

# **Лабораторная работа №13**

**Программирование в командном процессоре ОС UNIX. Ветвления и  
циклы**

Мухин Тимофей Владимирович

# Содержание

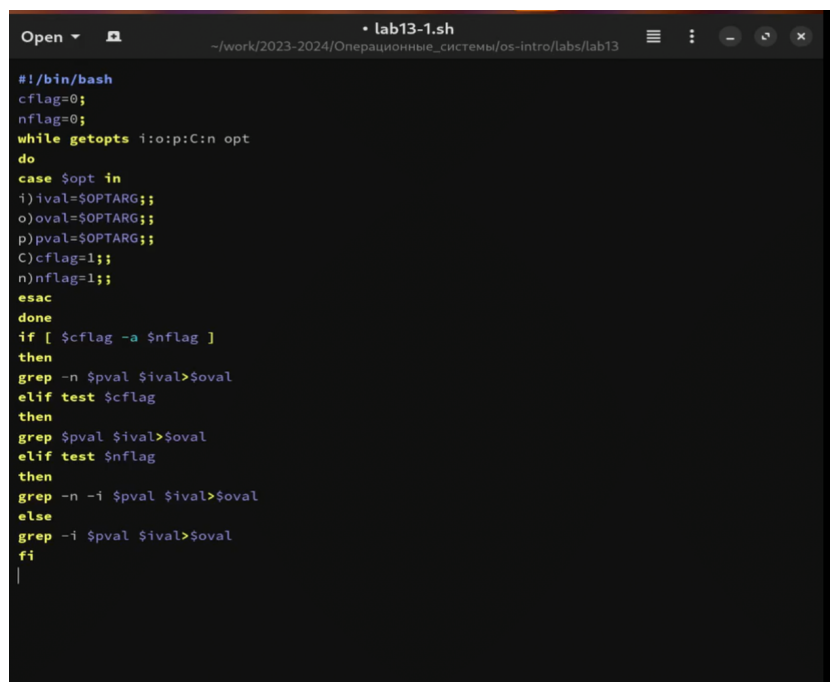
1	Цель работы	3
2	Выполнение лабораторной работы	4
3	Контрольные вопросы	8
4	Выводы	10

# 1 Цель работы

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

## 2 Выполнение лабораторной работы

1. Используя команды `getopts` `grep`, написали командный файл, который анализирует командную строку с ключами: `-iinputfile` — прочитать данные из указанного файла; `-ooutputfile` — вывести данные в указанный файл; `-ршаблон` — 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`.



```
Open ▾  • lab13-1.sh
~/work/2023-2024/Операционные_системы/os-intro/labs/lab13

#!/bin/bash
cflag=0;
nflag=0;
while getopts i:o:p:C:n opt
do
case $opt in
i)ival=$OPTARG;;
o)oval=$OPTARG;;
p)pval=$OPTARG;;
C)cflag=1;;
n)nflag=1;;
esac
done
if [ $cflag -a $nflag ]
then
grep -n $pval $ival>$oval
elif test $cflag
then
grep $pval $ival>$oval
elif test $nflag
then
grep -n -i $pval $ival>$oval
else
grep -i $pval $ival>$oval
fi
|
```

Рис. 2.1: Скрипт lab13-1

2. Запустим

```

txt
bash: ./lab13-1.sh: Permission denied
tvmukhin@fedora:~/work/2023-2024/Операционные_системы/os-intro/labs/lab13 $ chmod +x lab13-1.sh
tvmukhin@fedora:~/work/2023-2024/Операционные_системы/os-intro/labs/lab13 $ ./lab13-1.sh -test
txt
./lab13-1.sh: illegal option -- t
./lab13-1.sh: illegal option -- e
./lab13-1.sh: illegal option -- s
./lab13-1.sh: illegal option -- t
./lab13-1.sh: illegal option -- .
./lab13-1.sh: illegal option -- t
./lab13-1.sh: illegal option -- x
./lab13-1.sh: illegal option -- t
./lab13-1.sh: line 16: $oval: ambiguous redirect
tvmukhin@fedora:~/work/2023-2024/Операционные_системы/os-intro/labs/lab13 $ ./lab13-1.sh -i te
t.txt
./lab13-1.sh: line 16: $oval: ambiguous redirect
tvmukhin@fedora:~/work/2023-2024/Операционные_системы/os-intro/labs/lab13 $

```

Рис. 2.2: Запуск скрипта

3. 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. Команд- ный файл должен вызывать эту програм- му и, проанализировав с помощью команды `$?`, выдать сообщение о том, какое число было введено.

```

sey_chik@fedora:~/work/2023-2024/Операционные_системы/os-intro/labs/lab13
~/work/2023-2024/Операционные_системы/os-intro/labs/lab13
GNU nano 7.2 s2.c
#include <stdio.h>
#include <stdlib.h>
int main () {
    int o;
    scanf ("%d", &o);
    if (o < 0) {
        exit(1);
    }
    else if (o > 0) {
        exit(3);
    }
    exit (2);
}

```

Рис. 2.3: Программа на С

```
#!/bin/bash
gcc -c s2.c
gcc -o s2 s2.c
./s2
case $? in
    1) echo "отрицательное число";;
    2) echo "равно 0";;
    3) echo "отрицательное число";;
esac
```

Рис. 2.4: Скрипт lab13-2

4. Напишем 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)

```
s2.c | lab13-2.sh
#!/bin/bash
let i=$1+1
while (( i -=1 ))
do touch $i.tmp
done
let j=$2+1;
while (( j -= 1 ))
do rm $j.tmp
done
|
```

Рис. 2.5: Скрипт lab13-3

5. Запустим

```

lab13-1.sh lab13-2.sh lab13-3.sh presentation report s2.c
tvmukhin@fedora:~/work/2023-2024/Операционные_системы/os-intro/labs/lab13 $ chmod +x
lab13-3.sh
tvmukhin@fedora:~/work/2023-2024/Операционные_системы/os-intro/labs/lab13 $ ls
lab13-1.sh lab13-2.sh lab13-3.sh presentation report s2.c
tvmukhin@fedora:~/work/2023-2024/Операционные_системы/os-intro/labs/lab13 $ ./lab13-
3.sh 3
tvmukhin@fedora:~/work/2023-2024/Операционные_системы/os-intro/labs/lab13 $ ls
1.tmp 2.tmp 3.tmp lab13-1.sh lab13-2.sh lab13-3.sh presentation report s2.c
tvmukhin@fedora:~/work/2023-2024/Операционные_системы/os-intro/labs/lab13 $

```

Рис. 2.6: Запуск скрипта

6. Написать командный файл, который с помощью команды tar 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).

```

find $1 -mtime -7 | xargs tar -cf weekfiles.tar

```

Рис. 2.7: Скрипт lab13-4

7. Запустим

```

COURSE Makefile project-personal README.md
tvmukhin@fedora:~/work/2023-2024/Операционные_системы/os-intro $ cd labs/lab13
tvmukhin@fedora:~/work/2023-2024/Операционные_системы/os-intro/labs/lab13 $ ls
1.tmp 3.tmp lab13-2.sh lab13-4.sh report weekfiles.tar
2.tmp lab13-1.sh lab13-3.sh presentation s2.c
tvmukhin@fedora:~/work/2023-2024/Операционные_системы/os-intro/labs/lab13 $ rm wee

```

Рис. 2.8: Запуск скрипта

## 3 Контрольные вопросы

1. Команда `getopts` в `bash` используется для обработки параметров командной строки. Она облегчает разбор и извлечение значений из аргументов, переданных скрипту.
2. Метасимволы (wildcards) в `shell`-командах используются для соответствия шаблонам и генерации имен файлов. Например, символ `*` соответствует нулю или более символам, а `?` соответствует одному символу.
3. Операторы управления действиями в `bash` включают в себя условные операторы (`if-else`), циклы (`for`, `while`, `until`), операторы сравнения, операторы проверки файлов и директорий, операторы выхода из программы (`exit`), операторы переадресации вывода и ввода (`>`, `<`), операторы `pipe` (`|`), и многие другие.
4. Для прерывания цикла можно использовать оператор `break`. Он прерывает выполнение цикла и переводит выполнение программы за пределы цикла.
5. Команда `false` всегда возвращает ложное значение (код возврата 1), а `true` всегда возвращает истинное значение (код возврата 0). Они могут использоваться в скриптах для управления потоком выполнения.
6. Строка `if test -f man$s/$i.$s` в командном файле проверяет наличие файла с именем `man$s/$i.$s`. Здесь команда `test -f` проверяет, является ли файл существующим и обычным файлом.



7. Различия между конструкциями `while` и `until` в `bash` следующие:

- `while` выполняет блок кода, пока условие истинно (`true`).
- `until` выполняет блок кода, пока условие ложно (`false`), то есть продолжает выполнение, пока условие не станет истинным.

## 4 Выводы

В ходе выполнения работы я изучил основы программирования в ос Linux и научился писать более сложные командные файлы с ветвлениями и циклами.