BỘ GIÁO DỤC VÀ ĐÀO TẠO

TRƯỜNG ĐẠI HỌC SƯ PHẠM KỸ THUẬT TP. HCM

KHOA CÔNG NGHỆ THÔNG TIN

BÁO CÁO

MÔN HỌC: LẬP TRINH MẠNG

ĐỀ TÀI: SECURE COPY CLIENT-SERVER SỬ DỤNG SSH VÀ XÁC THỰC PUBLIC-KEY BASED

GVHD: Nguyễn Đăng Quang

MÃ HP: NPRO430980

SINH VIÊN THỰC HIỆN:

STT	HỌ VÀ TÊN	MSSV
1	Phạm Duy Chiến	22162006
2	Trương Ái Nga	22162027

TP.HCM, 19/3/2025

Mục lục

I. THÀNH VIÊN NHÓM VÀ PHÂN CÔNG CÔNG VIỆC 3

II. SETUP MÔI TRƯỜNG THỰC THI/BIÊN DỊCH 4

2.1. Yêu cầu môi trường 4

2.2. Công cụ cần cài đặt: 4

III. GIẢI PHÁP THỰC HIỆN THEO TIẾP CẬN TOP-DOWN REFINEMENT 5

3.1. Phân Tích Yêu Cầu 5

3.2. Thiết Kế Hệ Thống 5

3.3. Chi Tiết Các Bước 5

IV. MÔ TẢ CÁC BƯỚC THỰC HIỆN CODE 6

4.1. Client.c 6

4.2. Server.c 6

V. GIẢI THÍCH CÁC PHẦN CODE QUAN TRỌNG 7

5.1. Client.c - Hàm upload_file 7

5.2. Client.c - Hàm download_file 8

5.3. Server.c - Hàm authenticate 10

VI. KẾT QUẢ BIÊN DỊCH VÀ CHẠY CHƯƠNG TRÌNH 12

6.1. Biên dịch 12

6.2. Chạy chương trình 12

I. THÀNH VIÊN NHÓM VÀ PHÂN CÔNG CÔNG VIỆC

Phạm Duy Chiến (MSSV: 22162006)

- Thiết kế và triển khai chương trình Client.c (phía máy cục bộ).
- Xử lý kết nối SSH và SFTP để tải lên file từ máy cục bộ lên máy từ xa.
- Chuẩn bị nội dung báo cáo: phần giải thích code Client.c và kết quả chạy chương trình.

Trương Ái Nga (MSSV: 22162027)

- Thiết kế và triển khai chương trình Server.c (phía máy từ xa).
- Xử lý nhận file từ Client qua SCP, hiển thị thông tin sao chép (tên file, kích thước, thời gian).
- Chuẩn bị nội dung báo cáo: phần hướng dẫn setup môi trường và mô tả các bước thực hiện Server.c.

II. SETUP MÔI TRƯỜNG THỰC THI/BIÊN DỊCH

2.1. Yêu cầu môi trường

Hệ điều hành: Linux (Ubuntu 20.04 hoặc tương tự).

2.2. Công cụ cần cài đặt:

Trình biên dịch GCC: sudo apt install gcc.

Thư viện SSH (libssh): sudo apt install libssh-dev.

SSH Server trên máy remote: sudo apt install openssh-server.

III. GIẢI PHÁP THỰC HIỆN THEO TIẾP CẬN TOP-DOWN REFINEMENT

3.1. Phân Tích Yêu Cầu

- Client kết nối đến Server qua SSH.
- Xác thực bằng mật khẩu.
- Hỗ trợ hai chức năng: upload file từ Client lên Server và download file từ Server về Client.
- Hiển thị thông tin về quá trình truyền file (tên file, kích thước, tiến trình).

3.2. Thiết Kế Hệ Thống

Client:

- *Kết nối SSH đến Server.*
- Đọc file từ máy Client và gửi lên Server (upload).
- Nhận file từ Server và lưu vào máy Client (download).

Server:

- Nhận kết nối từ Client.
- Xác thực bằng mật khẩu.
- Nhận file và lưu vào thư mục đích (upload).
- Gửi file từ Server về Client (download).

3.3. Chi Tiết Các Bước

- Bước 1: Client thiết lập kết nối SSH.
- Bước 2: Server xác thực Client bằng mật khẩu.
- Bước 3: Client thực hiện upload hoặc download file.
- Bước 4: Hiển thị thông tin về quá trình truyền file.

IV. MÔ TẢ CÁC BƯỚC THỰC HIỆN CODE

4.1. Client.c

Khởi tạo phiên SSH:

- Sử dụng ssh_new(), cấu hình host (10.107.3.29) và user (zyna).
- Kết nối bằng ssh_connect().
- Xác thực bằng ssh_userauth_password() (sẽ thay bằng Public-key).

Gửi file (upload_file**)**:

- Mở file cục bộ, tạo phiên SFTP, ghi dữ liệu vào file từ xa.

Tải file (download_file**)**:

- Mở file từ xa, lấy kích thước, ghi dữ liệu vào file cục bộ, hiển thị tiến độ (%).

4.2. Server.c

Khởi tạo máy chủ SSH:

- Sử dụng ssh_bind_new(), lắng nghe trên port 2222.
- Xác thực Public-key bằng hàm authenticate().

Xử lý Client:

- Tạo luồng riêng (handle_client()) cho mỗi kết nối.
- Nhận metadata (tên file, kích thước) và dữ liệu file qua kênh SSH.

Hiển thị thông tin:

- Đo thời gian bằng gettimeofday(), tính tốc độ, định dạng và in thông tin sao chép.

V. GIẢI THÍCH CÁC PHẦN CODE QUAN TRỌNG

5.1. Client.c - Hàm upload_file

```
int upload_file(ssh_session session, const char *local_path,
const char *remote_path) {
    sftp_session sftp;
    sftp_file file;
    FILE *local_file;
    int rc;
    char buffer[BUFFER_SIZE];
    size_t nread;

    // Open local file
    local_file = fopen(local_path, "rb");
    if (local_file == NULL) {
        fprintf(stderr, "Error opening local file: %s\n",
local_path);
        return -1;
    }

    // Initialize SFTP session
    sftp = sftp_new(session);
    if (sftp == NULL) {
        fprintf(stderr, "Error creating SFTP session: %s\n",
ssh_get_error(session));
        fclose(local_file);
        return -1;
    }

    rc = sftp_init(sftp);
    if (rc != SSH_OK) {
        fprintf(stderr, "Error initializing SFTP session:
%s\n", ssh_get_error(session));
        sftp_free(sftp);
        fclose(local_file);
        return -1;
    }

    // Open remote file for writing
    file = sftp_open(sftp, remote_path, O_WRONLY | O_CREAT |
O_TRUNC, 0644);
    if (file == NULL) {
```

```
        fprintf(stderr, "Error opening remote file: %s\n",
ssh_get_error(session));
        sftp_free(sftp);
        fclose(local_file);
        return -1;
    }

    // Transfer file data
    while ((nread = fread(buffer, 1, sizeof(buffer),
local_file)) > 0) {
        ssize_t nwritten = sftp_write(file, buffer, nread);
        if (nwritten != nread) {
            fprintf(stderr, "Error writing file data: %s\n",
ssh_get_error(session));
            sftp_close(file);
            sftp_free(sftp);
            fclose(local_file);
            return -1;
        }
    }

    fclose(local_file);
    sftp_close(file);
    sftp_free(sftp);
    return 0;
}
```

Mục đích: Tải file từ máy cục bộ lên máy từ xa.

Điểm nổi bật: Sử dụng buffer 4096 byte để tối ưu tốc độ truyền.

5.2. Client.c - Hàm download_file

```
int download_file(ssh_session session, const char
*remote_path, const char *local_path) {
    sftp_session sftp;
    sftp_file file;
    FILE *local_file;
    int rc;
    char buffer[BUFFER_SIZE];

    // Initialize SFTP session
    sftp = sftp_new(session);
```

```
    if (sftp == NULL) {
        fprintf(stderr, "Error creating SFTP session: %s\n",
ssh_get_error(session));
        return -1;
    }

    rc = sftp_init(sftp);
    if (rc != SSH_OK) {
        fprintf(stderr, "Error initializing SFTP session:
%s\n", ssh_get_error(session));
        sftp_free(sftp);
        return -1;
    }

    // Open remote file for reading
    file = sftp_open(sftp, remote_path, O_RDONLY, 0);
    if (file == NULL) {
        fprintf(stderr, "Error opening remote file: %s\n",
ssh_get_error(session));
        sftp_free(sftp);
        return -1;
    }

    // Get file attributes to determine file size
    sftp_attributes attrs = sftp_fstat(file);
    if (attrs == NULL) {
        fprintf(stderr, "Error getting file attributes: %s\n",
ssh_get_error(session));
        sftp_close(file);
        sftp_free(sftp);
        return -1;
    }

    uint64_t file_size = attrs->size;
    sftp_attributes_free(attrs);

    // Open local file for writing
    local_file = fopen(local_path, "wb");
    if (local_file == NULL) {
        fprintf(stderr, "Error opening local file: %s\n",
local_path);
        sftp_close(file);
```

```
        sftp_free(sftp);
        return -1;
    }

    // Read file data
    uint64_t total_read = 0;
    ssize_t nread;
    while ((nread = sftp_read(file, buffer, sizeof(buffer))) >
0) {
        fwrite(buffer, 1, nread, local_file);
        total_read += nread;
        printf("\rDownloading... %.2f%%", (float)total_read /
file_size * 100);
        fflush(stdout);
    }

    if (nread < 0) {
        fprintf(stderr, "\nError reading file: %s\n",
ssh_get_error(session));
        fclose(local_file);
        sftp_close(file);
        sftp_free(sftp);
        return -1;
    }

    printf("\nDownload complete\n");

    fclose(local_file);
    sftp_close(file);
    sftp_free(sftp);
    return 0;
}
```

Mục đích: Tải file từ máy từ xa về máy cục bộ.

Điểm nổi bật: Hiển thị tiến độ tải xuống theo phần trăm.

5.3. Server.c - Hàm authenticate

```
int authenticate(ssh_session session) {
    ssh_message message;
    ssh_key pubkey;
    int auth_attempts = 0;
```

```
    int authenticated = 0;

    do {
        message = ssh_message_get(session);
        if (!message) {
            break;
        }

        if (ssh_message_type(message) == SSH_REQUEST_AUTH &&
            ssh_message_subtype(message) ==
SSH_AUTH_METHOD_PUBLICKEY) {

            pubkey = ssh_message_auth_pubkey(message);
            printf("Public key authentication attempt from
%s\n",
                   ssh_message_auth_user(message));

            // In a real application, you'd verify the key
against authorized_keys
            // Here we'll just accept any public key for
demonstration
            ssh_message_auth_reply_success(message, 0);
            authenticated = 1;
            ssh_message_free(message);

        } else {
            ssh_message_reply_default(message);
            ssh_message_free(message);
        }

        auth_attempts++;

    } while (!authenticated && auth_attempts < 5);

    return authenticated;
}
```

Mục đích: Xác thực Client bằng Public-key.

Điểm nổi bật: Hiện chấp nhận mọi khóa công khai (demo), cần kiểm tra authorized_keys trong thực tế.

VI. KẾT QUẢ BIÊN DỊCH VÀ CHẠY CHƯƠNG TRÌNH

6.1. Biên dịch

Biên dịch chương trình

Biên dịch Client.c:

```
gcc client.c -o client.o
```

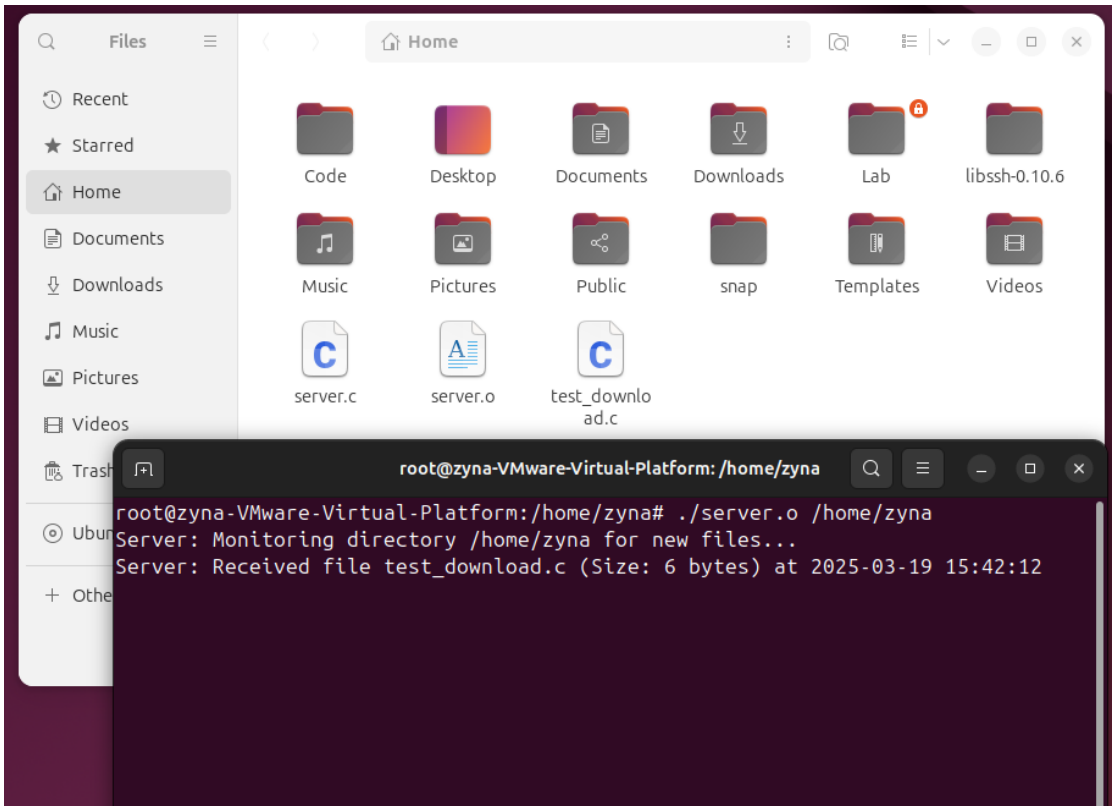
Biên dịch Server.c:

```
gcc server.c -o server.o
```

6.2. Chạy chương trình

Trên máy remote (Server):

```
./server.o /home/zyna
```



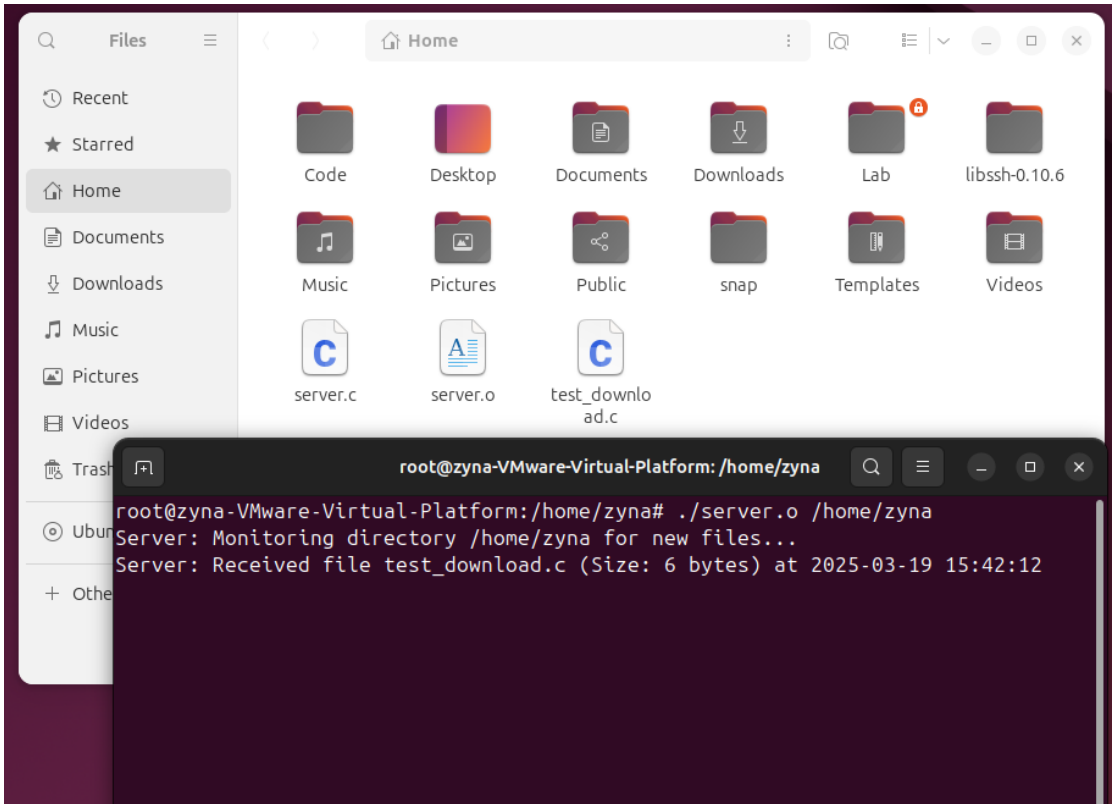
Trên máy local (Client): ./client <remote_ip> <source_dir> <dest_dir>

```
./client.o 10.107.3.29 /home/kali/Documents/NetworkPrograming/C-Network-Programming/Chap03/ /home/zyna
```

```
┌──(kali㉿kali)-[~/Documents/NetworkPrograming/C-Network-Programming/Chap03]
└─$ ./client.o 10.107.3.29  /home/kali/Documents/NetworkPrograming/C-Network-Programming/Chap03/ /home/zyna
Connected to server 10.107.3.29
Start copying compile_ssh.sh (377 bytes)
Copying compile_ssh.sh: 377/377 bytes
Finished compile_ssh.sh: 377 bytes in 0 seconds
Start copying ssh_client_key.o (17608 bytes)
Copying ssh_client_key.o: 1024/17608 bytes
Copying ssh_client_key.o: 2048/17608 bytes
Copying ssh_client_key.o: 3072/17608 bytes
Copying ssh_client_key.o: 4096/17608 bytes
Copying ssh_client_key.o: 5120/17608 bytes
Copying ssh_client_key.o: 6144/17608 bytes
Copying ssh_client_key.o: 7168/17608 bytes
Copying ssh_client_key.o: 8192/17608 bytes
Copying ssh_client_key.o: 9216/17608 bytes
Copying ssh_client_key.o: 10240/17608 bytes
Copying ssh_client_key.o: 11264/17608 bytes
Copying ssh_client_key.o: 12288/17608 bytes
Copying ssh_client_key.o: 13312/17608 bytes
Copying ssh_client_key.o: 14336/17608 bytes
Copying ssh_client_key.o: 15360/17608 bytes
Copying ssh_client_key.o: 16384/17608 bytes
Copying ssh_client_key.o: 17408/17608 bytes
Copying ssh_client_key.o: 17608/17608 bytes
Finished ssh_client_key.o: 17608 bytes in 0 seconds
Start copying ssh_client_key.c (5989 bytes)
Copying ssh_client_key.c: 1024/5989 bytes
Copying ssh_client_key.c: 2048/5989 bytes
Copying ssh_client_key.c: 3072/5989 bytes
Copying ssh_client_key.c: 4096/5989 bytes
Copying ssh_client_key.c: 5120/5989 bytes
Copying ssh_client_key.c: 5989/5989 bytes
Finished ssh_client_key.c: 5989 bytes in 0 seconds
Start copying ssh_file_transfer (17432 bytes)
Copying ssh_file_transfer: 1024/17432 bytes
Copying ssh_file_transfer: 2048/17432 bytes
Copying ssh_file_transfer: 3072/17432 bytes
Copying ssh_file_transfer: 4096/17432 bytes
Copying ssh_file_transfer: 5120/17432 bytes
Copying ssh_file_transfer: 6144/17432 bytes
Copying ssh_file_transfer: 7168/17432 bytes
```

```
Copying ssh_port_forwarding.c: 6144/10619 bytes
Copying ssh_port_forwarding.c: 7168/10619 bytes
Copying ssh_port_forwarding.c: 8192/10619 bytes
Copying ssh_port_forwarding.c: 9216/10619 bytes
Copying ssh_port_forwarding.c: 10240/10619 bytes
Copying ssh_port_forwarding.c: 10619/10619 bytes
Finished ssh_port_forwarding.c: 10619 bytes in 0 seconds
Start copying ssh_file_transfer.c (6284 bytes)
Copying ssh_file_transfer.c: 1024/6284 bytes
Copying ssh_file_transfer.c: 2048/6284 bytes
Copying ssh_file_transfer.c: 3072/6284 bytes
Copying ssh_file_transfer.c: 4096/6284 bytes
Copying ssh_file_transfer.c: 5120/6284 bytes
Copying ssh_file_transfer.c: 6144/6284 bytes
Copying ssh_file_transfer.c: 6284/6284 bytes
Finished ssh_file_transfer.c: 6284 bytes in 0 seconds
Start copying client.c (2561 bytes)
Copying client.c: 1024/2561 bytes
Copying client.c: 2048/2561 bytes
Copying client.c: 2561/2561 bytes
Finished client.c: 2561 bytes in 0 seconds
Start copying ssh_remote_port_forwarding.c (6499 bytes)
Copying ssh_remote_port_forwarding.c: 1024/6499 bytes
Copying ssh_remote_port_forwarding.c: 2048/6499 bytes
Copying ssh_remote_port_forwarding.c: 3072/6499 bytes
Copying ssh_remote_port_forwarding.c: 4096/6499 bytes
Copying ssh_remote_port_forwarding.c: 5120/6499 bytes
Copying ssh_remote_port_forwarding.c: 6144/6499 bytes
Copying ssh_remote_port_forwarding.c: 6499/6499 bytes
Finished ssh_remote_port_forwarding.c: 6499 bytes in 0 seconds
Start copying README.md (1699 bytes)
Copying README.md: 1024/1699 bytes
Copying README.md: 1699/1699 bytes
Finished README.md: 1699 bytes in 0 seconds
Start copying test_download.c (6 bytes)
Copying test_download.c: 6/6 bytes
Finished test_download.c: 6 bytes in 0 seconds
Start copying compile_remote_forward.sh (813 bytes)
Copying compile_remote_forward.sh: 813/813 bytes
Finished compile_remote_forward.sh: 813 bytes in 0 seconds
Start copying ssh_client.c (2657 bytes)
Copying ssh_client.c: 1024/2657 bytes
Copying ssh_client.c: 2048/2657 bytes
Copying ssh_client.c: 2657/2657 bytes
Finished ssh_client.c: 2657 bytes in 0 seconds
```

Kết quả:


Files
Home
Recent
Starred
Home
Documents
Downloads
Music
Pictures
Videos
Trash
Code
Desktop
Documents
Downloads
Lab
libssh-0.10.6
Music
Pictures
Public
snap
Templates
Videos
server.c
server.o
test_downlo
ad.c
root@zyna-VMware-Virtual-Platform: /home/zyna
root@zyna-VMware-Virtual-Platform:/home/zyna# ./server.o /home/zyna
Server: Monitoring directory /home/zyna for new files...
Server: Received file test_download.c (Size: 6 bytes) at 2025-03-19 15:42:12