





Server.c

#include<sys/socket.h>

#include<sys/types.h>

#include<stdio.h>

#include<string.h>

#include<netdb.h>

#include<signal.h>

#include<stdlib.h>

#include<unistd.h>

#include<arpa/inet.h>

#include<stdarg.h>

#include<errno.h>

#include<fcntl.h>

#include<sys/time.h>

#include<sys/ioctl.h>

//Std HTTP Port. We set it to 18000 because most os are picky about who will use 80.

#define SERVER\_PORT 18000

//Buffer Length

#define MAXLINE 4096

#define SA struct sockaddr

//Lets define read write from config.txt

//#define portsize 5

int portsize=5;

char data[100];

char port[5];

int portno;

char \* parse(){

FILE \*file;

long int size;

file=fopen("config.txt","r");

//This part moves to end of file, then ftell calculates steps, then we move again to start of file.

fseek(file,0,SEEK\_END);

size=ftell(file);

fseek(file,0,SEEK\_SET);

fgets(data,size,file);

//reads port address

fseek(file,5,SEEK\_SET); //seeks to 5th place from first place

fread(port,1,5,file); // reads 5 characters of 1 byte each and sets it into port array buffer

//printf("%s",port);

sscanf(port,"%d",&portno);

fclose(file);

return data;

}

//argc is the size of what we write in terminal and argv is an array of strings of what we write

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

char \*s;

s=parse();

//printf("%i",portno);

int ad=portno;

int listenfd,connfd,n;

struct sockaddr\_in servaddr;

uint8\_t buff[MAXLINE+1];

uint8\_t recvline[MAXLINE+1];

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

if (listenfd <0){

printf("Error in socket creation ...");

}

bzero(&servaddr,sizeof(servaddr));

servaddr.sin\_family=AF\_INET;

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

servaddr.sin\_port = htons(ad);

int status=0;

status=bind(listenfd, (struct sockaddr \*) &servaddr,sizeof(servaddr));

if(status<0){

printf("Error in binding ...");

}

status=0;

status=listen(listenfd,10);

if(status<0){

printf("Error in listening ...");

}

for( ; ; ){

fflush(stdout);

connfd = accept(listenfd, (struct sockaddr \*) NULL,NULL);

memset(recvline,0,MAXLINE);

while ( (n=read(connfd,recvline,MAXLINE-1))>0){

fprintf(stdout, "\n%s", recvline);

if(recvline[n-1]=='\n'){

break;

}

memset(recvline,0,MAXLINE);

}

char msg[]="<html><head><title>Abdul Basit Bhat</title></head><body><h1>Name: Abdul Basit Bhat</h1><h1>Roll No: 102003121</h1><h1>Group: COE5</h1><br><a href='page2.html'>Next Page</a><h3>Resources used for this project</h3><a href='https://www.youtube.com/watch?v=esXw4bdaZkc'>Creating a Web Server using Socket Programming</a><br><img src='https://drive.google.com/uc?export=view&id=1hmrQhv7XDguWnziuAvH8hrDj8j2PQXWy'><br></body></html>";

long int len=sizeof(msg);

snprintf((char\*)buff,sizeof(buff),"HTTP/1.0 200 OK\nContent-Type:text/html\nContent-Length: len\n\n<html><head><title>Abdul Basit Bhat</title></head><body><h1>Name: Abdul Basit Bhat</h1><h1>Roll No: 102003121</h1><h1>Group: COE5</h1><br><a href='page2.html'>Next Page</a><h3>Resources used for this project</h3><a href='https://www.youtube.com/watch?v=esXw4bdaZkc'>Creating a Web Server using Socket Programming</a><br><img src='https://drive.google.com/uc?export=view&id=1hmrQhv7XDguWnziuAvH8hrDj8j2PQXWy'></body></html>");

// write(connfd,,len);

write(connfd,(char \*)buff,strlen((char \*)buff));

close(connfd);

}

//exit(0);

}







