**《通信协议软件设计》课程作业三**

Jayce 杰斯

1. **请解释说明下列套接字函数的三个入口参数的作用。**

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

1. domain参数：

用于设置网络通信的域，函数根据这个参数选择通信协议的族。通信协议族在文件sys/socket.h中定义

|  |  |  |  |
| --- | --- | --- | --- |
| 名称 | 含义 | 名称 | 含义 |
| PF\_UNIX,PF\_LOCAL | 本地通讯 | PF\_X25 | ITU-T X25 / ISO-8208协议 |
| AF\_INET,PF\_INET | IPv4 Internet协议 | PF\_AX25 | Amateur radio AX.25 |
| PF\_INET6 | IPv6 Internet协议 | PF\_ATMPVC | 原始ATM PVC访问 |
| PF\_IPX | IPX-Novell协议 | PF\_APPLETALK | Appletalk |
| PF\_NETLINK | 内核用户界面设备 | PF\_PACKET | 底层包访问 |

1. type参数：

type用于设置套接字通信的类型，主要有**SOCKET\_STREAM（流式套接字）**、SOCK——DGRAM(数据包套接字)等。

|  |  |
| --- | --- |
| 名称 | 含义 |
| SOCK\_STREAM | Tcp连接，序列化、可靠、双向连接的字节流。支持带外数据传输 |
| SOCK\_DGRAM | 支持UDP连接（无连接状态的消息） |
| SOCK\_SEQPACKET | 序列化包,提供一个序列化、可靠、双向的基本连接的数据传输通道，数据长度定常。每次调用读系统调用时数据需要将全部数据读出 |
| SOCK\_RAW | RAW类型，提供原始网络协议访问 |
| SOCK\_RDM | 提供可靠的数据报文，不过可能数据会有乱序 |

1. protocol参数：

用于制定某个协议的特定类型，即**type**类型中的某个类型。通常某协议中只有一种特定类型，这样protocol参数仅能设置为0；但是有些协议有多种特定的类型，就需要设置这个参数来选择特定的类型。

* 类型为SOCK\_STREAM的套接字表示一个双向的字节流，与管道类似。流式的套接字在进行数据收发之前必须已经连接，连接使用connect()函数进行。一旦连接，可以使用read()或者write()函数进行数据的传输。流式通信方式保证数据不会丢失或者重复接收，当数据在一段时间内任然没有接受完毕，可以认为这个连接已经死掉。
* SOCK\_DGRAM和SOCK\_RAW 这个两种套接字可以使用函数sendto()来发送数据，使用recvfrom()函数接受数据，recvfrom()接受来自制定IP地址的发送方的数据。
* SOCK\_PACKET是一种专用的数据包，它直接从设备驱动接受数据。

1. **在Linux系统中，当在Linux系统中添加一个新的传输层协议软件时，该传输层协议软件提供的功能怎么能够被应用程序通过Socket套接字函数访问到？**

根据用户socket()的参数，匹配注册的协议族，使用对应的协议。

内核的文件系统中每个文件具有一个inode，socket也有，socket结构体中的file指针与相应的文件关联起来。Socket结构体如下：

struct socket {

    socket\_state        state;

    short            type;

    unsigned long       flags;

    struct file      \*file;

    struct sock      \*sk;

    const struct proto\_ops  \*ops;

    struct socket\_wq    wq;

};

总体流程为：

1. 用户调用glibc等提供的socket(),此时会产生系统调用，生成一个与文件绑定的socket。

2. 通过给定的参数family在net\_families中查找对应的协议簇指针pf。Pf有对应协议族的create函数，如inet\_create。

3. 调用pf->create来初始化socket，将对应的sock与其绑定。

4. Inet中，inet\_create首先根据sock->type和protocal在inetsw[]数组中查找对应的协议，再将对应协议的操作函数赋值给sock。

5. 之后就可以使用相应协议的功能了。

1. **在Linux系统中，当在Linux系统中添加一个新的传输层协议软件时，当IP协议接收到该传输层协议的IP分组后，IP协议软件怎么能够判断出该IP分组中携带了该传输层数据包？IP协议软件怎么能够将该传输层数据包提交给该传输层协议软件进行处理？**

需要提前注册传输层接口。

在传输层利用net\_protocal结构体来向ip层注册相应的传输层协议接口，当ip层收到数据后就会根据不同的协议调用对应的接口函数向上层传输数据。

对于ipv4来说应当在inet\_init中调用inet\_add\_protocol()进行注册。在inet\_add\_protocol()中将net\_protocal注册到inet\_protos[]这个数组中。同时inet\_init需要注册处理函数ip\_rcv，其作用是对数据包做各种合法性检查：协议头长度、协议版本、数据包长度、校验和等。然后调用网络过滤子系统的回调函数对数据包进行安全过滤。

如果数据包通过管理系统则调用ip\_rcv\_finish()对数据包进行实际处理。Ip\_rcv\_finish()中确定数据包是转发还是接收，如果是要接收的数据就调用ip\_local\_deliver()函数。此函数先重组数据包，接着调用ip\_local\_deliver\_finish()函数向上层协议分发。

Ip\_local\_deliver\_finish()首先从数据包读取protocol，然后在先前注册的inet\_protos[]数组中找到对应协议。找到后通过net\_protocol中指定的handler来将数据包向相应的传输层传送。例如如果是tcp\_protocol，则handler为tcp\_v4\_rcv，如果是udp\_protocol，则handler为udp\_rcv。通过handler来进入具体协议中处理。

1. **请说明Linux系统网络子系统中的数据结构sk\_buff的作用。**

sk\_buff:套接字缓冲区,用来在linux网络子系统中各层之间数据传递，起到了“神经中枢”的作用。

当发送数据包时，linux内核的网络模块必须建立一个包含要传输的数据包的sk\_buff,然后将sk\_buff传递给下一层，各层在 sk\_buff 中添加不同的协议头，直到交给网络设备发送。

同样，当接收数据包时，网络设备从物理媒介层接收到数据后，他必须将接收到的数据转换为sk\_buff，并传递给上层，各层剥去相应的协议头后直到交给用户。