什么是同步阻塞IO？

应用进程被阻塞，直到数据复制到应用进程缓冲区中才返回。

举例理解

你早上去买有现炸油条，你点单，之后一直等店家做好，期间你啥其它事也做不了。（你就是应用级别，店家就是操作系统级别， 应用被阻塞了不能做其它事）

什么是同步非阻塞IO？

应用进程执行系统调用之后，内核返回一个错误码。应用进程可以继续执行，但是需要不断的执行系统调用来获知 I/O 是否完成，这种方式称为轮询(polling)。

举例理解

你早上去买现炸油条，你点单，点完后每隔一段时间询问店家有没有做好，期间你可以做点其它事情。（你就是应用级别，店家就是操作系统级别，应用可以做其它事情并通过轮询来看操作系统是否完成）

----------------------------------------------

什么是多路复用IO？

系统调用可能是由多个任务组成的，所以可以拆成多个任务，这就是多路复用。

举例理解

你早上去买现炸油条，点单收钱和炸油条原来都是由一个人完成的，现在他成了瓶颈，所以专门找了个收银员下单收钱，他则专注在炸油条。（本质上炸油条是耗时的瓶颈，将他职责分离出不是瓶颈的部分，比如下单收银，对应到系统级别也时一样的意思）

----------------------------------------------------------

什么是信号驱动IO？

应用进程使用 sigaction 系统调用，内核立即返回，应用进程可以继续执行，也就是说等待数据阶段应用进程是非阻塞的。内核在数据到达时向应用进程发送 SIGIO 信号，应用进程收到之后在信号处理程序中调用 recvfrom 将数据从内核复制到应用进程中。

相比于非阻塞式 I/O 的轮询方式，信号驱动 I/O 的 CPU 利用率更高。

举例理解

你早上去买现炸油条，门口排队的人多，现在引入了一个叫号系统，点完单后你就可以做自己的事情了，然后等叫号就去拿就可以了。（所以不用再去自己频繁跑去问有没有做好了）

--------------------------------------------------------

什么是异步IO？

相对于同步IO，异步IO不是顺序执行。用户进程进行aio\_read系统调用之后，无论内核数据是否准备好，都会直接返回给用户进程，然后用户态进程可以去做别的事情。等到socket数据准备好了，内核直接复制数据给进程，然后从内核向进程发送通知。IO两个阶段，进程都是非阻塞的。

举例理解

你早上去买现炸油条， 不用去排队了，打开美团外卖下单，然后做其它事，一会外卖自己送上门。(你就是应用级别，店家就是操作系统级别, 应用无需阻塞，这就是非阻塞；系统还可能在处理中，但是立刻响应了应用，这就是异步)