import os

import socket

import threading

import time

IP = "localhost"

PORT = 4450

ADDR = (IP, PORT)

SIZE = 1024

FORMAT = "utf-8"

SERVER\_PATH = "server\_files" # Directory for storing server files

def handle\_client(conn, addr):

print(f"[NEW CONNECTION] {addr} connected.")

conn.send("OK@Welcome to the file sharing server".encode(FORMAT))

while True:

try:

data = conn.recv(SIZE).decode(FORMAT)

cmd, \*args = data.split("@")

send\_data = "OK@"

if cmd == "LOGOUT":

break

elif cmd == "UPLOAD":

filename, filesize = args

filesize = int(filesize)

if os.path.exists(os.path.join(SERVER\_PATH, filename)):

conn.send(f"ERROR@File {filename} already exists.".encode(FORMAT))

else:

with open(os.path.join(SERVER\_PATH, filename), "wb") as f:

received = 0

while received < filesize:

bytes\_read = conn.recv(SIZE)

f.write(bytes\_read)

received += len(bytes\_read)

conn.send(f"OK@Upload of {filename} complete.".encode(FORMAT))

elif cmd == "DOWNLOAD":

filename = args[0]

filepath = os.path.join(SERVER\_PATH, filename)

if os.path.exists(filepath):

filesize = os.path.getsize(filepath)

conn.send(f"OK@{filesize}".encode(FORMAT))

with open(filepath, "rb") as f:

while (bytes\_read := f.read(SIZE)):

conn.sendall(bytes\_read)

else:

conn.send(f"ERROR@File {filename} does not exist.".encode(FORMAT))

elif cmd == "DELETE":

filename = args[0]

filepath = os.path.join(SERVER\_PATH, filename)

if os.path.exists(filepath):

os.remove(filepath)

conn.send(f"OK@File {filename} deleted.".encode(FORMAT))

else:

conn.send(f"ERROR@File {filename} not found.".encode(FORMAT))

elif cmd == "DIR":

files = os.listdir(SERVER\_PATH)

send\_data += ", ".join(files) if files else "No files available."

conn.send(send\_data.encode(FORMAT))

except Exception as e:

conn.send(f"ERROR@{str(e)}".encode(FORMAT))

break

print(f"{addr} disconnected")

conn.close()

def main():

if not os.path.exists(SERVER\_PATH):

os.makedirs(SERVER\_PATH)

print("Starting the server")

server = socket.socket(socket.AF\_INET, socket.SOCK\_STREAM)

server.bind(ADDR)

server.listen()

print(f"Server is listening on {IP}: {PORT}")

while True:

conn, addr = server.accept()

thread = threading.Thread(target=handle\_client, args=(conn, addr))

thread.start()

if \_\_name\_\_ == "\_\_main\_\_":

main()