**Федеральное государственное автономное**

**Образовательное учреждение высшего образования**

**Российский Университет Дружбы Народов**

Математический университет имени Никольского

Факультет Физико-математических и Естественных наук

Кафедра Прикладной математики и информатики

Отчет по лабораторной работе № 11

“ Программирование в командном

процессоре ОС UNIX. Ветвления и циклы”

Выполнил:

Студент группы НПМбв-01-10

Адхамова Луиза Шухратовна

Москва

2024 год

# Цель работы:

Изучить основы программирования в оболочке ОС UNIX. Научится писать более

сложные командные файлы с использованием логических управляющих конструкций

и циклов.

# Выполнение:

1. Используя команды getopts grep, напишем командный файл, который анализирует

командную строку с ключами:

– -iinputfile — прочитать данные из указанного файла;

– -ooutputfile — вывести данные в указанный файл;

– -pшаблон — 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 1.1, 1.2, 1.3, 1.4)

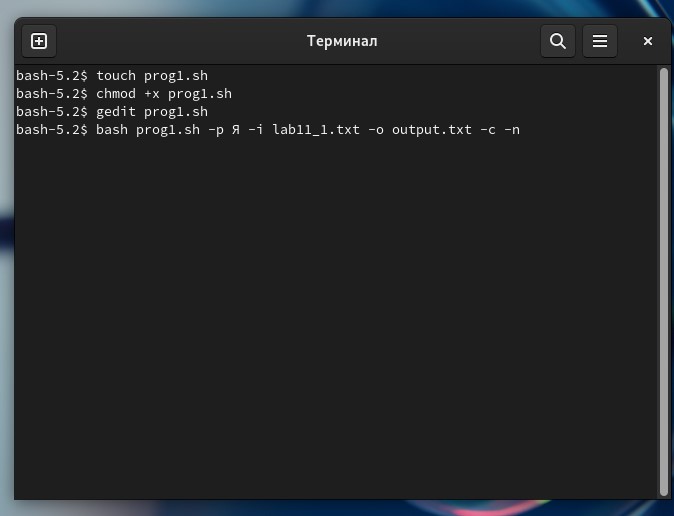


Рисунок 1.1. Файл.



Рисунок 1.2. Программа.

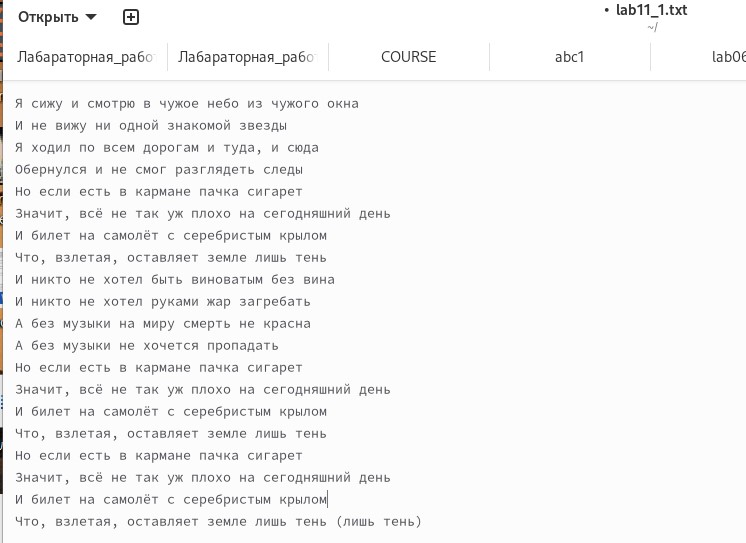


Рисунок 1.3. Файл, в котором выполнялась команда.

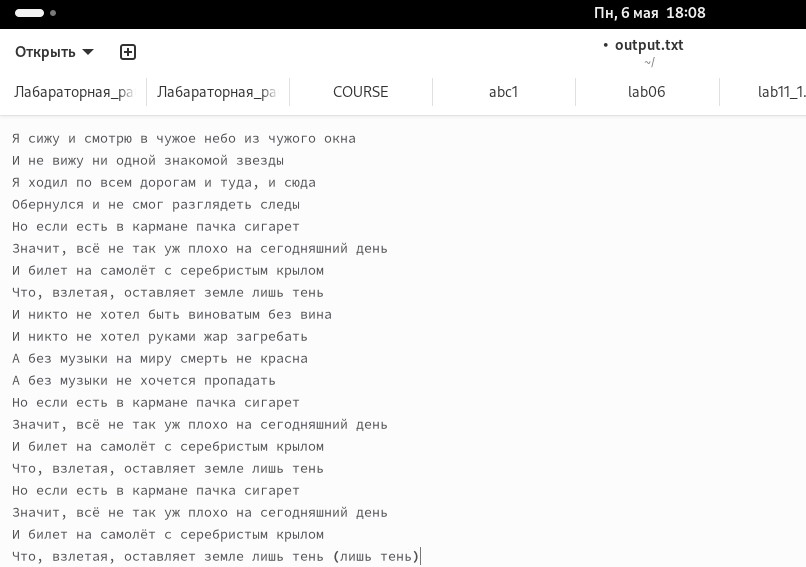


Рисунок 1.4. Результат.

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 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. 2.1, 2.2, 2.3)

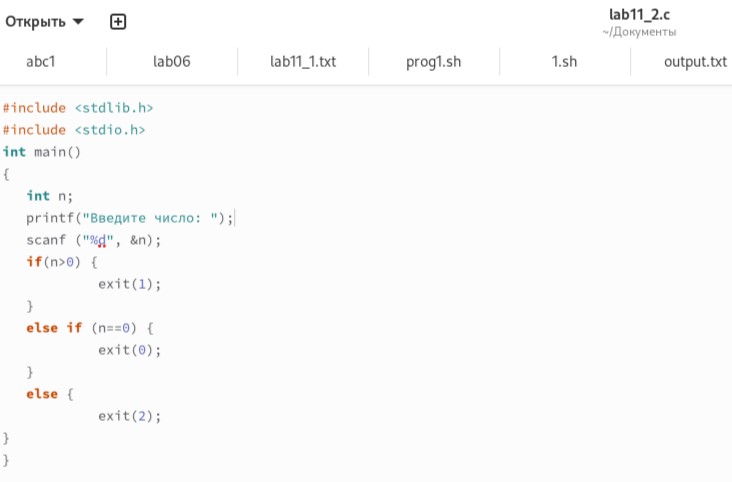


Рисунок 2.1. Программа на Си.

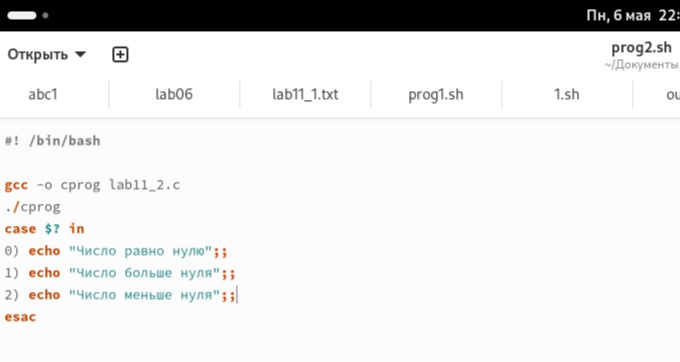


Рисунок 2.2. Программный файл.

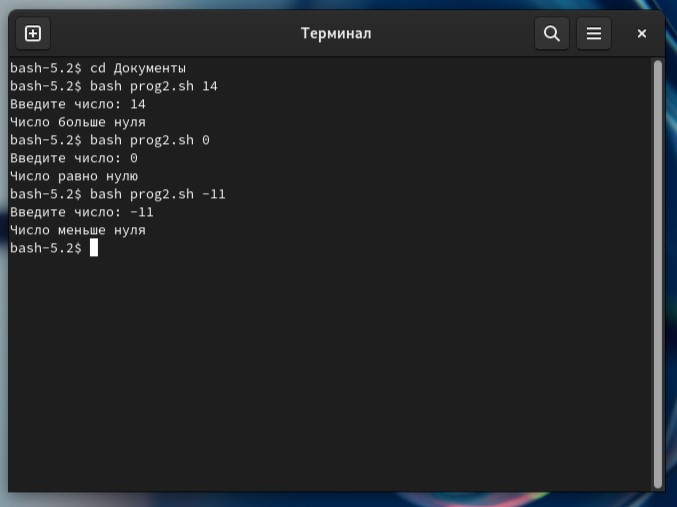


Рисунок 2.3. Результат.

3. 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) (рис. 3.1, 3.2).

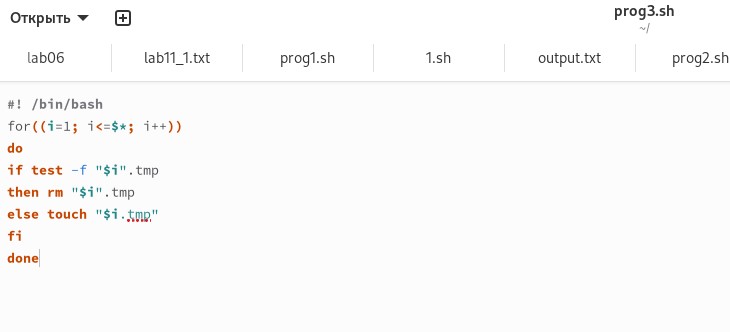


Рисунок 3.1. Программа.

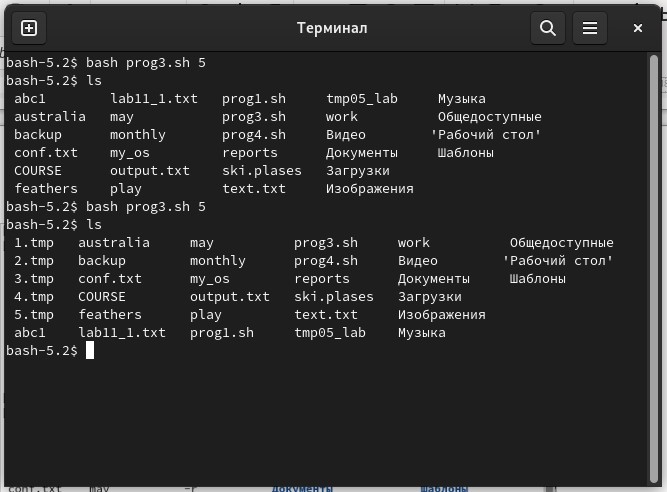


Рисунок 3.2. Результат.

4. Напишем командный файл, который с помощью команды tar запаковывает в архив

все файлы в указанной директории. Модифицируем его так, чтобы запаковывались

только те файлы, которые были изменены менее недели тому назад (использовать

команду find) (рис. 4.1, 4.2, 4.3, 4.4).

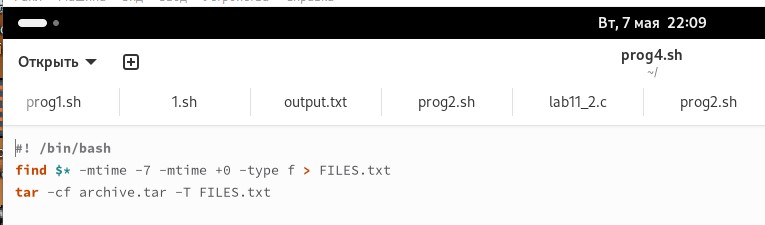


Рисунок 4.1. Программа.

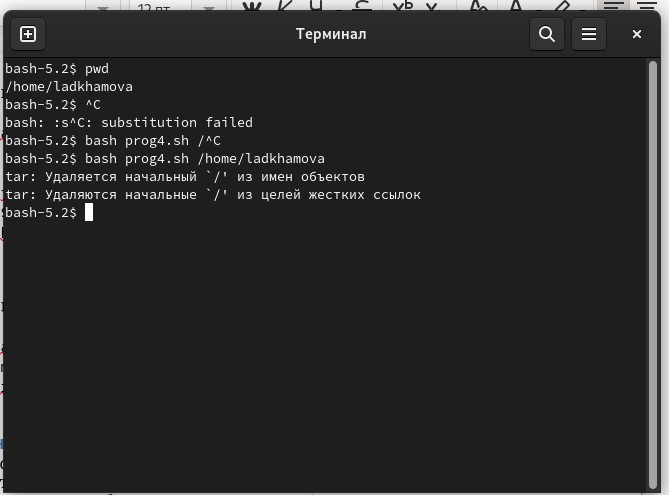


Рисунок 4.2. Текст программы.

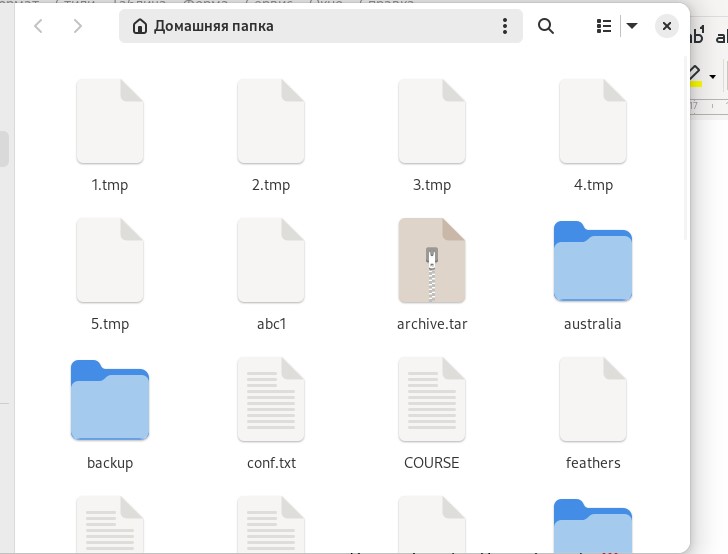


Рисунок 4.3. Созданный архив и файл.

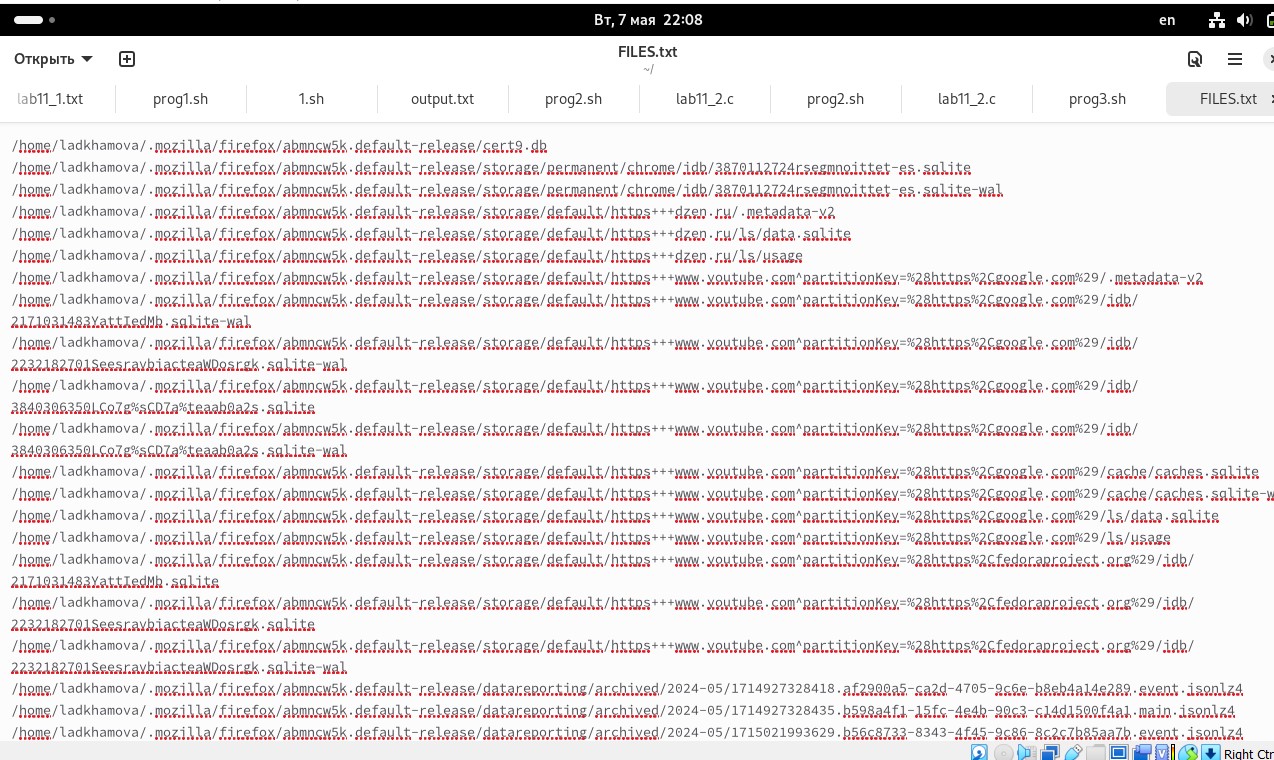


Рисунок 4.4. FILES.txt.

# Вывод:

Изучила основы программирования в оболочке ОС UNIX. Научилась писать более

сложные командные файлы с использованием логических управляющих конструкций

и циклов.

# Ответы на контрольные вопросы:

1. Каково предназначение команды getopts?

Осуществляет синтаксический анализ командной строки, выделяя флаги, и ис- пользуется для объявления переменных. Синтаксис команды следующий: getopts option-string variable [arg … ] Флаги – это опции командной строки, обычно помеченные знаком минус; Например, -F является флагом для команды ls -F. Иногда эти флаги имеют аргументы, связанные с ними. Программы интерпретируют эти флаги, соответствующим образом изменяя свое поведение. Строка опций option-string — это список возможных букв и чисел соответствующего флага. Если ожидается, что некоторый флаг будет сопровождаться некоторым аргументом, то за этой буквой должно следовать двоеточие. Соответствующей переменной при- сваивается буква данной опции. Если команда getopts может распознать аргумент, она возвращает истину. Принято включать getopts в цикл while и анализировать введенные данные с помощью оператора case.

1. Какое отношение метасимволы имеют к генерации имён файлов?

При перечислении имён файлов текущего каталога можно использовать сле- дующие символы: − соответствует произвольной, в том числе и пустой строке; ? − соответствует любому одинарному символу; [c1-c2] − соответствует любому символу, лексикографически находящемуся между символами с1 и с2. Например, echo \* − выведет имена всех файлов текущего каталога, что представляет собой простейший аналог команды ls; ls .c − выведет все файлы с последними двумя символами, совпадающими с .c. echo prog.? − выведет все файлы, состоящие из пяти или шести символов, первыми пятью символами которых являются prog.. [a-z] − соответствует произвольному имени файла в текущем каталоге, начинающемуся с любой строчной буквы латинского алфавита.

1. Какие операторы управления действиями вы знаете?

Часто бывает необходимо обеспечить проведение каких-либо действий цикли- чески и управление дальнейшими действиями в зависимости отрезультатов про- верки некоторого условия. Для решения подобных задач язык программирования bash предоставляет возможность использовать такие управляющие конструкции, как for, case, if и while. С точки зрения командного процессора эти управляю- щие конструкции являются обычными командами и могут использоваться как при создании командных файлов, так и при работе в интерактивном режиме. Команды, реализующие подобные конструкции, по сути, являются операторами языка программирования bash. Поэтому при описании языка программирова- ния bash термин оператор будет использоваться наравне с термином команда. Команды ОС UNIX возвращают код завершения, значение которого может быть использовано для принятия решения о дальнейших действиях. Команда test, например, создана специально для использования в командных файлах. Един- ственная функция этой команды заключается в выработке кода завершения.

1. Какие операторы используются для прерывания цикла?

Два несложных способа позволяют вам прерывать циклы в оболочке bash. Ко- манда break завершает выполнение цикла, а команда continue завершает данную итерацию блока операторов. Команда break полезна для завершения цикла while в ситуациях, когда условие перестаёт быть правильным. Команда continue исполь- зуется в ситуациях, когда больше нет необходимости выполнять блок операторов, но вы можете захотеть продолжить проверять данный блок на других условных выражениях.

1. Для чего нужны команды false и true?

Следующие две команды ОС UNIX используются только совместно с управляю- щими конструкциями языка программирования bash: это команда true, которая всегда возвращает код завершения, равный нулю (т.е. истина), и команда false, которая всегда возвращает код завершения, не равный нулю (т. е. ложь).

1. Что означает строка if test -f man𝑠/i.$s, встреченная в командном файле?

Строка if test -f mans/i.s, mans/i.s и является ли этот файл обычным файлом. Если данный файл является каталогом, то команда вернет нулевое значение (ложь).

1. Объясните различия между конструкциями while и until.

Выполнение оператора цикла while сводится к тому, что сначала выполня- ется последовательность команд (операторов), которую задаёт список-команд в строке, содержащей служебное слово while, а затем, если последняя выпол- ненная команда из этой последовательности команд возвращает нулевой код завершения (истина), выполняется последовательность команд (операторов), которую задаёт список-команд в строке, содержащей служебное слово do, после чего осуществляется безусловный переход на начало оператора цикла while. Вы- ход из цикла будет осуществлён тогда, когда последняя выполненная команда из последовательности команд (операторов), которую задаёт список-команд в строке, содержащей служебное слово while, возвратит ненулевой код заверше- ния (ложь). При замене в операторе цикла while служебного слова while на until условие, при выполнении которого осуществляется выход из цикла, меняется на противоположное. В остальном оператор цикла while и оператор цикла until идентичны