<https://blog.csdn.net/david_xtd/article/details/7087843>

**名称**

accept()

接收一个套接字中已建立的连接

#### 使用格式

#include <[sys/types.h](http://linux.die.net/include/sys/types.h)>

#include <sys/socket.h>

**int** accept(**int** sockfd,**struct sockaddr** \*addr,**socklen\_t** \*addrlen);

#### 功能参数描述

accept()系统调用主要用在基于连接的套接字类型，比如SOCK\_STREAM和SOCK\_SEQPACKET。它**提取出所监听套接字的等待连接队列中第一个连接请求**，**创建一个新的套接字，并返回指向该套接字的文件描述符**。新建立的套接字不在监听状态，原来所监听的套接字也不受该系统调用的影响。

备注：新建立的套接字准备发送send()和接收数据recv()。

参数：

**sockfd**,    利用系统调用socket()建立的套接字描述符，通过bind()绑定到一个本地地址(一般为服务器的套接字)，并且**通过listen()一直在监听连接**；

**addr**,    指向struct sockaddr的指针，该结构用通讯层服务器对等套接字的地址(一般为客户端地址)填写，返回地址addr的确切格式由套接字的地址类别(比如TCP或UDP)决定；若addr为NULL，没有有效地址填写，这种情况下，addrlen也不使用，应该置为NULL；

备注：addr是个指向局部数据结构sockaddr\_in的指针，这就是要求接入的信息本地的套接字(地址和指针)。

***addrlen***,    一个值结果参数，调用函数必须初始化为包含addr所指向结构大小的数值，函数返回时包含对等地址(一般为服务器地址)的实际数值；

备注：addrlen是个局部整形变量，设置为sizeof(struct   sockaddr\_in)。

如果队列中没有等待的连接，套接字也没有被标记为Non-blocking，accept()会**阻塞调用**函数**直到连接出现**；如果套接字被标记为Non-blocking，队列中也没有等待的连接，accept()返回错误**EAGAIN**或**EWOULDBLOCK**。

备注：一般来说，实现时accept()为阻塞函数，当监听socket调用accept()时，它先到自己的receive\_buf中查看是否有连接数据包；

若有，把数据拷贝出来，删掉接收到的数据包，创建新的socket与客户发来的地址建立连接；

若没有，就阻塞等待；

为了在套接字中有到来的连接时得到通知，可以使用select()或poll()。当尝试建立新连接时，系统发送一个可读事件，然后调用accept()为该连接获取套接字。另一种方法是，当套接字中有连接到来时设定套接字发送SIGIO信号。

#### 返回值

成功时，返回非负整数，该整数是接收到套接字的描述符；出错时，返回－1，相应地设定全局变量errno。

#### 错误处理

Linux下，accept()把已等待的网络错误传给新建立的连接，当作是accept()返回的错误。这于其他的BSD实现是不同的。为了可靠运行，应该在accept()之后检测协议已定义的一些网络错误，并把这些错误当作**EAGAIN**并重试。对于TCP／IP协议来说，主要有错误：**ENETDOWN**,**EPROTO**,**ENOPROTOOPT**,**EHOSTDOWN**,**ENONET**,**EHOSTUNREACH**,**EOPNOTSUPP**和**ENETUNREACH**。

原文连接：<http://linux.die.net/man/2/accept>