NBD 协议

1. 两个阶段

Nbd协议有两个阶段：

（1）握手阶段（在这个阶段，客户端与服务端建立连接，并对服务端export的设备进行参数以及协议自身的选项的协商）

（2）数据传输阶段（客户端与服务端进行数据传输）。

Linux主机上，第（1）个阶段在用户态完成，第（2）个阶段在内核态实现。

握手阶段，客户端至少需要执行两个ioctl命令：

ioctl(nbd, NBD\_SET\_SOCK, sock)

该命令主要将sock拷贝到内核，并设置到客户端内核nbd设备的数据结构上，以便第二阶段中，客户端内核直接与服务端进行数据传输。参数nbd是client端打开的文件描述符（一般对应/dev/nbdx设备），sock是到服务端的socket连接。

ioctl(nbd, NBD\_DO\_IT)

该命令启动客户端内核的数据循环，不会返回，直到客户端断开与服务端的连接才返回。

握手阶段还会视客户端启动参数情况，调用其它的ioctl命令，设置内核选项参数。

2. 数据传输阶段的两类消息

在数据传输阶段，nbd支持两类消息格式：

1. request请求(nbd客户端发送)

request请求有以下五种：

* NBD\_CMD\_READ

读请求

* NBD\_CMD\_WRITE

写请求

* NBD\_CMD\_DISC

断开连接

* NBD\_CMD\_FLUSH

刷盘请求

* NBD\_CMD\_TRIM

丢弃数据（discard）

Request请求的头部（header）格式如下：

32位的magic数字

32位的域字段（表示请求类型：见上述五种）

64位的handle（’handle’）

64位的偏移（请求的偏移地址，from）

32位的长度(请求操作的长度，len)

例如：写请求（NBD\_CMD\_WRITE）中，header后紧跟的就是len长度字节的数据。

NBD\_CMD\_FLUSH请求中，from和len被设置为0，表示整个设备。

1. response回复（nbd服务器发送）

Response回复的头部格式如下：

32位的magic数字

32位的error code

64位handle（'handle’）

例如：对读请求的回复中，error设置为0，回复的头部后面紧跟客户端请求长度的数据。