

Tarea N°1

Implementación de un OUI Lookup Tool

Integrantes:

Miguel Escobar

Alexis Loyola

Matías Zapata

Profesor:

Gabriel Astudillo

Diseño de la solución

Como primer paso, se toma el archivo con los datos y se añaden a una lista, de forma que un dato de ella queda con la siguiente estructura `datox=[mac, vendedor, empresa]` e inicializamos nuestras variables (`ip`, `mac`, `vendedor`) como `None` (lo hacemos de esta forma para saber cuál es la que toma valor después). Usamos `getopt()` para hacer la traducción de argumentos bash a Python, definimos nuestras variables como `'i'` y `'m'` y su forma larga `'ip='`, `'mac='` y `'help'` respectivamente y los almacenamos en una lista, podemos notar que `help` no tiene una variable corta asignada, esto es porque no usará ningún tipo de argumento. Con un ciclo `for` se recorre la lista de opciones tomando en cuenta sus argumentos, si el comando ingresado es `'-help'` ejecutara una función que muestra como se usa el programa, si el comando es `'--ip'` asignará el valor del argumento a la variable `ip` y si el comando es `'--mac'` asignara el argumento a la variable `mac`, en caso de no seleccionar nada solo muestra como debe usarse el programa. Al ya tener una variable asignada, se usa un condicional para ver cual de las variables fue asignada, comprobando cual es distinta de `None`. Si `ip` tiene valor, verifica si se encuentra en la red local, de ser así, busca la `mac` y luego compara en la lista de datos si coincide con algún `dato[0]` (según la estructura ese índice corresponde a la `mac`), si coincide se muestra por pantalla la `mac` y su vendedor (`dato[1]`). En caso de no encontrarse en la misma red lo informa por pantalla. Por otro lado, si `mac` tiene valor asignado, solo buscamos si está `mac` coincide con alguna nuestra base de datos (`dato[0]`) y se muestra en pantalla junto a su vendedor (`dato[1]`), en caso de no encontrarse muestra la `mac` y que no existe vendedor.

Implementación

Para el correcto uso y funcionamiento del programa, el archivo “datos.txt” debe estar en la misma carpeta que el código, ya que funciona de manera local, la implementación de este mismo funciona en Linux y su forma de uso se detalla a continuación:

```
python3 OUILookup.py --ip<ip> --mac<mac> [--help]
```

parámetros:

ip: si es ingresada busca por el numero ingresado.

mac: si es ingresado busca por la dirección ingresada.

help: muestra pantalla de uso y termina.

Explicación del código:

Recursos:

```
11 import sys
12 import socket #para ver cual es la ip desde donde se esta ejecutando
13 import getopt
```

Funciones:

```
15 #verificacion de la misma red (si se encuentra en la misma retorna 1 y si no 0):
16 def mismared(ipx):
17     redlocal = socket.gethostbyname(socket.gethostname()) #muestra la ip del equipo usando el nombre
18     #SI SON DE LA MISMA RED LAS PRIMERAS TRES SECCIONES DEBEN SER IGUALES, LA CUARTA PUEDE VARIAR
19     iplocal = redlocal.split('.') #separamos el ip local en 4 secciones
20     ipbusca = ipx.split('.') #separamos el ip a buscar en 4 secciones
21     for i in range(3):
22         if (iplocal[i] != ipbusca[i]):#comparamos las tres primeras secciones
23             return 0
24     return 1
25 #buscar mac en la lista (si existe devuelve su indice, si no existe retorna -1)
26 def buscarmac(lista,mac):
27     for i in lista:
28         if mac in i:
29             return lista.index(i)
30     return -1
31 #toma los datos del archivo, los guarda y retorna en una lista.
32 def generarbase():
33     lista = []
34     archivo = open("datos.txt","r")
35     while True:
36         linea = archivo.readline()
37         dato = linea.split(' ')
38         lista.append(dato)
39         if not linea: break
40     archivo.close()
41     return lista
```

```
84 # Funcion para dar instrucciones:
85 def usar():
86     print("Uso: Python3 " + sys.argv[0] + " --ip <IP> | --mac <IP> [--help] ")
87     print("\nParametros:")
88     print("    --ip: especificar la IP del host.")
89     print("    --maxtime: especificar la direccion MAC")
90     print("    --help: muestra esta pantalla y termina.")
91     exit(1)
```

Se toman los datos del archivo y se guardan en una lista, con la función “generarbase()”.

Para verificar si la ip esta en la misma red tenemos la función “mismared(ipx)”, es más fácil gracias al recurso “socket” al permitir detectar la ip desde donde se está ejecutando.

Al buscar la dirección mac en la lista se reconoce y en caso de encontrar retorna valores (-1 si no se encuentra y el índice si se encuentra) para confirmarlo en la función “buscarmac(lista,mac)”.

Para el uso y los parámetros a mostrar se hace uso de la función “usar()”.

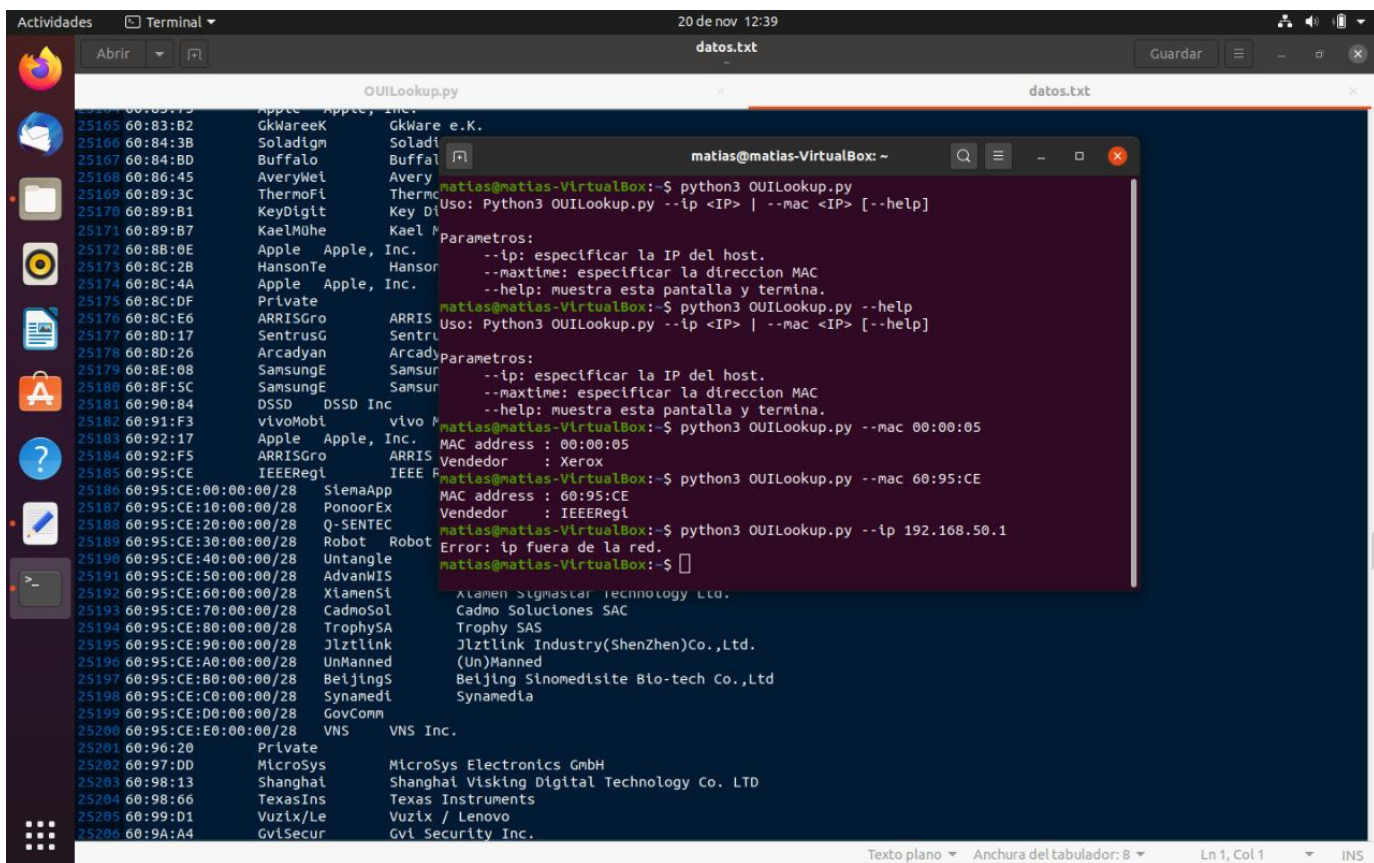
Main:

```
43 #CUERPO PRINCIPAL:
44 def main():
45     datos = generarbase() #creacion de base de datos
46     #variables
47     mac = None
48     ip = None
49     Vendedor = None
50
51     argv = sys.argv[1:]#toma argumentos desde el primero en adelante
52
53     try:
54         options, args = getopt.getopt(argv,"i,m",['ip=','mac=','help'])
55     except:
56         print("Error: parametros mal ingresados.")
57         usar()
58
59     for opt, arg in options:
60         if opt in ('--help'):
61             usar()
62         if opt in ('--ip'):
63             ip = str(arg)
64         elif opt in ('--mac'):
65             mac = str(arg)
66
67     #para ver cual dato fue ingresado.
68     if (ip != None):
69         if (mismared(ip)):
70             print("F")
71         else:
72             print("Error: ip fuera de la red.")
73     elif (mac != None):
74         indice = buscarmac(datos,mac)
75         if(indice !=-1):
76             print("MAC address : " + (datos[indice])[0])
77             print("Vendedor   : " + (datos[indice])[1])
78         else:
79             print("MAC address : " + mac)
80             print("Vendedor   : No Encontrado")
81     else:
82         usar()
```

Primero se genera la base (con la función previamente explicada) y se guarda en una variable, posteriormente se aplica las opciones mediante el “opt”.

Después de usan las condicionales para saber si los datos de ip y mac fueron ingresados correctamente, posteriormente se aplican las funciones de búsqueda en el archivo. Por ultimo se muestran por pantalla los datos requeridos y al final en caso de alguna entrada distinta, se muestra la función “usar()”, para mostrar el modo de uso.

Experimentación:



```
Actividades Terminal 20 de nov 12:39
datos.txt Guardar
OUILookup.py
25105 60:83:B2 GkWareK GkWare e.K.
25106 60:84:38 Soladign Soladign
25107 60:84:80 Buffalo Buffalo
25108 60:86:45 AveryWel AveryWel
25109 60:89:3C ThermoFl ThermoFl
25170 60:89:B1 KeyDigit KeyDigit
25171 60:89:B7 KaelMöhe KaelMöhe
25172 60:8B:0E Apple Apple, Inc.
25173 60:8C:2B HansorTe HansorTe
25174 60:8C:4A Apple Apple, Inc.
25175 60:8C:DF Private Private
25176 60:8C:E6 ARRISGro ARRISGro
25177 60:8D:17 SentrusG SentrusG
25178 60:8D:26 Arcadyan Arcadyan
25179 60:8E:08 SamsungE SamsungE
25180 60:8F:5C SamsungE SamsungE
25181 60:90:84 DSSD DSSD Inc
25182 60:91:F3 vivoMobi vivoMobi
25183 60:92:17 Apple Apple, Inc.
25184 60:92:F5 ARRISGro ARRISGro
25185 60:95:CE IEEERegl IEEERegl
25186 60:95:CE:00:00:00/28 SiemaApp SiemaApp
25187 60:95:CE:10:00:00/28 PonoorEx PonoorEx
25188 60:95:CE:20:00:00/28 Q-SENTEC Q-SENTEC
25189 60:95:CE:30:00:00/28 Robot Robot
25190 60:95:CE:40:00:00/28 Untangle Untangle
25191 60:95:CE:50:00:00/28 AdvanWIS AdvanWIS
25192 60:95:CE:60:00:00/28 XiamenSI XiamenSI
25193 60:95:CE:70:00:00/28 CadmoSol CadmoSol
25194 60:95:CE:80:00:00/28 TrophySA TrophySA
25195 60:95:CE:90:00:00/28 Jltzlink Jltzlink
25196 60:95:CE:A0:00:00/28 UnManned (Un)Manned
25197 60:95:CE:B0:00:00/28 BeijingS BeijingS
25198 60:95:CE:C0:00:00/28 Synamedia Synamedia
25199 60:95:CE:D0:00:00/28 GovComm GovComm
25200 60:95:CE:E0:00:00/28 VNS VNS Inc.
25201 60:96:20 Private Private
25202 60:97:00 MicroSys MicroSys Electronics GmbH
25203 60:98:13 Shanghal Shanghal Visking Digital Technology Co. LTD
25204 60:98:66 TexasIns Texas Instruments
25205 60:99:01 VuzixLe Vuzix / Lenovo
25206 60:9A:A4 GviSecur Gvi Security Inc.

matias@matias-VirtualBox: ~
matias@matias-VirtualBox:~$ python3 OUILookup.py
Uso: Python3 OUILookup.py --ip <IP> | --mac <IP> [--help]

Parametros:
--ip: especificar la IP del host.
--mac: especificar la direccion MAC
--help: muestra esta pantalla y termina.

matias@matias-VirtualBox:~$ python3 OUILookup.py --help
Uso: Python3 OUILookup.py --ip <IP> | --mac <IP> [--help]

Parametros:
--ip: especificar la IP del host.
--mac: especificar la direccion MAC
--help: muestra esta pantalla y termina.

matias@matias-VirtualBox:~$ python3 OUILookup.py --mac 00:00:05
MAC address : 00:00:05
Vendedor : Xerox

matias@matias-VirtualBox:~$ python3 OUILookup.py --mac 60:95:CE
MAC address : 60:95:CE
Vendedor : IEEERegl

matias@matias-VirtualBox:~$ python3 OUILookup.py --ip 192.168.50.1
Error: ip fuera de la red.

matias@matias-VirtualBox:~$
```

Pruebas con `--help`, `--mac 00:00:05`, `--mac 00:95:CE` y `--ip 192.168.50.1`

Diagrama de la solución

Posterior a la explicación de la tarea y el código, se presenta el diagrama de flujo de la solución.

