

# Отчет по лабораторной работе №11

Дисциплина: Операционные системы

Кашкин Иван Евгеньевич

# Содержание

Цель работы	5
Задание	6
Теоретическое введение	8
Выполнение лабораторной работы	9
Выводы	17
Список литературы	18

## Список иллюстраций

0.1	Создание первой программы . . . . .	9
0.2	Программа №1 . . . . .	10
0.3	Продолжение . . . . .	10
0.4	Ввод программы . . . . .	11
0.5	Создание программ . . . . .	11
0.6	Си . . . . .	12
0.7	sh . . . . .	12
0.8	Ввод программы . . . . .	13
0.9	Создание программы . . . . .	13
0.10	Программа №3 . . . . .	14
0.11	Ввод программы . . . . .	14
0.12	Создание программы . . . . .	15
0.13	Программа №4 . . . . .	15
0.14	Запуск программы . . . . .	16

## Список таблиц

## Цель работы

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

# 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;
  - `-r`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 `-r`.
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 `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`).

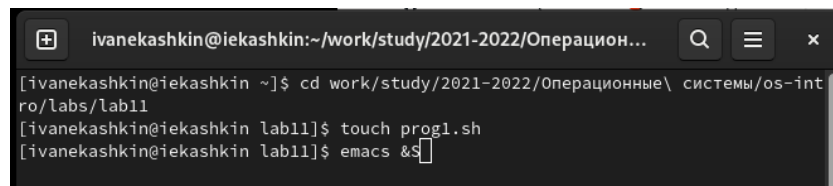
# Теоретическое введение

- Командный процессор (командная оболочка, интерпретатор команд shell) — это программа, позволяющая пользователю взаимодействовать с операционной системой компьютера. В операционных системах типа UNIX/Linux наиболее часто используются следующие реализации командных оболочек:
- оболочка Борна (Bourne shell или sh) — стандартная командная оболочка UNIX/Linux, содержащая базовый, но при этом полный набор функций;
- С-оболочка (или csh) — надстройка на оболочкой Борна, использующая С-подобный синтаксис команд с возможностью сохранения истории выполнения команд;
- оболочка Корна (или ksh) — напоминает оболочку С, но операторы управления программой совместимы с операторами оболочки Борна;
- BASH — сокращение от Bourne Again Shell (опять оболочка Борна), в основе своей совмещает свойства оболочек С и Корна (разработка компании Free Software Foundation). POSIX (Portable Operating System Interface for Computer Environments) — набор стандартов описания интерфейсов взаимодействия операционной системы и прикладных программ. Стандарты POSIX разработаны комитетом IEEE (Institute of Electrical and Electronics Engineers) для обеспечения совместимости различных UNIX/Linux-подобных операционных систем и переносимости прикладных программ на уровне исходного кода. POSIX-совместимые оболочки разработаны на базе оболочки Корна. Рассмотрим основные элементы программирования в оболочке bash. В других оболочках большинство команд будет совпадать с описанными ниже.



# Выполнение лабораторной работы

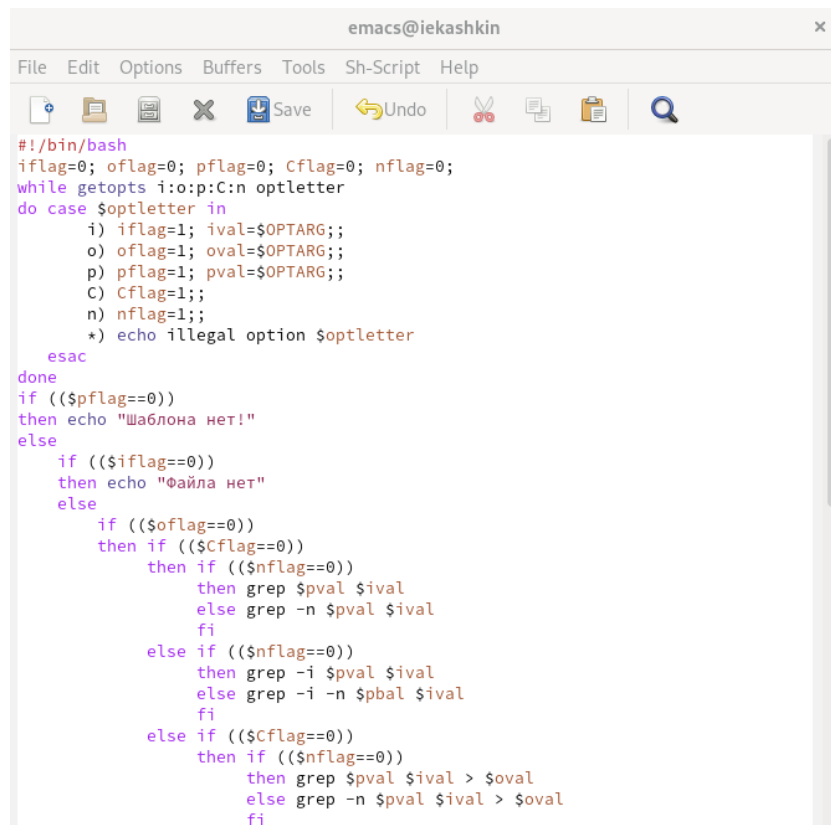
1. Зашёл в каталог лабораторной работы, создал файл 1 программы и открыл emacs (рис. [-@fig:001])



```
ivanekashkin@iekashkin:~/work/study/2021-2022/Операцион...
[ivanekashkin@iekashkin ~]$ cd work/study/2021-2022/Операционные\ системы/os-int
ro/labs/lab11
[ivanekashkin@iekashkin lab11]$ touch prog1.sh
[ivanekashkin@iekashkin lab11]$ emacs &[
```

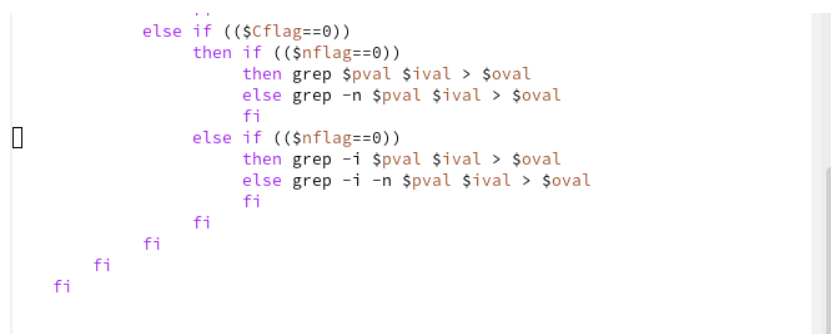
Рис. 0.1: Создание первой программы

- Используя команды getoptс grep, написал командный файл, который анализирует командную строку с ключами (рис. [-@fig:002]) (рис. [-@fig:003])



```
#!/bin/bash
iflag=0; oflag=0; pflag=0; Cflag=0; nflag=0;
while getopts i:o:p:C:n optletter
do case $optletter in
    i) iflag=1; ival=$OPTARG;;
    o) oflag=1; oval=$OPTARG;;
    p) pflag=1; pval=$OPTARG;;
    C) Cflag=1;;
    n) nflag=1;;
    *) echo illegal option $optletter
    esac
done
if (($pflag==0))
then echo "Шаблона нет!"
else
    if (($iflag==0))
    then echo "Файла нет"
    else
        if (($oflag==0))
        then if (($Cflag==0))
            then if (($nflag==0))
                then grep $pval $ival
                else grep -n $pval $ival
                fi
            else if (($nflag==0))
                then grep -i $pval $ival
                else grep -i -n $pval $ival
                fi
            else if (($Cflag==0))
                then if (($nflag==0))
                    then grep $pval $ival > $oval
                    else grep -n $pval $ival > $oval
                    fi
                fi
            fi
        fi
    fi
fi
```

Рис. 0.2: Программа №1



```
..
else if (($Cflag==0))
then if (($nflag==0))
    then grep $pval $ival > $oval
    else grep -n $pval $ival > $oval
    fi
else if (($nflag==0))
    then grep -i $pval $ival > $oval
    else grep -i -n $pval $ival > $oval
    fi
fi
fi
fi
fi
```

Рис. 0.3: Продолжение

- Далее сохранил файл и запустил его из командной строки, предварительно создав файл 1.txt и 2.txt. После прописываем команду “./prog1.sh ...” и добавляем маркеры для проверки файла. (рис. [-@fig:004])

```

[ivanekashkin@iekashkin lab11]$ chmod +x *.sh
[1]+  Завершён      emacs
[ivanekashkin@iekashkin lab11]$ cat 1.txt
"1984" is John Orwell's book
"Dead Souls" is Gogol's book
Show
"War and Peace" is Leo Tolstoy's book
HELL
"The Cherry Orchard" is Chekhov's book
[ivanekashkin@iekashkin lab11]$ ./prog1.sh -i 1.txt -o 2.txt -p book -C -n
[ivanekashkin@iekashkin lab11]$ cat 2.txt
"1984" is John Orwell's book
"Dead Souls" is Gogol's book
"War and Peace" is Leo Tolstoy's book
"The Cherry Orchard" is Chekhov's book
[ivanekashkin@iekashkin lab11]$ ./prog1.sh -i 1.txt -o 2.txt -p book -n
[ivanekashkin@iekashkin lab11]$ cat 2.txt
1:"1984" is John Orwell's book
2:"Dead Souls" is Gogol's book
4:"War and Peace" is Leo Tolstoy's book
6:"The Cherry Orchard" is Chekhov's book
[ivanekashkin@iekashkin lab11]$ ./prog1.sh -i 1.txt -C -n
Шаблона нет!
[ivanekashkin@iekashkin lab11]$ ./prog1.sh -o 2.txt -p book -n
Файла нет

```

Рис. 0.4: Ввод программы

2. Создал файлы для второй программы prog2.c и prog2.sh (рис. [-@fig:005])

```

[ivanekashkin@iekashkin lab11]$ touch prog2.c prog2.sh
[ivanekashkin@iekashkin lab11]$ emacs &s

```

Рис. 0.5: Создание программ

- Написал на языке Си программу, которая вводит число и определяет, является ли оно больше нуля, меньше нуля или равно нулю.
- Прописываю текст программы для файла Си (рис. [-@fig:006])
- После написал программу для файла .sh (рис. [-@fig:007])

```

#include <stdio.h>
#include <stdlib.h>

int main(){
    printf("Напишите число: ");
    int a;
    scanf("%d",&a);
    if (a<0) exit(0);
    if (a>0) exit(1);
    if (a==0) exit(2);
    return 0;
}

```

Рис. 0.6: Си

```

#!/bin/bash

gcc prog2.c -o prog2
./prog2
code=$?
case $code in
    0)echo "Меньше 0";;
    1)echo "Больше 0";;
    2)echo "Равно 0";;
esac

```

Рис. 0.7: sh

- Делае я я проверил работы программы, запустив ее из терминала “./prog2.sh” (рис. [-@fig:008])

```

[ivanekashkin@iekashkin lab11]$ chmod +x *.sh
[2]+  Завершён      emacs
[ivanekashkin@iekashkin lab11]$ ./prog2.sh
Напишите число: 4
Больше 0
[ivanekashkin@iekashkin lab11]$ ./prog2.sh
Напишите число: -1
Меньше 0
[ivanekashkin@iekashkin lab11]$ ./prog2.sh
Напишите число: 0
Равно 0

```

Рис. 0.8: Ввод программы

3. Создал файлы для третьей программы prog3.sh (рис. [-@fig:009])

```

[ivanekashkin@iekashkin lab11]$ touch prog3.sh
[ivanekashkin@iekashkin lab11]$ emacs &

```

Рис. 0.9: Создание программы

- Написал командный файл, создающий указанное число файлов, пронумерованных последовательно от 1 до N. Этот командный файл также удаляет все созданные им файлы. (рис. [-@fig:0010])

```
#!/bin/bash

opt=$1;
form=$2;
num=$3;
function Files(){
    for ((i=1; i<=$num;i++)) do
        file=$(echo $form | tr '#' "$i")
        if [ $opt == "-r" ]
        then
            rm -f $file
        elif [ $opt == "-c" ]
        then
            touch $file
        fi
    done
}
Files[]
```

Рис. 0.10: Программа №3

- Я сохранил командный файл, проверил содержимое дериктории командой “ls”, после создал файлы командой “./prog3.sh -c w#.txt 4” и удалил командой “./prog3.sh -r w#.txt 4” (рис. [-@fig:0011])

```
[ivaneekashkin@iekashkin lab11]$ chmod +x *.sh
[ivaneekashkin@iekashkin lab11]$ ls
1.txt  presentation  prog1.sh~  prog2.c  prog2.sh  prog3.sh  report
2.txt  prog1.sh      prog2      prog2.c~  prog2.sh~  prog3.sh~
[ivaneekashkin@iekashkin lab11]$ ./prog3.sh -c w#.txt 4
[ivaneekashkin@iekashkin lab11]$ ls
1.txt  prog1.sh  prog2.c  prog2.sh~  report  w3.txt
2.txt  prog1.sh~  prog2.c~  prog3.sh  w1.txt  w4.txt
presentation  prog2  prog2.sh  prog3.sh~  w2.txt
[ivaneekashkin@iekashkin lab11]$ ./prog3.sh -r w#.txt 4
[ivaneekashkin@iekashkin lab11]$ ls
1.txt  presentation  prog1.sh~  prog2.c  prog2.sh  prog3.sh  report
2.txt  prog1.sh      prog2      prog2.c~  prog2.sh~  prog3.sh~
```

Рис. 0.11: Ввод программы

4. Создал файлы для xtndthnjq программы prog4.sh (рис. [-@fig:0012])

```
[ivanekashkin@iekashkin lab11]$ touch prog4.sh
[ivanekashkin@iekashkin lab11]$ emacs &
```

Рис. 0.12: Создание программы

- Написал командный файл, который с помощью команды tar запаковывает в архив все файлы в указанной директории. (рис. [-@fig:0013])

```
#!/bin/bash

file=$(find ./ -maxdepth 1 -mtime -7)
listing=""
for file in "$files"; do
    file=$(echo "$file" | cut -c 3-)
    listing="$listing $file"
done
dir=$(basename $(pwd))
tar -cvf $dir.tar $listing
```

Рис. 0.13: Программа №4

- Далее проверил дерикторию командой “ls” и использовал командный файл с помощью двух команд “./prog4.sh” и “./tar -tf lab11.tar”(рис. [-@fig:0014])

```

[ivanekashkin@iekashkin lab11]$ chmod +x *.sh
[1]+  Завершён      emacs
[ivanekashkin@iekashkin lab11]$ ls -l
итого 64
-rw-rw-r--. 1 ivanekashkin ivanekashkin 145 мая 25 13:22 1.txt
-rw-rw-r--. 1 ivanekashkin ivanekashkin 143 мая 25 13:45 2.txt
drwxrwxr-x. 1 ivanekashkin ivanekashkin  0 апр 22 12:06 presentation
-rwxrwxr-x. 1 ivanekashkin ivanekashkin 976 мая 25 13:44 prog1.sh
-rwxrwxr-x. 1 ivanekashkin ivanekashkin 967 мая 25 12:50 prog1.sh~
-rwxrwxr-x. 1 ivanekashkin ivanekashkin 25096 мая 25 13:58 prog2
-rw-rw-r--. 1 ivanekashkin ivanekashkin 200 мая 25 13:54 prog2.c
-rw-rw-r--. 1 ivanekashkin ivanekashkin  0 мая 25 13:49 prog2.c~
-rwxrwxr-x. 1 ivanekashkin ivanekashkin 157 мая 25 13:57 prog2.sh
-rw-rw-r--. 1 ivanekashkin ivanekashkin  0 мая 25 13:49 prog2.sh~
-rwxrwxr-x. 1 ivanekashkin ivanekashkin 233 мая 25 14:06 prog3.sh
-rw-rw-r--. 1 ivanekashkin ivanekashkin  0 мая 25 14:00 prog3.sh~
-rwxrwxr-x. 1 ivanekashkin ivanekashkin 210 мая 25 14:33 prog4.sh
-rwxrwxr-x. 1 ivanekashkin ivanekashkin 209 мая 25 14:30 prog4.sh~
drwxrwxr-x. 1 ivanekashkin ivanekashkin  0 апр 23 21:50 report
[ivanekashkin@iekashkin lab11]$ sudo ~/work/study/2021-2022/Операционные\системы
/os-intro/labs/lab11/prog4.sh
sudo: /home/ivanekashkin/work/study/2021-2022/Операционныесистемы/os-intro/labs/
lab11/prog4.sh: command not found
[ivanekashkin@iekashkin lab11]$ ./prog4.sh
2.txt
1.txt
prog1.sh~
prog1.sh
prog2.c~
prog2.c
prog2.sh~
prog2.sh
prog2
prog3.sh~
prog3.sh
prog4.sh~
prog4.sh
[ivanekashkin@iekashkin lab11]$ tar -tf lab11.tar
tar: lab11.tar: Функция open завершилась с ошибкой: Нет такого файла или каталог
a
tar: Error is not recoverable: exiting now
[ivanekashkin@iekashkin lab11]$ tar -tf Операционные.tar
2.txt
1.txt
prog1.sh~
prog1.sh

```

Рис. 0.14: Запуск программы



## Выводы

- Изучил основы программирования в оболочке ОС UNIX. Научился писать более сложные командные файлы с использованием логических управляющих конструкций и циклов.

# Список литературы

::: {#Лабораторная работа No 10. Программирование в командном процессоре ОС UNIX. Командные файлы} :::