

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ
РОССИЙСКОЙ ФЕДЕРАЦИИ
Федеральное государственное автономное образовательное учреждение
высшего образования
«КРЫМСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ им. В. И. ВЕРНАДСКОГО»
ФИЗИКО-ТЕХНИЧЕСКИЙ ИНСТИТУТ
Кафедра компьютерной инженерии и моделирования

ОТЧЕТ ПО ПРАКТИЧЕСКОМУ ЗАДАНИЮ №6
«Сетевые средства мониторинга операционной системы GNU/Linux»

Практическая работа
по дисциплине «Системное программное обеспечение»
студента 3 курса группы ИВТ-б-о-222(2)
Чудопалова Богдана Андреевича

09.03.01 «Информатика и вычислительная техника»

Ход работы

1. Написать скрипт для сбора статистики с интерфейса. Обеспечить его постоянной загрузкой активностью: скачивание файла, открытие веб-страниц и т. п. Скрипт имеет вид:

```
sudo vim counter.sh

#!/bin/bash
INTERFACE=$1
STATS=$(cat /proc/net/dev | grep $INTERFACE)

IN_BYTES=$(echo $STATS | awk '{print $2}')
OUT_BYTES=$(echo $STATS | awk '{print $10}')

echo $IN_BYTES
echo $OUT_BYTES
```

Далее сделал его исполняемым

```
→ mrtg chmod +x ./counter.sh
→ mrtg
```

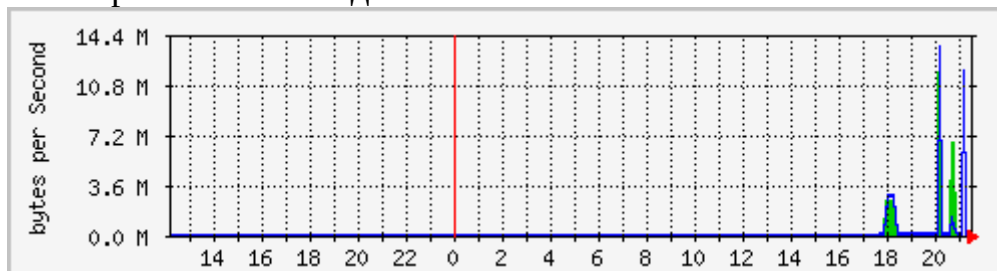
2. Обеспечить сбор данных для формирования графика активности с использованием утилиты mrtg. Для этого отредактировал конфиг следующим образом.

```
# Put your rules below or run cfmaker(1)
WriteExpires: Yes
Refresh: 300
WithPeak[^]: wym
Suppress[^]: y
Target[enpls0]: `/etc/mrtg/counter.sh enpls0 `
WorkDir: /var/www/html/mrtg
Options[enpls0]: growright
Title[enpls0]: enpls0 Traffic
PageTop[enpls0]: enpls0 Traffic
MaxBytes[enpls0]: 99999999
kilo[enpls0]: 1024
YLegend[enpls0]: bytes per Second
ShortLegend[enpls0]: bytes per Second
Legend0[enpls0]: enpls0 In Traffic :
LegendI[enpls0]: enpls0 Out Traffic :
~
```

Также воспользовался cron:

```
*/5 * * * * env LANG=C /usr/bin/mrtg /etc/mrtg/mrtg.cfg
```

График мониторинга имеет вид



4. Создать кольцевую БД `rrd` и обеспечить ее постоянное обновление данными сетевой активности. БД была создана следующей командой

```

→ netw rrdtool create wlp3s0.rrd \
--step 60 \
DS:input:COUNTER:180:0:U \
DS:output:COUNTER:180:0:U \
RRA:AVERAGE:0.5:1:1440 \
RRA:AVERAGE:0.5:5:2016 \
RRA:AVERAGE:0.5:60:8760

→ netw ls
netw.sh wlp3s0.rrd
→ netw █

```

Далее был создан скрипт для ее обновления и построения графика

```

#!/bin/bash
INPUT=`/sbin/ifconfig wlp3s0 |grep bytes| awk '/RX/{print $5}'`
OUTPUT=`/sbin/ifconfig wlp3s0 |grep bytes| awk '/TX/{print $5}'`
rrdtool update ./wlp3s0.rrd -t "input:output" N:$INPUT:$OUTPUT

```

Также был использован crontab

```

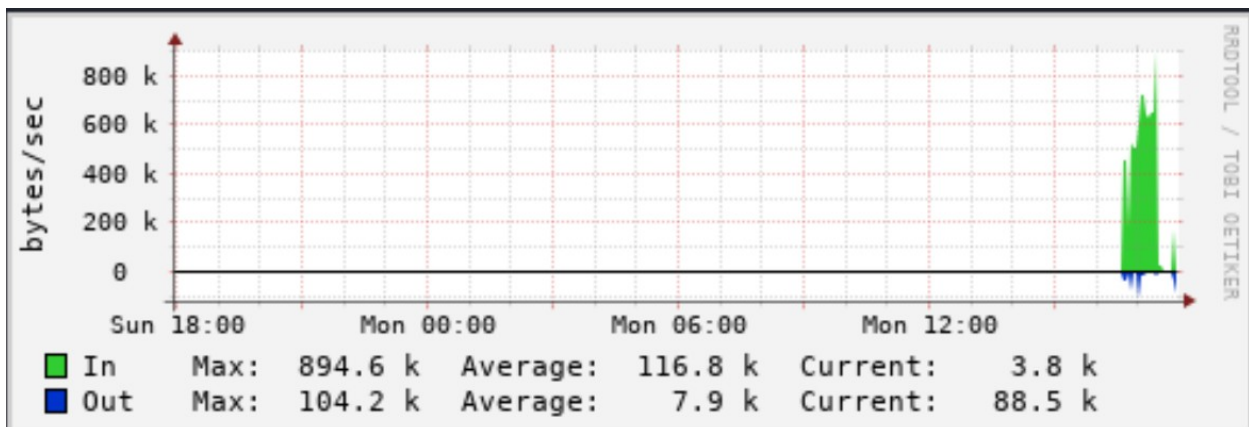
*/1 * * * * /home/bogdan/scripts_for_labs/netw/netw.sh
█

```

После некоторого времени командой визуализировал график

```
→ netw rrdtool graph net.png \
--start now-3h --end now \
--width 800 --height 300 \
--slope-mode \
DEF:input=wlp3s0.rrd:input:AVERAGE:step=60 \
DEF:output=wlp3s0.rrd:output:AVERAGE:step=60 \
CDEF:output_neg=output, -1, \*
```

График сетевой активности имел вид



Проверить работоспособность при обращении к некоторым общедоступным веб-ресурсам при помощи утилит трассировки маршрута. Наложить путь на географическую карту. В качестве веб-ресурса можно использовать сайты google.com, kernel.org, ubuntu.com. Для начала я извлек ip-адреса из команды traceroute google.com с помощью следующего скрипта

```
#!/bin/bash
OUTPUT_FILE="traceroute_ips.txt"
TRACEROUTE_OUTPUT=$(traceroute google.com)
IP_ADDRESSES=$(echo "$TRACEROUTE_OUTPUT" | grep -oE "[0-9]+\.[0-9]+\.[0-9]+\.[0-9]+")
echo "$IP_ADDRESSES" > "$OUTPUT_FILE"
```

После этого я преобразовал ip-адреса в координаты для карты с помощью python-скрипта


```

import requests

INPUT_FILE = "traceroute_ips.txt"
OUTPUT_FILE = "geo_coordinates.txt"

def get_geo_info(ip):
    url = f"http://ip-api.com/json/{ip}"
    try:
        response = requests.get(url)
        if response.status_code == 200:
            data = response.json()
            if data["status"] == "success":
                return data.get("lat"), data.get("lon"), data.get("city"), data.get("country")
            else:
                return None, None, "Unknown", "Unknown"
        else:
            return None, None, "Unknown", "Unknown"
    except Exception as e:
        print(f"Ошибка при запросе для IP {ip}: {e}")
        return None, None, "Unknown", "Unknown"

def process_ips(input_file, output_file):
    with open(input_file, "r") as infile, open(output_file, "w") as outfile:
        outfile.write("IP, Latitude, Longitude, City, Country\n")
        for line in infile:
            ip = line.strip()
            if ip:
                lat, lon, city, country = get_geo_info(ip)
                outfile.write(f"{ip}, {lat}, {lon}, {city}, {country}\n")
                print(f"IP: {ip} -> Координаты: {lat}, {lon} -> {city}, {country}")

if __name__ == "__main__":
    process_ips(INPUT_FILE, OUTPUT_FILE)
    print(f"Геолокационные данные сохранены в файл {OUTPUT_FILE}")

```

Наложение на карту я сделал также с помощью python

```

import folium

INPUT_FILE = "geo_coordinates.txt"
OUTPUT_MAP = "network_route_map.html"

def plot_coordinates_on_map(input_file, output_map):
    coordinates = []

    with open(input_file, "r") as file:
        next(file)
        for line in file:
            parts = line.strip().split(", ")
            if len(parts) >= 3:
                try:
                    latitude = float(parts[1])
                    longitude = float(parts[2])
                    coordinates.append((latitude, longitude))
                except ValueError:
                    print(f"Ошибка обработки строки: {line}")

    if not coordinates:
        print("Координаты не найдены в файле.")
        return

    map = folium.Map(location=coordinates[0], zoom_start=5)

    for coord in coordinates:
        folium.Marker(location=coord).add_to(map)
    folium.PolyLine(coordinates, color="blue", weight=2.5).add_to(map)

    map.save(output_map)
    print(f"Карта сохранена в файл {output_map}")

if __name__ == "__main__":
    plot_coordinates_on_map(INPUT_FILE, OUTPUT_MAP)

```

Итоговая карта маршрута имела вид

