Distribute System: Practical Work 1

Group 5

January 2022

1 File transfer system

Files: `send.txt`, `receive.txt`

I failed to modify provided chat system but I succeed in transferring file with TCP.

2 Design Protocol

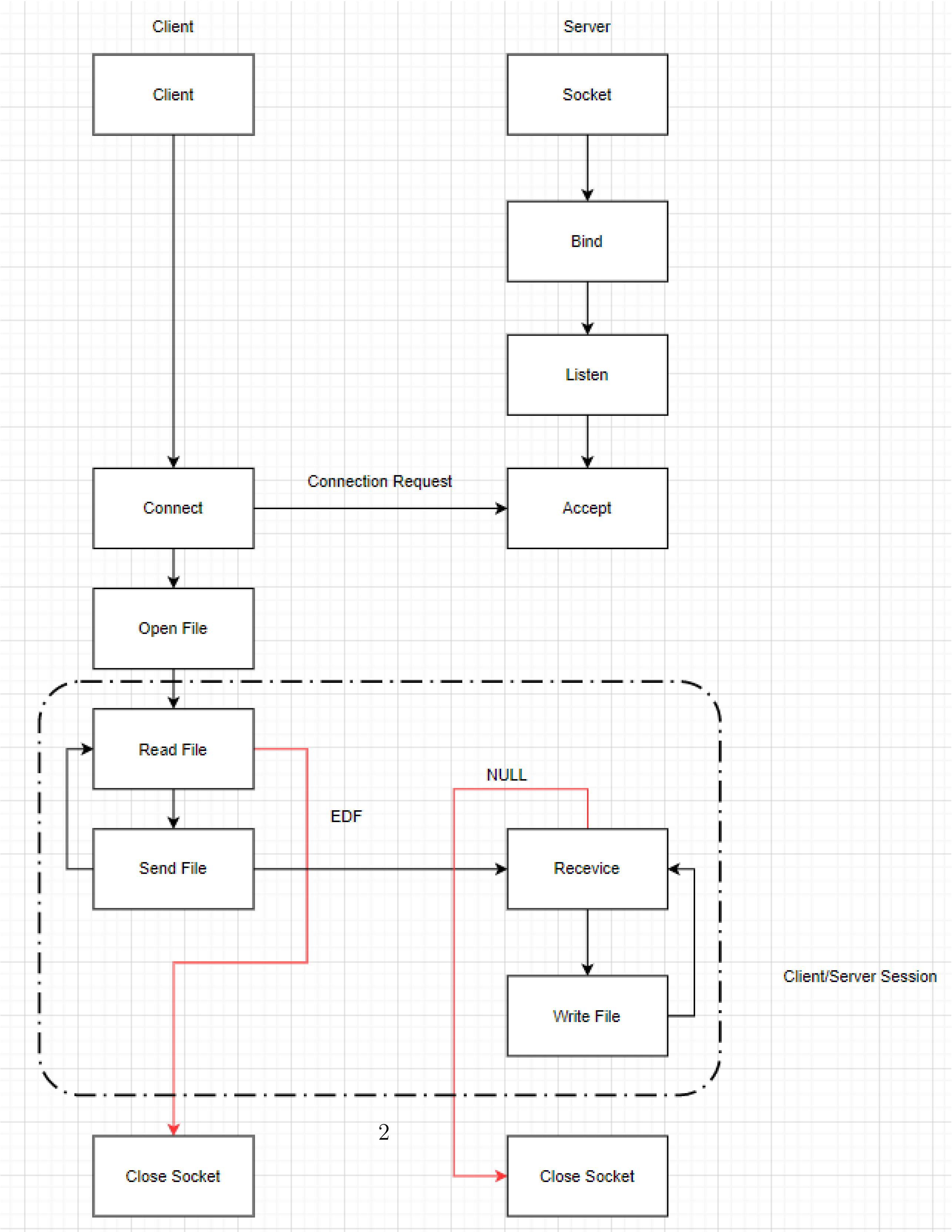

Figure 1: Protocol

3 Organizational system

```
+--------+
| Client |
+---+----+
    |
    |
    |
    | TCP /IP
    |
    |
    v
+---+----+
| Server |
+--------+
```

The server only listen to one client. The client send data as chunks to the server. After receiving a chunk, the server writes to a file. After finishing writing the file, the server closes itself.

4 Implementation

In `client.c`, the function to send file is implemented as followed.

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

```
#define SIZE 1024
#define localhost "127.0.0.1"
#define port 3306
#define IP_PROTOCOL 0
#define NET_BUF_SIZE 32
#define cipherKey 'S'
#define sendrecvflag 0
void sending(FILE *document, int file){
  char buf[SIZE] = {0};
  while(fgets(buf, SIZE, document) != NULL) {
    if (send(file, buf, sizeof(buf), 0) == -1) {
      printf("Sending error.\n");
    }
    bzero(buf, SIZE);
  }
}
int main(int argc, char* argv[])
{
        int file;
        struct sockaddr_in addr_of_server;
        char *filename="test.txt";
        FILE* document;
        file = socket(AF_INET, SOCK_STREAM, IP_PROTOCOL);

        if (file < 0)
                printf("\nFile not received!!\n");
        else
                printf("\nFile %d received\n", file);

        int addrlen = sizeof(addr_of_server);
        addr_of_server.sin_family = AF_INET;
        addr_of_server.sin_port = port;
        addr_of_server.sin_addr.s_addr = inet_addr(localhost);
        if (connect(file, (struct sockaddr*)&addr_of_server, addrlen)==-1){
                printf("Socket error\n");
        }
        printf("\n---------Data Received---------\n");
        document = fopen(filename, "r");
```

```
          if (document == NULL) {
                    printf("File reading error.\n");
          }
          sending(document, file);
          printf("File sent successfully.\n");
          close(file);
        return 0;
}
```

In servers.c, the function to write file is implemented as followed

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <arpa/inet.h>
#define SIZE 1024
#define localhost "127.0.0.1"
#define port 3306
#define IP_PROTOCOL 0
void create(int file){
  FILE *document;
  char *filename = "receives.txt";
  char bf[SIZE];
  document = fopen(filename, "w");
  while (1) {
    int n = recv(file, bf, SIZE, 0);
    if (n <= 0){
      break;
    }
    fprintf(document, "%s", bf);
    bzero(bf, SIZE);
  }
  return;
}
int main(int argc, char* argv[]){
  int file;
  int new_file;
  socklen_t size_of_address;
  char bf[SIZE];
  struct sockaddr_in ad, new_server_addr;
  size_of_address = sizeof(new_server_addr);
  file = socket(AF_INET, SOCK_STREAM, IP_PROTOCOL);
  if (file < 0)
        printf("\nFile not received!!\n");
    else
        printf("\nFile %d received\n", file);
  ad.sin_family = AF_INET;
  ad.sin_port = port;
  ad.sin_addr.s_addr = INADDR_ANY;
  if(bind(file, (struct sockaddr*)&ad, sizeof(ad))<0) {
    printf("Fail to bind\n");
  }
  printf("Binding successfully.\n");
  listen(file, 0);
  new_file = accept(file, (struct sockaddr*)&new_server_addr, &size_of_address);
  create(new_file);
  printf("Data written successfully.\n");
  return 0;
}
```