

**TRƯỜNG ĐẠI HỌC SƯ PHẠM KỸ THUẬT TP. HCM**  
**KHOA CÔNG NGHỆ THÔNG TIN**

**MÔN: LẬP TRÌNH MẠNG**



**HCMUTE**

**BÁO CÁO ĐỀ THI GIỮA KỲ**

**Sinh viên thực hiện:**

Tăng Huỳnh Chí Tân

Nguyễn Hoàng Tùng

**Mã sinh viên:**

22162040

22162053

Thành Phố Thủ Đức, Ngày 26 Tháng 11 Năm 2024

## I. Phân công nhiệm vụ

Thành viên	MSSV	Nhiệm vụ
Tăng Huỳnh Chí Tân	22162040	- Chuẩn bị môi trường cho máy chủ. - Thực hiện viết mã chương trình sao chép an toàn cho phía máy chủ, biên dịch mã và chạy chương trình. - Viết báo cáo về đoạn mã SSH phía máy chủ.
Nguyễn Hoàng Tùng	22162053	- Chuẩn bị môi trường cho máy khách. - Thực hiện viết mã chương trình sao chép an toàn cho phía máy khách, biên dịch mã và chạy chương trình. - Viết báo cáo về đoạn mã SSH phía máy khách.

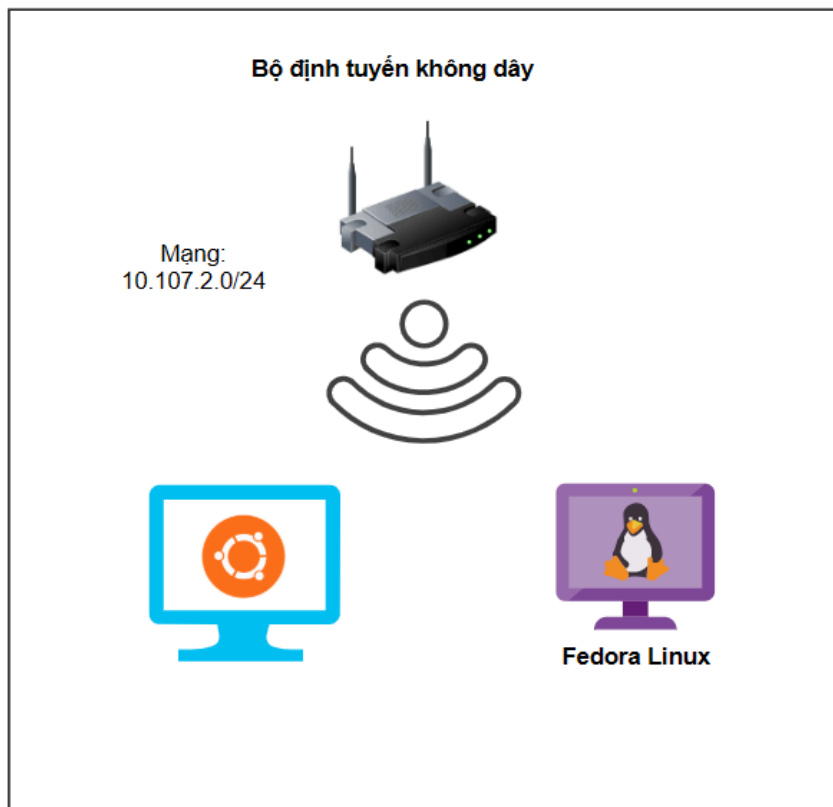
## II. Chuẩn bị môi trường

### a) Tổng quan môi trường

Để thực hành bài thi giữa kì chúng em sử dụng 2 máy tính lần lượt như sau:

- Máy khách: là máy Ubuntu 20.04.
- Máy chủ: là một máy Fedora Linux 41.

Mô hình mạng:



Để hai máy, máy khách và máy chủ có thể giao tiếp được với nhau, cụ thể là máy khách kết nối với máy chủ thông qua SSH và sao chép tệp từ máy khách sang máy chủ, chúng em cấu

hình hai máy cùng chung một mạng LAN (hoặc có thể sử dụng địa IP công khai của hai máy khách và máy chủ để kết nối với nhau, trong điều kiện tường lửa cho phép kết nối và có thể cấu hình thêm NAT).

### b) Môi trường phía máy khách

Chúng em đã chuẩn bị mạng có subnet là: 10.107.0.0/26, trong đó máy khách có địa chỉ 10.107.2.234:

```
2: ens33: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
    link/ether 00:0c:29:5f:26:60 brd ff:ff:ff:ff:ff:ff
    altnam enp2s1
    inet 10.107.3.41/16 brd 10.107.255.255 scope global dynamic noprefixroute ens33
        valid_lft 86396sec preferred_lft 86396sec
    inet6 fe80::d59a:fc0f:1f56:8530/64 scope link noprefixroute
        valid_lft forever preferred_lft forever
```

Đồng thời để có thể biên dịch chương trình từ mã nguồn C, chúng em tải trình biên dịch và thêm trình biên dịch GCC vào biến môi trường cho máy khách (máy Ubuntu).

```
ubuntu@ubuntu:~/Desktop$ gcc -v
Using built-in specs.
COLLECT_GCC=gcc
COLLECT_LTO_WRAPPER=/usr/lib/gcc/x86_64-linux-gnu/9/lto-wrapper
OFFLOAD_TARGET_NAMES=nvptx-none:hsa
OFFLOAD_TARGET_DEFAULT=1
Target: x86_64-linux-gnu
Configured with: ../src/configure -v --with-pkgversion='Ubuntu 9.4.0-1ubuntu1~20.04.2' --with-Thread model: posix
gcc version 9.4.0 (Ubuntu 9.4.0-1ubuntu1~20.04.2)
```

Để việc kết nối không bị chặn bởi tường lửa, chúng em cấu hình thêm các rule để cho phép máy chủ gửi gói tin tới cổng 22 (là cổng sẽ kết mở để trao đổi với máy chủ):

```
ubuntu@ubuntu:~/Desktop$ sudo ufw allow from 10.107.0.0/16 to 10.107.3.41/32 port 22
Rules updated
ubuntu@ubuntu:~/Desktop$ sudo ufw status
Status: inactive
ubuntu@ubuntu:~/Desktop$ sudo ufw enable
Firewall is active and enabled on system startup
ubuntu@ubuntu:~/Desktop$ sudo ufw status
Status: active

To Action From
--
10.107.3.41 22 ALLOW 10.107.0.0/16
```

### c) Môi trường máy chủ

## III. Các bước tiếp cận

### a) Xây dựng đoạn mã kết nối SSH trên Server

Đầu tiên, chúng em xây dựng một đoạn mã để Server chờ kết nối tới Client:

```
/* Tạo phiên SSH cho kết nối đến */
ssh_session session = ssh_new();
if (session == NULL) {
    fprintf(stderr, "Không tạo được phiên SSH.\n");
    ssh_bind_free(sshbind);
    exit(EXIT_FAILURE);
}

if (ssh_bind_accept(sshbind, session) != SSH_OK) {
    fprintf(stderr, "Lỗi chấp nhận kết nối: %s\n", ssh_get_error(sshbind));
    ssh_free(session);
    ssh_bind_free(sshbind);
    exit(EXIT_FAILURE);
}

/* Thực hiện bắt tay (key exchange) */
if (ssh_handle_key_exchange(session) != SSH_OK) {
    fprintf(stderr, "Lỗi bắt tay: %s\n", ssh_get_error(session));
    ssh_disconnect(session);
    ssh_free(session);
    ssh_bind_free(sshbind);
    exit(EXIT_FAILURE);
}
```

Sau đó thực hiện trao đổi khóa giữa Client và Server:

```
/* Xác thực người dùng. */
if (ssh_userauth_none(session, NULL) != SSH_AUTH_SUCCESS) {
    fprintf(stderr, "Xác thực thất bại: %s\n", ssh_get_error(session));
    ssh_disconnect(session);
    ssh_free(session);
    ssh_bind_free(sshbind);
    exit(EXIT_FAILURE);
}
printf("Client đã được xác thực thành công.\n");
```

Cuối cùng chúng em xác thực người dùng:

```

/* Xác thực người dùng. */
if (ssh_userauth_none(session, NULL) != SSH_AUTH_SUCCESS) {
    fprintf(stderr, "Xác thực thất bại: %s\n", ssh_get_error(session));
    ssh_disconnect(session);
    ssh_free(session);
    ssh_bind_free(sshbind);
    exit(EXIT_FAILURE);
}
printf("Client đã được xác thực thành công.\n");

```

## b) Xây dựng đoạn mã truyền tệp an toàn Server

Đầu tiên chúng em tạo một phiên Secure Copy File (SCP):

```

printf("Nhập thư mục lưu file trên server: ");
scanf("%255s", remote_directory);

/* Tạo phiên SCP ở chế độ đọc để nhận file */
ssh_scp scp = ssh_scp_new(session, SSH_SCP_READ, remote_directory);
if (scp == NULL) {
    fprintf(stderr, "Không thể tạo phiên SCP: %s\n", ssh_get_error(session));
    ssh_disconnect(session);
    ssh_free(session);
    ssh_bind_free(sshbind);
    exit(EXIT_FAILURE);
}
if (ssh_scp_init(scp) != SSH_OK) {
    fprintf(stderr, "Không khởi tạo được SCP: %s\n", ssh_get_error(session));
    ssh_scp_free(scp);
    ssh_disconnect(session);
    ssh_free(session);
    ssh_bind_free(sshbind);
    exit(EXIT_FAILURE);
}

```