

**计算机学院实验报告**

**班 级 网络工程1801 姓 名 吴斌 学 号 18408020129**

**时 间 2020.12.10 评 分 教师签名**

**课程名称 网络程序设计**

**1、实验名称； 广播与组播编程**

**2、实验目的；**

**通过在ubuntu上编程深入了解广播和组播的定义；以及掌握广播和组播分别在linux上的编程实现。**

**3、实验原理；**

**广播：在一个局域网内部，所有的终端都能够收到数据包。使用广播的发送数据，会使处于同一个局域网内部的所有用户都必须接收到数据，用户不能拒绝。  
 注意：在使用广播和组播时，发送方和接收方都要处于同一个局域网内。因为广播和组播只能在一个局域网内部发送数据。广播和组播与一般网络通信有所不同，网络通信传输的是数据包；广播传输的数据叫广播包，组播传输的数据叫组播包。  
 原理: 发送方发送广播包到交换机/路由器，交换机识别到是一个广播包, 于是由交换机转发N份到局域网中的每一个终端。ip 地址=网络号 +主机号，主机号全部为1的ip地址称为广播地址。发送方将ip包的目的地址改为广播地址，路由器收到数据包发现其目的地址是广播地址，于是就将广播包转发给局域网内部的每一台电脑。  
在广播中，只有发送端（send）和接收端（receive）一律使用udp协议传送数据。  
 编写广播程序的步骤：**

1. **设置允许发送；**
2. **将目的地址改为广播地址；**
3. **端口号和接收方一致。**

**组播：在路由器中建立一个分组，感兴趣的成员加入该组，往改组发送组播包, 路由器识别到组播包，就转发给组内的每一个成员。通过组播ip识别每一个分组。将数据包的目的ip改为该分组的ip地址，就可以将数据包发送给该组内成员。**

**4、实验内容与结果；源码一：server广播.c**

**#include<sys/socket.h>**

**#include<arpa/inet.h>**

**#include<sys/ioctl.h>**

**#include<stdio.h>**

**#include<stdlib.h>**

**#include<string.h>**

**#include<time.h>**

**#include<unistd.h>**

**#define BUFLEN 255**

**void getcurtime(char \*curtime){**

**time\_t tm;**

**time(&tm);**

**snprintf(curtime,BUFLEN,"%s\n",ctime(&tm));**

**}**

**int main(int argc,char \*\*argv){**

**struct sockaddr\_in peeraddr;**

**int sockfd,on=1;**

**int num,i;**

**char msg[BUFLEN+1];**

**if(argc!=3){**

**printf("usage :%s<ip address><port>\n",argv[0]);**

**exit(0);**

**}**

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

**if(sockfd<0){**

**fprintf(stderr,"socket creating error in tserv.c\n");**

**exit(1);**

**}**

**setsockopt(sockfd,SOL\_SOCKET,SO\_BROADCAST,&on,sizeof(int));**

**memset(&peeraddr,0,sizeof(struct sockaddr\_in));**

**peeraddr.sin\_family=AF\_INET;**

**if(inet\_pton(AF\_INET,argv[1],&peeraddr.sin\_addr)<=0){**

**printf("wrong dest ip address\n");**

**exit(0);**

**}**

**peeraddr.sin\_port=htons(atoi(argv[2]));**

**for(;;){**

**getcurtime(msg);**

**int a;**

**a=sendto(sockfd,msg,strlen(msg),0,(struct sockadr \*)&peeraddr,sizeof(struct sockaddr\_in));**

**printf("%d",a);**

**fflush(stdout);**

**sleep(3);**

**}**

**}**

**源码二：client广播.c**

**#include<sys/types.h>**

**#include<sys/socket.h>**

**#include<arpa/inet.h>**

**#include<stdio.h>**

**#include<stdlib.h>**

**#include<string.h>**

**#include<time.h>**

**#include<unistd.h>**

**#define BUFLEN 255**

**int main(int argc,char \*\*argv){**

**struct sockaddr\_in localaddr;**

**int sockfd,n;**

**char msg[BUFLEN+1];**

**if(argc!=2){**

**printf("usage:%s<port>\n",argv[0]);**

**exit(0);**

**}**

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

**if(sockfd<0){**

**fprintf(stderr,"socket creating error in tcli.c\n");**

**exit(1);**

**}**

**memset(&localaddr,0,sizeof(struct sockaddr\_in));**

**localaddr.sin\_port=htons(atoi(argv[1]));**

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

**int opt=SO\_REUSEADDR;**

**setsockopt(sockfd,SOL\_SOCKET,SO\_REUSEADDR,&opt,sizeof(opt));**

**if(bind(sockfd,(struct sockaddr \*)&localaddr,sizeof(struct sockaddr\_in))<0){**

**fprintf(stderr,"bind error in tcli.c\n");**

**exit(2);**

**}**

**n=read(sockfd,msg,BUFLEN);**

**if(n==-1){**

**fprintf(stderr,"read error in tcli.c\n");**

**exit(3);**

**}**

**else{**

**msg[n]=0;**

**printf("%s",msg);**

**}**

**}**

**实验源码三：server组播.c**

**#include<sys/socket.h>**

**#include<arpa/inet.h>**

**#include<sys/ioctl.h>**

**#include<stdio.h>**

**#include<stdlib.h>**

**#include<string.h>**

**#include<time.h>**

**#include<unistd.h>**

**#define BUFLEN 255**

**void getcurtime(char \*curtime){**

**time\_t tm;**

**time(&tm);**

**snprintf(curtime,BUFLEN,"%s\n",ctime(&tm));**

**}**

**int main(int argc,char \*\*argv){**

**struct sockaddr\_in peeraddr;**

**int sockfd,on=1;**

**int num,i;**

**char msg[BUFLEN+1];**

**if(argc!=3){**

**printf("usage :%s<ip address><port>\n",argv[0]);**

**exit(0);}**

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

**if(sockfd<0){**

**fprintf(stderr,"socket creating error in tserv.c\n");**

**exit(1);}**

**setsockopt(sockfd,SOL\_SOCKET,SO\_BROADCAST,&on,sizeof(int));**

**memset(&peeraddr,0,sizeof(struct sockaddr\_in));**

**peeraddr.sin\_family=AF\_INET;**

**if(inet\_pton(AF\_INET,argv[1],&peeraddr.sin\_addr)<=0){**

**printf("wrong dest ip address\n");**

**exit(0);}**

**peeraddr.sin\_port=htons(atoi(argv[2]));**

**for(;;){**

**getcurtime(msg);**

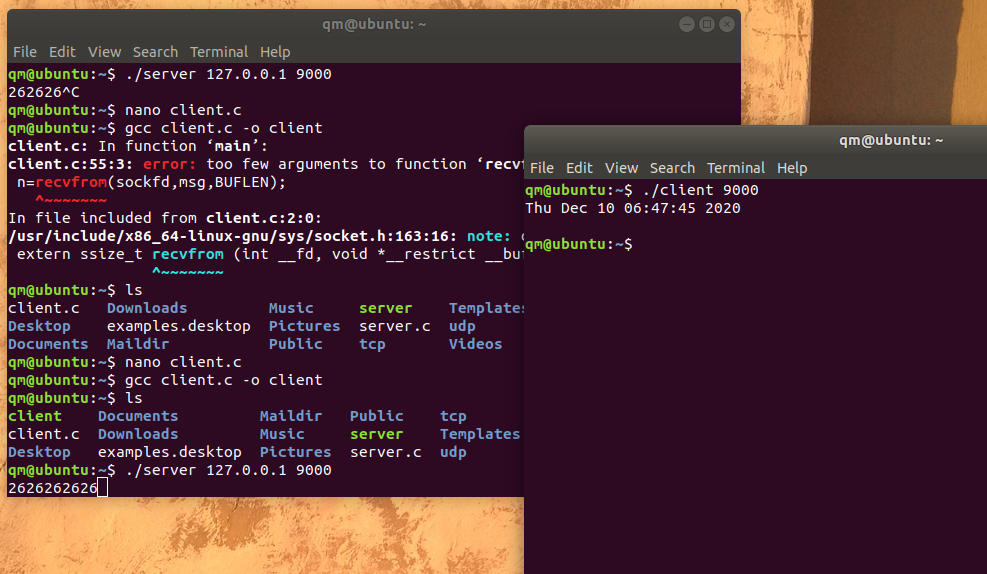
**int a;**

**a=sendto(sockfd,msg,strlen(msg),0,(struct sockadr \*)&peeraddr,sizeof(struct sockaddr\_in));**

**printf("%d",a);**

**fflush(stdout);**

**sleep(3);}}**



**5、总结与讨论；**

**通过在ubuntu上编程深入了解了广播和组播的定义；并且掌握了广播和组播分别在linux上的编程实现。**