

**ĐẠI HỌC QUỐC GIA THÀNH PHỐ HỒ CHÍ MINH**  
**TRƯỜNG ĐẠI HỌC KHOA HỌC TỰ NHIÊN**  
**KHOA CÔNG NGHỆ THÔNG TIN**



**ĐỒ ÁN THỰC HÀNH**  
**SOCKET**

**MÔN: MẠNG MÁY TÍNH**

Thành phố Hồ Chí Minh - 2024

**ĐẠI HỌC QUỐC GIA THÀNH PHỐ HỒ CHÍ MINH**  
**TRƯỜNG ĐẠI HỌC KHOA HỌC TỰ NHIÊN**  
**KHOA CÔNG NGHỆ THÔNG TIN**



**ĐỒ ÁN THỰC HÀNH**  
**SOCKET**

**GVHD: Ths. Nguyễn Thanh Quân và ThS. Lê Hà Minh**

Danh sách sinh viên thực hiện:

1. Trương Thanh Thuận
2. Nguyễn Kim Quốc 23120347
3. Nguyễn An Trí

Thành phố Hồ Chí Minh – 2024

MỤC LỤC

**Thông tin nhóm và bảng phân công công việc**

Họ và tên	MSSV
<b>Trương Thanh Thuận</b>	
<b>Nguyễn Kim Quốc</b>	
<b>Nguyễn An Trí</b>	

	Chức năng	Cơ bản	Mở rộng	MSSV
1	Kết nối			
2	Quản lý kết nối			
3	Đăng nhập			
4	Đăng ký			
5	Tra cứu			
6	Quản lý cơ sở dữ liệu			
7	Thoát			
8	Giao diện			

Mức độ hoàn thành

Chức năng	Mức độ hoàn thành	Chưa hoàn thành
1	100%	
2	100%	
3	100%	
4	100%	
5	90%	Server cập nhật thông tin liên tục 60 phút 1 lần
6	100%	
7	100%	
8	100%	

---

## I. Cấu trúc chương trình

---

### 1. Giới thiệu:

- Ứng dụng của nhóm là một hệ thống tải lên (upload) và tải xuống (download) tệp thông qua kết nối socket. Phần client và server sử dụng giao thức TCP/IP để truyền tải dữ liệu. Phần giao diện người dùng (GUI) sử dụng thư viện Tkinter để quản lý các chức năng liên quan đến đăng nhập và thao tác với các tệp.

### 2. Cấu trúc chương trình :

- Chương trình chia ra làm hai phần chính là client và server. Dưới đây là cấu trúc chương trình:

#### a. Phía server:

- Khởi tạo socket
- Chấp nhận kết nối
- Nhận và gửi tệp
- Thông báo lỗi và đóng kết nối

#### b. Phía client:

- Khởi tạo socket
- Gửi và nhận tệp
- Chọn chế độ Download/Upload
- Hiển thị thông tin

### 3. Phần giao diện người dùng:

- Login và đăng ký: một giao diện cho phép người dùng đăng nhập và đăng ký. Nếu đăng nhập sai thì chương trình báo lỗi. Đăng nhập đúng thì vào phần màn hình chính.
- Màn hình chính: nơi chứa các nút bấm để người dùng có thể lựa chọn hỗ trợ cho việc UPLOAD/DOWNLOAD một cách dễ dàng.

---

## II. Cách thức hoạt động của mỗi phần chính

---

### 1. Server:

- Tóm tắt quá trình: Khởi tạo socket → Lắng nghe kết nối → Chấp nhận kết nối từ client → Xử lý yêu cầu upload/download → Gửi/nhận tệp → Đóng kết nối.

#### a. Khởi tạo socket và lắng nghe kết nối:

- Server bắt đầu quá trình bằng cách tạo một socket mới sử dụng hàm `socket.socket()`. Socket này sử dụng giao thức TCP/IP, đảm bảo việc truyền tải dữ liệu ổn định.
- Sau khi tạo socket, server gọi `bind()` để gán một địa chỉ IP và cổng, cho phép client có thể tìm đến server qua địa chỉ này.
- Sau khi gán địa chỉ, server gọi `listen()` để chuyển sang chế độ chờ và lắng nghe các kết nối từ client.

#### b. Chấp nhận kết nối:

- Khi client kết nối đến server, server sử dụng phương thức `accept()` để chấp nhận kết nối. Lúc này, server nhận được một cặp socket (mới) đại diện cho kết nối giữa client và server.
- Server tiếp tục duy trì kết nối này để giao tiếp với client, cho phép thực hiện các thao tác upload hoặc download.

#### c. Xử lý yêu cầu Upload hoặc Download:

- Upload (Nhận tệp từ Client)
  - Nếu client gửi yêu cầu upload, server nhận dữ liệu từ client theo từng phần nhỏ theo phương thức `recv()`. Dữ liệu này được chia thành các đoạn nhỏ(chunk) để dễ dàng truyền tải qua mạng. Sau đó server lưu các đoạn dữ liệu này vào ổ đĩa, tạo thành một tệp hoàn chỉnh trên hệ thống.
- Download (Gửi tệp đến Client)
  - Nếu client gửi yêu cầu download, server mở tệp(file, folder) và chia nó thành các phần nhỏ. Dữ liệu tệp này sẽ được gửi tới client thông qua phương thức `send()`. Server gửi từng phần dữ liệu cho client.

#### d. Đóng kết nối:

- Sau khi hoàn tất việc nhận, gửi tệp thì server sẽ gọi `close()` để đóng kết nối. Điều này đảm bảo rằng tài nguyên của server được giải phóng sau mỗi lần giao tiếp.

### 2. Client:

- Tóm tắt quá trình: Tạo socket và kết nối tới server → Gửi yêu cầu upload/download → Gửi hoặc nhận tệp → Đóng kết nối.

## a. Kết nối tới server:

- Client tạo một socket TCP mới bằng `socket.socket()`. Sau đó, client sử dụng phương thức `connect()` để kết nối tới server thông và địa chỉ IP và cổng mà server đã chỉ định
- Sau khi kết nối thành công, client có thể gửi yêu cầu upload hoặc download tới server

## b. Gửi hoặc nhận tệp :

## • Upload:

- Client chọn tệp(folder) cần upload và đọc nội dung tệp đó từ ổ đĩa
- Client gửi tệp này qua kết nối socket, truyền tải dữ liệu theo từng phần (buffer) qua phương thức `send()`
- Quá trình gửi dữ liệu tiếp tục cho đến khi toàn bộ tệp được gửi hoàn toàn đến server.

## • Download:

- Client gửi yêu cầu đến server để yêu cầu download tệp.
- Khi server phản hồi, client nhận dữ liệu tệp qua phương thức `recv()`. Client sẽ nhận từng phần của tệp và lưu lại vào ổ đĩa của mình.
- Sau khi nhận toàn bộ tệp, client hoàn tất quá trình download.

## c. Đóng kết nối:

- Khi client hoàn tất thao tác upload hoặc download, client sẽ gọi `close()` để đóng kết nối với server. Điều này đảm bảo rằng tài nguyên của client được giải phóng và không có kết nối mở không cần thiết.

---

### III. Quy trình xử lý các yêu cầu tải lên, tải xuống và lỗi

---

## a. Quy trình xử lý yêu cầu tải lên và xử lý lỗi:

## • Bắt đầu yêu cầu:

- Client gửi yêu cầu upload đến Server bao gồm tên FILE và dung lượng của FILE( hoặc FOLDER)



- Server xác nhận khả năng xử lý:
  - + Kiểm tra dung lượng lưu trữ trên máy chủ.
  - + Xem xét tên tệp để tránh ghi đè tên tệp đã tồn tại
- Nếu yêu cầu hợp lệ, Server gửi phản hồi chấp nhận đến Client.
  - Truyền dữ liệu:
- Client chia nhỏ tệp thành các phần (chunk) và gửi lần lượt qua Socket.
- Server nhận từng phần ghi vào tệp đích trên hệ thống, đồng thời gửi phản hồi xác nhận từng phần nhận thành công.
  - Hoàn tất:
- Sau khi gửi toàn bộ tệp, client thông báo hoàn thành.
- Server kiểm tra tính toàn vẹn tệp (so sánh dung lượng hoặc checksum). Nếu thành công, gửi phản hồi xác nhận
  - Xử lý lỗi:
- Nếu mà mất kết nối, cả hai bên sẽ dừng tiến trình và lưu trạng thái:
  - + Client lưu trạng thái phần đã gửi để tiếp tục lần sau
  - + Server lưu trạng thái phần đã nhận và chờ kết nối lại
- Nếu xảy ra lỗi tệp (hỏng hoặc không tồn tại), Server gửi thông báo lỗi và yêu cầu Client gửi lại.

b. Quy trình xử lý yêu cầu tải xuống và xử lý lỗi:

- Bắt đầu yêu cầu:
- Client gửi yêu cầu download đến Server.
- Server mở bảng chọn các tệp từ thư mục lên hiện trên màn hình để cho Client có thể lựa chọn file để download.
- Truyền dữ liệu:

- Server chia nhỏ tệp thành các phần chunk và gửi từng phần socket.
- Client nhận từng phần ghi vào tệp đích và gửi xác nhận thành công về cho server.
  - Hoàn tất:
- Khi gửi xong toàn bộ dữ liệu, Server gửi thông báo hoàn tất.
- Client kiểm tra tính toàn vẹn tệp (checksum hoặc kích thước) và gửi xác nhận cuối cùng
  - Xử lý lỗi
- Nếu mất kết nối, Client lưu trạng thái phần đã nhận và gửi yêu cầu tiếp tục khi kết nối được khôi phục.
- Nếu server gặp lỗi tệp, Server gửi thông báo lỗi để Client hủy hoặc chọn tệp khác.

#### IV. Các thư viện và công nghệ đã sử dụng

```

import socket    # tạo kết nối giữa client và server
import os        # thao tác với hệ thống tệp(chẳng hạn duyệt thư mục, kiểm tra kích thước..)
import threading # Tạo các luồng (threads) để thực hiện nhiều tác vụ đồng thời. Ví dụ như nhiều
                 # ví dụ như nhiều kết nối cùng lúc
import time      # Xử lý các tác vụ liên quan đến thời gian, như tạo độ trễ, tính toán thời gian
import sys       # Cung cấp quyền truy cập vào các tham số và hàm liên quan đến hệ thống
import tkinter as tk # Tạo đồ họa giao diện cho ứng dụng
from tkinter.messagebox import askyesno # hiển thị thông báo hoặc câu hỏi trong GUI
from tqdm import tqdm # Hiển thị thanh tiến trình
import logging   # Quản lý việc ghi log để theo dõi hoạt động( debug, info, error)
from logging import basicConfig # Thiết lập cơ cấu cho bản ghi log
from tkinter import filedialog # Mở hộp thoại để chọn tên thư mục
from tkinter import scrolledtext # Tạo vùng văn bản cuộn
from tkinter import *
from tkinter import messagebox
from PIL import Image, ImageTk # Thư viện xử lý hình ảnh dùng để mở thay đổi kích thước và
                                # hiển thị ảnh trong giao diện

from time import sleep
from datetime import datetime # Lấy và xử lý thời gian

```

---

## **V. Các vấn đề gặp phải trong quá trình thực hiện và cách khắc phục**

---

### **1. Kết nối mạng không ổn định:**

- Mô tả vấn đề:
  - + Mất kết nối giữa client và server khi đang thực hiện việc truyền nhận dữ liệu
- Nguyên nhân: mạng yếu hoặc mất kết nối.
- Cách khắc phục:
  - + Thiết lập timeout
  - + Sử dụng cơ chế thử lại khi kết nối thất bại (retry)

### **2. Quá tải server khi xử lý nhiều kết nối:**

- Mô tả vấn đề:
  - + Server bị treo hoặc phản hồi chậm khi nhiều client kết nối đồng thời
- Nguyên nhân:
  - + Do server không tối ưu xử lý đa luồng
  - + Tài nguyên hệ thống bị giới hạn
- Cách khắc phục:
  - + Sử dụng threading để xử lý nhiều kết nối đồng thời.

### **3. Lỗi không tìm thấy tệp:**

- Mô tả vấn đề: client yêu cầu tải xuống tệp không tồn tại trên server
- Nguyên nhân: client không kiểm tra danh sách các tệp có trên server khi yêu cầu
- Cách khắc phục: hiển thị bảng chọn những tệp có trên server. Sau đó cho client chọn tệp cần tải về xuống.

#### **4. Giao diện UI không ổn định**

- Mô tả: bị lỗi treo hoặc không cập nhật kịp thời khi xử lý upload/download
- Nguyên nhân: thao tác quá nặng
- Cách khắc phục: sử dụng threading để xử lý các tác vụ nặng trên luồng phụ. Cập nhật giao diện.

---

## TÀI LIỆU THAM KHẢO

---

[1]	<a href="https://www.youtube.com/watch?v=yQSEXcf6s2I&amp;list=PLCC34OHNcOtoC6GglhF3ncJ5rLwQrLGnV">https://www.youtube.com/watch?v=yQSEXcf6s2I&amp;list=PLCC34OHNcOtoC6GglhF3ncJ5rLwQrLGnV</a>
[2]	<a href="https://github.com/duchieuvn/SOCKET-PUBLIC/blob/main/source%20code/client.py#L264">https://github.com/duchieuvn/SOCKET-PUBLIC/blob/main/source%20code/client.py#L264</a>
[3]	<a href="https://www.pythontutorial.net/tkinter/tkinter-treeview/">https://www.pythontutorial.net/tkinter/tkinter-treeview/</a>
[4]	<a href="https://drive.google.com/file/d/1A1IsrfZuzOxfEaW3ukxblOfVYUmG1T8i/view">https://drive.google.com/file/d/1A1IsrfZuzOxfEaW3ukxblOfVYUmG1T8i/view</a>





