**//concurrent server.c program**

#include"header.h"

struct sockaddr\_in c\_addr;

void sort\_data(char \*s)

{

char t;

int i,j;

for(i=0;i<strlen(s)-1;i++)

{

for(j=0;j<strlen(s)-1-i;j++)

{

if(s[j]>s[j+1])

{

t=s[j];

s[j]=s[j+1];

s[j+1]=t;

}

}

}

}

void client\_info()

{

printf(\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\n");

printf("client port no:%d\n",ntohs(c\_addr.sin\_port));

printf("client ip\_address:%s\n",inet\_ntoa(c\_addr.sin\_addr));

printf("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\n");

}

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

{

if(argc!=3)

{

printf("usage: ./server port\_num ip\_address\n");

return 0;

}

int sfd,nsfd,len,b,l,i;

char s[20];

struct sockaddr\_in s\_addr;

/\*\*\*\*\*\*\*\*\*\* socket() \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

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

if(sfd<0)

{

perror("Socket:");

return 0;

}

/\*\*\*\*\*\*\*\*\*\*\* bind() \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

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

s\_addr.sin\_port= htons(atoi(argv[1]));

s\_addr.sin\_addr.s\_addr=inet\_addr(argv[2]);

len=sizeof(s\_addr);

b= bind(sfd,(struct sockaddr\*) &s\_addr,len);

if(b<0)

{

perror("Bind:");

return 0;

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\* listen() \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

l= listen(sfd,5);

if(l<0)

{

perror("Listen:");

return 0;

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* accept() \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

while(1)

{

printf("waiting for the client incoming request\n" );

nsfd=accept(sfd,(struct sockaddr\*)&c\_addr,&len);

if(nsfd<0)

{

perror("Accept:");

return 0;

}

printf("client request Accepted\n");

if(fork()==0)

{

client\_info();

while(1)

{

read(nsfd,s,sizeof(s));

if(strcmp(s,"bye")==0)

break;

sort\_data(s);

write(nsfd,s,strlen(s)+1);

}

}

}

close(sfd);

close(nsfd);

}