ПРИЛОЖЕНИЕ Б

(обязательное)

Исходный текст программной части

Файл main.py:

import socket

from commands import client\_commands

import os

import os.path

import sys

import errno

import time

import re

HOST = ''

PORT = 9001

BUFFER\_SIZE = 1024

TIMEOUT = 20

OK\_STATUS = 200

def wait\_ok():

while (client.recv(2).decode('utf-8') != "OK"):

print("wait for OK")

def send\_ok():

client.send("OK".encode('utf-8'))

def get\_data():

return client.recv(BUFFER\_SIZE).decode('utf-8')

def send\_data(data):

client.send(str(data).encode('utf-8'))

def handle\_input\_request(request):

command = request.split()

name\_command = command[0]

if (len(command) == 2):

body = command[1]

if (client\_commands.get(name\_command) == "echo"):

send\_data(request)

if (wait\_for\_ack(name\_command) == False):

return

echo(body)

if (client\_commands.get(name\_command) == "time"):

send\_data(request)

if (wait\_for\_ack(name\_command) == False):

return

get\_time()

if (client\_commands.get(name\_command) == "download"):

send\_data(request)

if (wait\_for\_ack(name\_command) == False):

return

download(body, request)

if (client\_commands.get(name\_command) == "upload"):

if (is\_file\_exist(body)):

send\_data(request)

if (wait\_for\_ack(name\_command) == False):

return

upload(body, request)

else:

show\_error\_message("No such file exists")

if (client\_commands.get(name\_command) == "delete"):

send\_data(request)

if (wait\_for\_ack(name\_command) == False):

return

delete(body, request)

if (client\_commands.get(name\_command) == "exit"):

send\_data(request)

if (wait\_for\_ack(name\_command) == False):

return

client.close()

os.\_exit(1)

def wait\_for\_ack(command\_to\_compare):

while True:

response = client.recv(BUFFER\_SIZE).decode('utf-8').split(" ", 2)

if not response:

return False

sent\_request = response[0]

status = response[1]

if (len(response) > 2):

message = response[2]

else:

message = None

if (command\_to\_compare == sent\_request and int(status) == OK\_STATUS):

return True

elif (message):

print(message)

return False

else:

return False

def is\_server\_available(request, command):

global client

client.close()

i = TIMEOUT

while(i > 0):

try:

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

client.connect((HOST, PORT))

client.send(request.encode('utf-8'))

wait\_for\_ack(command)

return True

except socket.error as er:

sys.stdout.write("Waiting for a server: %d seconds \r" % i)

sys.stdout.flush()

i -= 1

time.sleep(1)

sys.stdout.flush()

print("\nServer was disconnected")

sys.stdout.flush()

return False

def is\_file\_exist(file\_name):

return os.path.exists(file\_name)

def echo(body):

send\_data(body)

print(get\_data())

def get\_time():

print(get\_data())

def download(file\_name, request):

size = int(get\_data()) # 1

send\_data(0) # 3

data\_size\_recv = int(get\_data()) # 4

if (data\_size\_recv == 0):

f = open(file\_name, "wb")

else:

f = open(file\_name, "rb+")

while (data\_size\_recv < size):

try:

data = client.recv(BUFFER\_SIZE)

f.seek(data\_size\_recv, 0)

f.write(data)

data\_size\_recv += len(data)

progress = (data\_size\_recv / (size)) \* 100

sys.stdout.write("Download progress: %d%% \r" % progress)

sys.stdout.flush()

except socket.error as e:

if(is\_server\_available(request, "download")):

size = int(get\_data())

send\_data(data\_size\_recv)

data\_size\_recv = int(get\_data())

print("\n")

else:

f.close()

client.close()

os.\_exit(1)

except KeyboardInterrupt:

print("KeyboardInterrupt was handled")

f.close()

client.close()

os.\_exit(1)

f.close()

print("\n" + file\_name + " was downloaded")

def upload(file\_name, request):

f = open(file\_name, "rb+")

size = int(os.path.getsize(file\_name))

send\_data(size) # 1

wait\_ok() # 2

send\_data(0) # 3

data\_size\_recv = int(get\_data()) # 4

wait\_ok() # 5

f.seek(data\_size\_recv, 0)

while (data\_size\_recv < size):

try:

data\_file = f.read(BUFFER\_SIZE)

client.send(data\_file)

received\_data = get\_data()

progress = (data\_size\_recv / size) \* 100

sys.stdout.write("Upload progress: %d%% \r" % progress)

sys.stdout.flush()

except socket.error as e:

if(is\_server\_available(request, "upload")):

send\_data(size)

wait\_ok()

send\_data(data\_size\_recv)

data\_size\_recv = int(get\_data())

wait\_ok()

print("\n")

else:

f.close()

client.close()

os.\_exit(1)

except KeyboardInterrupt:

print("KeyboardInterrupt was handled")

f.close()

client.close()

os.\_exit(1)

if (received\_data):

data\_size\_recv = int(received\_data)

f.seek(data\_size\_recv, 0)

f.close()

print("\n" + file\_name + " was uploaded")

def delete(file\_name):

pass

def exit():

pass

def check\_valid\_request(request):

command = request.split()

if (len(command) == 0):

return False

else:

return True

def show\_status():

pass

def show\_error\_message(error):

print(error)

def show\_start\_message():

print("\nWelcome to client cli!")

is\_valid\_address = False

REGULAR\_IP = '^(([0-9]|[1-9][0-9]|1[0-9]{2}|2[0-4][0-9]|25[0-5])\.){3}([0-9]|[1-9][0-9]|1[0-9]{2}|2[0-4][0-9]|25[0-5])$'

regex = re.compile(REGULAR\_IP)

while (is\_valid\_address == False):

addr = input("\nInput host addres: ")

if (regex.match(addr)):

is\_valid\_address = True

HOST = addr

else:

try:

HOST = socket.gethostbyname(addr)

is\_valid\_address = True

except socket.error:

print("Please, input valid address")

is\_valid\_address = False

show\_start\_message()

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

client.connect((HOST, PORT))

while True:

try:

request = input()

if (check\_valid\_request(request)):

handle\_input\_request(request)

except KeyboardInterrupt:

print("KeyboardInterrupt was handled")

client.close()

os.\_exit(1)