/\* --- Project 10 ---

--- Remote Calcy Project ---

--- BY Manpreet Singh Bhamra 104692811, Dikshay Palta 104644896, Jaspreet Singh 104658273, Mehak Grover 104663943, Monika 104706503----GROUP 10

--- Server accepting clients from android and unix command line interface ---

\*/

// Header files

#include<stdio.h>

#include<netinet/in.h>

#include<sys/types.h>

#include<sys/socket.h>

#include<netdb.h>

#include<stdlib.h>

#include<string.h>

#define MAX 80

//#define PORT 43454

#define SA struct sockaddr

void error(char \*msg) //function to display errors

{

perror(msg);

exit(1);

}

void calculate(int sockfd) //main function that perform airthmetic operation

{

char buff[MAX];

int n;

for(;;)

{

bzero(buff,MAX); //emptying the buffer

read(sockfd,buff,sizeof(buff)); //reading data from socket

printf("Expression From client: %s \n ",buff);

static int i,size1=1,size2=1,opp;

char \*op1,\*op2;

op1=malloc(size1);

op2=malloc(size1);

for(i=0;i<strlen(buff);i++)

{

if(buff[i]!='\*' & buff[i]!='/' & buff[i]!='+' & buff[i]!='-')

{

op1[i]=(char)buff[i];

size1++;

op1=realloc(op1,size1);

}

else

{

int j,s=0;

opp=i;

for(j=i+1;j<strlen(buff);j++)

{

op2[s]=(char)buff[j];

size2++;

op1=realloc(op1,size2);

s++;

}

break;

}

}

int opr1,opr2,result;

opr1=atoi(op1); //converting the string to integer

opr2=atoi(op2);

if(buff[opp]=='\*')

{

result=opr1\*opr2;

bzero(buff,MAX);

snprintf(buff,10,"%d",result);

printf("Result: %s",buff);

write(sockfd,buff,sizeof(buff)); //writing the result onto buffer

fprintf(stderr,"sending result from server : %s\n : ",buff);

}

else if(buff[opp]=='/')

{

result=opr1/opr2;

bzero(buff,MAX);

snprintf(buff,10,"%d",result);

write(sockfd,buff,sizeof(buff)); //writing the result onto buffer

fprintf(stderr,"sending result from server.Result : %s :\n ",buff);

}

else if(buff[opp]=='+')

{

result=opr1+opr2;

bzero(buff,MAX);

snprintf(buff,10,"%d",result);

write(sockfd,buff,sizeof(buff)); //writing the result onto buffer

fprintf(stderr,"sending result from server.Result : %s : \n",buff);

}

else if(buff[opp]=='-')

{

result=opr1-opr2;

bzero(buff,MAX);

snprintf(buff,10,"%d",result);

write(sockfd,buff,sizeof(buff)); //writing the result onto buffer

fprintf(stderr,"sending result from server.Result : %s : \n",buff);

}

else

{

write(sockfd,"Wrong Expression try it again\n",29);

}

if(strncmp("exit",buff,4)==0)

{

printf("Server Exit...\n");

break;

}

}

}

//Main Function

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

{

int sockfd,connfd,len,portno,pid;

struct sockaddr\_in servaddr,cli;

if(argc<2)

{

fprintf(stderr,"Error,no port provided\n");

exit(1);

}

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

if(sockfd==-1)

{

printf("socket creation failed...\n");

exit(0);

}

else

printf("Socket successfully created..\n");

bzero(&servaddr,sizeof(servaddr));

portno=atoi(argv[1]); //argv[1] is in char array so converting it into integer using atoi.

servaddr.sin\_family=AF\_INET;

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

servaddr.sin\_port=htons(portno);

if((bind(sockfd,(SA\*)&servaddr, sizeof(servaddr)))!=0) // binding the address with the socket

{

printf("socket bind failed...\n");

exit(0);

}

else

printf("Socket successfully binded..\n");

if((listen(sockfd,5))!=0) //socket is ready to listen any new client request

{

printf("Listen failed...\n");

exit(0);

}

else

printf("Server listening..\n");

len=sizeof(cli);

for(;;)

{

connfd=accept(sockfd,(SA \*)&cli,&len);//socket will going to accept any new client request

if(connfd<0)

{

printf("server acccept failed...\n");

exit(0);

}

else

printf("server acccept the client...\n");

if((pid==fork())==0)

calculate(connfd);

else

close (connfd);

}

close(sockfd);//closing the socket created

}