

# Отчёт по лабораторной работе №11

Операционные системы

Балханова Алтана Юрьевна

# Содержание

Цель работы	5
Выполнение лабораторной работы	6
Контрольные вопросы	11
Выводы	12

## Список иллюстраций

0.1	Командный файл . . . . .	6
0.2	Проверка файла . . . . .	7
0.3	Программа . . . . .	8
0.4	Командный файл . . . . .	9
0.5	Проверка файла . . . . .	10

## Список таблиц

## Цель работы

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

## Выполнение лабораторной работы

1. Я написала, используя команду `getopts grep`, командный файл, который анализирует командную строку с ключами: `-iinputfile` — прочитать данные из указанного файла; `-ooutputfile` — вывести данные в указанный файл; `-rшаблон` — указать шаблон для поиска; `-C` — различать большие и малые буквы; `-n` — выдавать номера строк. а затем ищет в указанном файле нужные строки, определяемые ключом `-p` (рис. 0.1).

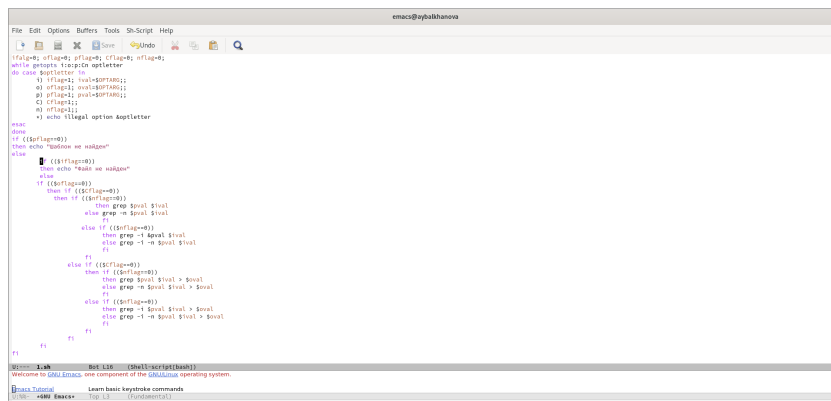
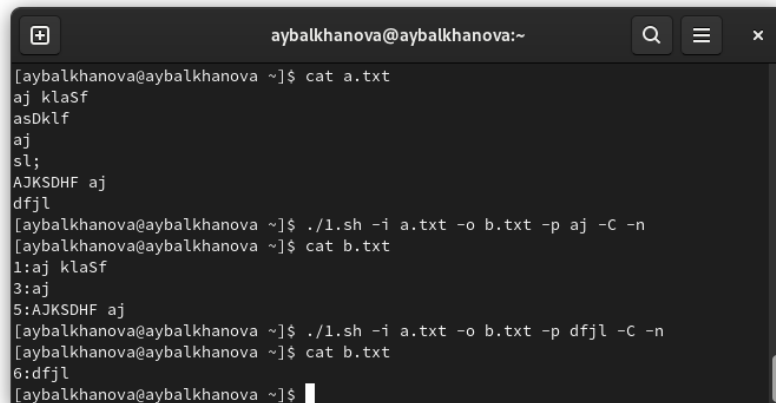


Рис. 0.1: Командный файл

2. Проверила его работу (рис. 0.2).

A terminal window titled 'aybalkhanova@aybalkhanova:~' with search, menu, and close buttons. It shows a sequence of commands and their outputs: 'cat a.txt' lists file contents; './l.sh -i a.txt -o b.txt -p aj -C -n' processes the file; 'cat b.txt' shows the output with line numbers. The process is repeated for 'dfjl'.

```
[aybalkhanova@aybalkhanova ~]$ cat a.txt
aj klaSf
asDklf
aj
sl;
AJKSDHF aj
dfjl
[aybalkhanova@aybalkhanova ~]$ ./l.sh -i a.txt -o b.txt -p aj -C -n
[aybalkhanova@aybalkhanova ~]$ cat b.txt
1:aj klaSf
3:aj
5:AJKSDHF aj
[aybalkhanova@aybalkhanova ~]$ ./l.sh -i a.txt -o b.txt -p dfjl -C -n
[aybalkhanova@aybalkhanova ~]$ cat b.txt
6:dfjl
[aybalkhanova@aybalkhanova ~]$
```

Рис. 0.2: Проверка файла

3. Написала на языке Си программу (рис. 0.3), 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 (рис. 0.4).

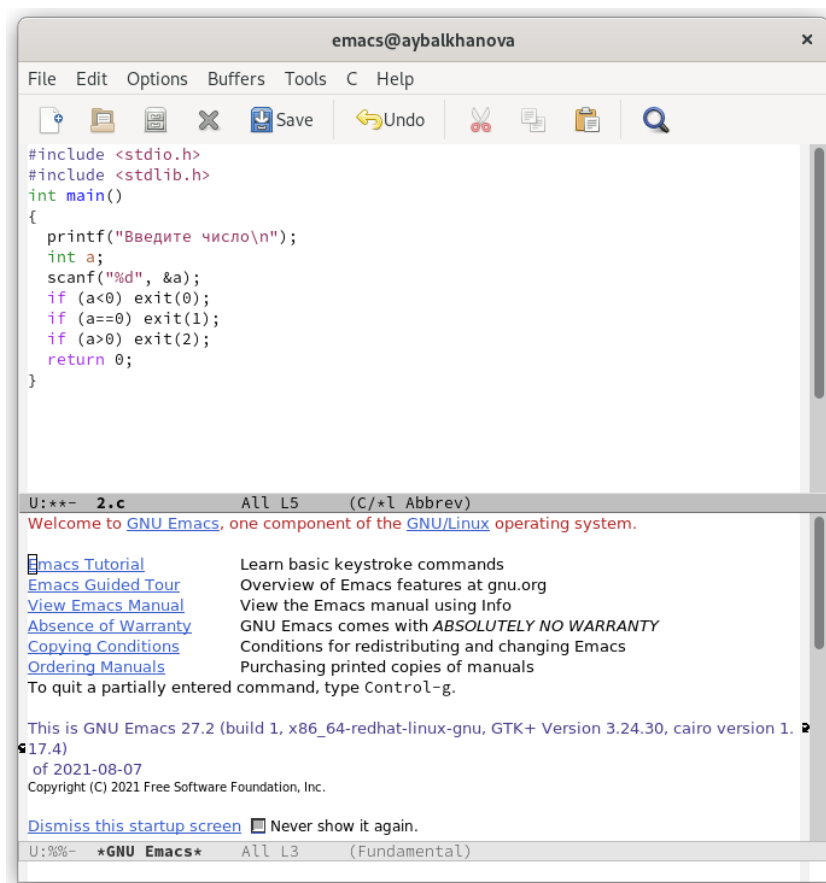


Рис. 0.3: Программа



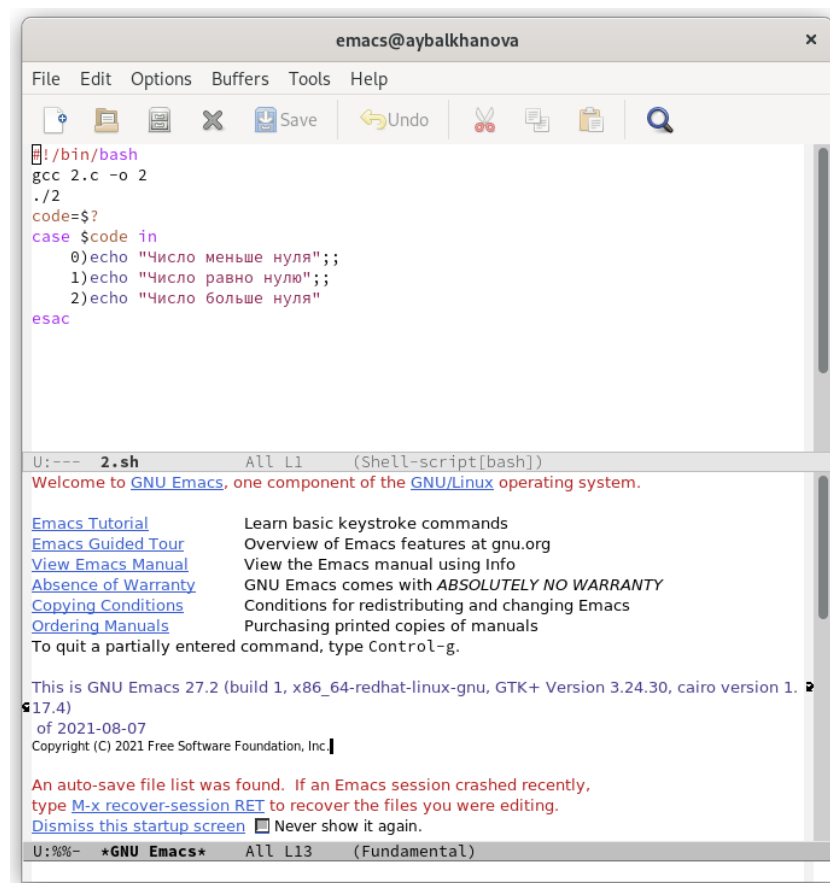
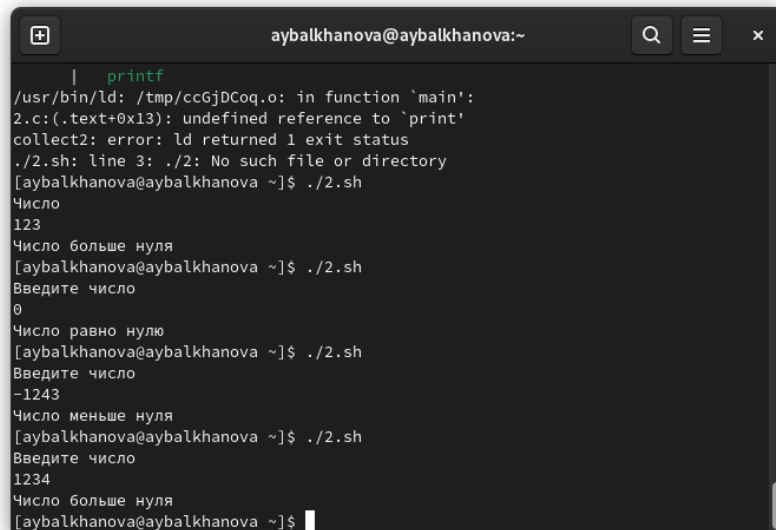


Рис. 0.4: Командный файл

4. Проверила его работу (рис. 0.5).



```
aybalkhanova@aybalkhanova:~  
| printf  
/usr/bin/ld: /tmp/ccGjDCoq.o: in function `main':  
2.c:(.text+0x13): undefined reference to `print'  
collect2: error: ld returned 1 exit status  
./2.sh: line 3: ./2: No such file or directory  
[aybalkhanova@aybalkhanova ~]$ ./2.sh  
Число  
123  
Число больше нуля  
[aybalkhanova@aybalkhanova ~]$ ./2.sh  
Введите число  
0  
Число равно нулю  
[aybalkhanova@aybalkhanova ~]$ ./2.sh  
Введите число  
-1243  
Число меньше нуля  
[aybalkhanova@aybalkhanova ~]$ ./2.sh  
Введите число  
1234  
Число больше нуля  
[aybalkhanova@aybalkhanova ~]$
```

Рис. 0.5: Проверка файла

# Контрольные вопросы

1. Команда `getopts` является встроенной командой командной оболочки `bash`, предназначенной для разбора параметров сценариев. Она обрабатывает исключительно однобуквенные параметры как с аргументами, так и без них и этого вполне достаточно для передачи сценариям любых входных данных.
2. Метасимволы - символы, имеющие специальное значение для интерпретатора. После всех подстановок в каждом слове команды ищутся символы `*`, `?`, и `[`. Если находится хотя бы один из них, то это слово рассматривается как шаблон имен файлов и заменяется именами файлов, удовлетворяющих данному шаблону (в алфавитном порядке). Если ни одно имя файла не удовлетворяет шаблону, то он остается неизменным.
3. Условные операторы `if`, оператор выбора `switch`, операторы цикла `for`, `while`, `do`.
4. `break`, `Continue`, `return`
5. Их задача - всегда возвращаться с заданным статусом выхода.
6. Она определяет, является ли этот файл обычным файлом и существует ли.
7. Цикл используется для выполнения заданный набор команд неизвестное число раз до тех пор , как данное условие принимает значение истинно. `until` используется для выполнения заданного набора команд до тех пор, пока данное условие оценивается как ложное.

## Выводы

Я изучила основы программирования в оболочке ОС UNIX. Научилась писать более сложные командные файлы с использованием логических управляющих конструкций и циклов.