```c
// File transfer: file_server.c

#include <stdio.h>
#include <unistd.h>
#include <string.h>
#include <sys/socket.h>
#include <netinet/in.h>
#include <sys/types.h>

#define SERV_PORT 5576

int main(int argc,char **argv) {
   int i, j ;
   ssize_t n ;
   FILE *fp ;
   char s[80], f[80] ;
   struct sockaddr_in servaddr, cliaddr ;

   int listenfd, connfd, clilen ;
   listenfd = socket(AF_INET, SOCK_STREAM, 0) ;
   bzero(&servaddr, sizeof(servaddr)) ;
   servaddr.sin_family = AF_INET ;
   servaddr.sin_port = htons(SERV_PORT) ;

   bind(listenfd, (struct sockaddr*)&servaddr,
                              sizeof(servaddr)) ;
   listen(listenfd, 1) ;
   clilen = sizeof(cliaddr) ;
   connfd = accept(listenfd, (struct sockaddr*)&cliaddr,
                                                &clilen) ;
   printf("Client connected!\n") ;

   read(connfd, f, 80) ;
   fp = fopen(f, "r") ;
   printf("Filename: %s\n", f) ;
   while(fgets(s, 80, fp) != NULL) {
      printf("%s", s) ;
      write(connfd, s, sizeof(s)) ;
   }

   close(listenfd) ;
   fclose(fp) ;
}
```

```
$ # Running on the server (terminal-1)
$ gcc file_server.c -o server
$ ./server
```

```c
// File transfer: file_client.c

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

#define SERV_PORT 5576

int main(int argc,char **argv) {
   int i, j ;
   ssize_t n ;
   char filename[80], recvline[80] ;
   struct sockaddr_in servaddr ;

   int sockfd ;
   sockfd = socket(AF_INET, SOCK_STREAM, 0);
   bzero(&servaddr, sizeof(servaddr)) ;
   servaddr.sin_family = AF_INET;
   servaddr.sin_port = htons(SERV_PORT) ;
   inet_pton(AF_INET, argv[1], &servaddr.sin_addr) ;
   connect(sockfd, (struct sockaddr*)&servaddr,
                                      sizeof(servaddr)) ;

   printf("Enter the filename: ") ;
   scanf("%s", filename) ;
   write(sockfd, filename, sizeof(filename)) ;
   printf("Data from server: \n");

   while(read(sockfd, recvline, 80) != 0)
      fputs(recvline, stdout) ;

   close(sockfd) ;
}
```

```
$ # Running on the client (terminal-2)
$ gcc file_client.c -o client
$ # Run the server before running the client
$ ./client localhost
```