# 实验六 网络编程

## **一、实验目的**

1、握端口及Socket的使用；

2、掌握面向连接的TCP编程；

3、掌握面向非连接的UDP编程；

4、掌握Ｉ/O多路利用的控制；

5、掌握复杂网络程序的实现。

## 二、实验目标与要求

1、网络参数的获取；

2、TCP协议下网络通信程序设计；

3、客户端与服务器端网络通信程序设计；

4、文件传送的网络程序设计；

5、UDP方式的网络程序设计。

## 三、实验工具与准备

计算机PC机，Linux Redhat 9.0或者其他的linux操作系统

## **四、实验步骤与操作指导**

**任务一：程序设计**

编写一个基于TCP协议的网络通信程序，要求服务器通过socket连接后，并要求输入用户，判断为tim时，才向客户端发送字符串“ Hello，you are connected!”。在服务器上显示客户端的IP地址或域名。

**任务二：调试程序**

server.c的作用是从客户端读字符，然后将每个字符转换为大写并回送给客户端。

/\* server.c \*/

#include <stdio.h>

#include <stdlib.h>

#include <string.h>

#include <unistd.h>

#include <sys/socket.h>

#include <netinet/in.h>

#define MAXLINE 80

#define SERV\_PORT 8000

int main(void)

{

struct sockaddr\_in servaddr, cliaddr;

socklen\_t cliaddr\_len;

int listenfd, connfd;

char buf[MAXLINE];

char str[INET\_ADDRSTRLEN];

int i, n;

listenfd = socket(AF\_INET, SOCK\_STREAM, 0);

bzero(&servaddr, sizeof(servaddr));

servaddr.sin\_family = AF\_INET;

servaddr.sin\_addr.s\_addr = htonl(INADDR\_ANY);

servaddr.sin\_port = htons(SERV\_PORT);

bind(listenfd, (struct sockaddr \*)&servaddr, sizeof(servaddr));

listen(listenfd, 20);

printf("Accepting connections ...\n");

while (1)

{

cliaddr\_len = sizeof(cliaddr);

connfd = accept(listenfd,

(struct sockaddr \*)&cliaddr, &cliaddr\_len);

n = read(connfd, buf, MAXLINE);

printf("received from %s at PORT %d\n",

inet\_ntop(AF\_INET, &cliaddr.sin\_addr, str, sizeof(str)),

ntohs(cliaddr.sin\_port));

for (i = 0; i < n; i++)

buf[i] = toupper(buf[i]);

write(connfd, buf, n);

close(connfd);

}

}

client.c的作用是从命令行参数中获得一个字符串发给服务器，然后接收服务器返回的字符串并打印。

/\* client.c \*/

#include <stdio.h>

#include <stdlib.h>

#include <string.h>

#include <unistd.h>

#include <sys/socket.h>

#include <netinet/in.h>

#define MAXLINE 80

#define SERV\_PORT 8000

int main(int argc, char \*argv[])

{

struct sockaddr\_in servaddr;

char buf[MAXLINE];

int sockfd, n;

char \*str;

if (argc != 2)

{

fputs("usage: ./client message\n", stderr);

exit(1);

}

str = argv[1];

sockfd = socket(AF\_INET, SOCK\_STREAM, 0);

bzero(&servaddr, sizeof(servaddr));

servaddr.sin\_family = AF\_INET;

inet\_pton(AF\_INET, "127.0.0.1", &servaddr.sin\_addr);

servaddr.sin\_port = htons(SERV\_PORT);

connect(sockfd, (struct sockaddr \*)&servaddr, sizeof(servaddr));

write(sockfd, str, strlen(str));

n = read(sockfd, buf, MAXLINE);

printf("Response from server:\n");

write(STDOUT\_FILENO, buf, n);

close(sockfd);

return 0;

}

问题

（1）编辑、编译程序，用以下方式执行程序：

在服务器运行：

#./server

# netstat -apn|grep 8000

b.在客户端运行：

#./client abcd

回到server所在的终端，看看server的输出，再次运行服务器端程序：

#./server

客户端的端口号是自动分配的。现在把客户端所连接的服务器IP改为其它主机的IP，试试两台主机的通讯。将Client端的地址改为外网地址后，运行结果如何？

（2）在客户端的connect()代码之后插一个while(1);死循环，使客户端和服务器都处于连接中的状态，用netstat命令查看。

在服务器运行：

# ./server &

# Accepting connections ...

b.在客户端运行：

#./client abcd &

# netstat -apn|grep 8000

**任务三：程序设计**

编写一个用于两人之间的通讯程序：双方都可以从终端上输入一个字符串，程序将时间信息加入该字符串，然后通过UDP的方式发送到对方。程序在运行时须输入本地、目标的IP地址和端口号。