Server

//UDPServer.c

/\*

\* gcc -o server UDPServer.c

\* ./server

\*/

#include <arpa/inet.h>

#include <netinet/in.h>

#include <stdio.h>

#include <sys/types.h>

#include <sys/socket.h>

#include <unistd.h>

#include <stdlib.h>

#include <string.h>

#define BUFLEN 512

#define PORT 9930

void err(char \*str)

{

perror(str);

exit(1);

}

int main(void)

{

struct sockaddr\_in my\_addr, cli\_addr;

int sockfd, i;

socklen\_t slen=sizeof(cli\_addr);

char buf[BUFLEN];

if ((sockfd = socket(AF\_INET, SOCK\_DGRAM, IPPROTO\_UDP))==-1)

err("socket");

else

printf("Server : Socket() successful\n");

bzero(&my\_addr, sizeof(my\_addr));

my\_addr.sin\_family = AF\_INET;

my\_addr.sin\_port = htons(PORT);

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

if (bind(sockfd, (struct sockaddr\* ) &my\_addr, sizeof(my\_addr))==-1)

err("bind");

else

printf("Server : bind() successful\n");

while(1)

{

if (recvfrom(sockfd, buf, BUFLEN, 0, (struct sockaddr\*)&cli\_addr, &slen)==-1)

err("recvfrom()");

printf("Received packet from %s:%d\nData: %s\n\n",

inet\_ntoa(cli\_addr.sin\_addr), ntohs(cli\_addr.sin\_port), buf);

}

close(sockfd);

return 0;

}

CLIENT

//UDPClient.c

/\*

\* gcc -o client UDPClient.c

\* ./client <server-ip>

\*/

#include <arpa/inet.h>

#include <netinet/in.h>

#include <stdio.h>

#include <sys/types.h>

#include <sys/socket.h>

#include <unistd.h>

#include <stdlib.h>

#include <string.h>

#define BUFLEN 512

#define PORT 9930

void err(char \*s)

{

perror(s);

exit(1);

}

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

{

struct sockaddr\_in serv\_addr;

int sockfd, i, slen=sizeof(serv\_addr);

char buf[BUFLEN];

if(argc != 2)

{

printf("Usage : %s <Server-IP>\n",argv[0]);

exit(0);

}

if ((sockfd = socket(AF\_INET, SOCK\_DGRAM, IPPROTO\_UDP))==-1)

err("socket");

bzero(&serv\_addr, sizeof(serv\_addr));

serv\_addr.sin\_family = AF\_INET;

serv\_addr.sin\_port = htons(PORT);

if (inet\_aton(argv[1], &serv\_addr.sin\_addr)==0)

{

fprintf(stderr, "inet\_aton() failed\n");

exit(1);

}

while(1)

{

printf("\nEnter data to send(Type exit and press enter to exit) : ");

scanf("%[^\n]",buf);

getchar();

if(strcmp(buf,"exit") == 0)

exit(0);

if (sendto(sockfd, buf, BUFLEN, 0, (struct sockaddr\*)&serv\_addr, slen)==-1)

err("sendto()");

}

close(sockfd);

return 0;

}

