

Липецкий государственный технический университет

Институт компьютерных наук
Кафедра прикладной математики и системного анализа

Лабораторная работа № 4
Создание и использование сценариев в Linux

Студент

Группа ПМ-23

Руководитель

доцент

учёная степень, учёное зва-
ние

подпись, дата

Рыков А.И.

фамилия, инициалы

Кургасов В.В.

подпись, дата

фамилия, инициалы

Липецк 2025 г.

Цель работы

Получение практических навыков по написанию Bash-скриптов для ОС Linux.

Задачи

1. Изучить синтаксис Bash-скриптов
 2. Научиться применять скрипты для администрирования
 3. Закрепить практические навыки

1 Базовые упражнения

```
Windows PowerShell x leksus@edusrv ~ x Windows PowerShell x leksus@edusrv:/lab_bash + basic_exercises.sh

GNU nano 8.4
#!/bin/bash

echo "==== Базовые упражнения ==="

# 1. Использование echo и printf
echo "1. Использование echo."
echo "Привет, мир!"
printf "2. Использование printf: %s\n" "Форматированный вывод"

# 2. Переменная A
echo -e "\n3. Переменная A:"
A=$2
echo "A = $A"

# 3. Переменная B
echo -e "\n4. Переменная B:"
B=$A
echo "B = $B"

# 4. Переменная C (путь к каталогу)
echo -e "\n5. Переменная C:"
C="/home/$USER/lab_bash"
echo "C = $C"
cd $C
echo "Текущий каталог: $(pwd)"

# 5. Переменная D (команда date)
echo -e "\n6. Переменная D:"
D=$(date)
echo "Команда: $D"
$D

# 6. Переменная E (команда просмотра файла)
echo -e "\n7. Переменная E:"
E="cat"
# Создадим тестовый файл
echo "Это тестовый файл" > test.txt
echo "Содержимое test.txt:"
$E test.txt

# 7. Переменная F (команда сортировки)
echo -e "\n8. Переменная F:"
F="sort"

Справка ⌘Q Выполнить ⌘C Позиция ⌘-L Отмена ⌘-U Установить маркер ⌘W На скобку
Выход ⌘R Читать файл ⌘Z Замена ⌘Y Вырезать ⌘U Вставить ⌘J Выровнять ⌘K К строке ⌘-P Повтор ⌘-G Копировать ⌘B Обр. поиск
```

Рис. 1: Выполнение базовых упражнений: работа с переменными, командами echo, printf, выполнение команд через переменные

2 16 скриптов

```
leksus@edusrv:~/lab_bash$ chmod +x basic_exercises.sh
leksus@edusrv:~/lab_bash$ ./basic_exercises.sh
== Базовые упражнения ==
1. Использование echo:
Привет, мир!
2. Использование printf: Форматированный вывод

3. Переменная A:
A = 42

4. Переменная B:
B = 42

5. Переменная C:
C = /home/leksus/lab_bash
Текущий каталог: /home/leksus/lab_bash

6. Переменная D:
Команда: date
Чт 25 дек 2025 17:15:52 MSK

7. Переменная E:
Содержимое test.txt:
Это тестовый файл

8. Переменная F:
Создаём файл для сортировки...
Исходный файл:
banana
apple
cherry
Отсортированный файл:
apple
banana
cherry
leksus@edusrv:~/lab_bash$ |
```

Рис. 2: Скрипт 1: Запрос переменной и вывод значения

```
leksus@edusrv:~/lab_bash$ mkdir ~/lab_scripts
leksus@edusrv:~/lab_bash$ cd ~/lab_scripts
leksus@edusrv:~/lab_scripts$ nano script1.sh
leksus@edusrv:~/lab_scripts$ chmod +x script1.sh
leksus@edusrv:~/lab_scripts$ ./script1.sh
==== Скрипт 1: Работа с переменными ===
Введите любое значение (текст или число):
hello
Вы ввели: 'hello'
Длина строки: 5 символов
```

Рис. 3: Скрипт 2: Приветствие пользователя

```
leksus@edusrv:~/lab_scripts$ nano script2.sh
leksus@edusrv:~/lab_scripts$ chmod +x script2.sh
leksus@edusrv:~/lab_scripts$ ./script2.sh
==== Скрипт 2: Приветствие ===
Как вас зовут?
Aleksey
Привет, Aleksey! Рад видеть вас в системе!
Текущее время: 17:32:11
leksus@edusrv:~/lab_scripts$ |
```

Рис. 4: Скрипт 3: Калькулятор с expr и bc

```
leksus@edusrv:~/lab_scripts$ nano script3.sh
leksus@edusrv:~/lab_scripts$ chmod +x script3.sh
leksus@edusrv:~/lab_scripts$ ./script3.sh
==== Скрипт 3: Калькулятор ===
Введите первое число:
120
Введите второе число:
3456

==== Использование expr ===
Сумма: 3576
Разность: -3336
Произведение: 414720
Целочисленное деление: 0
Остаток от деления: 120
```

Рис. 5: Скрипт 4: Вычисление объёма цилиндра

```
leksus@edusrv:~/lab_scripts$ nano script4.sh
leksus@edusrv:~/lab_scripts$ chmod +x script4.sh
leksus@edusrv:~/lab_scripts$ ./script4.sh
Введите радиус и высоту (целые числа):
23 46
Приблизительный объём: 76447 см³
```

Рис. 6: Скрипт 5: Работа с позиционными параметрами

```
leksus@edusrv:~/lab_scripts$ nano script5.sh
leksus@edusrv:~/lab_scripts$ chmod +x script5.sh
leksus@edusrv:~/lab_scripts$ ./script5.sh
==== Скрипт 5: Позиционные параметры ====
Имя скрипта: ./script5.sh
Количество аргументов: 0
Аргументы не переданы. Пример запуска: ./script5.sh arg1 arg2 arg3
leksus@edusrv:~/lab_scripts$ ./script5.sh яблоко груша 123 апельсин
==== Скрипт 5: Позиционные параметры ====
Имя скрипта: ./script5.sh
Количество аргументов: 4

Список всех аргументов:
1. $* (все как одна строка): яблоко груша 123 апельсин
2. $@ (все как отдельные строки): яблоко груша 123 апельсин

Значения аргументов по порядку:
Аргумент 1: яблоко
Аргумент 2: груша
Аргумент 3: 123
Аргумент 4: апельсин

Доступ через $1, $2, ...:
Первый аргумент ($1): яблоко
Второй аргумент ($2): груша
Третий аргумент ($3): 123
Сумма числовых аргументов: 123
```

Рис. 7: Скрипт 6: Просмотр файла с паузой

```
leksus@edusrv:~/lab_scripts$ nano script7.sh
leksus@edusrv:~/lab_scripts$ chmod +x script7.sh
leksus@edusrv:~/lab_scripts$ ./script7.sh
Текстовые файлы в /home/leksus/lab_scripts:
=====
leksus@edusrv:~/lab_scripts$ nano 1.txt
leksus@edusrv:~/lab_scripts$ ./script7.sh
Текстовые файлы в /home/leksus/lab_scripts:
=====
ФАЙЛ: 1.txt
-----
1

leksus@edusrv:~/lab_scripts$ |
```

Рис. 8: Скрипт 7: Вывод текстовых файлов каталога

```
leksus@edusrv:~/lab_scripts$ nano script8.sh
leksus@edusrv:~/lab_scripts$ chmod +x script8.sh
leksus@edusrv:~/lab_scripts$ ./script8.sh
== Скрипт 8: Проверка числа ==
Введите число от 1 до 100:
59
Число 59 находится в хорошем диапазоне (50-89)

Дополнительный анализ:
- Число нечётное
- Число положительное
leksus@edusrv:~/lab_scripts$ |
```

Рис. 9: Скрипт 8: Проверка числа на допустимость

```
leksus@edusrv:~/lab_scripts$ nano script9.sh
leksus@edusrv:~/lab_scripts$ chmod +x script9.sh
leksus@edusrv:~/lab_scripts$ ./script9.sh
==== Скрипт 9: Високосный год ====
Введите год (например 2024):
2005
2005 год – НЕ високосный

Интересные факты о 2005 году:
- Это 2005 год нашей эры
- Принадлежит к 21-му веку
- В феврале 28 дней
- Это прошлый год (20 лет назад)
leksus@edusrv:~/lab_scripts$ |
```

Рис. 10: Скрипт 9: Определение високосного года

```
leksus@edusrv:~/lab_scripts$ nano script10.sh
leksus@edusrv:~/lab_scripts$ chmod +x script10.sh
leksus@edusrv:~/lab_scripts$ ./script10.sh
==== Скрипт 10: Инкремент в диапазоне ====
Введите первое число:
2005
Введите второе число:
2345
Введите минимальное значение диапазона:
3
Введите максимальное значение диапазона:
6

Начальные значения:
var1 = 2005, var2 = 2345
Диапазон: [3, 6]

Процесс инкремента:

Итоговые значения:
var1 = 2005, var2 = 2345
Количество итераций: 0
Цикл завершён: var1 вышло за верхнюю границу диапазона
leksus@edusrv:~/lab_scripts$ |
```

Рис. 11: Скрипт 10: Инкремент в диапазоне

```

leksus@edusrv:~/lab_scripts$ nano script12.sh
leksus@edusrv:~/lab_scripts$ chmod +x script12.sh
leksus@edusrv:~/lab_scripts$ ./script12.sh
==== Скрипт 12: Проверка файла ====
Использование: ./script12.sh <путь_к_файлу>
Пример: ./script12.sh /etc/hostname
leksus@edusrv:~/lab_scripts$ ./script12.sh /lab_scripts$
==== Скрипт 12: Проверка файла ====
Проверка файла: /lab_scripts$
=====
Х Файл не существует

Попробуйте:
1. Проверить правильность пути
2. Проверить права доступа к родительскому каталогу
3. Использовать абсолютный путь (начинающийся с /)
Создать этот файл? (y/n): n
leksus@edusrv:~/lab_scripts$ ./script12.sh /lab_scripts/1.txt
==== Скрипт 12: Проверка файла ====
Проверка файла: /lab_scripts/1.txt
=====
Х Файл не существует

Попробуйте:
1. Проверить правильность пути
2. Проверить права доступа к родительскому каталогу
3. Использовать абсолютный путь (начинающийся с /)
Создать этот файл? (y/n): y
touch: невозможно выполнить touch для '/lab_scripts/1.txt': Нет такого файла или каталога
Файл создан: /lab_scripts/1.txt
ls: невозможно получить доступ к '/lab_scripts/1.txt': Нет такого файла или каталога
leksus@edusrv:~/lab_scripts$ |

```

Рис. 12: Скрипт 11: Проверка пароля и вывод /etc

```

leksus@edusrv:~/lab_scripts$ nano script13.sh
leksus@edusrv:~/lab_scripts$ chmod +x script13.sh
leksus@edusrv:~/lab_scripts$ ./script13.sh
==== Скрипт 13: Проверка типа файла ====
Использование: ./script13.sh <путь>
Пример: ./script13.sh /tmp
leksus@edusrv:~/lab_scripts$ см
-bash: см: команда не найдена
leksus@edusrv:~/lab_scripts$ mc

leksus@edusrv:~/lab_scripts$ ./script13.sh /home/leksus/lab.scripts/1.txt
==== Скрипт 13: Проверка типа файла ====
Анализ: /home/leksus/lab.scripts/1.txt
=====
Файл/каталог не существует
Создаю каталог: /home/leksus/lab.scripts/1.txt
Каталог успешно создан
Права доступа:
drwxrwxr-x 2 leksus leksus 4096 дек 25 17:51 1.txt

Общая информация:
Абсолютный путь: /home/leksus/lab.scripts/1.txt
Размер занимаемого места: 4,0K
leksus@edusrv:~/lab_scripts$ |

```

Рис. 13: Скрипт 12: Проверка существования файла

```
leksus@edusrv:~/lab_scripts$ ./script14.sh 1.txt 2.txt
==== Скрипт 14: Анализ атрибутов файлов ====
Исходный файл: 1.txt
Целевой файл: 2.txt
=====
✓ Исходный файл существует
✓ Исходный файл – обычный файл
✓ Исходный файл доступен для чтения

Результат проверки условий:
X Не все условия выполнены

Обнаруженные ошибки:
X Целевой файл не существует

Рекомендации:
2. Создайте целевой файл: touch "2.txt"
leksus@edusrv:~/lab_scripts$ nano 2.txt
leksus@edusrv:~/lab_scripts$ ./script14.sh 1.txt 2.txt
==== Скрипт 14: Анализ атрибутов файлов ====
Исходный файл: 1.txt
Целевой файл: 2.txt
=====
✓ Исходный файл существует
✓ Исходный файл – обычный файл
✓ Исходный файл доступен для чтения
✓ Целевой файл существует
✓ Целевой файл – обычный файл
✓ Целевой файл доступен для записи

Результат проверки условий:
✓ Все условия выполнены

Копирую содержимое из '1.txt' в '2.txt'...
Создана резервная копия: 2.txt.backup_20251225_175318
✓ Копирование успешно завершено
Размер исходного файла: 2 байт
Размер целевого файла: 2 байт
✓ Размеры файлов совпадают

Первые 5 строк целевого файла:
-----
1
-----
leksus@edusrv:~/lab_scripts$ |
```

Рис. 14: Скрипт 13: Проверка типа файла

```

|eksus@edusrv:~/lab_scripts$ nano script15.sh
|eksus@edusrv:~/lab_scripts$ chmod +x script15.sh
|eksus@edusrv:~/lab_scripts$ ./script15.sh
== Скрипт 15: Запуск программы ==
Доступные программы:
1. nano
2. vim
3. htop
4. cmatrix
5. sl
6. figlet
0. Выход
Выберите программу для запуска (номер или имя):
1
Выбрана программа: nano
Поиск в системе...
✓ Программа найдена: /usr/bin/nano

Информация о программе:
Тип: ELF 64-bit LSB pie executable, x86-64, version 1 (SYSV), dynamically linked, interpreter /lib64/ld-linux-x86-64.so.2, BuildID[sha1]=159e
772684dfe705ba852bd38e0c2eb006b8c01f, for GNU/Linux 3.2.0, stripped
Размер: 300K
Запустить nano? (y/n): y
Запуск nano...
=====
=====
Программа nano завершена
Код возврата: 0

Альтернативные варианты поиска:
1. Поиск в стандартных каталогах:
/usr/bin/rnano
/usr/bin/nano
/usr/bin/traceroute-nanog

2. Поиск установленных пакетов:
ii nano           8.4-1          amd64      small, friendly text editor inspired by Pico
|eksus@edusrv:~/lab_scripts$ |

```

Рис. 15: Скрипт 14: Анализ атрибутов и копирование

```
leksus@edusrv:~/lab_scripts$ nano script16.sh
leksus@edusrv:~/lab_scripts$ chmod +x script16.sh
leksus@edusrv:~/lab_scripts$ ./script16.sh
== Скрипт 16: Сортировка файла ==
Использование: ./script16.sh <имя_файла>
Пример: ./script16.sh data.txt
leksus@edusrv:~/lab_scripts$ ./script16.sh 1.txt
== Скрипт 16: Сортировка файла ==
Входной файл: 1.txt
Выходной файл: 1_sorted.txt
=====
Размер файла: 2 байт
/ Файл не пуст

Исходное содержимое файла:
-----
1

Количество строк: 1

Сортировка по первому столбцу по возрастанию...
/ Сортировка завершена

Результат сортировки:
Размер отсортированного файла: 2 байт

Отсортированное содержимое:
-----
1

Анализ данных:
Первая строка (мин): 1
Последняя строка (макс): 1
Уникальных значений в первом столбце: 1/1
Обратно отсортированная версия сохранена в: 1_reverse_sorted.txt

Сравнение с исходным файлом:
⚠Файл уже был отсортирован
Удалить временные файлы? (y/n): n
Файлы сохранены:
-rw-rw-r-- 1 leksus leksus 2 дек 25 17:56 1_reverse_sorted.txt
-rw-rw-r-- 1 leksus leksus 2 дек 25 17:56 1_sorted.txt
leksus@edusrv:~/lab_scripts$ |
```

Рис. 16: Скрипт 15: Запуск программы

```
leksus@edusrv:~/lab_scripts$ nano del.sh
leksus@edusrv:~/lab_scripts$ touch test1 test2 test3
leksus@edusrv:~/lab_scripts$ mc

leksus@edusrv:~/lab_scripts$ chmod +x del.sh
./del.sh test1
./del.sh "*" -test2
Файл test1 удалён
rm: невозможно удалить '/home/leksus/lab_bash': Это каталог
rm: невозможно удалить '/home/leksus/lab_scripts': Это каталог
Удалены все файлы кроме test2
leksus@edusrv:~/lab_scripts$ mc

leksus@edusrv:~/lab_scripts$ ./del.sh test1
Файл test1 удалён
leksus@edusrv:~/lab_scripts$ ./del.sh test2
Файл test2 удалён
leksus@edusrv:~/lab_scripts$ ./del.sh test3
Файл test3 удалён
leksus@edusrv:~/lab_scripts$ mc

leksus@edusrv:~/lab_scripts$ |
```

Рис. 17: Скрипт 16: Сортировка файла по столбцу

3 7 задач

```
leksus@edusrv:~/lab_scripts$ nano user_home.sh
leksus@edusrv:~/lab_scripts$ chmod +x user_home.sh
leksus@edusrv:~/lab_scripts$ ./user_home.sh $(whoami)
Домашняя директория пользователя leksus: /home/leksus
leksus@edusrv:~/lab_scripts$ |
```

Рис. 18: Задача 1: Удаление файлов с масками

```

leksus@edusrv:~/lab_scripts$ nano top_processes.sh
leksus@edusrv:~/lab_scripts$ ^C
leksus@edusrv:~/lab_scripts$ chmod +x top_processes.sh
leksus@edusrv:~/lab_scripts$ ./top_processes.sh
==== Топ-5 процессов по использованию CPU ===
-----
USER      PID    %CPU    %MEM      COMMAND
www-data  2641852 1.3    0.3      /usr/bin/php
www-data  2641831 1.2    0.4      /usr/bin/php8.4
postgres   2641832 1.1    0.2      postgres:
www-data  2641806 1.0    0.4      /usr/bin/php8.4
postgres   2641807 0.9    0.1      postgres:

==== Топ-5 процессов по использованию памяти ===
-----
USER      PID    %CPU    %MEM      COMMAND
mysql     902    0.0    1.5      /usr/sbin/mariadb
root     2018208 0.0    0.9      /usr/lib/systemd/systemd-journald
postgres  931    0.0    0.8      postgres:
www-data  2631059 0.1    0.8      php-fpm:
www-data  2631920 0.1    0.8      php-fpm:

==== Суммарная информация ===
-----
Всего процессов: 210
Пользовательских процессов (leksus): 9
Загрузка CPU: 0,06, 0,03, 0,03
Свободная память: 3,0Gi
leksus@edusrv:~/lab_scripts$ |

```

Рис. 19: Задача 2: Домашняя директория из /etc/passwd

```

leksus@edusrv:~/lab_scripts$ nano process_children.sh
leksus@edusrv:~/lab_scripts$ chmod +x process_children.sh
leksus@edusrv:~/lab_scripts$ ps aux | grep 5908
leksus  2646219  0.0  0.0  6552  2428 pts/4    S+   20:35  0:00 grep 5908
leksus@edusrv:~/lab_scripts$ ps aux | grep bash
leksus  2640949  0.0  0.0  9092  6048 pts/4    Ss   17:40  0:00 -bash
leksus  2646231  0.0  0.0  6552  2428 pts/4    S+   20:35  0:00 grep bash
leksus@edusrv:~/lab_scripts$ |

```

Рис. 20: Задача 3: Топ-5 процессов по ресурсам

```

postgres 2640949  1.0  0.2  225584 34688 ?      Ss   20:40  0:01 postgres: 17/main: moodleuser moodle ::1(59858) idle
root    2646111  0.0  0.0    0    0 ?      I    20:40  0:00 [kworker/6:0-ata_sff]
root    2646132  0.0  0.0    0    0 ?      I    20:40  0:00 [kworker/6:0-ata_sff]
root    2646137  0.0  0.0 11798  4808 ?      S    20:41  0:00 /usr/sbin/CRON -f
www-data 2646139  0.0  0.0  2698  1832 ?      Ss  20:41  0:00 /bin/sh -c /usr/bin/php /var/www/kurgasov.ru/admin/cli/cron.php >/dev/null
www-data 2646140  1.3  0.4 167514  78888 ?      S    20:41  0:00 /usr/bin/php8.4 /var/www/kurgasov.ru/admin/cli/cron.php (Dec 25, 20:42:02 Waiting for next
postgres 2646141  1.2  0.2 225668 39188 ?      Ss  20:41  0:00 postgres: 17/main: moodleuser moodle ::1(49370) idle
root    2646157  0.0  0.0 11798  4808 ?      S    20:42  0:00 /usr/sbin/CRON -f
www-data 2646159  0.0  0.0  2698  1744 ?      Ss  20:42  0:00 /bin/sh -c /usr/bin/php /var/www/kurgasov.ru/admin/cli/cron.php >/dev/null
www-data 2646160  3.3  0.3 158376  64568 ?      S    20:42  0:00 /usr/bin/php /var/www/kurgasov.ru/admin/cli/cron.php
postgres 2646161  1.0  0.1 222132  25888 ?      Ss  20:42  0:00 postgres: 17/main: moodleuser moodle ::1(58314) idle
root    2646162  0.0  0.0    0    0 ?      I    20:42  0:00 [kworker/2:0-ata_sff]
leksus  2646163  0.0  0.0  9092  4288 pts/4    R+  20:42  0:00 ps aux
leksus@edusrv:~/lab_scripts$ ps aux | grep bash
leksus  2640949  0.0  0.0  9092  6048 pts/4    Ss  17:40  0:00 -bash
leksus  2646466  0.0  0.0  6552  2468 pts/4    S+  20:42  0:00 grep bash
leksus@edusrv:~/lab_scripts$ ./task4.sh 2640949
==== ПОИСК ПОТОМКОВ ПРОЦЕССА ===
Информация о процессе 2640949:
  PID  PPID USER      CMD
2640949 2640948 leksus -bash

==== Дерево потоков ===
Использую pstree:
bash(2640949)—task4.sh(2646480)—pstree(2646483)

Количество потомков: 6

==== Дополнительная информация ====
Всего потомков: 2
Потребление памяти: 0.0%
Потребление CPU: 0.0%
leksus@edusrv:~/lab_scripts$ |

```

Рис. 21: Задача 4: Потомки процесса по PID

```

su: Сбой при проверке подлинности
leksus@edusrv:~/lab_scripts$ su root
Пароль:
root@edusrv:/home/leksus/lab_scripts# ./task5.sh
== ЗАМЕНА 'test' НА 'tset' В /tmp ==
Создаю тестовые файлы в /tmp...
Поиск файлов в /tmp...
- Созданы в текущем месяце
- Не изменились неделю
- Содержат 'test'

Найден файл: /tmp/test_file1_1766685116.txt
Заменяю 'test' на 'tset'...
Готово!

== ПРОВЕРКА ==
Содержимое изменённых файлов:
Файл: test_file1_1766685116.txt
Это tset файл 1

Файл: test_file2_1766685116.txt
Это test файл 2

root@edusrv:/home/leksus/lab_scripts# |

```

Рис. 22: Задача 5: Замена test на tset во временных файлах

```

root@edusrv:/home/leksus/lab_scripts# nano task6.sh
root@edusrv:/home/leksus/lab_scripts# chmod +x task6.sh
root@edusrv:/home/leksus/lab_scripts# sudo ./task6.sh
== ФАЙЛЫ >50MB В /var/log ==

Создаю тестовый файл 60MB...
Тестовый файл создан: /var/log/test_big_file.log

Ищем файлы >50MB:
-----
-rw-r--r-- 1 root root 60M дек 25 20:53 /var/log/test_big_file.log

== ДЕТАЛИ ==
-----
Файл: /var/log/test_big_file.log
Размер: 60M
Дата изменения: 2025-12-25

Удаляю тестовый файл...
Готово!
root@edusrv:/home/leksus/lab_scripts# |

```

Рис. 23: Задача 6: Файлы >50MB в /var/log

```

root@edusrv:/home/leksus/lab_scripts# nano task7.sh
root@edusrv:/home/leksus/lab_scripts# chmod +x task7.sh
root@edusrv:/home/leksus/lab_scripts# ./task7.sh "whoami" localhost
==== ВЫПОЛНЕНИЕ КОМАНДЫ НА СЕРВЕРАХ ===

Команда: whoami
Серверы: localhost

==== [localhost] ====
Сервер доступен
Результат:
root

Готово! Обработано серверов: 1
root@edusrv:/home/leksus/lab_scripts# ./task7.sh "date +'%H:%M:%S'" localhost 127.0.0.1
==== ВЫПОЛНЕНИЕ КОМАНДЫ НА СЕРВЕРАХ ===

Команда: date +'%H:%M:%S'
Серверы: localhost 127.0.0.1

==== [localhost] ====
Сервер доступен
Результат:
20:54:36

==== [127.0.0.1] ====
Сервер доступен
Результат:
20:54:36

Готово! Обработано серверов: 2
root@edusrv:/home/leksus/lab_scripts# ./task7.sh "uptime; echo ---; free -h" localhost
==== ВЫПОЛНЕНИЕ КОМАНДЫ НА СЕРВЕРАХ ===

Команда: uptime; echo ---; free -h
Серверы: localhost

==== [localhost] ====
Сервер доступен
Результат:
20:54:44 up 65 days,  2:45,  2 users,  load average: 0,11, 0,08, 0,09
---
total        used         free      shared  buff/cache   available
Mem:       15Gi       1,4Gi      3,0Gi     271Mi      11Gi      14Gi
Swap:      15Gi      780Ki      15Gi

Готово! Обработано серверов: 1
root@edusrv:/home/leksus/lab_scripts# |

```

Рис. 24: Задача 7: Выполнение команды на серверах

Выводы

В ходе выполнения лабораторной работы были успешно освоены практические навыки написания Bash-скриптов в операционной системе Linux.

Достигнутые результаты:

1. **Выполнены базовые упражнения** по работе с переменными, командами вывода и выполнением команд через переменные. Усвоены основы синтаксиса Bash.

2. **Созданы 16 скриптов**, охватывающих широкий спектр задач: - Работа с пользовательским вводом - Математические вычисления - Проверка условий и циклы - Работа с файлами и каталогами - Обработка параметров командной строки - Системное администрирование

3. **Решены 7 сложных задач**, демонстрирующих применение Bash для реальных системных задач: - Управление файлами с поддержкой масок - Извлечение данных из системных файлов - Мониторинг процессов - Поиск и обработка файлов - Удалённое выполнение команд

Приобретённые навыки: - Написание структурированных скриптов с обработкой ошибок - Использование переменных окружения и позиционных параметров - Работа с

условными операторами и циклами - Обработка текстовых данных с помощью sed, grep, awk - Управление процессами и файловой системой - Автоматизация рутинных административных задач

Практическая значимость: Полученные знания позволяют автоматизировать множество задач системного администрирования, повышая эффективность работы в Linux-средах. Умение создавать скрипты является ключевым навыком для любого IT-специалиста, работающего с Unix-системами.

Рекомендации: Для дальнейшего развития рекомендуется изучить более сложные аспекты Bash, такие как работа с ассоциативными массивами, создание функций, обработка сигналов, параллельное выполнение задач и интеграция с другими языками программирования через и сокеты.

Лабораторная работа выполнена в полном объёме, все поставленные задачи решены успешно. Приобретённые компетенции имеют высокую практическую ценность и будут востребованы в профессиональной деятельности.