Министерство науки и высшего образования Российской Федерации

Федеральное государственное бюджетное образовательное учреждение высшего образования

 «Ульяновский государственный технический университет»

Кафедра «Вычислительная техника»

Дисциплина «Сети и телекоммуникации»

**Лабораторная работа №1**

«Использование сетевых утилит ОС Windows для проверки и настройки локальной сети»

Выполнил студент

группы ИВТАПбд-41:

Галацков И.А.

Проверил:

Мартынов А. И.

Ульяновск, 2024

**Задание на лабораторную работу**

Необходимо научиться пользоваться вышеперечисленными утилитами. Типовыми заданиями являются:

* Просмотр и анализ настроек стека протоколов TCP/IP с использованием утилиты ipconfig.
* Тестирование сетевого соединения утилитой ping (конечный адрес может быть, напр., IP-адрес местного Proxy-сервера, адрес одной из машин в сети, доменное имя некоторого узла в Сети и др.).
* Прослеживание маршрута пакетов до заданного узла утилитой tracert.
* Отображение и изменение таблицы хранения соответствия МАС- и IP адресов с помощью утилиты arp.
* Просмотр текущих сетевых соединений, портов, TCP/IP-статистики утилитой netstat.
* Отображение и изменение таблицы маршрутов утилитой route (необходимо вывести все маршруты, добавить свой маршрут, изменить его, а затем удалить).

**Описание реализации**

Создаем графическое приложение с использованием библиотеки tkinter, которое предоставляет удобный интерфейс для работы с популярными сетевыми утилитами. Основное окно программы содержит четыре вкладки, каждая из которых отвечает за выполнение определенной сетевой команды и отображение результатов.

Во вкладке **Ping** пользователь может ввести адрес или доменное имя и нажать кнопку для выполнения команды ping. После этого результаты команды выводятся построчно в таблице, что позволяет легко оценить доступность указанного узла в сети.

Во вкладке **Ipconfig** предусмотрена кнопка для выполнения команды ipconfig. После ее нажатия команда выполняется, а результаты разбиваются на отдельные блоки (по интерфейсам) и отображаются в виде текста на подстраницах внутри вкладки. Каждая подстраница соответствует одному сетевому интерфейсу, что упрощает анализ конфигурации.

Во вкладке **Tracert** пользователь может ввести адрес или доменное имя, а затем запустить команду tracert, которая отслеживает маршрут к указанному узлу. Результаты выполнения отображаются построчно в таблице, где виден каждый шаг маршрута.

Во вкладке **Netstat** пользователь может нажать кнопку для выполнения команды netstat -an, которая выводит информацию о текущих сетевых соединениях. Результаты команды отображаются в текстовом поле в полном объеме.

Для выполнения всех системных команд используется модуль subprocess, который позволяет запускать команды операционной системы и захватывать их результаты. Полученные данные отображаются в соответствующих виджетах: таблицах или текстовых полях. Таким образом, приложение обеспечивает удобный доступ к сетевым утилитам через графический интерфейс, что делает его полезным для анализа и диагностики сетевых проблем.

**Приложение**

import tkinter as tk  
from tkinter import ttk  
import subprocess  
  
# Функция для выполнения системных команд  
def run\_command(command):  
 process = subprocess.Popen(command, stdout=subprocess.PIPE, stderr=subprocess.PIPE, text=True, encoding='cp866')  
 output, error = process.communicate()  
 return output if not error else error  
  
# --- PING ---  
def ping\_command():  
 target = entry\_ping.get()  
 if not target:  
 return  
 output = run\_command(['ping', target])  
 for line in output.split('\n'):  
 if line.strip():  
 tree\_ping.insert('', 'end', values=(line,))  
  
# --- IPCONFIG ---  
def ipconfig\_command():  
 output = run\_command(['ipconfig'])  
 interfaces = output.split("\n\n") # Разделение по интерфейсам  
 for i, interface in enumerate(interfaces):  
 tab = ttk.Frame(ipconfig\_notebook)  
 ipconfig\_notebook.add(tab, text=f"Интерфейс {i + 1}")  
 text = tk.Text(tab, wrap='word')  
 text.insert('1.0', interface)  
 text.pack(fill='both', expand=True)  
  
# --- TRACERT ---  
def tracert\_command():  
 target = entry\_tracert.get()  
 if not target:  
 return  
 output = run\_command(['tracert', target])  
 for line in output.split('\n'):  
 if line.strip():  
 tree\_tracert.insert('', 'end', values=(line,))  
  
# --- NETSTAT ---  
def netstat\_command():  
 output = run\_command(['netstat', '-an'])  
 text\_netstat.delete('1.0', 'end')  
 text\_netstat.insert('1.0', output)  
  
# --- Основной интерфейс ---  
root = tk.Tk()  
root.title("Сетевые утилиты")  
root.geometry("800x600")  
  
# Создание вкладок  
notebook = ttk.Notebook(root)  
notebook.pack(fill='both', expand=True)  
  
# --- Вкладка PING ---  
frame\_ping = ttk.Frame(notebook)  
notebook.add(frame\_ping, text="Ping")  
  
lbl\_ping = ttk.Label(frame\_ping, text="Введите адрес или доменное имя:")  
lbl\_ping.pack(pady=5)  
  
entry\_ping = ttk.Entry(frame\_ping, width=30)  
entry\_ping.pack(pady=5)  
  
btn\_ping = ttk.Button(frame\_ping, text="Выполнить Ping", command=ping\_command)  
btn\_ping.pack(pady=5)  
  
tree\_ping = ttk.Treeview(frame\_ping, columns=("result",), show="headings")  
tree\_ping.heading("result", text="Результат")  
tree\_ping.pack(fill='both', expand=True)  
  
# --- Вкладка IPCONFIG ---  
frame\_ipconfig = ttk.Frame(notebook)  
notebook.add(frame\_ipconfig, text="Ipconfig")  
  
btn\_ipconfig = ttk.Button(frame\_ipconfig, text="Выполнить Ipconfig", command=ipconfig\_command)  
btn\_ipconfig.pack(pady=5)  
  
ipconfig\_notebook = ttk.Notebook(frame\_ipconfig)  
ipconfig\_notebook.pack(fill='both', expand=True)  
  
# --- Вкладка TRACERT ---  
frame\_tracert = ttk.Frame(notebook)  
notebook.add(frame\_tracert, text="Tracert")  
  
lbl\_tracert = ttk.Label(frame\_tracert, text="Введите адрес или доменное имя:")  
lbl\_tracert.pack(pady=5)  
  
entry\_tracert = ttk.Entry(frame\_tracert, width=30)  
entry\_tracert.pack(pady=5)  
  
btn\_tracert = ttk.Button(frame\_tracert, text="Выполнить Tracert", command=tracert\_command)  
btn\_tracert.pack(pady=5)  
  
tree\_tracert = ttk.Treeview(frame\_tracert, columns=("result",), show="headings")  
tree\_tracert.heading("result", text="Результат")  
tree\_tracert.pack(fill='both', expand=True)  
  
# --- Вкладка NETSTAT ---  
frame\_netstat = ttk.Frame(notebook)  
notebook.add(frame\_netstat, text="Netstat")  
  
btn\_netstat = ttk.Button(frame\_netstat, text="Выполнить Netstat", command=netstat\_command)  
btn\_netstat.pack(pady=5)  
  
text\_netstat = tk.Text(frame\_netstat, wrap='word')  
text\_netstat.pack(fill='both', expand=True)  
  
# Запуск основного цикла  
root.mainloop()