

Отчёта по лабораторной работе №10:

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

Кононов Алексей Сергеевич

Содержание

1	Цель работы	4
2	Задание	5
3	Выполнение лабораторной работы	7
4	Контрольные вопросы	12
5	Выводы	14
	Список литературы	15

Список иллюстраций

3.1	Задание 1	7
3.2	Выполнение и результат 1	8
3.3	Задание 2-1	9
3.4	Задание 2-2	9
3.5	Выполнение и результат 2	9
3.6	Задание 3	10
3.7	Выполнение и результат 3	10
3.8	Задание 4	11
3.9	Выполнение и результат 4	11

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;
- `-p` - 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 `-p`.

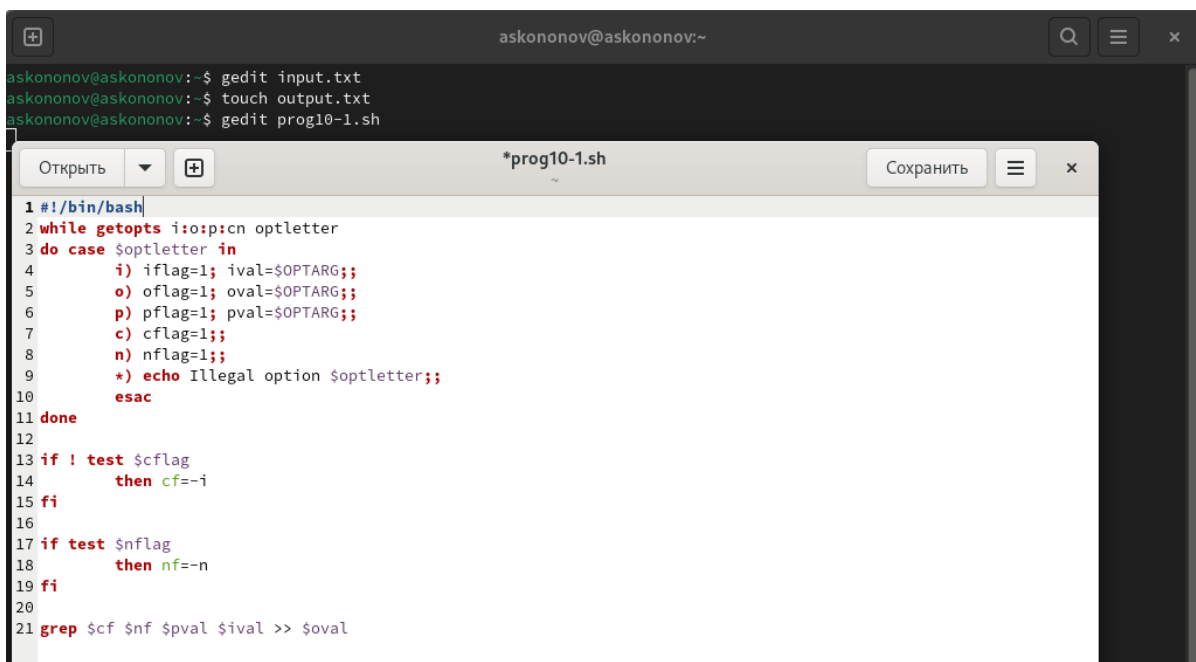
2. Написать на языке Си программу, которая вводит число и определяет, является ли оно больше нуля, меньше нуля или равно нулю. Затем программа завершается с помощью функции `exit(n)`, передавая информацию о коде завершения в оболочку. Командный файл должен вызывать эту программу и, проанализировав с 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`).

3 Выполнение лабораторной работы

1. Напишем командный файл `getopts` `grep`, который анализирует командную строку с ключами (`-i`, `-o`, `-p`, `-c`, `-n`), а затем ищет в указанном файле нужные строки, определяемые ключом `-p` (рис. 3.1):



```
askononov@askononov:~$ gedit input.txt
askononov@askononov:~$ touch output.txt
askononov@askononov:~$ gedit prog10-1.sh

Открыть *prog10-1.sh Сохранить x

1 #!/bin/bash
2 while getopts i:o:p:cn optletter
3 do case $optletter in
4     i) iflag=1; ival=$OPTARG;;
5     o) oflag=1; oval=$OPTARG;;
6     p) pflag=1; pval=$OPTARG;;
7     c) cflag=1;;
8     n) nflag=1;;
9     *) echo Illegal option $optletter;;
10    esac
11 done
12
13 if ! test $cflag
14 then cf=-i
15 fi
16
17 if test $nflag
18 then nf=-n
19 fi
20
21 grep $cf $nf $pval $ival >> $oval
```

Рис. 3.1: Задание 1

Делаем текстовый файл “input.txt” для ввода и “output.txt” для вывода. Делаем файл “prog10-1.sh” исполняемым и выводим результат (рис. 3.2).

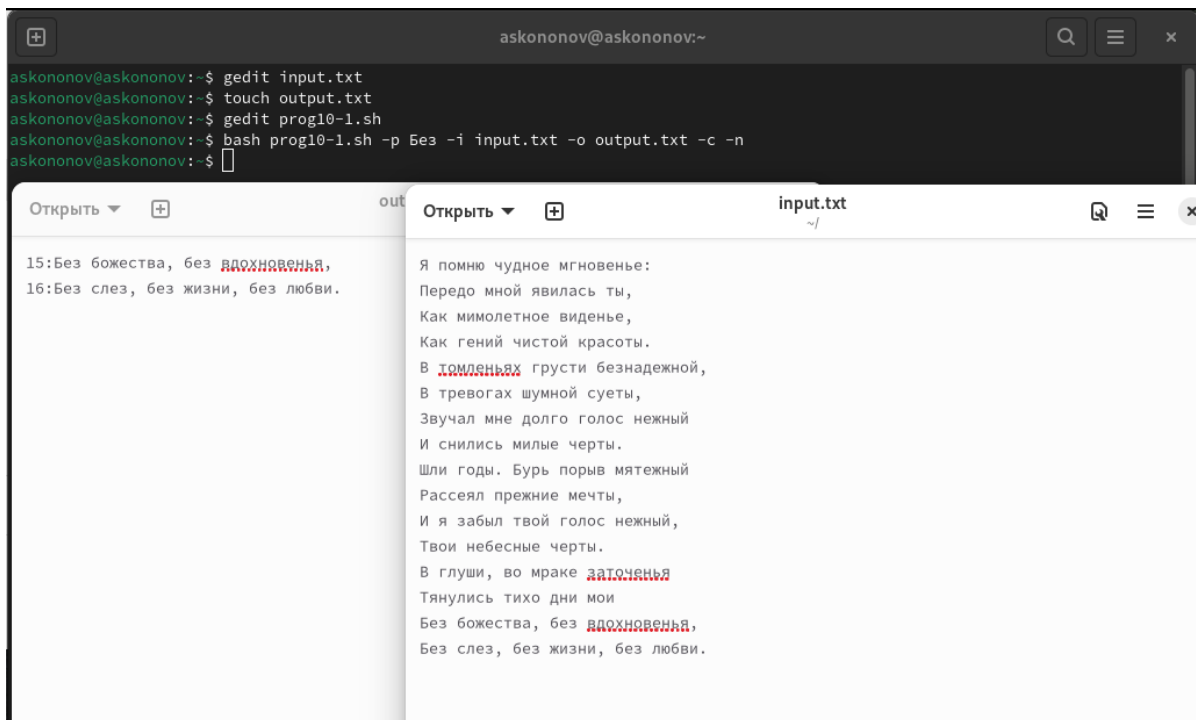


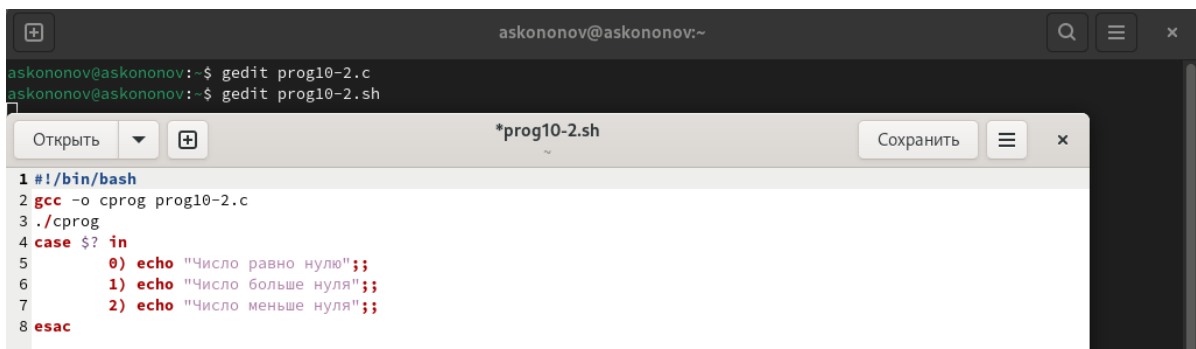
Рис. 3.2: Выполнение и результат 1

2. Напишем 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 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. 3.3), (рис. 3.4):



```
askononov@askononov:~$ gedit prog10-2.c
*prog10-2.c
1 #include <stdlib.h>
2 #include <stdio.h>
3
4 int main () {
5     int n;
6     printf("Велите число: ");
7     scanf ("%d", &n);
8     if (n > 0){
9         exit(1);
10    }
11    else if (n == 0) {
12        exit(0);
13    }
14    else {
15        exit(2);
16    }
17 }
```

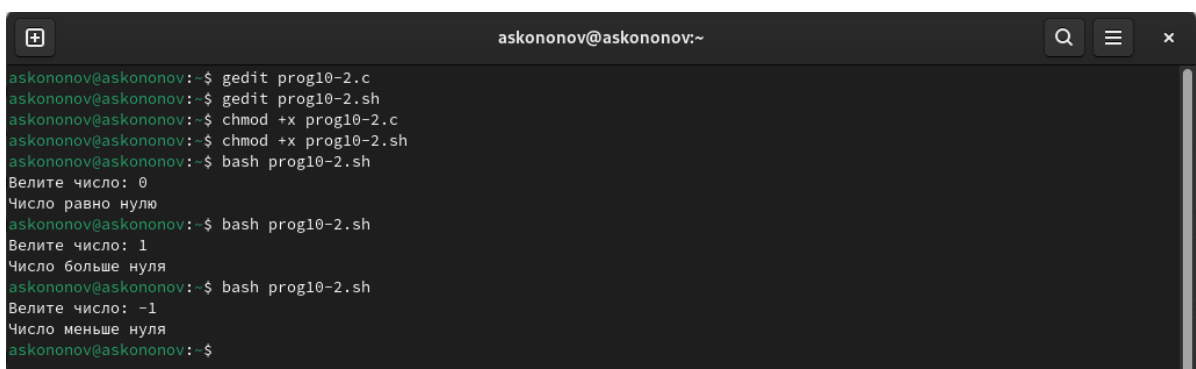
Рис. 3.3: Задание 2-1



```
askononov@askononov:~$ gedit prog10-2.c
askononov@askononov:~$ gedit prog10-2.sh
*prog10-2.sh
1 #!/bin/bash
2 gcc -o cprog prog10-2.c
3 ./cprog
4 case $? in
5     0) echo "Число равно нулю";;
6     1) echo "Число больше нуля";;
7     2) echo "Число меньше нуля";;
8 esac
```

Рис. 3.4: Задание 2-2

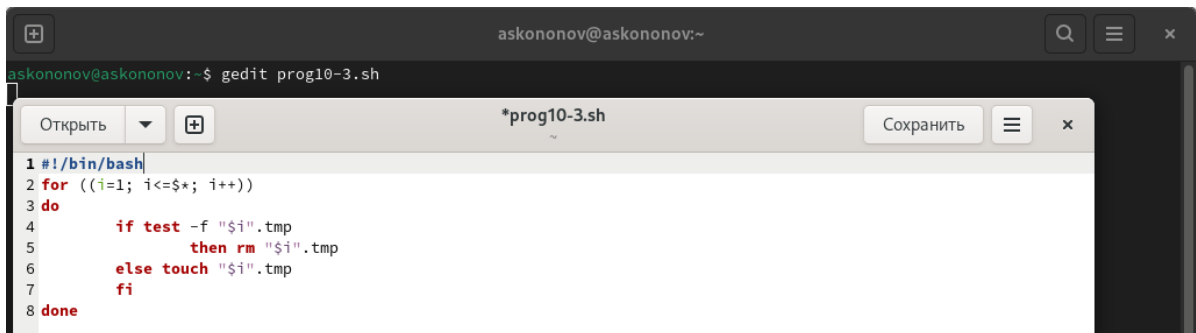
Делаем файлы исполняемыми и выводим результат (рис. 3.5).



```
askononov@askononov:~$ gedit prog10-2.c
askononov@askononov:~$ gedit prog10-2.sh
askononov@askononov:~$ chmod +x prog10-2.c
askononov@askononov:~$ chmod +x prog10-2.sh
askononov@askononov:~$ bash prog10-2.sh
Велите число: 0
Число равно нулю
askononov@askononov:~$ bash prog10-2.sh
Велите число: 1
Число больше нуля
askononov@askononov:~$ bash prog10-2.sh
Велите число: -1
Число меньше нуля
askononov@askononov:~$
```

Рис. 3.5: Выполнение и результат 2

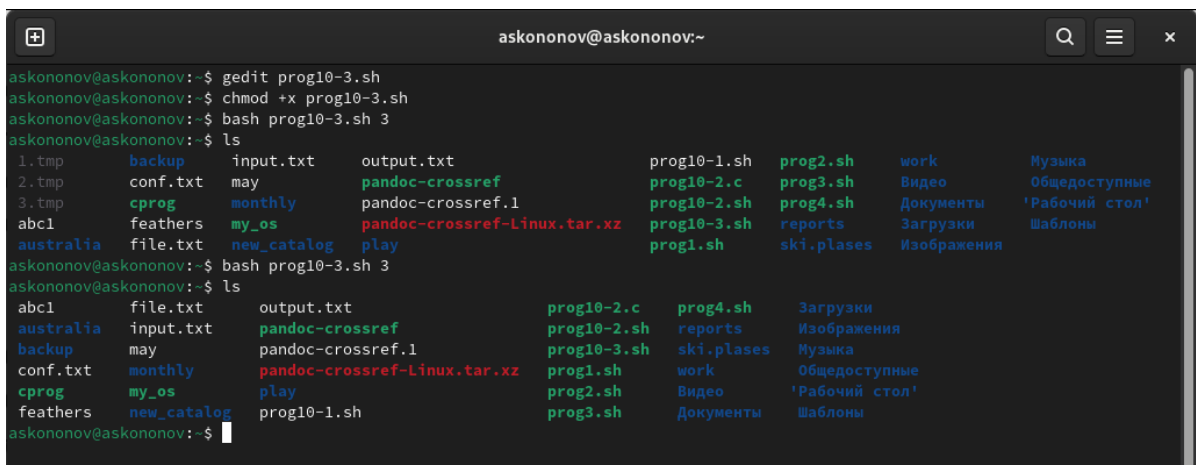
3. Напишем командный файл, создающий указанное число файлов, пронумерованных последовательно от 1 до N. Число файлов, которые необходимо создать, передаётся в аргументы командной строки. Этот же командный файл должен уметь удалять все созданные им файлы (рис. 3.6):



```
askononov@askononov:~$ gedit prog10-3.sh
1 #!/bin/bash
2 for ((i=1; i<=$*; i++))
3 do
4     if test -f "$i".tmp
5     then rm "$i".tmp
6     else touch "$i".tmp
7     fi
8 done
askononov@askononov:~$ bash prog10-3.sh 3
```

Рис. 3.6: Задание 3

Делаем файлы исполняемыми и выводим результат (рис. 3.7).



```
askononov@askononov:~$ gedit prog10-3.sh
askononov@askononov:~$ chmod +x prog10-3.sh
askononov@askononov:~$ bash prog10-3.sh 3
askononov@askononov:~$ ls
1.tmp      backup      input.txt   output.txt   prog10-1.sh  prog2.sh    work        Музыка
2.tmp      conf.txt    may         pandoc-crossref  prog10-2.c   prog3.sh    Видео      Общедоступные
3.tmp      cprog      monthly    pandoc-crossref.1  prog10-2.sh  prog4.sh    Документы  'Рабочий стол'
abcl      feathers   my_os      pandoc-crossref-Linux.tar.xz  prog10-3.sh  reports     Загрузки   Шаблоны
australia  file.txt   new_catalog play         prog1.sh     ski.places  Изображения

askononov@askononov:~$ bash prog10-3.sh 3
askononov@askononov:~$ ls
abcl      file.txt    output.txt   prog10-2.c   prog4.sh    Загрузки
australia input.txt   pandoc-crossref  prog10-2.sh  reports     Изображения
backup    may         pandoc-crossref.1  prog10-3.sh  ski.places  Музыка
conf.txt  monthly    pandoc-crossref-Linux.tar.xz  prog1.sh     work        Общедоступные
cprog     my_os      play         prog2.sh     Видео      'Рабочий стол'
feathers   new_catalog prog10-1.sh   prog3.sh     Документы  Шаблоны
askononov@askononov:~$
```

Рис. 3.7: Выполнение и результат 3

4. Напишем 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, используя команду find (рис. 3.8):

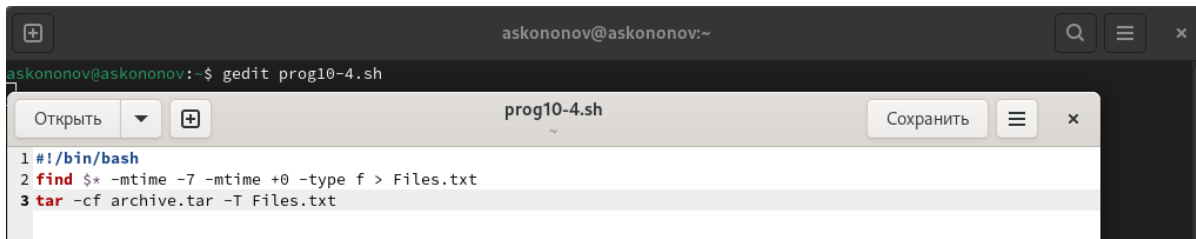


Рис. 3.8: Задание 4

Делаем файлы исполняемыми и выводим результат (рис. 3.9).

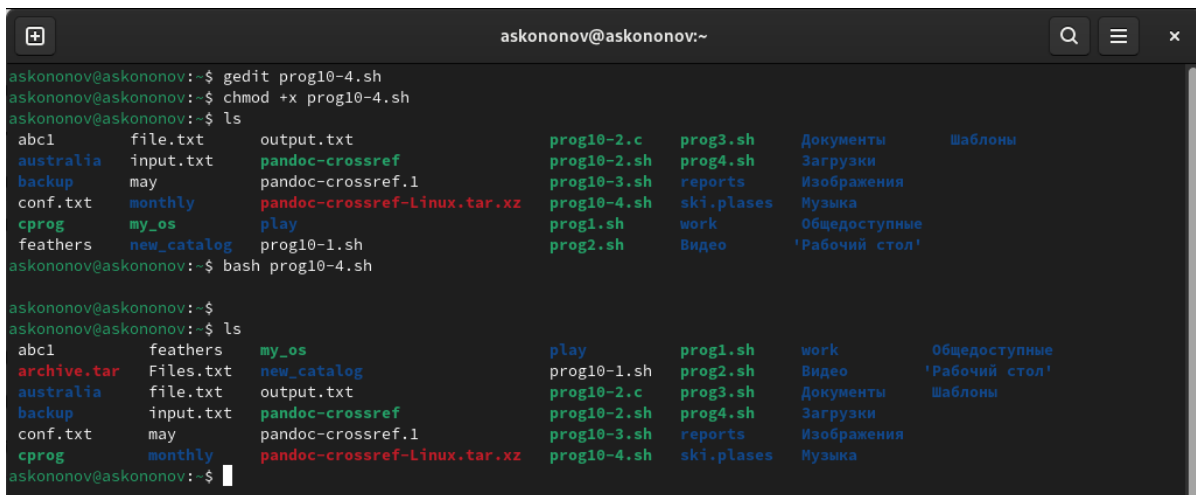


Рис. 3.9: Выполнение и результат 4

4 Контрольные вопросы

1. Каково предназначение команды `getopts`? С помощью `getopts` можно достаточно легко произвести разбор флагов переданных скрипту.
2. Какое отношение метасимволы имеют к генерации имён файлов? Метасимволы играют важную роль в генерации имен файлов, особенно в командных оболочках *Unix* и *Linux*. Эти символы представляют собой специальные символы, которые интерпретируются оболочкой для выполнения шаблонного поиска и подстановки. Например:
 - `*` - соответствует любому количеству символов в имени файла.
 - `?` - соответствует ровно одному символу.
 - `[...]` - соответствуют любому одному символу из перечисленных в скобках.
3. Какие операторы управления действиями вы знаете?
 - Условные операторы:
 - `if`, `elif` и `else` - позволяют выполнять команды, основываясь на условиях.
 - `case` - выбор действий в зависимости от значения переменной.
 - Циклы:
 - `for` - выполняет команды для списка значений.
 - `while` - цикл который выполняется пока условие истинно.
 - `until` - цикл который выполняется пока условие ложно.

4. Какие операторы используются для прерывания цикла? В Bash для прерывания циклов используются следующие операторы:

- `break` - прерывает выполнение текущего цикла и передает управление на команду, следующую за циклом. Вы можете указать `break n`, где `n` - это количество уровней цикла, которые нужно прервать.
- `continue` - пропускает оставшуюся часть тела текущего цикла и переходит к следующей итерации. Аналогично `break`, можно использовать `continue n` для пропуска итераций во вложенных циклах.

5. Для чего нужны команды `false` и `true`? Команды `true` и `false` в Bash являются простыми утилитами, которые возвращают статус выхода. Они используются в скриптах и условных операторах для управления логикой выполнения.

- `true` всегда возвращает статус выхода 0, что означает успех. Эта команда может быть использована в местах, где требуется гарантированно успешный результат, например, в бесконечных циклах (`while true; do ... done`) или как заглушка для функции, которая еще не реализована.
- `false` всегда возвращает статус выхода 1, что означает неудачу. Эта команда может быть использована для преднамеренного вызова ошибки или как условие, которое никогда не будет выполнено.

6. Что означает строка `if test -f man$s/$i.$s`, встреченная в командном файле? В этой строке проверяется существование файла `man$s/$i.$s`, где `$s` и `$i` переменные подставляющиеся в имя файла.

7. Объясните различия между конструкциями `while` и `until`.

- `while` — цикл который выполняется пока условие истинно.
- `until` — цикл который выполняется пока условие ложно.

5 Выводы

В этой работе мы поближе познакомились с циклами в языке `bash`. Научились писать более сложные командные файлы используя логические управляющие конструкции.

Список литературы