

Roll No:- 102003087

Name :- Gaurav Pahwa

Group No:- 3COE4

Output

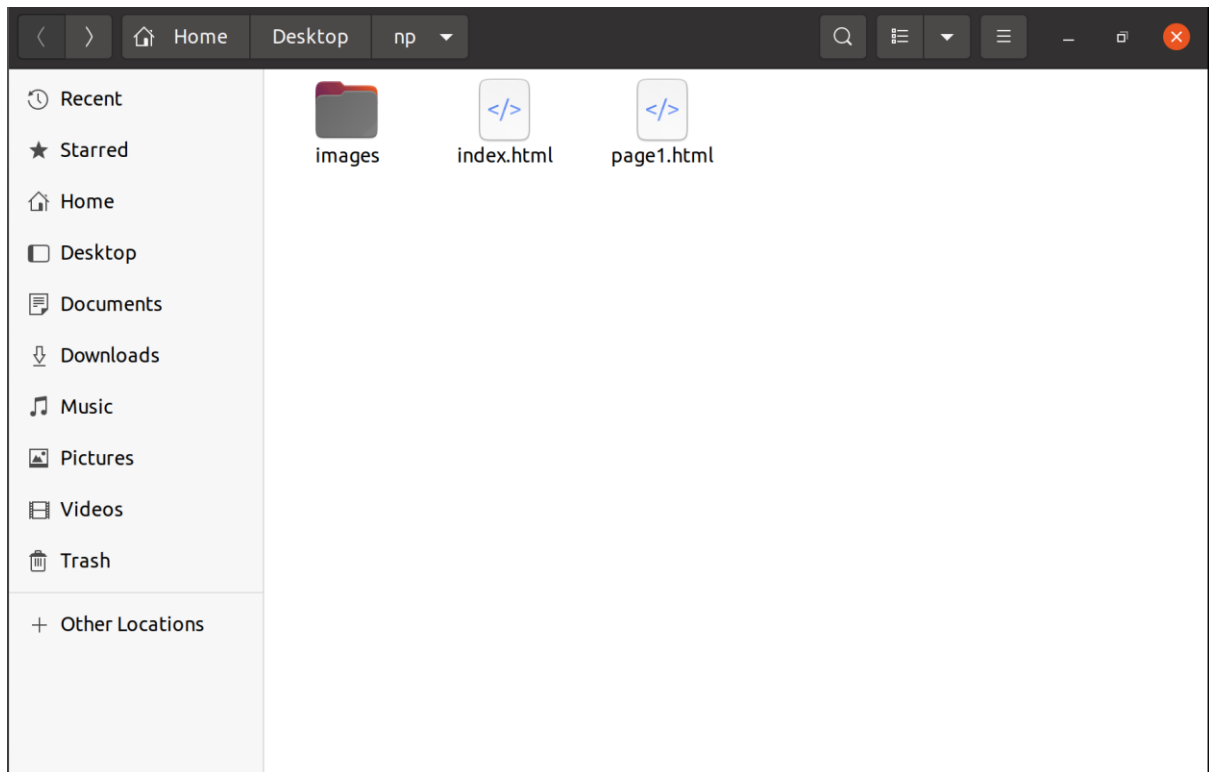
```
gauravv2204@ubuntu:~/Desktop$ gcc -o httpserverk httpserverk.c
gauravv2204@ubuntu:~/Desktop$ ./httpserverk

+++++++ Waiting for new connection +++++++
█
```

config.txt

```
1 port 22000
2 path np/
```

Plain Text ▼ Tab Width: 8 ▼ Ln 2, Col 9 ▼ INS



```
httpserverk.c  ×  *index.html  ×  *config.txt  ×
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4   <meta charset="UTF-8">
5   <meta http-equiv="X-UA-Compatible" content="IE=edge">
6   <meta name="viewport" content="width=device-width, initial-scale=1.0">
7   <title>Document</title>
8 </head>
9 <body>
10  <h1>Gaurav Pahwa</h1>
11  <h1>102003087</h1>
12  <h1>COE 4</h1>
13  
14  <a href="page1.html">Go to Page 1</a>
15 </body>
16 </html>
```

HTML ▾ Tab Width: 8 ▾ Ln 13, Col 34 ▾ INS

Gaurav Pahwa

102003087

COE 4



[Go to Page 1](#)

Status	Method	Domain	File	Initiator	Type	Transferred	Size	0 ms
200	GET			document	html	515 B	496 B	3 ms

3 requests 46.03 KB / 47.27 KB transferred Finish: 1.36 s DOMContentLoaded: 375 ms load: 1.08 s

This is Page 1

[Go to home page](#)

Status	Method	Domain	File	Initiator	Type	Transferred	Size	0 ms
200	GET	192.168.216.128:22000	page1.html	document	html	352 B	333 B	7 ms
	GET	192.168.216.128:22000	favicon.ico	FaviconLoader.jsm:186...		0 B	0 B	973 ms

2 requests 333 B / 352 B transferred Finish: 1.19 s load: 110 ms



Working Code

```
#include <stdio.h>
```

```
#include <sys/socket.h>
```

```
#include <unistd.h>
```

```
#include <stdlib.h>
```

```
#include <netinet/in.h>
```

```
#include <string.h>
```

```
void getadr(char buffer[],char fle[]){
```

```
    int i=5;
```

```
    while(buffer[i]!=' '){
```

```
        int len = strlen(fle);
```

```
        fle[len] = buffer[i];
```

```
        fle[len+1] = '\0';
```

```
        i++;
```

```
    }  
}
```

```
size_t file_size(const char* file_path) {  
    FILE* fin = fopen(file_path, "rb");  
    if (fin == NULL) {  
        printf("open %s failed.", file_path);  
        exit(0);  
    }  
}
```

```
    fseek(fin, 0L, SEEK_END);  
    size_t size = ftell(fin);  
    fclose(fin);  
    return size;  
}
```

```
void read_file(unsigned char* buffer, size_t size, const char*  
file_path) {  
    FILE* fin = fopen(file_path, "rb"); // r for read, b for binary  
    if (fin == NULL) {  
        printf("open %s failed.", file_path);  
        return;  
    }  
}
```

```

fseek(fin, 0L, SEEK_SET);
int ret = fread(buffer, 1, size, fin);
//printf("%d %d\n",strlen(buffer),ret); // read sizeof(buffer)
elements to our buffer
}

```

```

void setHttpHeader(char httpHeader[], char fle[], char PATH[])
{
    // File object to return

    if(strstr(fle,"images")){
        FILE *htmlData = fopen(fle, "rb");
        char line[100];
        char responseData[800000]="";
        int i=0;
        while (fgets(line,2, htmlData) != NULL) {
            strcat(responseData, line);
        }
        strcat(httpHeader, responseData);
        bzero(&responseData,sizeof(responseData));
    }
    else if(strlen(fle)==strlen(PATH)){

```

```

        strcat(fle,"index.html");
        FILE *htmlData = fopen(fle, "r");
        char line[100];
        char responseData[8000]="";
        while (fgets(line, 100, htmlData) != 0) {
            strcat(responseData, line);
        }
        strcat(httpHeader, responseData);
        bzero(&responseData,sizeof(responseData));
    }
    else{
        FILE *htmlData = fopen(fle, "r");
        char line[100];
        char responseData[8000]="";
        while (fgets(line, 100, htmlData) != 0) {
            strcat(responseData, line);
        }
        strcat(httpHeader, responseData);
        bzero(&responseData,sizeof(responseData));
    }
}

```

```

int main(int argc, char const *argv[])
{

```

```

int server_fd, new_socket; long valread;
struct sockaddr_in address;
int addrlen = sizeof(address);

// Only this line has been changed. Everything is same.

//char *hello = "HTTP/1.1 200 OK\nContent-Type:
text/plain\nContent-Length: 12\n\nHello world!";

// Creating socket file descriptor
if ((server_fd = socket(AF_INET, SOCK_STREAM, 0)) == 0)
{
    perror("In socket");
    exit(EXIT_FAILURE);
}

int PORT = 8080;
char * line = NULL;
size_t len = 0;
ssize_t rea;
FILE *file = fopen("config.txt", "r");
char PATH[80000]="";
if(file != NULL)
{

```



```

/* Line-by-line read cache file */
while ((rea = getline(&line, &len, file)) != -1)
{
    /* Find key in file */
    if(strstr(line, "port") != NULL)
    {
        /* Get port */
        PORT = atoi(line+4);
    }
    if(strstr(line, "path") != NULL)
    {
        /* Get port */
        strcat(PATH, line+5);
        PATH[strlen(PATH)-1]='\0';
    }
}

/* Close file */
fclose(file);
}

//printf("%d",PORT);
address.sin_family = AF_INET;
address.sin_addr.s_addr = INADDR_ANY;

```

```
address.sin_port = htons( PORT );
```

```
memset(address.sin_zero, '\0', sizeof address.sin_zero);
```

```
if (bind(server_fd, (struct sockaddr *)&address, sizeof(address))<0)
```

```
{
```

```
    perror("In bind");
```

```
    exit(EXIT_FAILURE);
```

```
}
```

```
if (listen(server_fd, 10) < 0)
```

```
{
```

```
    perror("In listen");
```

```
    exit(EXIT_FAILURE);
```

```
}
```

```
while(1)
```

```
{
```

```
    printf("\n+++++++ Waiting for new connection ++++++\n\n");
```

```
    if ((new_socket = accept(server_fd, (struct sockaddr *)&address,  
(socklen_t*)&addrlen))<0)
```

```
    {
```

```
        perror("In accept");
```

```
        exit(EXIT_FAILURE);
```

```
    }
```

```

char buffer[30000] = {0};
valread = read( new_socket , buffer, 30000);
printf("%s\n",buffer );
char fle[800] = "";
getadr(buffer,fle);
char httpHeader[850000] = "";
printf("%s\n",fle);
if(strstr(fle,"images")){
    size_t si=file_size(fle);
    char temp[500] = "";
    //sprintf(temp,"HTTP/1.1 200 OK\nContent-Type:
image/gif\r\nContent-Length: %ld\r\n\r\n",si);
    sprintf(temp,"HTTP/1.1 200 OK\nContent-Type:
image/gif\r\n\r\n");
    //printf("%s\n",temp);
    strcat(httpHeader,temp);
}
else
    strcat(httpHeader,"HTTP/1.1 200 OK\r\n\r\n");
char tpath[80000]="";
strcat(tpath,PATH);
strcat(tpath,fle);
//printf("%s\n",tpath);

```

```

        setHttpHeader(httpHeader,tpath,PATH);
write(new_socket , httpHeader , strlen(httpHeader));
printf("-----Hello message sent-----");
bzero(&httpHeader,sizeof(httpHeader));
bzero(&fle,sizeof(fle));
bzero(&buffer,sizeof(buffer));
bzero(&tpath,sizeof(tpath));
close(new_socket);
}
return 0;
}

```

```

GET /index.html HTTP/1.1
Host: 192.168.216.128:22000
User-Agent: Mozilla/5.0 (X11; Ubuntu; Linux x86_64; rv:104.0) Gecko/20100101 Firefox/104.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Connection: keep-alive
Referer: http://192.168.216.128:22000/page1.html
Upgrade-Insecure-Requests: 1

```

```

GET /page1.html HTTP/1.1
Host: 192.168.216.128:22000
User-Agent: Mozilla/5.0 (X11; Ubuntu; Linux x86_64; rv:104.0) Gecko/20100101 Firefox/104.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Referer: http://192.168.216.128:22000/index.html
Connection: keep-alive
Upgrade-Insecure-Requests: 1

```