据:

ssize_t sendto(int sockfd, const void *buf, size_t len, int flags,
const struct sockaddr *dest_addr, socklen_t addrlen);

dest_addr: 目的地的地址信息

addrlen: 结构体大小

收数据:

ssize_t recvfrom(int sockfd, void *buf, size_t len, int flags,
struct sockaddr *src_addr, socklen_t *addrlen);

src_addr: 对方的地址信息

addrlen: 结构体大小的地址

5 创建报式套接字

socket

int socket(int domain, int type, int protocol);

参数:

domain : AF_INET
type : SOCK_DGRAM
protocol : 0

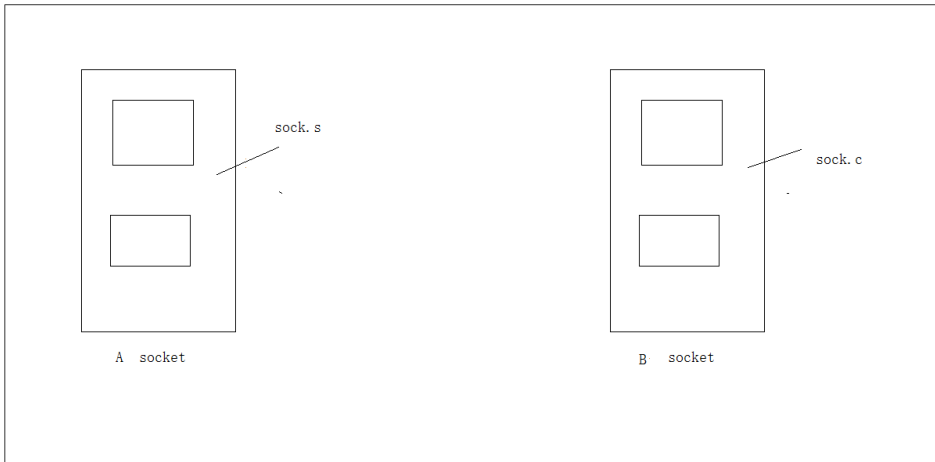
6 本地套接字

unix domain socket

本地套接字通信

全双工

套接字用文件来标识, 这个文件在绑定之前是不能存在



同一台主机

7 本地套接字实现tcp通信

创建本地套接字

绑定

监听

提取

读写

关闭

创建本地套接字用于tcp通信

int socket(int domain, int type, int protocol);

参数:

domain : AF_UNIX
type : SOCK_STREAM
protocol : 0

```
绑定
int bind(int sockfd, const struct sockaddr *addr,
        socklen_t addrlen);

sockfd: 本地套接字
addr: 本地套接字结构体地址
struct sockaddr_un {
    sa_family_t sun_family;    /* AF_UNIX */
    char        sun_path[108]; /* pathname *///文件的路径名
};
addrlen: sockaddr_un大小\
```

```
提取
int accept(int sockfd, struct sockaddr *addr, socklen_t *addrlen);

addr: struct sockaddr_un 结构体地址
```

需要注意的点:
客户端可以隐式绑定,但是服务器不可以
绑定指定文件时m,这个文件必须不存在,如果存在绑定失败