

Отчёт по лабораторной работе №13

Дисциплина: Операционные системы

Долгаев Евгений Сергеевич

Содержание

1	Цель работы	5
2	Задание	6
3	Выполнение лабораторной работы	8
3.1	Задача 1	9
3.2	Задача 2	10
3.3	Задача 3	12
3.4	Задача 4	13
4	Выводы	14

Список иллюстраций

3.1	Задача 1. Код	9
3.2	Задача 1. Результат выполнения	9
3.3	Задача 2. Код на С	10
3.4	Задача 2. Код на bash	11
3.5	Задача 2. Результат выполнения	11
3.6	Задача 3. Код	12
3.7	Задача 3. Результат выполнения	12
3.8	Задача 4. Код	13
3.9	Задача 4. Результат выполнения	13

Список таблиц

1 Цель работы

Изучить основы программирования в оболочке ОС UNIX. Научится писать более сложные командные файлы с использованием логических управляющих конструкций и циклов.

2 Задание

1. Используя команды `getopts` `grep`, написать командный файл, который анализирует командную строку с ключами:
 - `-i inputfile` — прочитать данные из указанного файла;
 - `-o outputfile` — вывести данные в указанный файл;
 - `-р шаблон` — указать шаблон для поиска;
 - `-C` — различать большие и малые буквы;
 - `-n` — выдавать номера строк, а затем ищет в указанном файле нужные строки, определяемые ключом `-р`.
2. Написать на языке Си программу, которая вводит число и определяет, является ли оно больше нуля, меньше нуля или равно нулю. Затем программа завершается с помощью функции `exit(n)`, передавая информацию в о коде завершения в оболочку. Командный файл должен вызывать эту программу и, проанализировав с помощью команды `$?`, выдать сообщение о том, какое число было введено.
3. Написать командный файл, создающий указанное число файлов, пронумерованных последовательно от 1 до n (например `1.tmp`, `2.tmp`, `3.tmp`, `4.tmp` и т.д.). Число файлов, которые необходимо создать, передаётся в аргументы командной строки. Этот же командный файл должен уметь удалять все созданные им файлы (если они существуют).
4. Написать командный файл, который с помощью команды `tag` запаковывает в архив все файлы в указанной директории. Модифицировать его так, чтобы

запаковывались только те файлы, которые были изменены менее недели тому назад (использовать команду find).

3 Выполнение лабораторной работы

Далее в отчете будут прикреплены скриншоты выполнения работы.

3.1 Задача 1

```
script1.sh [-M--] 2 L:[ 1+27 28/ 40] *(651 / 858b) 0010 0x00A
input_file=""
output_file=""
pattern=""
case_sensitive=0
line_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) line_numbers=1;;
        ?)
            echo "Использование: $0 -i inputfile -o outputfile -p шаблон [-C] [-n]"
            exit 1;;
    esac
done

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

if [[ ! -f "$input_file" ]] then
    echo "Файл '$input_file' не найден."
    exit 1
fi

grep_option=""

if [[ $case_sensitive -eq 0 ]] then
    grep_option="-i"
fi

if [[ $line_numbers -eq 1 ]] then
    grep_option+="-n"
fi

grep $grep_option -- "$pattern" "$input_file" > "$output_file"
```

Рис. 3.1: Задача 1. Код

```
[root@esdolgaev ~]# /home/esdolgaev/lab13/script1.sh
Использование: /home/esdolgaev/lab13/script1.sh -i inputfile -o outputfile -p шаблон [-C] [-n]
[root@esdolgaev ~]#
```

Рис. 3.2: Задача 1. Результат выполнения


```
#!/bin/bash

./check_number
status=$?

if [ status -eq 1 ]; then
    echo "Введенное число больше 0"
elif [ status -eq 0 ]; then
    echo "Введенное число равно 0"
elif [ status -eq 2 ]; then
    echo "Введенное число меньше 0"
else
    echo "Ошибка ввода"
fi
```

Рис. 3.4: Задача 2. Код на bash

```
[root@esdolgaev lab13]# ./script2.sh
Введите число: 0
Введенное число равно 0
[root@esdolgaev lab13]# ./script2.sh
Введите число: 3
Введенное число больше 0
[root@esdolgaev lab13]# ./script2.sh
Введите число: -3
Введенное число меньше 0
[root@esdolgaev lab13]#
```

Рис. 3.5: Задача 2. Результат выполнения

3.3 Задача 3

```
/home/esdolgaev/lab13/script3.sh
if [ "$1" == "delete" ]; then
    for file in [0-9]*.tmp; do
        if [ -e "$file" ]; then
            rm "$file"
        fi
    done
    exit 0
fi

N="$1"

for (( i=1; i<=N; i++ )); do
    filename="{i}.tmp"
    touch "$filename"
    echo "Файл '$filename' создан"
done
```

Рис. 3.6: Задача 3. Код

```
[root@esdolgaev lab13]# ./script3.sh
[root@esdolgaev lab13]# ./script3.sh 4
[root@esdolgaev lab13]# ls
1.tmp 2.tmp 3.tmp 4.tmp  check_number  check_number.c  script1.sh  script2.sh  script3.sh
[root@esdolgaev lab13]# ./script3.sh delete
[root@esdolgaev lab13]# ls
ld: отсутствуют входные файлы
[root@esdolgaev lab13]# ls
check_number  check_number.c  script1.sh  script2.sh  script3.sh
[root@esdolgaev lab13]#
```

Рис. 3.7: Задача 3. Результат выполнения

3.4 Задача 4

```
/home/esdolgaev/lab13/script4.sh
#!/bin/bash
if [ $# -ne 2 ]; then
    echo "Использование: $0 <директория> <имя архива>"
    exit 1
fi

directory="$1"
archive="$2"

find "$directory" -type f -mtime -7 -print0 | \
tar --null --files-from= -czf "$archive"

echo "Архив $archive создан в папке $directory"
```

Рис. 3.8: Задача 4. Код

```
[root@esdolgaev lab13]# ./script4.sh /home/esdolgaev/lab13 lab13
tar: : Функция stat завершилась с ошибкой: Нет такого файла или каталога
tar: Error is not recoverable: exiting now
Архив lab13 создан в папке /home/esdolgaev/lab13
[root@esdolgaev lab13]# ls
check_number  check_number.c  lab13  script1.sh  script2.sh  script3.sh  script4.sh
[root@esdolgaev lab13]#
```

Рис. 3.9: Задача 4. Результат выполнения

4 Выводы

Таким образом, я научился писать более сложные программы с использованием логических управляющих конструкций и циклов.