**ESCOM**

Practica 1

Axel

Estrada Pichardo Jonatan Isúi

Andrés Rodarte López

3CM#

Introducción

Los sockets TCP son puertos virtuales que son utilizados en comunicaciones TCP para identificar conexiones únicas. Son puertos virtuales por que un solo conector físico puede servir varias conexiones. Cada lado de la conexión usa su propio puerto, el que no cambia durante el ciclo de vida de la conexión. El puerto y la IP identifican a un servidor. Cuando se junta el cliente y el servidor se puede hablar de que se creó un socket de cada lado.

Desarrollo

Esta es una aplicación que permite al cliente guardar archivos en el servidor y se envían a través de un socket

**Servidor**

#!/usr/bin/env python3

import socket

import os

import pickle #serialize the list to send it tru socket

import zipfile

HOST = '127.0.0.1' # Standard loopback interface address (localhost)

PORT = 65432 # Port to listen on (non-privileged ports are > 1023)

f = 0

dataSend = 0

ServerDirectory = './ServerDummy/'

option = -1

fileList = 0

def folderContent():

fileList = os.listdir(ServerDirectory)

conn.sendall(pickle.dumps(fileList))

def UploadAFile():

flnm = 0

while True:

data = conn.recv(1024)

if flnm == 0:

f = open(ServerDirectory + data.decode(), 'wb')

flnm = flnm + 1

else:

f.write(data)

if len(data) < 1024:

flnm = 0

break

f.close()

conn.sendall(b'Send was succesful\n')

print("Done Receiving")

fantasy\_zip = zipfile.ZipFile(ServerDirectory + '/archive.zip')

fantasy\_zip.extractall(ServerDirectory)

fantasy\_zip.close()

os.remove( ServerDirectory + '/archive.zip')

def DownloadFile():

folderContent()

filesname = pickle.loads(conn.recv(1024))

for item in filesname:

file2D = conn.recv(1024).decode()

#print('Se enviara el archvio:', fileList[file2D])

f = open(ServerDirectory + os.path.basename(file2D), 'rb')

chonk = f.read(1024)

while chonk:

conn.sendall(chonk)

chonk = f.read(1024)

with socket.socket(socket.AF\_INET, socket.SOCK\_STREAM) as s:

if not os.path.exists(ServerDirectory):

os.mkdir(ServerDirectory)

s.setsockopt(socket.SOL\_SOCKET, socket.SO\_REUSEADDR, 1)

s.bind((HOST, PORT))

s.listen()

fileList = os.listdir(ServerDirectory)

switcher = {

0: folderContent,

1: UploadAFile,

2: DownloadFile

}

print('Welcome to the DropPle')

while True:

conn, addr = s.accept()

with conn:

print('Direccion', addr)

while True:

if option < 0:

option = int(conn.recv(1024).decode())

elif option >= 3:

option = -1

break

else:

switcher[option]()

option = -1

#while True:

s.close()

#

**Cliente**

#!/usr/bin/env python3

from tkinter import Tk

from tkinter.filedialog import askopenfilenames

from tkinter.filedialog import askdirectory

import zipfile

import pickle

import socket

import sys

import os

import shutil

HOST = '127.0.0.1' # The server's hostname or IP address

PORT = 65432 # The port used by the server

option = -1

fileList = 0

ServerDirectory = './.TempServerDummy/'

def folderContent():

fileList = pickle.loads(s.recv(1024))

print('\nEl directorio contiene:')

for i in range(len(fileList)):

print('\t ', i, ')', fileList[i])

print()

return fileList

def UploadAFile():

Tk().withdraw() # we don't want a full GUI, so keep the root window from appearing

filename = askopenfilenames(title='Select the files') # show an "Open" dialog box and return the path to the selected file

fantasy\_zip = zipfile.ZipFile('./archive.zip', 'w')

for \_file in list(filename):

fantasy\_zip.write(\_file, os.path.basename(\_file))

fantasy\_zip.close()

#print('Se enviara el archvio:', os.path.basename(filename))

f = open('./archive.zip', 'rb')

s.sendall('archive.zip'.encode())

chonk = f.read(1024)

while chonk:

s.sendall(chonk)

chonk = f.read(1024)

print(s.recv(1024).decode())

f.close()

#while True:

# try:

os.remove('./archive.zip')

# break

# except:

# print('Elimiando')

def DownloadFile():

fileList = folderContent()

if not os.path.exists(ServerDirectory):

os.mkdir(ServerDirectory)

for elem in fileList:

f = open(ServerDirectory + elem, 'wb')

f.close()

Tk().withdraw() # we don't want a full GUI, so keep the root window from appearing

filename = askopenfilenames(initialdir = ServerDirectory,title='Select the files')

s.sendall(pickle.dumps(filename))

#file2D = input('Choose the file >')

dirname = askdirectory(title='Open the folder to save')

for target\_list in filename:

s.sendall(target\_list.encode())

f = open(dirname + '/' + os.path.basename(target\_list), 'wb')

while True:

data = s.recv(1024)

f.write(data)

if len(data) < 1024:

break

f.close()

print("Done Receiving", end='\n\n')

shutil.rmtree(ServerDirectory, ignore\_errors=True)

with socket.socket(socket.AF\_INET, socket.SOCK\_STREAM) as s:

switcher = {

0: folderContent,

1: UploadAFile,

2: DownloadFile

}

s.connect((HOST, PORT))

print('Welcome to the DropPle')

while True:

if option < 0:

option = input('Select 0 to see what is in the folder\nSelect 1 to upload a file\nSelect 2 to dowload a file\n> ')

s.sendall(option.encode())

option = int(option)

elif option == 3:

break

else:

switcher[option]()

option = -1

#

s.close()

Conclusiones

Los sockets son útiles para mandar datos a través del internet, es importante definir bien que datos vamos a mandar para que el servidor pueda hacer el uso correcto de los datos que reciba. También debemos tomar en cuenta el tamaño de lo que queremos mandar por el socket, las tramas TCP solo pueden contener un máximo de 1600 kilobytes por lo que si el archivo que deseamos mandar pesa mas que eso, debemos separarlo en partes para poder enviarlo.