

# Distributed System Practical Work 1

TCP File Transfer

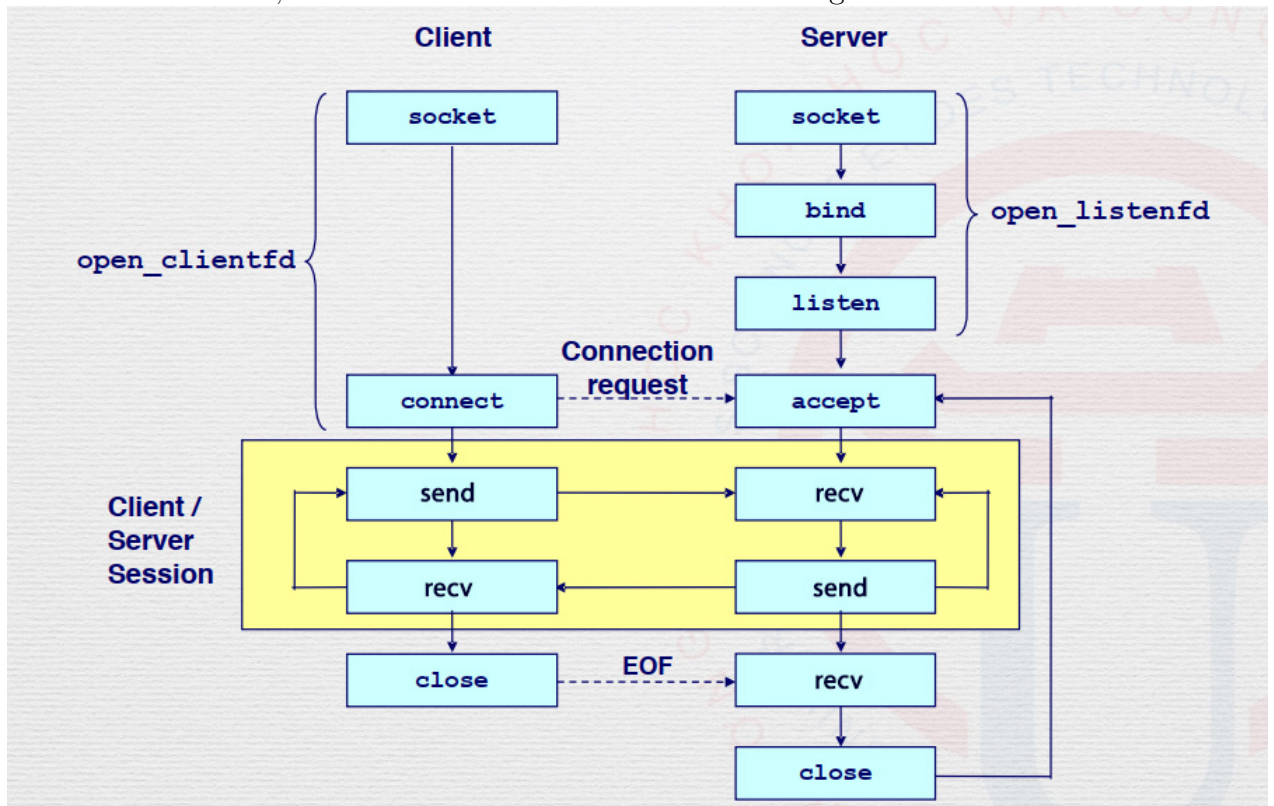
March 11, 2020

## Contents

<b>1</b>	<b>Protocol Design</b>	<b>2</b>
<b>2</b>	<b>Codes</b>	<b>3</b>
2.1	Server . . . . .	3
2.2	Client . . . . .	4

# 1 Protocol Design

Due to being based on the provided Chat System, the protocol design is somewhat the same, however there are still some small changes.



## 2 Codes

### 2.1 Server

```
while (1) {
    FILE *fp;
    char s[MAXCHAR];
    // it's client turn to chat, I wait and read message from client
    recv(cli, s, sizeof(s), 0);
    printf("client says: %s\n",s);
    fp = fopen("serverReceive.txt","w");
    if (fp == NULL){
        printf("Could not open file");
    }
    fprintf(fp,"%s\n",s);
    fclose(fp);
    // now it's my (server) turn
    char *filename;
    int i=0;
    printf("server>");
    scanf("%s", filename);
    fp = fopen(filename,"r");
    if (fp == NULL){
        printf("Could not open file %s",filename);
    }
    while (fgets(s,MAXCHAR,fp)!=NULL){
        fscanf(fp,"%s",&s[i]);
        i++;
    }
    fclose(fp);
    send(cli,&s,sizeof(s),0);
}
return 0;
```

## 2.2 Client

```
while (1) {
    // after connected, it's client turn to chat
    FILE *fp;
    char s[MAXCHAR];
    char *filename;
    int i=0;
    // send some data to server
    printf("client>");
    scanf("%s", filename);
    fp = fopen(filename,"r");
    if (fp == NULL){
        printf("Could not open file %s",filename);
    }
    while (fgets(s,MAXCHAR,fp)!=NULL){
        fscanf(fp,"%s",&s[i]);
        i++;
    }
    fclose(fp);
    send(serv,&s,sizeof(s),0);
    // then it's server turn
    fp = fopen("clientReceive.txt","w");
    if (fp == NULL){
        printf("Could not open file");
    }
    recv(serv, s, sizeof(s), 0);
    fprintf(fp,"%s\n",s);
    fclose(fp);
    printf("server says: %s\n", s);
}
```