106118106_Socket_Programming_Assignment

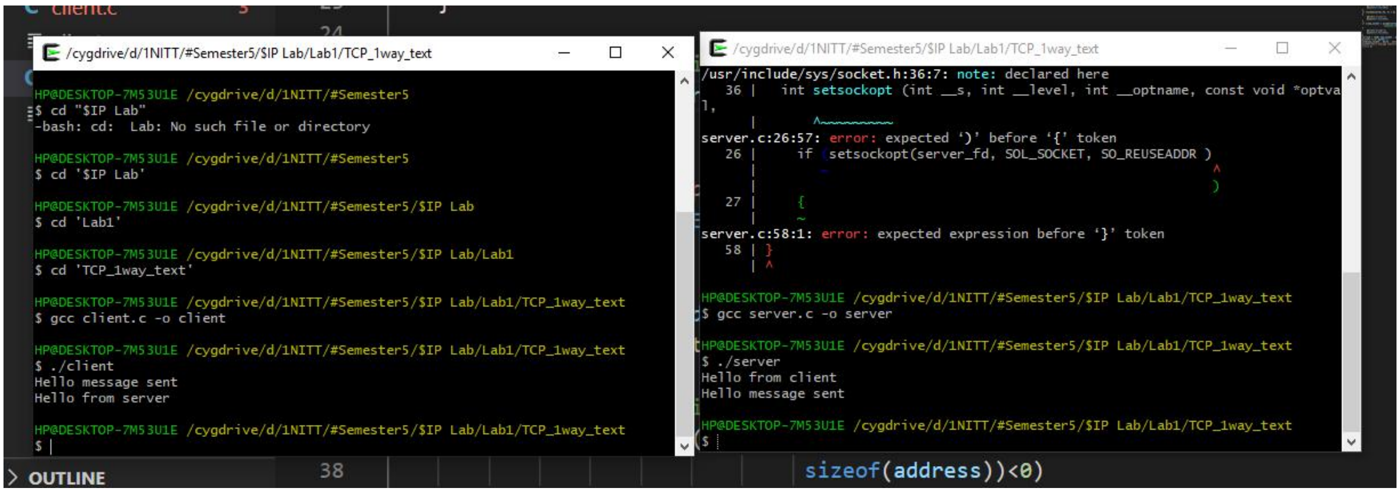
One Way text transfer TCP

server.c

```c
#include <unistd.h>
#include <stdio.h>
#include <sys/socket.h>
#include <stdlib.h>
#include <netinet/in.h>
#include <string.h>
#define PORT 8080
int main(int argc, char const *argv[])
{
    int server_fd, new_socket, valread;
    struct sockaddr_in address;
    int opt = 1;
    int addrlen = sizeof(address);
    char buffer[1024] = {0};
    char *hello = "Hello from server";

    if ((server_fd = socket(AF_INET, SOCK_STREAM, 0)) == 0)
    {
        perror("socket failed");
        exit(EXIT_FAILURE);
    }
    if (setsockopt(server_fd, SOL_SOCKET, SO_REUSEADDR,&opt, sizeof(opt)))
    {
        perror("setsockopt");
        exit(EXIT_FAILURE);
    }
    address.sin_family = AF_INET;
    address.sin_addr.s_addr = INADDR_ANY;
    address.sin_port = htons( PORT );
    if (bind(server_fd, (struct sockaddr *)&address,
                                 sizeof(address))<0)
    {
        perror("bind failed");
        exit(EXIT_FAILURE);
    }
    if (listen(server_fd, 3) < 0)
    {
        perror("listen");
        exit(EXIT_FAILURE);
    }
    if ((new_socket=accept(server_fd, (struct sockaddr *)&address,socklen_t*)&addrlen))<0)
    {
        perror("accept");
        exit(EXIT_FAILURE);
    }
    valread = read( new_socket , buffer, 1024);
    printf("%s\n",buffer );
    send(new_socket , hello , strlen(hello) , 0 );
    printf("Hello message sent\n");
    return 0; }
```

client.c

```
#include <stdio.h>
#include <sys/socket.h>
#include <arpa/inet.h>
#include <unistd.h>
#include <string.h>
#define PORT 8080

int main(int argc, char const *argv[])
{
    int sock = 0, valread;
    struct sockaddr_in serv_addr;
    char *hello = "Hello from client";
    char buffer[1024] = {0};
    if ((sock = socket(AF_INET, SOCK_STREAM, 0)) < 0)
    {
        printf("\n Socket creation error \n");
        return -1;
    }

    serv_addr.sin_family = AF_INET;
    serv_addr.sin_port = htons(PORT);

    if(inet_pton(AF_INET, "127.0.0.1", &serv_addr.sin_addr)<=0)
    {
        printf("\nInvalid address/ Address not supported \n");
        return -1;
    }

    if (connect(sock, (struct sockaddr *)&serv_addr, sizeof(serv_addr)) < 0)
    {
        printf("\nConnection Failed \n");
        return -1;
    }
    send(sock , hello , strlen(hello) , 0 );
    printf("Hello message sent\n");
    valread = read( sock , buffer, 1024);
    printf("%s\n",buffer );
    return 0;
}
```



```
/cygdrive/d/1NITT/#Semester5/$IP Lab/Lab1/TCP_1way_text

HP@DESKTOP-7M53U1E /cygdrive/d/1NITT/#Semester5
$ cd "$IP Lab"
-bash: cd:  Lab: No such file or directory

HP@DESKTOP-7M53U1E /cygdrive/d/1NITT/#Semester5
$ cd '$IP Lab'

HP@DESKTOP-7M53U1E /cygdrive/d/1NITT/#Semester5/$IP Lab
$ cd 'Lab1'

HP@DESKTOP-7M53U1E /cygdrive/d/1NITT/#Semester5/$IP Lab/Lab1
$ cd 'TCP_1way_text'

HP@DESKTOP-7M53U1E /cygdrive/d/1NITT/#Semester5/$IP Lab/Lab1/TCP_1way_text
$ gcc client.c -o client

HP@DESKTOP-7M53U1E /cygdrive/d/1NITT/#Semester5/$IP Lab/Lab1/TCP_1way_text
$ ./client
Hello message sent
Hello from server

HP@DESKTOP-7M53U1E /cygdrive/d/1NITT/#Semester5/$IP Lab/Lab1/TCP_1way_text
$ |
```

```
/cygdrive/d/1NITT/#Semester5/$IP Lab/Lab1/TCP_1way_text
/usr/include/sys/socket.h:36:7: note: declared here
   36 |   int setsockopt (int __s, int __level, int __optname, const void *optva
l,
      |       ^~~~~~~~~~
server.c:26:57: error: expected ')' before '{' token
   26 |     if (setsockopt(server_fd, SOL_SOCKET, SO_REUSEADDR )
      |        ~                                                ^
      |                                                         )
   27 |     {
      |     ~
server.c:58:1: error: expected expression before '}' token
   58 | }
      | ^

HP@DESKTOP-7M53U1E /cygdrive/d/1NITT/#Semester5/$IP Lab/Lab1/TCP_1way_text
$ gcc server.c -o server

HP@DESKTOP-7M53U1E /cygdrive/d/1NITT/#Semester5/$IP Lab/Lab1/TCP_1way_text
$ ./server
Hello from client
Hello message sent

HP@DESKTOP-7M53U1E /cygdrive/d/1NITT/#Semester5/$IP Lab/Lab1/TCP_1way_text
$
```

One Way file transfer TCP

receive_file.c

```c
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <unistd.h>
#include <arpa/inet.h>
#include <netinet/in.h>
#include <sys/socket.h>
#include "transfer.h"

void writefile(int sockfd, FILE *fp);
ssize_t total=0;
int main(int argc, char *argv[])
{
    int sockfd = socket(AF_INET, SOCK_STREAM, 0);
    if (sockfd == -1)
    {
        perror("Can't allocate sockfd");
        exit(1);
    }

    struct sockaddr_in clientaddr, serveraddr;
    memset(&serveraddr, 0, sizeof(serveraddr));
```

```c
    serveraddr.sin_family = AF_INET;
    serveraddr.sin_addr.s_addr = htonl(INADDR_ANY);
    serveraddr.sin_port = htons(SERVERPORT);
```

```c
    if (bind(sockfd, (const struct sockaddr *) &serveraddr, sizeof(serveraddr)) == -1)
    {
        perror("Bind Error");
        exit(1);
    }
```

```c
    if (listen(sockfd, LINSTENPORT) == -1)
    {
        perror("Listen Error");
        exit(1);
    }
```

```c
    socklen_t addrlen = sizeof(clientaddr);
    int connfd = accept(sockfd, (struct sockaddr *) &clientaddr, &addrlen);
    if (connfd == -1)
    {
        perror("Connect Error");
        exit(1);
    }
    close(sockfd);
```

```c
    char filename[BUFFSIZE] = {0};
    if (recv(connfd, filename, BUFFSIZE, 0) == -1)
    {
        perror("Can't receive filename");
        exit(1);
    }
```

```c
    FILE *fp = fopen(filename, "wb");
    if (fp == NULL)
    {
        perror("Can't open file");
        exit(1);
    }

    char addr[INET_ADDRSTRLEN];
    printf("Start receive file: %s from %s\n", filename, inet_ntop(AF_INET, &clientaddr.si
n_addr, addr, INET_ADDRSTRLEN));
    writefile(connfd, fp);
    printf("Receive Success, NumBytes = %ld\n", total);

```

```c
    fclose(fp);
    close(connfd);
    return 0;
}
```

```c
void writefile(int sockfd, FILE *fp)
{
    ssize_t n;
    char buff[MAX_LINE] = {0};
    while ((n = recv(sockfd, buff, MAX_LINE, 0)) > 0)
    {
        total+=n;
        if (n == -1)
        {
            perror("Receive File Error");
            exit(1);
        }

        if (fwrite(buff, sizeof(char), n, fp) != n)
        {
            perror("Write File Error");
            exit(1);
        }
        memset(buff, 0, MAX_LINE);
    }
}
```

Send_file.c

```c
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <libgen.h>
#include <unistd.h>
#include <arpa/inet.h>
#include <netinet/in.h>
#include <sys/socket.h>
#include "transfer.h"

void sendfile(FILE *fp, int sockfd);
ssize_t total=0;
int main(int argc, char* argv[])
{
    if (argc != 3)
    {
        perror("usage:send_file filepath <IPaddress>");
        exit(1);
    }
```

```c
    int sockfd = socket(AF_INET, SOCK_STREAM, 0);
    if (sockfd < 0)
    {
        perror("Can't allocate sockfd");
        exit(1);
    }
```

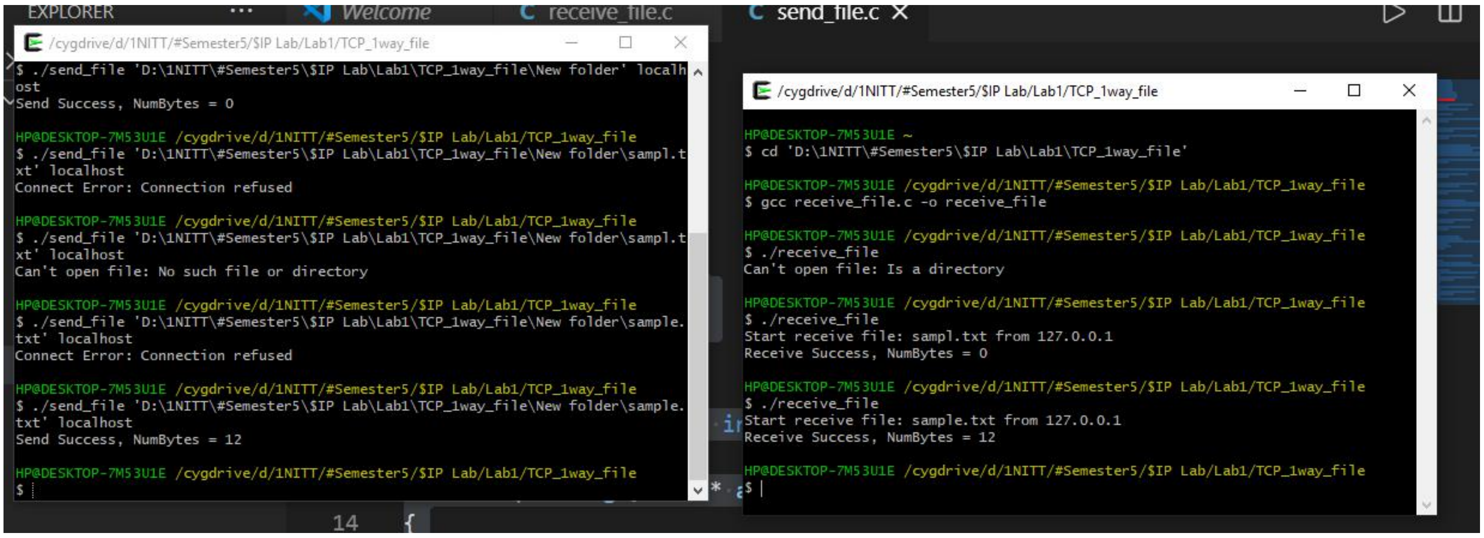
```c
    struct sockaddr_in serveraddr;
    memset(&serveraddr, 0, sizeof(serveraddr));
    serveraddr.sin_family = AF_INET;
    serveraddr.sin_port = htons(SERVERPORT);
    if (inet_pton(AF_INET, argv[2], &serveraddr.sin_addr) < 0)
    {
        perror("IPaddress Convert Error");
        exit(1);
    }
```

```
    if (connect(sockfd, (const struct sockaddr *) &serveraddr, sizeof(serveraddr)) < 0)
    {
        perror("Connect Error");
        exit(1);
    }

    char *filename = basename(argv[1]);
    if (filename == NULL)
    {
        perror("Can't get filename");
        exit(1);
    }

    char buff[BUFFSIZE] = {0};
    strncpy(buff, filename, strlen(filename));
    if (send(sockfd, buff, BUFFSIZE, 0) == -1)
    {
        perror("Can't send filename");
        exit(1);
    }

    FILE *fp = fopen(argv[1], "rb");
    if (fp == NULL)
    {
        perror("Can't open file");
        exit(1);
    }
```

```
    sendfile(fp, sockfd);
    //puts("Send Success");
    printf("Send Success, NumBytes = %ld\n", total);
    fclose(fp);
    close(sockfd);
    return 0;
}
```

```
void sendfile(FILE *fp, int sockfd)
{
    int n;
    char sendline[MAX_LINE] = {0};
    while ((n = fread(sendline, sizeof(char), MAX_LINE, fp)) > 0)
    {
        total+=n;
        if (n != MAX_LINE && ferror(fp))
        {
            perror("Read File Error");
            exit(1);
        }

```

```c
        if (send(sockfd, sendline, n, 0) == -1)
        {
            perror("Can't send file");
            exit(1);
        }
        memset(sendline, 0, MAX_LINE);
    }
}
```



EXPLORER
Welcome
receive_file.c
send_file.c
/cygdrive/d/1NITT/#Semester5/$IP Lab/Lab1/TCP_1way_file
/cygdrive/d/1NITT/#Semester5/$IP Lab/Lab1/TCP_1way_file

```
$ ./send_file 'D:\1NITT\#Semester5\$IP Lab\Lab1\TCP_1way_file\New folder' localh
ost
Send Success, NumBytes = 0

HP@DESKTOP-7M53U1E /cygdrive/d/1NITT/#Semester5/$IP Lab/Lab1/TCP_1way_file
$ ./send_file 'D:\1NITT\#Semester5\$IP Lab\Lab1\TCP_1way_file\New folder\sampl.t
xt' localhost
Connect Error: Connection refused

HP@DESKTOP-7M53U1E /cygdrive/d/1NITT/#Semester5/$IP Lab/Lab1/TCP_1way_file
$ ./send_file 'D:\1NITT\#Semester5\$IP Lab\Lab1\TCP_1way_file\New folder\sampl.t
xt' localhost
Can't open file: No such file or directory

HP@DESKTOP-7M53U1E /cygdrive/d/1NITT/#Semester5/$IP Lab/Lab1/TCP_1way_file
$ ./send_file 'D:\1NITT\#Semester5\$IP Lab\Lab1\TCP_1way_file\New folder\sample.
txt' localhost
Connect Error: Connection refused

HP@DESKTOP-7M53U1E /cygdrive/d/1NITT/#Semester5/$IP Lab/Lab1/TCP_1way_file
$ ./send_file 'D:\1NITT\#Semester5\$IP Lab\Lab1\TCP_1way_file\New folder\sample.
txt' localhost
Send Success, NumBytes = 12

HP@DESKTOP-7M53U1E /cygdrive/d/1NITT/#Semester5/$IP Lab/Lab1/TCP_1way_file
$
```

```
HP@DESKTOP-7M53U1E ~
$ cd 'D:\1NITT\#Semester5\$IP Lab\Lab1\TCP_1way_file'

HP@DESKTOP-7M53U1E /cygdrive/d/1NITT/#Semester5/$IP Lab/Lab1/TCP_1way_file
$ gcc receive_file.c -o receive_file

HP@DESKTOP-7M53U1E /cygdrive/d/1NITT/#Semester5/$IP Lab/Lab1/TCP_1way_file
$ ./receive_file
Can't open file: Is a directory

HP@DESKTOP-7M53U1E /cygdrive/d/1NITT/#Semester5/$IP Lab/Lab1/TCP_1way_file
$ ./receive_file
Start receive file: sampl.txt from 127.0.0.1
Receive Success, NumBytes = 0

HP@DESKTOP-7M53U1E /cygdrive/d/1NITT/#Semester5/$IP Lab/Lab1/TCP_1way_file
$ ./receive_file
Start receive file: sample.txt from 127.0.0.1
Receive Success, NumBytes = 12

HP@DESKTOP-7M53U1E /cygdrive/d/1NITT/#Semester5/$IP Lab/Lab1/TCP_1way_file
$
```