|  |  |
| --- | --- |
|  | Sockets En python |
|  |  |
| P.1.4 | Laboratorio de Redes y Servicios |
|  | INTEGRANTES DEL GRUPO:  Nombre: MANUEL MONTOYA CATALÁ  Nombre: ANDRÉS BEATO OLLERO |

ÍNDICE

[INTRODUCCIÓN 2](#_Toc388182928)

[Código del servidor 3](#_Toc388182929)

[Código del CLIENTE 3](#_Toc388182930)

[bibliografía 4](#_Toc388182931)

# INTRODUCCIÓN

En esta práctica vamos a realizar una implementación similiar al programa realizado en la práctica 1.1, pero realizado en el lenguaje Python en vez de C. El objetivo de esta práctica es familiarizarse con la programación de aplicaciones de red con Python y compararla con C. No explicaremos en esta práctica el concepto de socket y de su lógica de funcionamiento ya que estos conceptos se explicaron en la P1.1, si bien el juego de funciones de sockets de Python es diferente y de más alto nivel que C. Tampoco explicaremos en esta práctica los principios de la programación en Python en sí, nos centraremos solo a las funciones de sockets y su comparativa con los de C.

Lo primero que tenemos que hacer es importar la librería de sockets para poder utilizar las funciones:

**import** socket

A continuación explicaremos las funciones de esta librería que hemos utilizado en la práctica:

- **Funciones para ponernos a escuchar por un puerto:**

srvsock **=** socket**.**socket**(** socket**.**AF\_INET**,** socket**.**SOCK\_STREAM **)**

srvsock**.**bind**(** **(**''**,** 2000 **)**

srvsock**.**listen**(** 5 **)**

Estas tres funciones:

1) Crea un socket TCP/IP 🡪 socket() de C

2) Asocia al socket el puerto 2000 🡪 bind() de C

3) Pone al socket a la escucha. 🡪 listen() de C

En cuanto a las funciones, su utilizacion y parametros son muy similares, sin embargo, en C se requiere una serie de estructuras sockaddr\_in que hay que inicializar y utilizar.

**- Funciones para conectarse a un servidor:**

clisock **=** socket**.**socket**(** socket**.**AF\_INET**,** socket**.**SOCK\_STREAM **)**

clisock**.**connect**(** **(**dir\_IP**,** int **(**sys**.**argv**[**2**]))** **)**

Estas dos funciones:

1) Crea un socket TCP/IP 🡪 socket() de C

2) Conectarse a una maquina 🡪 connect() de C

En C se requiere una serie de estructuras sockaddr\_in que hay que inicializar y utilizar.

**- Funciones de envío y recepcón de datos:**

clisock**.**send**(** str\_env**)**

str **=** clisock**.**recv**(**100**)**

Son parecidas a las funciones de C, solo que envían y reciben cadenas:

1) Envia la cadena 🡪 send() de C

2) Recibe la cadena 🡪 recv() de C

# Código del servidor

Código del fichero ping\_server.py.

**import** socket

**import** sys

**import** atexit

**if(**len**(**sys**.**argv**)** **>** 1**):**

**print** 'servidor escuchando en puerto :'**,** sys**.**argv**[**1**]**

srvsock **=** socket**.**socket**(** socket**.**AF\_INET**,** socket**.**SOCK\_STREAM **)**

srvsock**.**bind**(** **(**''**,** int **(**sys**.**argv**[**1**]))** **)**

srvsock**.**listen**(** 5 **)**

clisock**,** **(**remhost**,** remport**)** **=** srvsock**.**accept**()**

**while** 1**:**

str **=** clisock**.**recv**(**100**)**

str\_env **=** 'Packet Received with contnent: ' **+** str

**print** str\_env

clisock**.**send**(** str\_env**)**

**def** exit\_func**():**

clisock**.**close**()**

atexit**.**register**(**exit\_func**)**

# Código del CLIENTE

Código del fichero ping.py.

**import** socket

**import** sys

**import** time

**import** atexit

**if(**len**(**sys**.**argv**)** **>** 1**):**

dir\_IP **=** socket**.**gethostbyname**(**sys**.**argv**[**1**])**

**print** 'conexion con servidor'**,** sys**.**argv**[**1**],** 'en el puerto :'**,** sys**.**argv**[**2**]**

clisock **=** socket**.**socket**(** socket**.**AF\_INET**,** socket**.**SOCK\_STREAM **)**

clisock**.**connect**(** **(**dir\_IP**,** int **(**sys**.**argv**[**2**]))** **)**

average\_delay **=** 0

packets\_sent **=** 0

**while** 1**:**

packets\_sent**+=** 1

start **=** time**.**time**()**

clisock**.**send**(**"TCP Ping Packet !!"**)**

answere **=** clisock**.**recv**(**100**)**

end **=** time**.**time**()**

average\_delay **+=** end **-** start

**print** "Packet sent with Return delay " **+** str**(**end **-** start**)**

time**.**sleep**(**1**)**

**def** exit\_func**():**

clisock**.**close**()**

average\_delay **/=** paquets\_sent

**print** "\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*"

**print** "The average delay is " **+** str**(**average\_delay**)**

**print** "\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*"

atexit**.**register**(**exit\_func**)**

# bibliografía

[1] Sockets programming in Python © Copyright IBM Corporation 1994, 2006.

[2] Documentación oficial de Python: <http://www.python.org/doc/>