//Design using poll server application to multiplex TCP and UDP requests for converting a given text into upper case.

//server program

#include<stdio.h>

#include<netinet/in.h>

#include<sys/types.h>

#include<string.h>

#include<stdlib.h>

#include<sys/socket.h>

#include<sys/select.h>

#include<unistd.h>

#define MAXLINE 20

#define SERV\_PORT 8114

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

{

int i,j,maxi,maxfd,listenfd,connfd,sockfd;

int nready,client[FD\_SETSIZE];

ssize\_t n;

fd\_set rset,allset;

char line[MAXLINE];

socklen\_t clilen;

struct sockaddr\_in cliaddr,servaddr;

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,1);

maxfd=listenfd;

maxi=-1;

for(i=0;i<FD\_SETSIZE;i++)

client[i]=-1;

FD\_ZERO(&allset);

FD\_SET(listenfd,&allset);

for(;;)

{

rset=allset;

nready=select(maxfd+1,&rset,NULL,NULL,NULL);

if(FD\_ISSET(listenfd,&rset))

{

clilen=sizeof(cliaddr);

connfd=accept(listenfd,(struct sockaddr \*)&cliaddr,&clilen);

for(i=0;i<FD\_SETSIZE;i++)

if(client[i]<0)

{

client[i]=connfd;

break;

}

if(i==FD\_SETSIZE)

{

printf("too many clients");

exit(0);

}

FD\_SET(connfd,&allset);

if(connfd>maxfd)

maxfd=connfd;

if(i>maxi)

maxi=i;

if(--nready<=0)

continue;

}

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

{

if((sockfd=client[i])<0)

continue;

if(FD\_ISSET(sockfd,&rset))

{

if((n=read(sockfd,line,MAXLINE))==0)

{

close(sockfd);

FD\_CLR(sockfd,&allset);

client[i]=-1;

}

else

{

printf("line received from client:%s\n",line);

for(j=0;line[j]!='\0';j++)

line[j]=toupper(line[j]);

write(sockfd,line,MAXLINE);

}

if(--nready<=0)

break;

}

}

}

}