

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ
федеральное государственное автономное образовательное учреждение высшего образования
«САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
АЭРОКОСМИЧЕСКОГО ПРИБОРОСТРОЕНИЯ»

ИНСТИТУТ НЕПРЕРЫВНОГО И ДИСТАНЦИОННОГО ОБРАЗОВАНИЯ

КАФЕДРА КОМПЬЮТЕРНЫХ ТЕХНОЛОГИЙ И ПРОГРАММНОЙ ИНЖЕНЕРИИ

ОЦЕНКА

ПРЕПОДАВАТЕЛЬ

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

ОТЧЕТ О ЛАБОРАТОРНОЙ РАБОТЕ № 1

Работа с текстовыми потоками в командном интерпретаторе Bash

по дисциплине: Операционные системы

РАБОТУ ВЫПОЛНИЛ

| | | | |
|---------------|--------------|---------------|-------------------|
| СТУДЕНТ ГР. № | Z1432K | | Куралесов Ф.А. |
| | номер группы | подпись, дата | инициалы, фамилия |

| | |
|----------------------|-----------|
| Студенческий билет № | 2021/3806 |
|----------------------|-----------|

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 на лабораторную работу:

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 дальнейшую 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`.

Вариант задания:

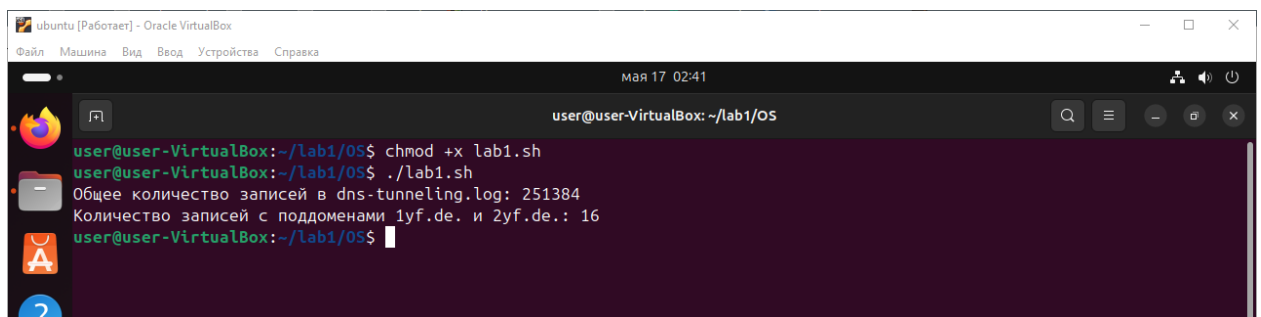
9. Написать скрипт с использованием `grep`, `sed`, `awk` (необходимо использовать не менее одной из указанных утилит; использовать все три необязательно) для переконвертирования данных в формат JSON со следующей структурой:

```
{
  "dnslog": [
    {
      "timestamp": "Отметка времени, когда поступил запрос",
      "client ip": "IP адрес пользователя",
      "url": "Запрашиваемый URL"
    },
    {
      "timestamp": "Отметка времени, когда поступил запрос",
      "client ip": "IP адрес пользователя",
      "url": "Запрашиваемый URL"
    },
    ...
  ]
}
```

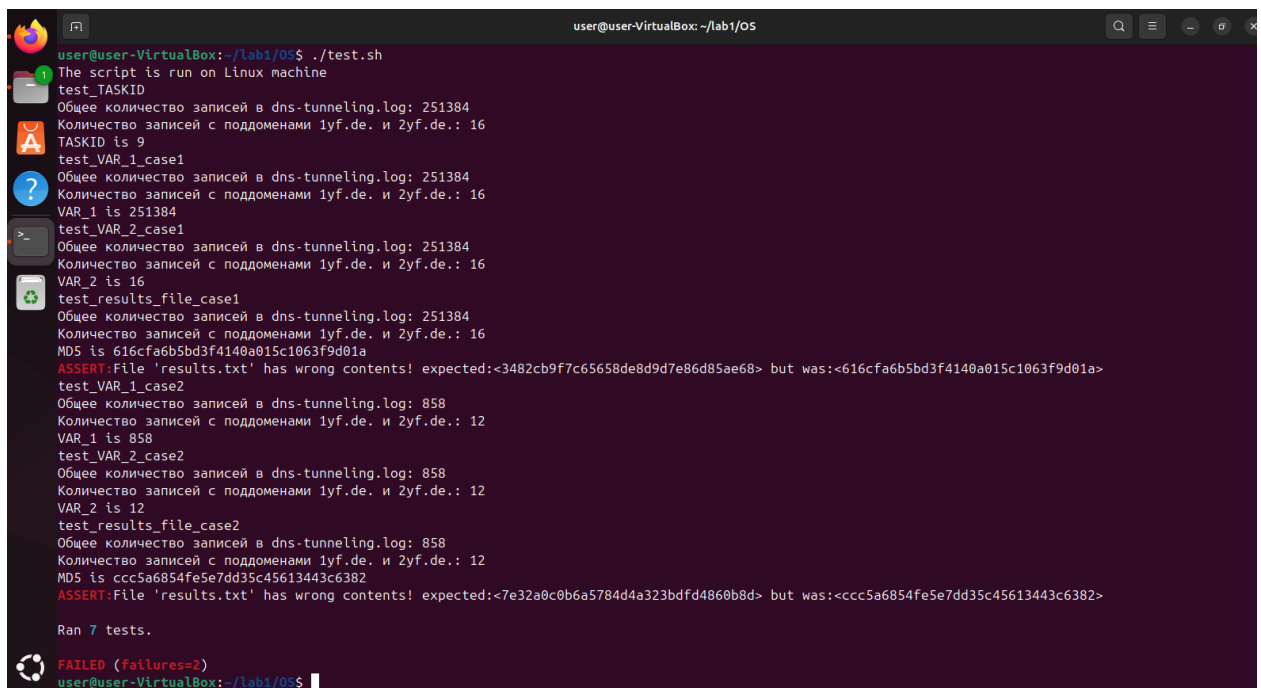
Взятый в кавычки русский текст необходимо заменить соответствующими значениями из логов. В формате JSON необходимо все ключи и все строковые значения помещать в двойные кавычки, числовые же значения (`int`, `float`) следует вставлять без кавычек. Для отступов использовать символ табуляции (4 пробела в примере выше = одна табуляция). Сохранить в файле `results.txt` результат применения написанного скрипта к строкам с 15 по 30-ю (включительно) файла `dns-tunneling.log`.

В переменную `VAR_2` записать количество записей в получившемся текстовом файле, которые содержат обращение к любым поддоменам `1yf.de.` и `2yf.de..`

1. Результат выполнения работы:



```
user@user-VirtualBox:~/lab1/OS$ chmod +x lab1.sh
user@user-VirtualBox:~/lab1/OS$ ./lab1.sh
Общее количество записей в dns-tunneling.log: 251384
Количество записей с поддоменами 1yf.de. и 2yf.de.: 16
user@user-VirtualBox:~/lab1/OS$
```



```
user@user-VirtualBox:~/lab1/OS$ ./test.sh
The script is run on Linux machine
test_TASKID
Общее количество записей в dns-tunneling.log: 251384
Количество записей с поддоменами 1yf.de. и 2yf.de.: 16
TASKID is 9
test_VAR_1_case1
Общее количество записей в dns-tunneling.log: 251384
Количество записей с поддоменами 1yf.de. и 2yf.de.: 16
VAR_1 is 251384
test_VAR_2_case1
Общее количество записей в dns-tunneling.log: 251384
Количество записей с поддоменами 1yf.de. и 2yf.de.: 16
VAR_2 is 16
test_results_file_case1
Общее количество записей в dns-tunneling.log: 251384
Количество записей с поддоменами 1yf.de. и 2yf.de.: 16
MD5 is 616cfa6b5bd3f4140a015c1063f9d01a
ASSERT:File 'results.txt' has wrong contents! expected:<3482cb9f7c65658de8d9d7e86d85ae68> but was:<616cfa6b5bd3f4140a015c1063f9d01a>
test_VAR_1_case2
Общее количество записей в dns-tunneling.log: 858
Количество записей с поддоменами 1yf.de. и 2yf.de.: 12
VAR_1 is 858
test_VAR_2_case2
Общее количество записей в dns-tunneling.log: 858
Количество записей с поддоменами 1yf.de. и 2yf.de.: 12
VAR_2 is 12
test_results_file_case2
Общее количество записей в dns-tunneling.log: 858
Количество записей с поддоменами 1yf.de. и 2yf.de.: 12
MD5 is ccc5a6854fe5e7dd35c45613443c6382
ASSERT:File 'results.txt' has wrong contents! expected:<7e32a0cb6a5784d4a323bdfd4860b8d> but was:<ccc5a6854fe5e7dd35c45613443c6382>

Ran 7 tests.
FAILED (failures=2)
user@user-VirtualBox:~/lab1/OS$
```

Исходный код программы:

```
#!/bin/bash
```

```
TASKID=9
```

```
# Динамически считаем общее количество строк
```

```
VAR_1=$(wc -l < dns-tunneling.log)
```

```
# Берём строки с 15 по 30 (включительно)
```

```
LINES=$(sed -n '15,30p' dns-tunneling.log)
```

```
# Формируем JSON в файл results.txt
```

```
{
  echo '{'
  echo -e "\tdnslog": ['
```

```
# Обрабатываем строки, преобразуем в JSON-объекты
```

```
echo "$LINES" | awk '
```

```
BEGIN {
```

```
    indent = "\t\t"
```

```
    first = 1
```

```
}
```

```
{
```

```
    timestamp = $1
```

```
    client_ip = $2
```

```
    # Для url всё начиная с третьего поля до конца строки
```

```
    url = ""
```

```
    for(i=3; i<=NF; i++) {
```

```
        url = url $i
```

```
        if(i<NF) url = url " "
```

```
    }
```

```
    # Экранируем двойные кавычки в url
```

```
    gsub(/"/, "\\\"", url)
```

```
    if (first == 0) {
```

```
        print ","
```

```
    } else {
```

```
        first = 0
```

```
    }
```

```
    print indent "{"
```

```
    print indent "\t\t\"timestamp\": \"" timestamp "\","
```

```
    print indent "\t\t\"client ip\": \"" client_ip "\","
```

```
    print indent "\t\t\"url\": \"" url "\""
```

```
    print indent "}"
```

```
}
```

```
END {
```

```
    print ""
```

```
}
```

```
'
```

```
echo -e '\t'
```

```
echo '}'
```

```
} > results.txt
```

```
# Подсчитываем количество записей с поддоменами 1yf.de. или 2yf.de.
```

```
# grep ищет совпадения, wc -l считает строки
```

```
VAR_2=$(grep -E '([0-9a-zA-Z.-]+\.)?(1yf\.de\|2yf\.de\.)' results.txt | wc -l)
```

```
echo "Общее количество записей в dns-tunneling.log: $VAR_1"
```

```
echo "Количество записей с поддоменами 1yf.de. и 2yf.de.: $VAR_2"
```


Выводы

В лабораторной работе было изучено использование команд `grep`, `awk` и `sed` для обработки текстовых файлов, что позволило получить важные навыки работы с текстовыми потоками в командной строке GNU/Linux.