



МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РФ
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ
ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
«ЛИПЕЦКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»

Институт (факультет)
Кафедра

Институт компьютерных наук
Автоматизированные системы управления

ЛАБОРАТОРНАЯ РАБОТА №4

По дисциплине: «Операционные системы»

На тему: «Создание и использование сценариев (скриптов) в Linux»

Студент

АИ-23

Группа

подпись, дата

Жданов М.С.

фамилия, инициалы

Руководитель

кандидат наук

ученая степень, ученое звание

подпись, дата

Кургасов В.В.

фамилия, инициалы

Липецк 2025

Задание:

Задача 1

Написать скрипт на Bash, который должен принимать в качестве аргумента любую строку и удалять из /home/"user" файл с именем, соответствующим переданной строке без необходимости отвечать утвердительно на вопросы системы. Так же, скрипту можно передать параметр «*» (удалит все файлы) и «-» кроме файлов имена который начинаются с параметра после «-». Т.е Sdel.sh file # удалит файл file

Sdel.sh * -file # удалит все, кроме файла file

Задача 2

Написать скрипт, который из /etc/passwd выведет домашнюю директорию для пользователя заданного в параметре скрипта, пример файла /etc/passwd:

```
sync:x:4:65534:sync:/bin:/bin/sync games:x:5:60:games:/usr/games:/bin/sh
man:x:6:12:man:/var/cache/man:/bin/sh
```

```
lp:x:7:7:lp:/var/spool/lpd:/bin/sh
```

```
mail:x:8:8:mail:/var/mail:/bin/sh
```

```
news:x:9:9:news:/var/spool/news:/bin/sh uucp:x:10:10:uucp:/var/spool/uucp:/bin/sh
```

```
proxy:x: 13:13:proxy:/bin:/bin/sh
```

```
www-data:x:33:33:www-data:/var/www:/bin/sh
```

```
backup:x:34:34:backup:/var/backups:/bin/sh
```

```
list:x:38:38:Mailing List Manager:/var/list:/bin/shirc:x:39:39:ircd:/var/run/ircd:/bin/sh
```

```
gnats:x:41:41:Gnats
```

Bug-Reporting

System

```
(admin):/var/lib/gnats:/bin/sh
```

```
nobody:x:65534:65534:nobody:/nonexistent:/bin/sh
```

```
libuuid:x:100:101::/var/lib/libuuid:/bin/sh
```

```
syslog:x: 101:103::/home/syslog:/bin/false
```

```
messagebus:x:102:105::/var/run/dbus:/bin/false
```

Задача 3

процессора в системе.

Написать скрипт, который выведет top 5 процессов, которые потребляют больше всего памяти и

Задача 4

Написать скрипт, который выведет всех потомков процесса по его PID.

Задача 5

Написать скрипт, который для всех файлов во временной директории, которые были созданы за текущий месяц, и не изменялись в течение недели при нахождении в них строки test поменяет ее на tset.

Задача 6

Написать скрипт, которые выведет на экран все файлы, размер которых больше 50 Мб в

директории /var/log

Задача 7

Написать скрипт, который в качестве первого параметра принимает команду, которую нужно выполнить, а в качестве остальных параметров адреса серверов, на которых эта команда должна выполняться.

Ход работы:

1. Написать скрипт на Bash, который должен принимать в качестве аргумента любую строку и удалять из /home/"user" файл с именем, соответствующим переданной строке без необходимости отвечать утвердительно на вопросы системы. Так же, скрипту можно передать параметр «*» (удалит все файлы) и «-» кроме файлов имена который начинаются с параметра после «-». Т.е Sdel.sh file # удалит файл file Sdel.sh * -file # удалит все, кроме файла file – рисунок 1:

```
GNU nano 7.2
dir="/home/tjtn"

if [ "$1" = "*" ] && [-z "$2" ]; then
    rm -f "$dir"/*
    exit 0
fi

if [ "$1" = "*" ] && [[ "$2" == -* ]]; then
    KEEP="${2#-}"
    for f in "$dir"/*; do
        base=$(basename "$f")
        [[ "$base" == "$KEEP"* ]] || rm -f "$1"
    done
    exit 0
fi

rm -f "$dir/$1"
```

Рисунок 1 – Задача 1

```
tjtn@localhost:~$ ls
1.txt 2.txt 3.txt file.txt script.sh
tjtn@localhost:~$ ./script.sh 1.txt
tjtn@localhost:~$ ls
2.txt 3.txt file.txt script.sh
tjtn@localhost:~$ _
```

Рисунок 2 – Результат выполнения скрипта для задачи 1

2. Написать скрипт, который из /etc/passwd выведет домашнюю директорию для пользователя заданного в параметре скрипта – рисунок 3:

```
GNU nano 7.2
USER="$1"
grep "^$USER:" /etc/passwd | cut -d: -f6
```

Рисунок 3 – Задача 2

```
tjtn@localhost:~$ ./script.sh tjtn
/home/tjtn
tjtn@localhost:~$ _
```

Рисунок 4 – Результат выполнения скрипта для задачи 2

3. Написать скрипт, который выведет top 5 процессов, которые потребляют больше всего памяти и процессора в системе:

```
GNU nano 7.2
ps -eo pid,comm,%cpu,%mem --sort=-%cpu | head -n 6
```

Рисунок 5 – Задача 3

```
tjtt@localhost:~$ ./script.sh
PID COMMAND          %CPU %MEM
 25 kworker/1:0-eve    0.0  0.0
 758 kworker/0:6-eve   0.0  0.0
  59 kworker/u4:4-ev   0.0  0.0
 366 multipathd        0.0  0.6
   1 systemd           0.0  0.3
tjtt@localhost:~$
```

Рисунок 6 – Результат выполнения скрипта для задачи 3

4. Написать скрипт, который выведет всех потомков процесса по его PID.

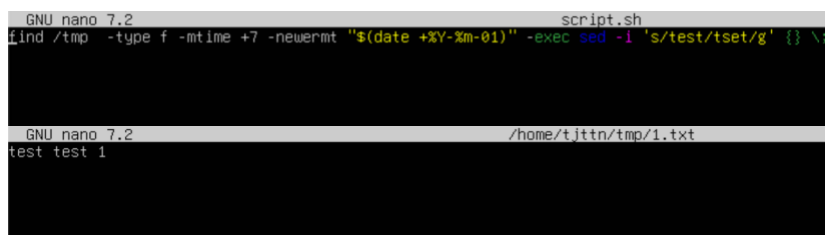
```
GNU nano 7.2
ps --ppid "$1" -o pid=
```

Рисунок 7 – Задача 4

```
tjtt@localhost:~$ ./script.sh 1
311
366
390
538
542
545
682
687
695
701
715
755
765
831
865
999
8384
tjtt@localhost:~$ _
```

Рисунок 8 – Результат выполнения скрипта для задачи 4

5. Написать скрипт, который для всех файлов во временной директории, которые были созданы за текущий месяц, и не изменялись в течение недели при нахождении в них строки test поменяет ее на tset.



```
GNU nano 7.2 script.sh
find /tmp -type f -mtime +7 -newermt "$(date +%Y-%m-01)" -exec sed -i 's/test/tset/g' {} \;
```

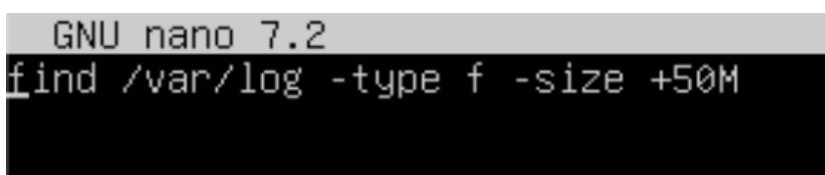
Рисунок 9 – Скрипт для задачи 5 и исходные данные файла



```
GNU nano 7.2 /home/tjtn/tmp/1.txt
tset tset 1
```

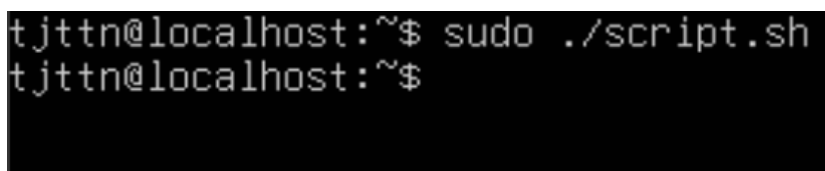
Рисунок 10 – Результат работы скрипта для задачи 5

6. Написать скрипт, которые выведет на экран все файлы, размер которых больше 50 Мб в директории /var/log



```
GNU nano 7.2
find /var/log -type f -size +50M
```

Рисунок 11 – Задача 6



```
tjtn@localhost:~$ sudo ./script.sh
tjtn@localhost:~$
```

Рисунок 12 – Результат работы скрипта для задачи 6

Пояснение к рисунку 12. Так как в системе у меня нет файлов с весом больше 50 МБ, поэтому вывод пустой.

7. Написать скрипт, который в качестве первого параметра принимает команду, которую нужно выполнить, а в качестве остальных параметров адреса серверов, на которых эта команда должна выполняться.

```
GNU nano 7.2
cmd="$1"
shift

for host in "$@"; do
    ssh "$host" "$cmd"
done
```

Рисунок 13 – Задача 7

```
tjtn@localhost:~$ ./script.sh "uptime" localhost
ssh: connect to host localhost port 22: Connection refused
tjtn@localhost:~$ uptime
 18:42:14 up 57 min,  1 user,  load average: 0,00, 0,00, 0,00
tjtn@localhost:~$
```

Рисунок 14 – Результат работы скрипта для задачи 7

Так как мы работаем на локальной ОС можно просто прописать команду ‘uptime’, чтобы посмотреть время в сети.

Вывод:

В ходе лабораторной работы я получил практические навыки по написанию bash-скриптов для ОС Linux.