SERVER

```
#include <stdio.h>
#include <stdlib.h>
#include <unistd.h>
#include <sys/types.h>
#include <sys/socket.h>
#include <netinet/in.h>
#include <fcntl.h>
#include <string.h>
#include <errno.h>
#include <arpa/inet.h> // Add this header for inet_ntoa

#define PORT 8888
#define BUFFER_SIZE 1024

void handle_client(int client_socket) {
    char buffer[BUFFER_SIZE];
    ssize_t bytes_received;
    char *file_name;
    int file_fd;
    ssize_t bytes_sent;

    // Receive the file name from the client
    bytes_received = recv(client_socket, buffer, BUFFER_SIZE - 1, 0);
    if (bytes_received < 0) {
        perror("recv");
        close(client_socket);
        return;
    }

    buffer[bytes_received] = '\0';
    file_name = buffer;

    // Open the file
    file_fd = open(file_name, O_RDONLY);
    if (file_fd < 0) {
        // Send a message indicating the file does not exist
        send(client_socket, "File not found", strlen("File not found"), 0);
        close(client_socket);
        return;
    } else {
        // Send the process ID to the client
        sprintf(buffer, "PID: %d\n", getpid());
        send(client_socket, buffer, strlen(buffer), 0);

        // Send the file contents to the client
        while ((bytes_sent = sendfile(client_socket, file_fd, NULL, BUFFER_SIZE)) > 0) {
            // Continue sending until the file is fully sent
        }

        if (bytes_sent < 0) {
            perror("sendfile");
        }

        // Close the file
        close(file_fd);
    }

    // Close the client socket
    close(client_socket);
}
```

```
int main() {
    int server_socket, client_socket;
    struct sockaddr_in server_addr, client_addr;
    socklen_t client_len = sizeof(client_addr);
    pid_t pid;

    // Create a TCP socket
    server_socket = socket(AF_INET, SOCK_STREAM, 0);
    if (server_socket < 0) {
        perror("socket");
        exit(EXIT_FAILURE);
    }

    // Bind the socket to a specific IP address and port
    server_addr.sin_family = AF_INET;
    server_addr.sin_addr.s_addr = htonl(INADDR_ANY);
    server_addr.sin_port = htons(PORT);
    if (bind(server_socket, (struct sockaddr *)&server_addr, sizeof(server_addr)) < 0) {
        perror("bind");
        exit(EXIT_FAILURE);
    }

    // Enable the server to accept connections
    if (listen(server_socket, 5) < 0) {
        perror("listen");
        exit(EXIT_FAILURE);
    }

    printf("Server PID: %d\n", getpid());

    while (1) {
        // Accept a client connection
        client_socket = accept(server_socket, (struct sockaddr *)&client_addr, &client_len);
        if (client_socket < 0) {
            perror("accept");
            continue; // Continue accepting other clients
        }
        printf("Connection from %s:%d\n", inet_ntoa(client_addr.sin_addr),
ntohs(client_addr.sin_port));

        // Create a new process to handle the client
        pid = fork();
        if (pid < 0) {
            perror("fork");
            close(client_socket);
            continue; // Continue accepting other clients
        }

        if (pid == 0) {
            // Child process
            handle_client(client_socket);
            exit(EXIT_SUCCESS);
        } else {
            // Parent process
            close(client_socket);
        }
    }

    // Close the server socket
    close(server_socket);
    return 0;
}
```

CLIENT

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

#define SERVER_IP "127.0.0.1"
#define PORT 8888
#define BUFFER_SIZE 1024

int main() {
    int client_socket;
    struct sockaddr_in server_addr;
    struct hostent *server;
    char file_name[BUFFER_SIZE];
    ssize_t bytes_sent, bytes_received;
    char buffer[BUFFER_SIZE];

    // Create a TCP socket
    client_socket = socket(AF_INET, SOCK_STREAM, 0);
    if (client_socket < 0) {
        perror("socket");
        exit(EXIT_FAILURE);
    }

    // Get the server's IP address
    server = gethostbyname(SERVER_IP);
```

```
if (server == NULL) {
    fprintf(stderr, "Error: no such host\n");
    exit(EXIT_FAILURE);
}

// Configure the server address
server_addr.sin_family = AF_INET;
server_addr.sin_port = htons(PORT);
memcpy(&server_addr.sin_addr.s_addr, server->h_addr, server->h_length);

// Connect to the server
if (connect(client_socket, (struct sockaddr *)&server_addr, sizeof(server_addr)) < 0) {
    perror("connect");
    exit(EXIT_FAILURE);
}

// Get the file name from the user
printf("Enter file name: ");
scanf("%s", file_name);

// Send the file name to the server
bytes_sent = send(client_socket, file_name, strlen(file_name), 0);
if (bytes_sent < 0) {
    perror("send");
    exit(EXIT_FAILURE);
}

// Receive the server's response
bytes_received = recv(client_socket, buffer, BUFFER_SIZE, 0);
if (bytes_received < 0) {
    perror("recv");
```

```
        exit(EXIT_FAILURE);
    }

    buffer[bytes_received] = '\0';
    printf("%s\n", buffer);

    // Close the socket
    close(client_socket);

    return 0;
}
```

OUTPUT

```
SERVER
Server PID:5781
Connection from 127.0.0.1:49562

CLIENT
Enter the file name: hello.txt
PID:5336
Hello
```

ABY PIOUS VINOY

S6 CSE

ROLL NO : 57