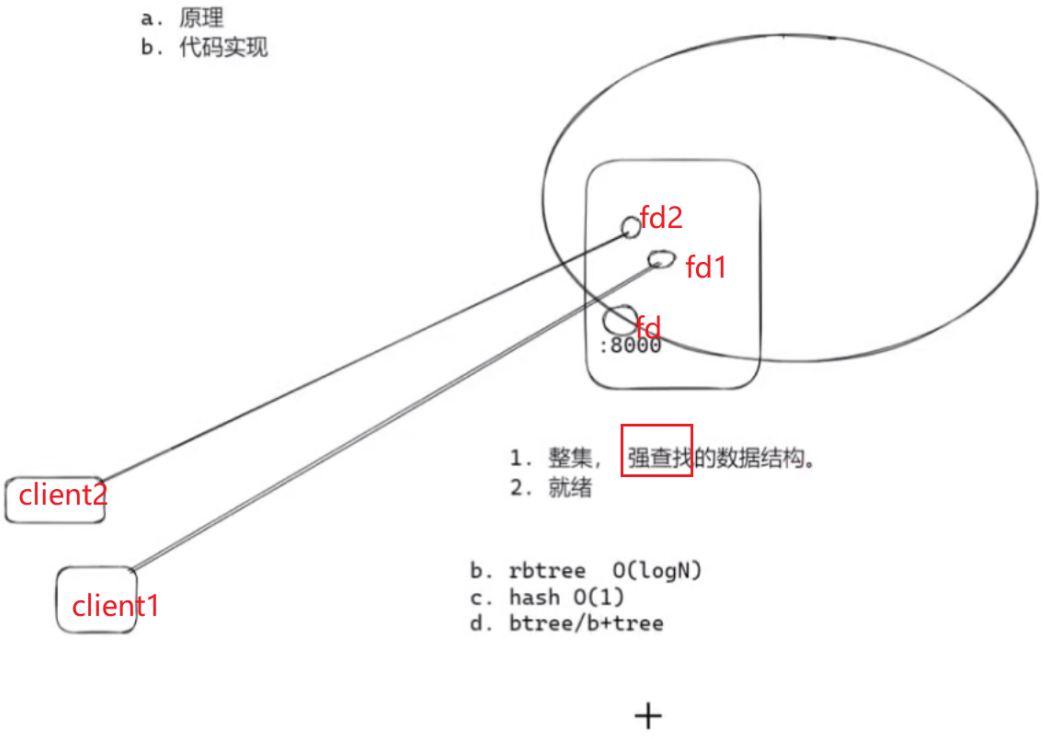
1.实现epoll 的问题：

epoll ⟶ 如何实现

1. epoll的数据结构如何选择
2. 以tcp，网络io的可读可写如何判断?
3. epoll如何做到线程安全?
4. 水平触发与边沿触发如何实现?

1.数据结构如何选择



a.**整集**需要**强查找**的数据结构

a. list, array O(N)b. rbtree O(logN)
c. hash O(1)
d. btree

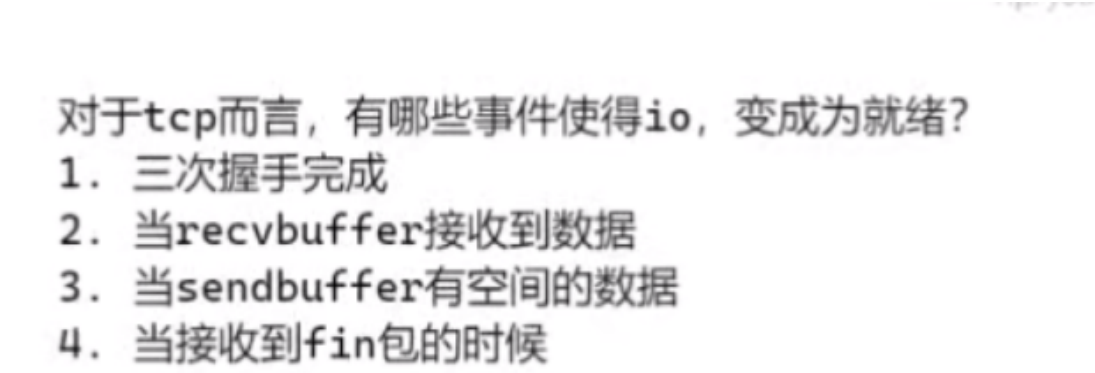
首先a的复杂度可以达到O(n)，**太慢**直接排除，其次**整集io的数量不确定**，该结构需要能够兼容数量大的连接，但在链接数量很少的情况下也能高效，此时哈希的查找性能很高，但是**牺牲了内存。**(如果能确定io数量在10w级以上，此时hash性能更好)b/b+tree更多用在**磁盘io**，效率这里没有rbtree高，（为什么不选择跳表skiptable？）

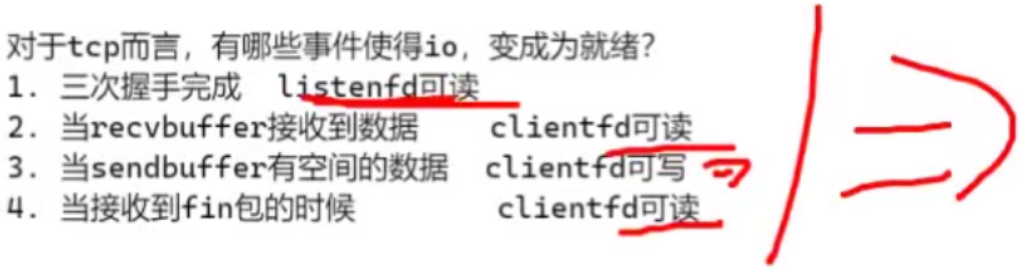
所以rbtree是首选！！！

b.**就绪**需要queue（用list实现），可以**均匀处理**就绪队列

所以需要queue！！！

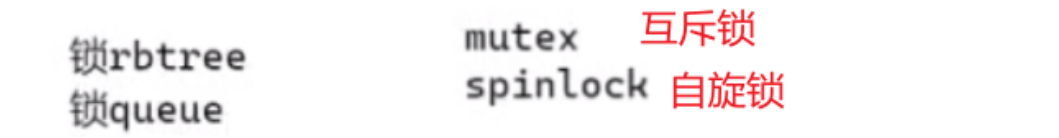
2.对于tcp，哪些事件使得io变为就绪?





3.线程安全

两把锁，一把锁rbtree，一把锁queue list

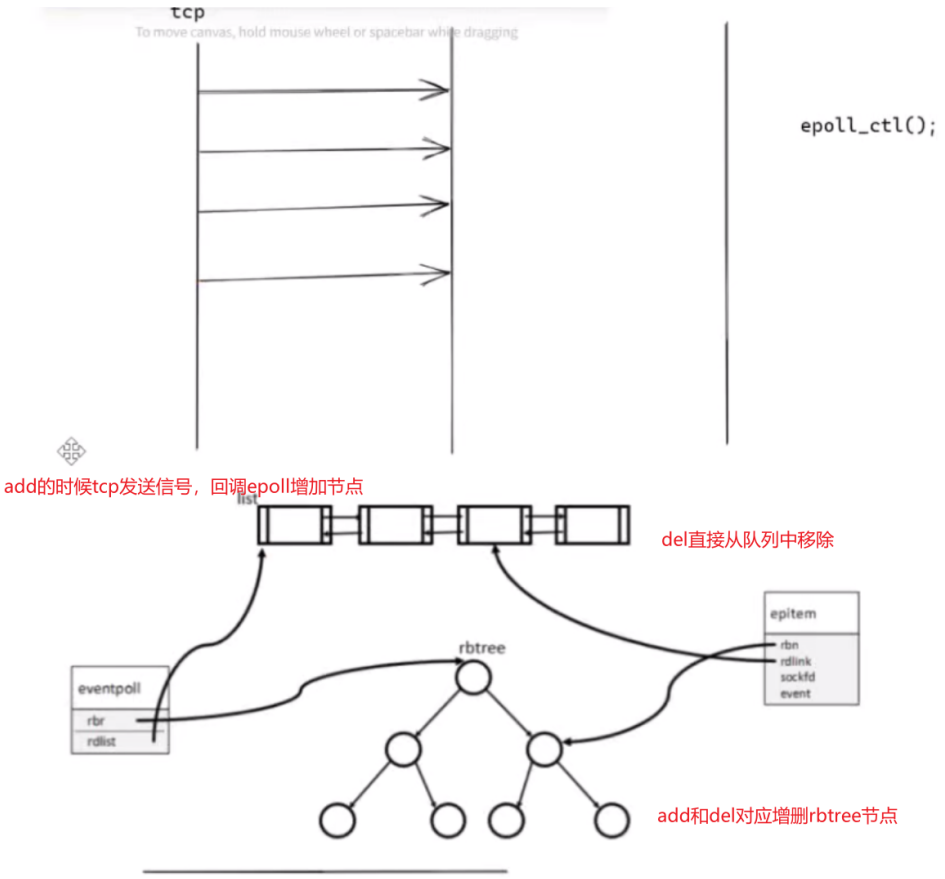


mutex:当有线程占用时会直接休眠

spinlock:当有线程占用时一直等待

因为红黑数增删的过程复杂不可控，选择mutex，避免一直等待
queue的操作只有两个指针作为临界资源，轻量级，选择spinlock

4.添加和删除



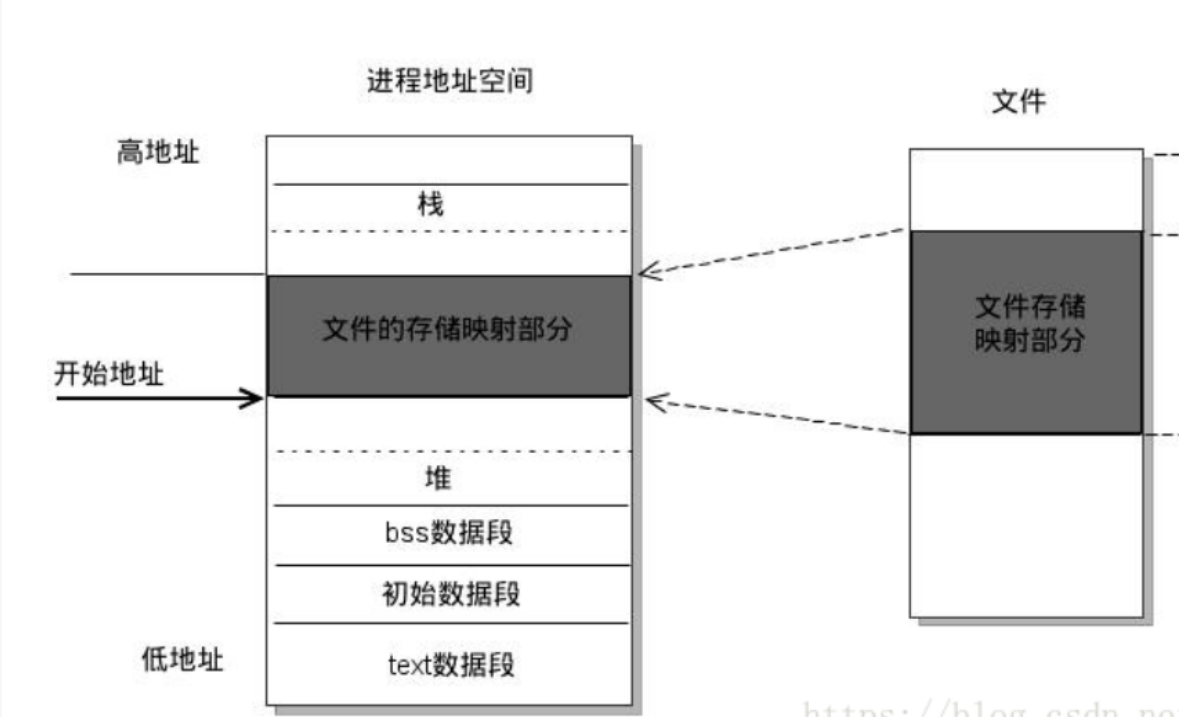
5.水平触发和边缘触发

水平触发：recv_buf有数据就触发

边缘触发： recv_buf来一个数据才触发一次

思考：epoll实现是否采用了mmap?

mmap是一种内存映射文件的方法，即将一个文件或者其它对象映射到进程的地址空间，实现文件磁盘地址和进程虚拟地址空间中一段虚拟地址的一一对映关系。



6.面试技巧

问到epoll如何实现的：

1.数据结构的选择

2.网络io的可读可写如何判断的

3.线程安全如何实现

4.水平触发和边缘触发如何实现的