МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РФ

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ

ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ

«ВЯТСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»

ФАКУЛЬТЕТ КОМПЬЮТЕРНЫХ И ФИЗИКО-МАТЕМАТИЧЕСКИХ НАУК

КАФЕДРА ПРИКЛАДНОЙ МАТЕМАТИКИ И ИНФОРМАТИКИ

Отчёт по лабораторной работе №2 по дисциплине «Протоколы компьютерного взаимодействия»

**ICMP, ARP**

|  |  |
| --- | --- |
| Выполнил: студент группы ФИб-4302-51-00 | / Д.А. Савин / |
| Проверил: преподаватель каф. ПМиИ | / А.Б. Белиц / |

Киров 2021

Задание №1

Реализуйте отправку ICMP сообщений на заданный узел сети с получением ответов на эти сообщения. Аналог команды "ping адрес -t". Во время работы приложения отключите сеть, проанализируйте результаты. Программа должна быть реализована на языке Python, без прямого консольного вызова команды ping.

Код:

|  |
| --- |
| **from** **icmplib** **import** ping  print("Input address:")  address = input()  print("**\n**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**\n**")  **if** (address != " "):  print (ping(address, count=**10**, interval=**0.2**))  **else**:  print ("Вы ничего не ввели.") |

Пример работы:

Изображение выглядит как текст

Автоматически созданное описание

**Входные данные:**

Ya.ru

**Выходные данные:**

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

87.250.250.242

------------------------------------------------------------

Packets sent: 10

Packets received: 10

Packet loss: 0.0%

Round-trip times: 19.017 ms / 19.017 ms / 19.018 ms

Jitter: 0.0 ms

------------------------------------------------------------

Задание №2

Разработайте GUI приложение отправляющее ICMP сообщения и выводящее результат в таблицу. Аналог команды "tracert 8.8.8.8". Программа должна быть реализована на языке Python, без прямого консольного вызова команды tracert.

Код:

|  |
| --- |
| **from** **tkinter** **import** \*  **from** **tkinter.scrolledtext** **import** ScrolledText  **from** **icmplib** **import** traceroute  #adress = ping('google.com', count=10, interval=0.2)  **def** **Testproc**():  #txt.insert(INSERT, 'WAIT..')  tmpStr = message.get()    **if** (tmpStr != " " **or** tmpStr != ""):  txt.insert(INSERT, 'WAIT..')  hops = traceroute(tmpStr)  txt.delete(**1.0**, END)  txt.insert(INSERT, 'Distance/TTL Address Average round-trip time')  last\_distance = **0**  **for** hop **in** hops:  **if** last\_distance + **1** != hop.distance:  txt.insert(INSERT,'**\n**Some gateways are not responding')  txt.insert(INSERT, "**\n**")  txt.insert(INSERT, f'{hop.distance} {hop.address} {hop.avg\_rtt} ms')  last\_distance = hop.distance  window = Tk()  window.title("")  lbl = Label(window, text="Введите IP-адрес:")  lbl.place(relx=.**15**, rely=.**15**, anchor="c")  message = StringVar()  message\_entry = Entry(textvariable=message)  message\_entry.place(relx=.**5**, rely=.**15**, anchor="c")  Button\_1 = Button(window, text="Ввести", command=Testproc)  Button\_1.place(relx=.**85**, rely=.**15**, anchor="c")  txt = ScrolledText(window, width=**60**, height=**10**)  txt.place(relx=.**5**, rely=.**6**, anchor="c")  window.geometry(f"{500}x{250}+500+300")  window.mainloop() |

Пример работы:

Изображение выглядит как текст

Автоматически созданное описание

**Входные данные:**

8.8.8.8

**Выходные данные:**

Distance/TTL Address Average round-trip time

Some gateways are not responding

24 8.8.8.8 31.028 ms

Задание №3

Разработайте GUI приложение выводящее текущую ARP таблицу. Аналог команды "arp -a". Для реализации программы необходимо использовать язык Python и библиотеку PyQT.

Код:

|  |
| --- |
| **from** **ctypes** **import** \*  **from** **ctypes** **import** wintypes **as** w  **import** **struct**  **import** **sys**  **from** **PyQt5** **import** QtCore  **from** **PyQt5.QtWidgets** **import** (QPlainTextEdit, QWidget, QToolTip, QLineEdit, QPushButton, QApplication)  **from** **PyQt5.QtGui** **import** QFont  **from** **PyQt5** **import** QtWidgets  MAXLEN\_PHYSADDR = **8**  TYPE = {**1**:'other',**2**:'invalid',**3**:'dynamic',**4**:'static'}  **class** **MIB\_IPNETROW**(Structure):  \_fields\_ = (('dwIndex',w.DWORD),  ('dwPhysAddrLen',w.DWORD),  ('bPhysAddr',w.BYTE \* MAXLEN\_PHYSADDR),  ('dwAddr',w.DWORD),  ('dwType',w.DWORD))  **def** **\_\_repr\_\_**(self):  ip = struct.pack('<L',self.dwAddr)  ip = f'{ip[0]}.{ip[1]}.{ip[2]}.{ip[3]}'  mac = bytes(self.bPhysAddr)[:self.dwPhysAddrLen]  mac = '-'.join(f'{b:02x}' **for** b **in** mac)  **return** f"#| {mac}**\t**| {ip}**\t**|**\t**{TYPE[self.dwType]}"  **def** **TABLE**(n):  **class** **\_MIB\_IPNETTABLE**(Structure):  \_fields\_ = (('dwNumEntries',w.DWORD),  ('table',MIB\_IPNETROW \* n))  **return** \_MIB\_IPNETTABLE  **def** **Testproc**():  MIB\_IPNETTABLE = TABLE(**0**)  dll = WinDLL('iphlpapi')  dll.GetIpNetTable.argtypes = POINTER(MIB\_IPNETTABLE),w.PULONG,w.BOOL  dll.GetIpNetTable.restype = w.ULONG  size = w.DWORD(**0**)  dll.GetIpNetTable(**None**,byref(size),**True**)  buf = cast(create\_string\_buffer(b'',size=size.value),POINTER(MIB\_IPNETTABLE))  dll.GetIpNetTable(buf,byref(size),**True**)  buf = cast(buf,POINTER(TABLE(buf.contents.dwNumEntries)))  tmpStr =result  **for** t **in** buf.contents.table:  **if** t.dwType != **2** **and** t.dwPhysAddrLen:  tmpStr+=str(t)  tmpStr+='**\n**'    **return** tmpStr    result = "#|**\t**MAC**\t**|**\t**IP**\t**|**\t**Type**\n**"  **class** **Example**(QWidget):  **def** **\_\_init\_\_**(self):  super().\_\_init\_\_()  self.initUI()  **def** **initUI**(self):  self.setWindowTitle("ARP -a")  desktop = QtWidgets.QApplication.desktop()  QToolTip.setFont(QFont('SansSerif', **10**))  self.textbox = QPlainTextEdit(self)  self.textbox.move(**10**, **50**)  self.textbox.resize(**450**,**200**)    self.textbox.setReadOnly(**True**)      self.btn = QPushButton('Поиск', self)  self.btn.resize(self.btn.sizeHint())  self.btn.move(**10**, **10**)  self.btn.clicked.connect(self.click)  self.setGeometry((int)(desktop.width()/**2**) - (int)(self.width()/**2**), (int)(desktop.height()/**2**) - (int)(self.height()/**2**), **470**, **260**)  self.show()  **def** **click**(self):  result = Testproc()  self.textbox.clear()  self.textbox.insertPlainText(result)  **if** \_\_name\_\_ == '\_\_main\_\_':  app = QApplication(sys.argv)  ex = Example()  sys.exit(app.exec\_()) |

Пример работы:

Изображение выглядит как стол

Автоматически созданное описание

**Выходные данные:**

#| MAC | IP | Type

#| f0-b4-d2-42-a3-bb | 192.168.0.1 | dynamic

#| ff-ff-ff-ff-ff-ff | 192.168.0.255 | static

#| 01-00-5e-00-00-02 | 224.0.0.2 | static

#| 01-00-5e-00-00-16 | 224.0.0.22 | static

#| 01-00-5e-00-00-fb | 224.0.0.251 | static

#| 01-00-5e-00-00-fc | 224.0.0.252 | static

#| 01-00-5e-7f-ff-fa | 239.255.255.250 | static

#| ff-ff-ff-ff-ff-ff | 255.255.255.255 | static