ГУАП

КАФЕДРА № 43

ОТЧЕТ
ЗАЩИЩЕН С ОЦЕНКОЙ

ПРЕПОДАВАТЕЛЬ

старший преподаватель		М. Д. Поляк
должность, уч. степень, звание	подпись, дата	инициалы, фамилия

ОТЧЕТ О ЛАБОРАТОРНОЙ РАБОТЕ №1

Лабораторная работа №1: работа с текстовыми потоками в командном интерпретаторе Bash

по курсу: Операционные системы

РАБОТУ ВЫПОЛНИЛ

СТУДЕНТ гр. №	4233К	09.02.2025	С. В. Голанова
		подпись, дата	инициалы, фамилия

Санкт-Петербург 2025

Цель работы

Изучение принципов работы с командным интерпретатором GNU/Linux и основ обработки текстовых файлов с помощью команд grep, awk, sed.

Задание

1. Подготовить операционную систему GNU/Linux к работе. При необходимости, воспользоваться средством виртуализации VMware Workstation/Player, Oracle VirtualBox или др. В качестве дистрибутива рекомендуется взять Ubuntu.
2. Создать в домашней директории пользователя /home/user/ каталог lab1 с помощью команды mkdir. Перейти в этот каталог с помощью команды cd. Всю дальнейшную работу вести внутри этого каталога.
3. С помощью команды git clone <путь_к_репозиторию_на_github> создать локальную копию репозитория. Перейти в каталог с копией репозитория (при клонировании репозитория каталог будет иметь такое же название, как и сам репозиторий на github.com).
4. Скачать архив с логами DNS-сервера с использованием команды wget https://github.com/markpolyak/datasets/raw/main/data/dns-tunneling.log.tar.bz2 или curl -L -O https://github.com/markpolyak/datasets/raw/main/data/dns-tunneling.log.tar.bz2. При отсутствии в используемой ОС утилит wget и curl установить их самостоятельно. В Ubuntu это можно сделать с помощью команд sudo apt-get install wget и sudo apt-get install curl соответственно. В этом случае перед вызовом команды sudo apt-get install ... настоятельно рекомендуется выполнить команду sudo apt-get update.
5. С помощью команды tar -xjf dns-tunneling.log.tar.bz2 разархивировать файл dns-tunneling.log.
6. Ознакомиться с форматом файла dns-tunneling.log и хранящимися в нем данными с помощью команд cat, head, tail, more, less и т.п.
7. Подсчитать количество записей в файле dns-tunneling.log.
8. Вычислить номер варианта задания как остаток от деления порядкового номера студента по списку в журнале на количество вариантов заданий. Если остаток равен нулю, необходимо брать последнее задание.
9. Объявить в файле lab1.sh переменную TASKID, указав в качестве ее значения номер варианта задания в виде целого числа.
10. Объявить в файле lab1.sh переменную VAR_1, указав в качестве ее значения количество записей в файле dns-tunneling.log (данное значение должно вычисляться динамически во время выполнения скрипта).

11. Дописать в файл lab1.sh код для обработки данных из файла dns-tunneling.log в соответствии с вариантом задания используя исключительно команды командного интерпретатора bash. Использовать другие интерпретируемые или компилируемые языки запрещается. Скрипт должен читать файл dns-tunneling.log и все результаты должны вычисляться на его основе в реальном времени, использовать в скрипте заранее полученные значения запрещается. В случае использования циклов в bash для обработки данных, итоговая оценка за работу может быть снижена по усмотрению преподавателя.
12. Сделать файлы lab1.sh и test.sh исполняемыми с помощью команды chmod.
13. Отладить скрипт lab1.sh и убедиться, что тесты из test.sh выполняются без ошибок. Следует иметь в виду, что в случае принудительного прерывания выполнения тестов (например, командой Ctrl+C с клавиатуры, или путем выключения питания компьютера) следует самостоятельно восстановить файл dns-tunneling.log из бэкапа командой mv dns-tunneling.log.bak dns-tunneling.log. Также восстановить файл можно повторно выполнив команду разархивации tar -xjf dns-tunneling.log.tar.bz2 (при условии, что архив не был удален) или с помощью команды git checkout HEAD dns-tunneling.log, если до этого файл с логами был добавлен в репозиторий.
14. С помощью команд git add lab1.sh, git commit и git push добавить файл с написанным кодом в репозиторий. Убедиться, что тест в репозитории пройден успешно.
15. Подготовить отчет о выполнении лабораторной работы и загрузить его под именем report.pdf в репозиторий. В случае использования системы компьютерной верстки LaTeX также загрузить исходный файл report.tex.

Индивидуальное задание

Вариант 3.

Написать скрипт с использованием grep, sed, awk (необходимо использовать не менее одной из указанных утилит; использовать все три необязательно) для переконвертирования данных в формат XML со следующей структурой:

```
<dnslog>
<row>
    <timestamp>Отметка времени, когда поступил запрос</timestamp>
    <client_ip>IP адрес пользователя</client_ip>
    <client_port>Порт пользователя</client_port>
</row>
<row>
    <timestamp>Отметка времени, когда поступил запрос</timestamp>
    <client_ip>IP адрес пользователя</client_ip>
```

```
    <client_port>Порт пользователя</client_port>
</row>
<row>
    ...
</row>
</dnslog>
```

Текст внутри тегов `<timestamp>`, `<client_ip>` и `<client_port>` необходимо заменить соответствующими значениями из логов. Для отступов использовать символ табуляции (4 пробела в примере выше = одна табуляция). Сохранить в файле `results.txt` результат применения написанного скрипта к первым 20 строкам файла `dns-tunneling.log`. В переменную `VAR_2` записать количество записей в получившемся текстовом файле, которые содержат запросы от пользователей с IP-адресами из подсети `10.1.*.*`.

Описание входных данных

Файл dns-tunneling.log содержит логи DNS-сервера, представленные в виде текстового файла, в котором каждая строка соответствует записи о поступившем на вход сервера запросе. В логах сохраняются следующие параметры запроса, разделенные символом табуляции:

1. Название провайдера телекоммуникационных услуг: character array,
2. Название узла, на котором хранятся данные: character array,
3. Порядковый номер запроса: long,
4. Отметка времени, когда поступил запрос: два числа long, разделенных точкой; первое число – количество секунд, прошедших с 1 января 1970 года; второе число – количество микросекунд; т.е. фактически это тип данных float,
5. IP-адрес пользователя: character array,
6. Порт пользователя: int,
7. Локальный IP-адрес, на который поступил запрос: character array,
8. Локальный порт: int,
9. Название оборудования DNS-сервера: character array,
10. Класс запроса: int,
11. Тип запроса: int,
12. Код возвращаемого значения: int,
13. Флаги: int,
14. Вспомогательный идентификатор: int,
15. Запрашиваемый URL: character array,
16. Зона: character array,

17. Вспомогательное поле 1: character array,
18. Вспомогательное поле 2: character array,
19. Вспомогательное поле 3: character array,
20. Вспомогательное поле 4: character array,
21. Ответ сервера: character array,
22. Вспомогательное поле 5: character array,
23. Вспомогательное поле 6: character array,
24. Длина ответа: int

Фрагмент файла dns-tunneling.log (head dns-tunneling.log):

telecom-provider vnode 2103118491 1451001600.446786 10.119.76.168 42487 10.122.64.60 53 sprod1 1 10 0 524289 21410 cF.UNNwG3IPQ0ihAI8r4O8b6H7HPILgxM5o0Yzv4XPB5Y1YJCnB5VP2uRXSNx9kU4u.UUIp603GZT8lLRwSG10B70eGaFivUbIHugw6QfLMnpCxFR63Cy9WB4G1sRd8jtc.RdFx9WN8kXLYxq7UgbDUtPMAXSFKs3izL9h2eLq7iWt5tTPAs058hoZ90li.s37.1yf.de. product-tunnels-policy-1 1

telecom-provider vnode 2103118362 1451001600.059012 10.116.74.176 41280 10.122.64.60 53 sprod1 1 10 0 524289 2260 aD.Di2rYt4B7AmpHYBYqzsW0veNSn0EUbzQJQtktgtO.s24.1yf.de. product-tunnels-policy-1 1

telecom-provider vnode 2103118504 1451001600.473921 10.47.144.132 43301 10.122.64.60 53 sprod1 1 10 0 524289 3687 aD.DMDLAld51egwLOcgs0QS1hwY8jbbdzi6sQeQEi.s06.1yf.de. product-tunnels-policy-1 1

telecom-provider vnode 2103118525 1451001600.562429 10.119.76.168 45399 10.122.64.60 53 sprod1 1 10 0 524289 21412 aD.DshodpMc8LhAyVdTHPkuX1VNOn7fT690bAkNtgtO.s37.1yf.de. product-tunnels-policy-1 1

telecom-provider vnode 2103118545 1451001600.634430 10.47.144.132 43301 10.122.64.60 53 sprod1 1 10 0 524289 3691 aD.DUn7W8wxbSZRjHiKg9UrGEc3akbbdzi6sQeQEi.s06.1yf.de. product-tunnels-policy-1 1

telecom-provider vnode 2103118573 1451001600.756670 10.119.76.168 50476 10.122.64.60 53 sprod1 1 10 0 524289 21414 aD.DRHXjvlDkpsTjfn6GKLTTdhQPn7fT690bAkNtgtO.s37.1yf.de. product-tunnels-policy-1 1

telecom-provider vnode 2103118669 1451001600.105204 10.47.144.132 43301 10.122.64.60 53 sprod1 1 10 0 524289 3673 aT.T3IOoGH49YsCHtlvolATfFIV0Ef3KSiNcntNiRedtwn7tNOAwJx7k49dIleb.s06.1yf.de. product-tunnels-policy-1 1

telecom-provider vnode 2103118685 1451001600.154305 10.47.144.132 43301 10.122.64.60 53 sprod1 1 10 0 524289 3675 aD.DshTbRpr99UDn7X2pEbtonOM3jbbdzi6sQeQEi.s06.1yf.de. product-tunnels-policy-1 1
telecom-provider vnode 2103118706 1451001600.225852 10.47.144.132 43301 10.122.64.60 53 sprod1 1 10 0 524289 3631 aD.DaQnATdSCQy8Pj1bhewOQIY1Ijbbdzi6sQeQEi.s06.1yf.de. product-tunnels-policy-1 1
telecom-provider vnode 2103118708 1451001600.232628 10.119.76.168 59208 10.122.64.60 53 sprod1 1 10 0 524289 21408 aD.DPkBcvnt6cN9ZobJa1eebQALNn7fT690bAkNtgtO.s37.1yf.de. product-tunnels-policy-1 107:28

Результат выполнения работы

В результате выполнения файла lab1.sh получены следующие данные:

TASKID=3

VAR_1=251384

VAR_2=0

Также полученные данные приведены на рисунке 1 ниже.

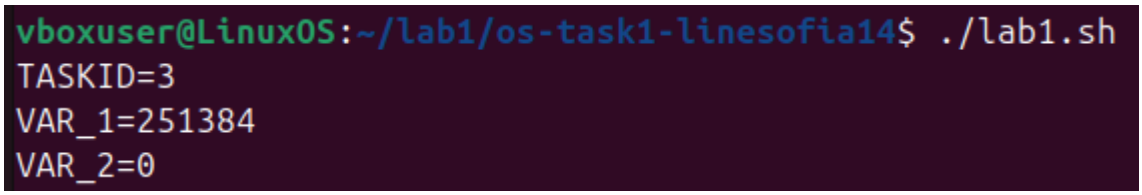

Рисунок 1 – Результат исполнения lab1.sh (терминал)

Данные, которые записаны в файл results.txt в результате работы программы представлены ниже.

Содержание файла results.txt:

```
<dnslog>
<row>
      <timestamp>1451001600.446786</timestamp>
      <client_ip>10.119.76.168</client_ip>
      <client_port>42487</client_port>
</row>
<row>
      <timestamp>1451001600.059012</timestamp>
      <client_ip>10.116.74.176</client_ip>
      <client_port>41280</client_port>
</row>
<row>
      <timestamp>1451001600.473921</timestamp>
      <client_ip>10.47.144.132</client_ip>
      <client_port>43301</client_port>
</row>
<row>
```

```
      <timestamp>1451001600.562429</timestamp>
      <client_ip>10.119.76.168</client_ip>
      <client_port>45399</client_port>
</row>
<row>
      <timestamp>1451001600.634430</timestamp>
      <client_ip>10.47.144.132</client_ip>
      <client_port>43301</client_port>
</row>
<row>
      <timestamp>1451001600.756670</timestamp>
      <client_ip>10.119.76.168</client_ip>
      <client_port>50476</client_port>
</row>
<row>
      <timestamp>1451001600.105204</timestamp>
      <client_ip>10.47.144.132</client_ip>
      <client_port>43301</client_port>
</row>
<row>
      <timestamp>1451001600.154305</timestamp>
      <client_ip>10.47.144.132</client_ip>
      <client_port>43301</client_port>
</row>
<row>
      <timestamp>1451001600.225852</timestamp>
      <client_ip>10.47.144.132</client_ip>
      <client_port>43301</client_port>
</row>
<row>
      <timestamp>1451001600.232628</timestamp>
      <client_ip>10.119.76.168</client_ip>
      <client_port>59208</client_port>
</row>
<row>
      <timestamp>1451001600.785342</timestamp>
      <client_ip>10.47.144.132</client_ip>
      <client_port>43301</client_port>
</row>
<row>
      <timestamp>1451001601.164861</timestamp>
      <client_ip>10.47.144.132</client_ip>
      <client_port>43301</client_port>
</row>
<row>
      <timestamp>1451001601.400704</timestamp>
      <client_ip>10.119.76.168</client_ip>
      <client_port>51056</client_port>
</row>
<row>
      <timestamp>1451001601.504543</timestamp>
      <client_ip>10.47.144.132</client_ip>
      <client_port>43301</client_port>
</row>
<row>
      <timestamp>1451001601.858664</timestamp>
      <client_ip>10.116.74.176</client_ip>
      <client_port>41280</client_port>
</row>
```

```
<row>
      <timestamp>1451001601.373581</timestamp>
      <client_ip>10.47.144.132</client_ip>
      <client_port>43301</client_port>
</row>
<row>
      <timestamp>1451001601.584901</timestamp>
      <client_ip>10.47.144.132</client_ip>
      <client_port>43301</client_port>
</row>
<row>
      <timestamp>1451001601.883377</timestamp>
      <client_ip>10.47.144.132</client_ip>
      <client_port>43301</client_port>
</row>
<row>
      <timestamp>1451001601.998364</timestamp>
      <client_ip>10.116.74.176</client_ip>
      <client_port>41280</client_port>
</row>
<row>
      <timestamp>1451001602.058561</timestamp>
      <client_ip>10.116.74.176</client_ip>
      <client_port>41280</client_port>
</row>
</dnslog>
```

Исходный код программы с комментариями

```
#!/usr/bin/env bash

# edit the code below and add your code
# отредактируйте код ниже и добавьте свой

# Переменная с номером варианта (константа):
# Объявляем TASKID (Вариант 3)
TASKID=3

# Дополнительные переменные (должны вычисляться динамически):
# Объявляем VAR_1 и вычисляем количество записей в dns-tunneling.log
VAR_1=$(wc -l < dns-tunneling.log)

# Преобразование данных в формат XML
head -n 20 dns-tunneling.log | awk '{
  timestamp = $4;
  client_ip = $5;
  client_port = $6;

  printf "<row>\n";
  printf "\t<timestamp>%s</timestamp>\n", timestamp;
  printf "\t<client_ip>%s</client_ip>\n", client_ip;
  printf "\t<client_port>%s</client_port>\n", client_port;
  printf "</row>\n";
}' | sed '1s/^/<dnslog>\n/g' | sed '$s/$/\n<\/dnslog>/g' > results.txt
```

```
# Подсчет записей с IP-адресами из подсети 10.1.*.*
VAR_2=$(grep "<client_ip>10\.1\." results.txt | wc -l)

echo "TASKID=$TASKID"
echo "VAR_1=$VAR_1"
echo "VAR_2=$VAR_2"
```

Выводы

В ходе лабораторной работы освоены базовые навыки работы с командной строкой GNU/Linux, включая навигацию, создание каталогов и установку программ. Изучены и применены инструменты grep, awk и sed для фильтрации, обработки и преобразования текстовых данных. Разработан bash-скрипт для автоматического извлечения и преобразования данных из лог-файла в XML-подобный формат, а также для выполнения статистического анализа (подсчет количества IP-адресов из заданной подсети).