

Отчет по лабораторной работе 13

Программирование в командном процессоре ОС UNIX. Ветвления и циклы.

Власов Артем Сергеевич

Содержание

1	Цель работы	1
2	Задание.....	1
3	Выполнение лабораторной работы 13.	1
4	Контрольные вопросы	9
5	Выводы.....	9
	Список литературы.....	9

1 Цель работы

Изучить основы программирования в оболочке ОС UNIX. Научиться писать небольшие командные файлы с ветвлениями и циклами.

2 Задание

Выполнить последовательность действий по заданному сценарию, написать 4 скрипта для разных целей.

3 Выполнение лабораторной работы 13.

Создание файла первого скрипта и изменение его прав доступа. (рис. fig. 1).

```
[vlasovas@vbox 13lab]$ touch scr1.sh
[vlasovas@vbox 13lab]$ ./scr1.sh -i text -p "love"
bash: ./scr1.sh: Отказано в доступе
[vlasovas@vbox 13lab]$ chmod 777 scr1.sh
```

Рис. 1: Создание файла первого скрипта и изменение его прав доступа

Код первого скрипта. (рис. fig. 2).

```

input_file=""
output_file=""
pattern=""
case_sensitive=0
show_numbers=0

while getopts ":i:o:p:Cn" opt; do
    case $opt in
        i) input_file="$OPTARG";;
        o) output_file="$OPTARG";;
        p) pattern="$OPTARG";;
        C) case_sensitive=1;;
        n) show_numbers=1;;
        \?) echo "Неверный ключ: -$OPTARG" >&2;
    exit 1;;
        :) echo "Ключ -$OPTARG требует аргумента" >&2;
    exit 1;;
    esac
done

if [ -z "$input_file" ] || [ -z "$pattern" ]
; then
    echo "Использование: $0 -i inputfile -p pattern [-o outputfile] [-C] [-n]"
    exit 1
fi

```

Рис. 2: Первый скрипт

Проверка работы первого скрипта. (рис. fig. 3).

```

[vlasovas@vbox 13lab]$ ./scr1.sh -i text -p "hello"
Hello
[vlasovas@vbox 13lab]$ ./scr1.sh -i text -p "love"
I love coding

```

Рис. 3: Проверка

```

[vlasovas@vbox 13lab]$ ./scr1.sh -i text -p "love" -C
I love coding
[vlasovas@vbox 13lab]$ ./scr1.sh -i text -p "Love" -C
[vlasovas@vbox 13lab]$

```

Рис. 4: Проверка

```

[vlasovas@vbox 13lab]$ ./scr1.sh -i text -p "love" -n
3:I love Mother
6:I love coding
[vlasovas@vbox 13lab]$

```

Рис. 5: Проверка

```

[vlasovas@vbox 13lab]$ ./scr1.sh -i text -p "love" -o result.txt
[vlasovas@vbox 13lab]$ cat result.txt
I love Mother
I love coding
[vlasovas@vbox 13lab]$

```

Рис. 6: Проверка

```

[vlasovas@vbox 13lab]$ ./scr1.sh -i text -p "love" -C -n -o result.txt
[vlasovas@vbox 13lab]$ cat result.txt
3:I love Mother
6:I love coding
[vlasovas@vbox 13lab]$

```

Рис. 7: Проверка

Код второго скрипта на С (рис. fig. 8).

```

1 #include <stdio.h>
2 #include <stdlib.h>
3
4 int main() {
5     int number;
6     printf("Введите число: ");
7     scanf("%d", &number);
8
9     if (number > 0) {
10         exit(1);
11     } else if (number < 0) {
12         exit(2);
13     } else {
14         exit(0);
15     }
16 }

```

Рис. 8: Код второго скрипта на С

Код второго скрипта(командный файл). (рис. fig. 4fig:009?).

```

if [ ! -f number_checker ]; then
    gcc number_checker.c -o number_checker
fi

./number_checker

case $? in
    0) echo "Введен ноль";;
    1) echo "Введено положительное число";;
    2) echo "Введено отрицательное число";;
    *) echo "Неизвестный код завершения";;
esac

```

Рис. 9: Второй скрипт

Проверка работы второго скрипта. (рис. fig. 11).

```

[vlasovas@vbox 13lab]$ ./scr2.sh
Введите число: 4
Введено положительное число
[vlasovas@vbox 13lab]$ ./scr2.sh
Введите число: -6
Введено отрицательное число
[vlasovas@vbox 13lab]$ ./scr2.sh
Введите число: 0
Введен ноль
[vlasovas@vbox 13lab]$

```

Рис. 10: Проверка

Код третьего скрипта (рис. fig. 11).

```

if [ $# -ne 1 ]; then
    echo "Использование: $0 [create N|clean]"
    exit 1
fi

case $1 in
    create*)

        N=$(echo $1 | cut -d' ' -f2)
        if ! [[ "$N" =~ ^[0-9]+$ ]]; then
            echo "Некорректное число файлов: $N"
            exit 1
        fi

        for i in $(seq 1 $N); do
            touch "$i.tmp"
        done
        echo "Создано $N файлов"
        ;;
    clean)

        count=$(ls *.tmp 2>/dev/null | wc -l)
        rm -f *.tmp
        echo "Удалено $count файлов"
        ;;
    *)
        echo "Неверная команда. Используйте create N
или clean"
        exit 1
        ;;
esac

```

Рис. 11: Третий скрипт

Проверка работа третьего скрипта(рис. fig. **fig:014?**).

```

[vlasovas@vbox 13lab]$ ./scr3.sh "create 5"
Создано 5 файлов

```

Рис. 12: Проверка

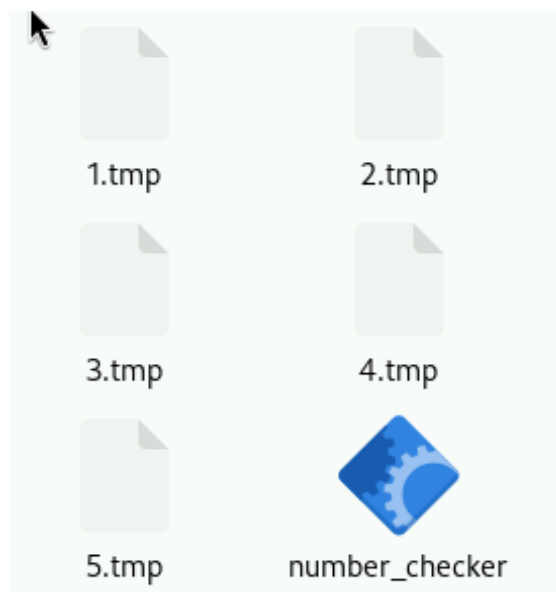


Рис. 13: Проверка

```
[vlasovas@vbox 13lab]$ ./scr3.sh clean  
Удалено 5 файлов
```

Рис. 14: Проверка

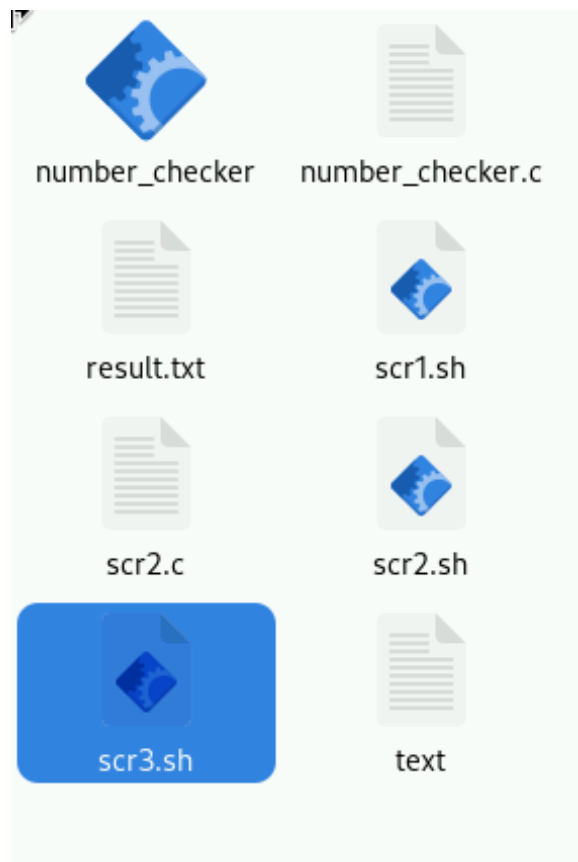


Рис. 15: Проверка

Код четвертого скрипта. (рис. fig. 16).

```

if [ $# -ne 1 ]; then
    echo "Использование: $0 директория"
    exit 1
fi

directory="$1"
archive_name="backup_$(date +%Y%m%d_%H%M%S).tar.gz"

if [ ! -d "$directory" ]; then
    echo "Директория $directory не существует"
    exit 1
fi

#tar -czf "$archive_name" "$directory"

find "$directory" -type f -mtime -7 -print0 | tar -czf
"$archive_name" --null -T -

```

Рис. 16: Четвертый скрипт

Проверка работы четвертого скрипта. (рис. fig. 17).

```

[vlasovas@vbox 13lab]$ ./scr4.sh .
[vlasovas@vbox 13lab]$

```

Рис. 17: Проверка

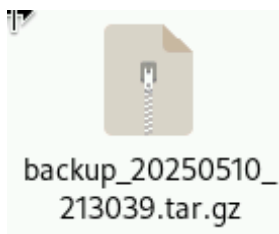


Рис. 18: Проверка

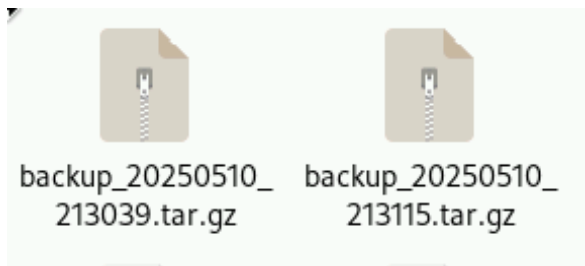


Рис. 19: Проверка

4 Контрольные вопросы

1. getoptс Обрабатывает аргументы командной строки в скриптах.
2. Метасимволы (*, ?, {}, []) автоматизируют генерацию имён файлов по шаблону.
3. Операторы управления if-else, case, for, while, until, break, continue.
4. Прерывание циклов break - прерывает цикл, continue - пропускает итерацию.
5. true/false true возвращает 0 (успех), false возвращает 1 (ошибка) для управления выполнением.
6. Проверка файла if test -f marks/i.s проверяет существование файла с именем из переменных в папке marks.
7. while vs until while выполняется пока условие истинно, until - пока ложно.

5 Выводы

Мы изучили основы программирования в оболочке ОС UNIX. Научились писать небольшие командные файлы с ветвлениями и циклами.

Список литературы