EX NO: 5

21/09/2021

**ITERATIVE APPLICATION**

**AIM:**

To create a TCP/IP iterative server program to count number of words, characters and prepositions in the file name mentioned by client and send to it the client.

**ALGORITHM:  
SERVER:**

1. Include header files, initialize the required variables and specify the family, protocol, IP address and port number.
2. Create a socket using socket() function.
3. Bind the IP address and port number and listen iteratively to the client’s request for connection.
4. Count the number of words, characters and prepositions in the file name mentioned by the client.
5. send the counted values to the client.

**CLIENT:**

1. Include header files, initialize the required variables and specify the family, protocol, IP address and port number.
2. Create a socket using socket() function.
3. Call the connect() function.
4. Scan the input message(file name) entered by the client.
5. Display the counted values received from the server.

**PROGRAM:**

**SERVER:**

#include<stdio.h>

#include<netdb.h>

#include<string.h>

#include<stdlib.h>

#include<sys/types.h>

#include<netinet/in.h>

#include <sys/socket.h>

#include <netinet/in.h>

#include <arpa/inet.h>

char \*inet\_ntoa(struct in\_addr in);

void main(){

int port=8080;

int sockfd,connfd,clilen,bno;

struct sockaddr\_in server\_addr,client\_addr;

char buffer[1024];

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

if(sockfd<0){

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

exit(1);

}

printf("Server connected successfully...\n");

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

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

server\_addr.sin\_port = htons(port);

bno=bind(sockfd,(struct sockaddr\*)&server\_addr,sizeof(server\_addr));

if(bno<0){

printf("Binding error...\n");

exit(1);

}

printf("Bind successful...\n");

if(listen(sockfd,5)==0)

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

clilen=sizeof(client\_addr);

while(1){

printf("\n...Server waiting for client connection...\n");

connfd=accept(sockfd,(struct sockaddr\*)&client\_addr,&clilen);

if(connfd>0){

printf("\nClient accepted...");

}

while(1){

bzero(buffer,1024);

recv(connfd,buffer,sizeof(buffer),0);

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

break;

int an=0,the=0,charac=0,word=0,i,message[4];

for(i=0;i<sizeof(buffer);i++){

if(i<(sizeof(buffer)-1) && (buffer[i]=='a' || buffer[i]== 'A') && buffer[i+1]=='n' && buffer[i+2]==' ')

an++;

else if(i<(sizeof(buffer)-2) && (buffer[i]=='T' || buffer[i]=='t' ) && buffer[i+1]=='h' && buffer[i+2]=='e' && buffer[i+3]==' ')

the++;

else if(i<(sizeof(buffer)-1) && buffer[i+1]==' ')

word++;

if(buffer[i]!='\0 ')

charac++;

}

printf("\n\nTotal number of words: %d\nTotal number of characters: %d\nTotal number of Preposition:\n\tCount of AN: %d\n\tCount of THE: %d\n",word,charac,an,the);

bzero(message,4);

message[0]=charac;

message[1]=word;

message[2]=an;

message[3]=the;

send(connfd,message,sizeof(message),0);

}

printf("\nClient exit...\n");

}

}

**CLIENT:**

#include<stdio.h>

#include<stdlib.h>

#include<string.h>

#include<sys/socket.h>

#include<sys/types.h>

#include<netinet/in.h>

#include<netdb.h>

#include <arpa/inet.h>

char \*inet\_ntoa(struct in\_addr in);

void main(){

int port=8080,message[4];;

char buffer[1024];

int connfd;

struct sockaddr\_in server\_addr;

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

if(connfd<0){

printf("Error in client connection...\n");

exit(1);

}

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

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

server\_addr.sin\_port = htons(port);

if (connect(connfd,(struct sockaddr\*)&server\_addr,sizeof(server\_addr))>0){

printf("Server connection failed...\n");

exit(0);

}

printf("Connected to the server...\n");

FILE \*fp;

bzero(message,4);

while(1){

bzero(buffer,1024);

printf("Enter the file name: ");

gets(buffer);

if(strncmp(buffer,"exit",4)==0){

send(connfd,buffer,sizeof(buffer),0);

break;

}

fp=fopen(buffer,"r");

if(fp==NULL){

puts("Cannot open the file!");

exit(1);

}

bzero(buffer,1024);

char input[1024];

while(fgets(input,1024,fp))

strcat(buffer,input);

send(connfd,buffer,sizeof(buffer),0);

fclose(fp);

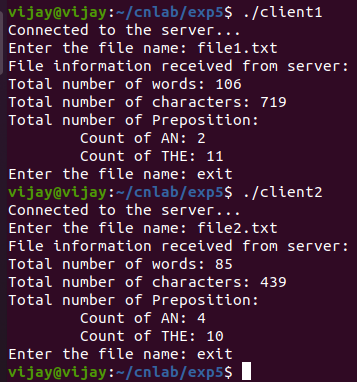
bzero(message,4);

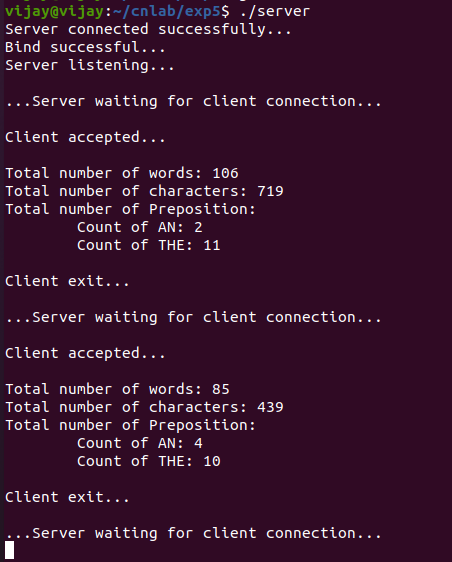
recv(connfd,message,sizeof(message),0);

printf("File information received from server: \nTotal number of words: %d\nTotal number of characters: %d\nTotal number of Preposition:\n\tCount of AN: %d\n\tCount of THE: %d\n",message[1],message[0],message[2],message[3]);

}

}

**SAMPLE OUTPUT:**



**RESULT:**

Hence the TCP/IP iterative server program was created to find the number of words, characters and prepositions in the file name provided by clients and the output was verified.