

УДК 004.7

<https://doi.org/10.33619/2414-2948/130/16>

РОЛЬ PYTHON В АДМИНИСТРИРОВАНИИ КОМПЬЮТЕРНЫХ СЕТЕЙ

©**Закирова Д. А.**, ORCID: 0009-0002-6723-690X, SPIN-код: 4158-0909,
Ошский технологический университет им. М. М. Адышева,
г. Ош, Кыргызстан, zakirovadinara03@gmail.com
©**Кудайбердиев М. Ж.**, Лицей им. Алымбек Датка,
г. Ош, Кыргызстан, muamedalikudajberdiev805@gmail.com
©**Маткаликов А. М.**, Ошский технологический университет
им. М. М. Адышева, г. Ош, Кыргызстан

THE ROLE OF PYTHON IN COMPUTER NETWORK ADMINISTRATION

©**Zakirova D.**, ORCID: 0009-0002-6723-690X, SPIN-code: 4158-0909,
Osh Technological University named after M.M. Adyshev,
Osh, Kyrgyzstan, zakirovadinara03@gmail.com
©**Kudaiberdiev M.**, Lyceum named after A. Datka,
Osh, Kyrgyzstan, muamedalikudajberdiev805@gmail.com
©**Matkalikov A.**, Osh Technological University named after M. Adyshev, Osh, Kyrgyzstan

Аннотация. Рассматриваются возможности применения языка программирования Python в задачах администрирования компьютерных сетей. Проведен анализ основных направлений использования Python. Показано, что применение Python позволяет повысить эффективность работы системных и сетевых администраторов за счет автоматизации рутинных операций и сокращения времени обработки информации. В работе представлена программа, которая предназначена для сканирования локальной сети и проверки доступности устройств по заданным IP-адресам. Приложение позволяет пользователю указать диапазон адресов, выполнить их проверку с помощью команды ping и отобразить результаты в таблице. Также предусмотрена возможность сохранения полученных данных в CSV-файл. Графический интерфейс обеспечивает удобный ввод параметров и просмотр результатов сканирования. Также добавлена программа для мониторинга сети, позволяющая просматривать данные об отправке и получении информации по сети.

Abstract. This article examines the potential of the Python programming language for computer network administration tasks. It analyzes the main areas of Python use and demonstrates that Python can improve the efficiency of system and network administrators by automating routine operations and reducing information processing time. The article presents a program designed to scan a local network to check the availability of devices at specified IP addresses. The application allows users to specify a range of addresses, ping them, and display the results in a table. It also provides the option to save the resulting data to a CSV file. The graphical interface allows for easy parameter entry and viewing of scan results. A network monitoring program is also included for viewing data sent and received over the network.

Ключевые слова: Python, сетевое администрирование, автоматизация, мониторинг сети, компьютерные сети, информационные технологии.

Keywords: Python, Network administration, automation, network monitoring, computer networks, information technology.

Современные компьютерные сети являются неотъемлемой частью информационной инфраструктуры организаций, образовательных учреждений и предприятий. С ростом количества сетевых устройств увеличивается сложность их обслуживания и администрирования. Выполнение рутинных операций вручную требует значительных временных затрат и повышает вероятность возникновения ошибок. Одним из эффективных способов решения данной проблемы является автоматизация процессов администрирования. Среди современных средств автоматизации особое место занимает язык программирования Python, который, благодаря простому синтаксису, высокой производительности и большому количеству специализированных библиотек, получил широкое распространение в области сетевых технологий. Целью данной работы является исследование роли Python в администрировании компьютерных сетей и анализ его практического применения для решения задач мониторинга, управления и обеспечения безопасности сетевой инфраструктуры. В ряде статей анализируются особенности использования Python для построения сетевых моделей. Рассматриваются понятие и сущность сетевых моделей, а также ключевые свойства языка программирования Python. Построение и анализ сетевой модели комплекса работ позволяет оптимизировать затраты на их выполнение. Анализ сетевой модели начинается с расчета ее временных параметров. Предложен один из методов расчета временных параметров событий. Рассматривается построение сетевых графиков для наглядного отображения этапов работ, выявления узких мест и сокращения сроков реализации проектов. Рассматриваются основные области использования Python, а также анализируются преимущества и недостатки его применения в различных сферах. Исследование заключается в разработке автоматизированной системы учета и мониторинга активного сетевого оборудования для телекоммуникационной компании. Рассматривается разработка алгоритма настройки маршрутизаторов компьютерных сетей, в том числе добавление статического маршрута через IP-адрес конкретного провайдера. Данный алгоритм реализуется с помощью языка программирования Python [1-6].

Python позволяет автоматизировать процессы настройки маршрутизаторов, коммутаторов и серверов. Вместо ручного ввода команд администратор может использовать программные сценарии для выполнения типовых операций. Одной из наиболее востребованных задач является мониторинг состояния сети. Python предоставляет средства для получения информации о сетевых интерфейсах, объеме передаваемых данных и количестве активных соединений. Python может использоваться для анализа проходящего сетевого трафика. С помощью специализированных библиотек можно определять количество пакетов, используемые протоколы и наиболее активные узлы сети. Полученная информация используется для оптимизации работы сети и повышения уровня безопасности. Сетевые администраторы регулярно выполняют сбор данных о работе оборудования и сетевых сервисов. Python позволяет автоматически сохранять статистику в базы данных, электронные таблицы или текстовые файлы. Это обеспечивает ведение журналов событий и формирование отчетов. Python широко применяется для разработки средств контроля безопасности компьютерных сетей. С его помощью можно создавать системы обнаружения вторжений, анализаторы журналов событий и инструменты контроля доступа. Автоматический анализ сетевой активности позволяет оперативно выявлять потенциальные угрозы и принимать меры по их устранению. Для исследования возможностей Python была разработана программа мониторинга сетевой активности персонального компьютера. Она собирает статистику об объеме переданных и полученных данных, рассчитывает скорость передачи информации и сохраняет результаты в файл для последующего анализа.

Полученные результаты показали возможность использования Python в качестве эффективного инструмента сетевого администрирования даже на небольших объектах, таких как учебные кабинеты. Во-первых, необходимо установить Python и PyCharm. PyCharm — это среда разработки с отладчиком. Во-вторых, нужно установить необходимые библиотеки для создания программы. Для решения данной задачи часто используются библиотеки: tkinter, subprocess, platform, threading, csv, time, datetime, psutil. Чтобы использовать эти библиотеки, их нужно установить через терминал, введя команду `pip install psutil`. Ниже показан примерный код программы:

```
import tkinter as tk
from tkinter import ttk
import subprocess
import platform
import threading
import csv
# проверка одного IP адреса
def ping(ip):
    param = "-n" if platform.system().lower() == "windows" else "-c"

    result = subprocess.run(
        ["ping", param, "1", ip],
        stdout=subprocess.PIPE,
        stderr=subprocess.PIPE,
        text=True
    )
    return "TTL=" in result.stdout or "ttl=" in result.stdout
# Сканирование сети
def scan_network():
    start = int(entry_start.get())
    end = int(entry_end.get())
    base = entry_base.get()
    for i in range(start, end + 1):
        ip = f"{base}.{i}"
        status = ping(ip)
        if status:
            tree.insert("", "end", values=(ip, "ДОСТУПЕН"))
        else:
            tree.insert("", "end", values=(ip, "НЕТ СВЯЗИ"))
# Запуск
def start_scan():
    threading.Thread(target=scan_network).start()
# Сохранение данных в CSV файле
def save_file():
    with open("scan_result.csv", "w", newline="", encoding="utf-8") as f:
        writer = csv.writer(f)
        writer.writerow(["IP-адрес", "Статус"])
        for row in tree.get_children():
            writer.writerow(tree.item(row)["values"])
# Окно программы
root = tk.Tk()
root.title("Сканирование локальной сети")
root.geometry("600x400")
```

```
# Ввод IP диапазона
frame = tk.Frame(root)
frame.pack(pady=10)
tk.Label(frame, text="Сеть:").grid(row=0, column=0)
entry_base = tk.Entry(frame, width=15)
entry_base.insert(0, "192.168.1")
entry_base.grid(row=0, column=1)
tk.Label(frame, text="От:").grid(row=0, column=2)
entry_start = tk.Entry(frame, width=5)
entry_start.insert(0, "1")
entry_start.grid(row=0, column=3)
tk.Label(frame, text="До:").grid(row=0, column=4)
entry_end = tk.Entry(frame, width=5)
entry_end.insert(0, "30")
entry_end.grid(row=0, column=5)
# Кнопки
tk.Button(root, text="Сканировать сеть", command=start_scan).pack(pady=5)
tk.Button(root, text="Сохранить отчет", command=save_file).pack(pady=5)
# Таблица
tree = ttk.Treeview(root, columns=("IP", "Status"), show="headings")
tree.heading("IP", text="IP-адрес")
tree.heading("Status", text="Статус")
tree.pack(fill="both", expand=True)
root.mainloop()
```

После компиляции кода программы, в окне появится следующее окно.

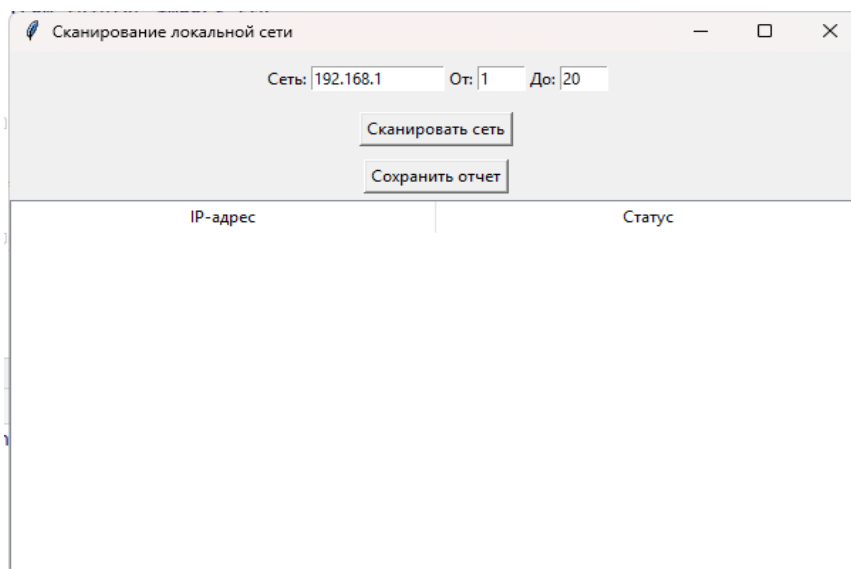


Рисунок 1. Окно сканирование локальной сети

Выбрав кнопку «Сканировать сеть», у нас идет сканирование сети в диапазоне от 1 до 20. На Рисунке 2 показаны данные о сети. Коммутатор отключен, вот поэтому на рисунке показана «нет связи».

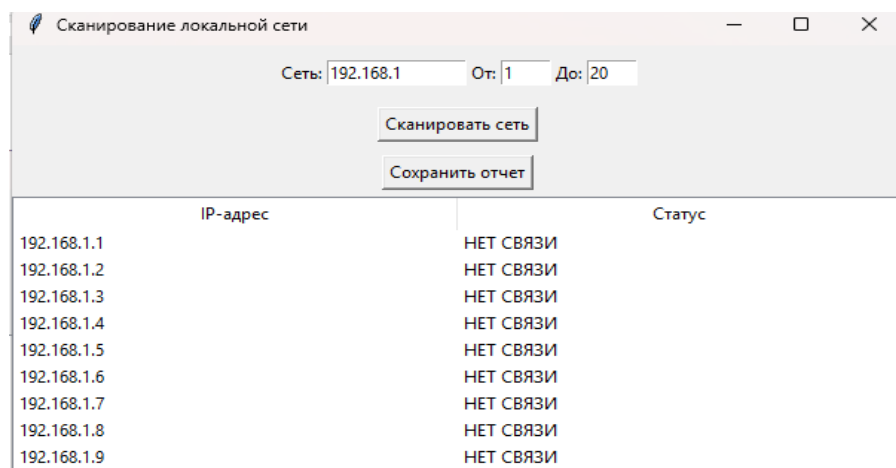


Рисунок 2. Информация о сканирование сети

В следующем рисунке можно увидеть измененный сеть в диапазоне от 1 до 30. 192.168.12 – эти цифры означает адрес подсети. На рисунке 3 показана, что адрес подсети изменен и связь есть. Коммутатор включен, некоторые IP адреса доступны, потому что компьютеры были включены. Заново сканируем сеть и у нас появится следующее окно.

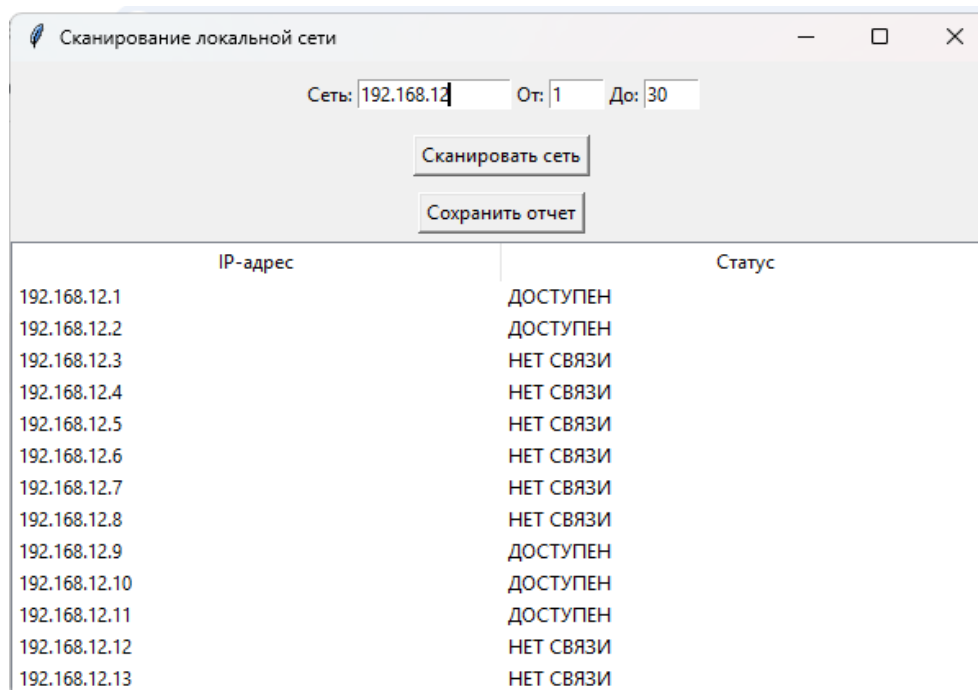


Рисунок 3. Повторное сканирование сети

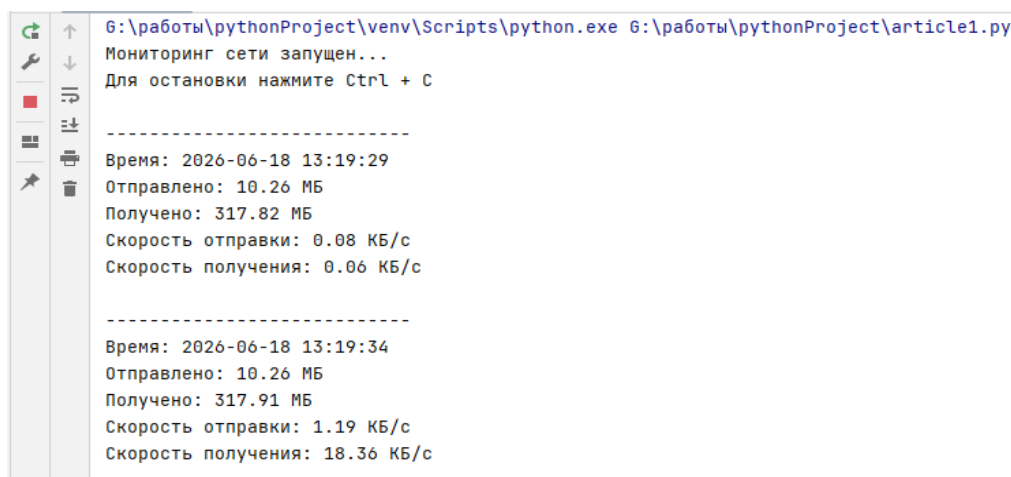
Кнопка «Сохранить отчет» отвечает для сохранение данных в scan_result.csv формате.

А также для мониторинга сети, можно использовать следующий код программы:

```
import psutil
import csv
import time
from datetime import datetime
# Файл для сохранения статистики
LOG_FILE = "network_stats.csv"
def get_network_usage():
```

```
""""Получение статистики сети""""
net = psutil.net_io_counters()
return net.bytes_sent, net.bytes_recv
# Создание CSV-файла и запись заголовков
with open(LOG_FILE, "w", newline="", encoding="utf-8") as file:
    writer = csv.writer(file)
    writer.writerow([
        "Дата и время",
        "Отправлено (МБ)",
        "Получено (МБ)",
        "Скорость отправки (КБ/с)",
        "Скорость получения (КБ/с)"
    ])
print("Мониторинг сети запущен...")
print("Для остановки нажмите Ctrl + C")
# Начальные значения
old_sent, old_recv = get_network_usage()
try:
    while True:
        # Интервал измерения
        time.sleep(5)
        # Новые значения
        new_sent, new_recv = get_network_usage()
        # Расчёт скорости
        sent_speed = (new_sent - old_sent) / 1024 / 5
        recv_speed = (new_recv - old_recv) / 1024 / 5
        # Общий объём трафика
        sent_mb = round(new_sent / 1024 / 1024, 2)
        recv_mb = round(new_recv / 1024 / 1024, 2)
        current_time = datetime.now().strftime("%Y-%m-%d %H:%M:%S")
        print("\n-----")
        print("Время:", current_time)
        print("Отправлено:", sent_mb, "МБ")
        print("Получено:", recv_mb, "МБ")
        print("Скорость отправки:", round(sent_speed, 2), "КБ/с")
        print("Скорость получения:", round(recv_speed, 2), "КБ/с")
        # Сохранение данных
        with open(LOG_FILE, "a", newline="", encoding="utf-8") as file:
            writer = csv.writer(file)
            writer.writerow([
                current_time,
                sent_mb,
                recv_mb,
                round(sent_speed, 2),
                round(recv_speed, 2)
            ])
        # Обновление значений
        old_sent = new_sent
        old_recv = new_recv
except KeyboardInterrupt:
    print("\nМониторинг завершён.")
```

После компиляции кода можем увидеть результат программы.



```
6:\работы\pythonProject\venv\Scripts\python.exe 6:\работы\pythonProject\article1.py
Мониторинг сети запущен...
Для остановки нажмите Ctrl + C

-----
Время: 2026-06-18 13:19:29
Отправлено: 10.26 МБ
Получено: 317.82 МБ
Скорость отправки: 0.08 КБ/с
Скорость получения: 0.06 КБ/с

-----
Время: 2026-06-18 13:19:34
Отправлено: 10.26 МБ
Получено: 317.91 МБ
Скорость отправки: 1.19 КБ/с
Скорость получения: 18.36 КБ/с
```

Рисунок 4. Мониторинг сети

В программах мы использовали несколько библиотек языка и показали сканирование и мониторинг сети. Эти данные использовались в учебном кабинете ВУЗа.

Заключение

Проведенное исследование показало, что язык программирования Python играет важную роль в современном сетевом администрировании. Благодаря широкому набору библиотек и простоте разработки, Python позволяет автоматизировать рутинные задачи, осуществлять мониторинг сетевой инфраструктуры и обеспечивать безопасность компьютерных сетей. Использование Python способствует повышению эффективности работы администраторов и снижению трудозатрат на обслуживание сетевых систем.

Список литературы:

1. Зайцев Д. С. Использование Python для построения сетевых моделей // E-Scio. 2022. №11(74). С. 90-96.
2. Двоерядкина Н. Н. Расчет временных параметров событий в сетевой модели // Вестник Амурского государственного университета. Серия: Гуманитарные науки. 2020. №89. С. 22-27.
3. Будникова И. К., Приймак Е. В. Моделирование управляемых процессов с применением методов сетевого планирования // Вестник технологического университета. 2018. Т. 21. №1. С. 115-118.
4. Мадиярбекова А. Python как высокоуровневый язык программирования // Вестник науки. 2024. Т. 4. №11 (80). С. 1114-1126.
5. Киргизова Е. В., Фирер А. В. Автоматизация учета и мониторинга активного сетевого оборудования // Международный научно-исследовательский журнал. 2024. №10 (148). С. 32.
6. Джонмахмадов И. Т., Амзаев О. О., Комили А. А. Разработка алгоритма администрирования компьютерных сетей с помощью языка Python // Инновационная наука. 2019. №4. С. 39-43.

References:

1. Zajcev, D. S. (2022). Ispol'zovanie Python dlya postroeniya setevykh modelej. *E-Scio*, (11 (74)), 90-96. (in Russian).

2. Dvoeryadkina, N. N. (2020). Raschet vremennykh parametrov sobytij v setevoy modeli. *Vestnik Amurskogo gosudarstvennogo universiteta. Seriya: Gumanitarnye nauki*, (89), 22-27. (in Russian).
3. Budnikova, I. K., & Prijmak, E. V. (2018). Modelirovanie upravlyaemykh processov s primeneniem metodov setevogo planirovaniya. *Vestnik tekhnologicheskogo universiteta*, 21(1), 115-118. (in Russian).
4. Madiyarbekova, A. (2024). Python kak vysokourovnevnyy yazyk programmirovaniya. *Vestnik nauki*, 4(11 (80)), 1114-1126. (in Russian).
5. Kirgizova, E. V., & Firer, A. V. (2024). Avtomatizatsiya ucheta i monitoringa aktivnogo setevogo oborudovaniya. *Mezhdunarodnyy nauchno-issledovatel'skij zhurnal*, (10 (148)), 32. (in Russian).
6. Dzhonmahmadov, I. T., Amzaev, O. O., & Komili, A. A. (2019). Razrabotka algoritma administrirovaniya komp'yuternykh setej s pomoshch'yu yazyka Python. *Innovacionnaya nauka*, (4), 39-43. (in Russian).

Поступила в редакцию
23.06.2026 г.

Принята к публикации
06.07.2026 г.

Ссылка для цитирования:

Закирова Д. А., Кудайбердиев М. Ж., Маткаликов А. М. Роль Python в администрировании компьютерных сетей // Бюллетень науки и практики. 2026. Т. 12. №9. С. 160-167. <https://doi.org/10.33619/2414-2948/130/16>

Cite as (APA):

Zakirova, D., Kudaiberdiev, M., & Matkalikov, A. (2026). The Role of Python in Computer Network Administration. *Bulletin of Science and Practice*, 12(9), 160-167. (in Russian). <https://doi.org/10.33619/2414-2948/130/16>