Лабораторная работа 11

Отчет по лабораторной работе 11

Куркина Евгения Вячеславовна

Содержание

# 1 Цель работы

Здесь приводится формулировка цели лабораторной работы. Формулировки цели для каждой лабораторной работы приведены в методических указаниях.

Цель данной лабораторной работы— Изучить основы программирования в оболочке OC UNIX. Научиться писать более сложные командные файлы с использованием логических управляющих конструкций и циклов.

# 2 Задание

Здесь приводится описание задания в соответствии с рекомендациями методического пособия и выданным вариантом.

# 3 Выполнение лабораторной работы

1)Создала файл для первого скрипта, перешла в emacs(рис. 1). С помощью команды getopts grep, написала ткст скрипта, который анализирует строку с ключами: -inputfile-прочитывает данные из указанного файла -aotputfile-вывод данных в указанный файл -p шаблон -указать шаблон поиска -С- различить строчные и заглавные буквы -n- выдть номера строк, а затем поиск указанных строк, которые определяются ключом -p.(рис. 2). Затем сохранила файл (рис. 3). Создала два файла, зполнила их словами(рис. 4) (рис. 5). Предоставила доступ на исполнение (рис. 6), проверила корректную работу скрипта (рис. 7).

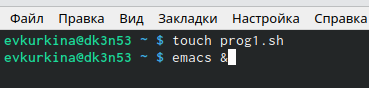


Рис. 1: Создание файла, переход в emacs

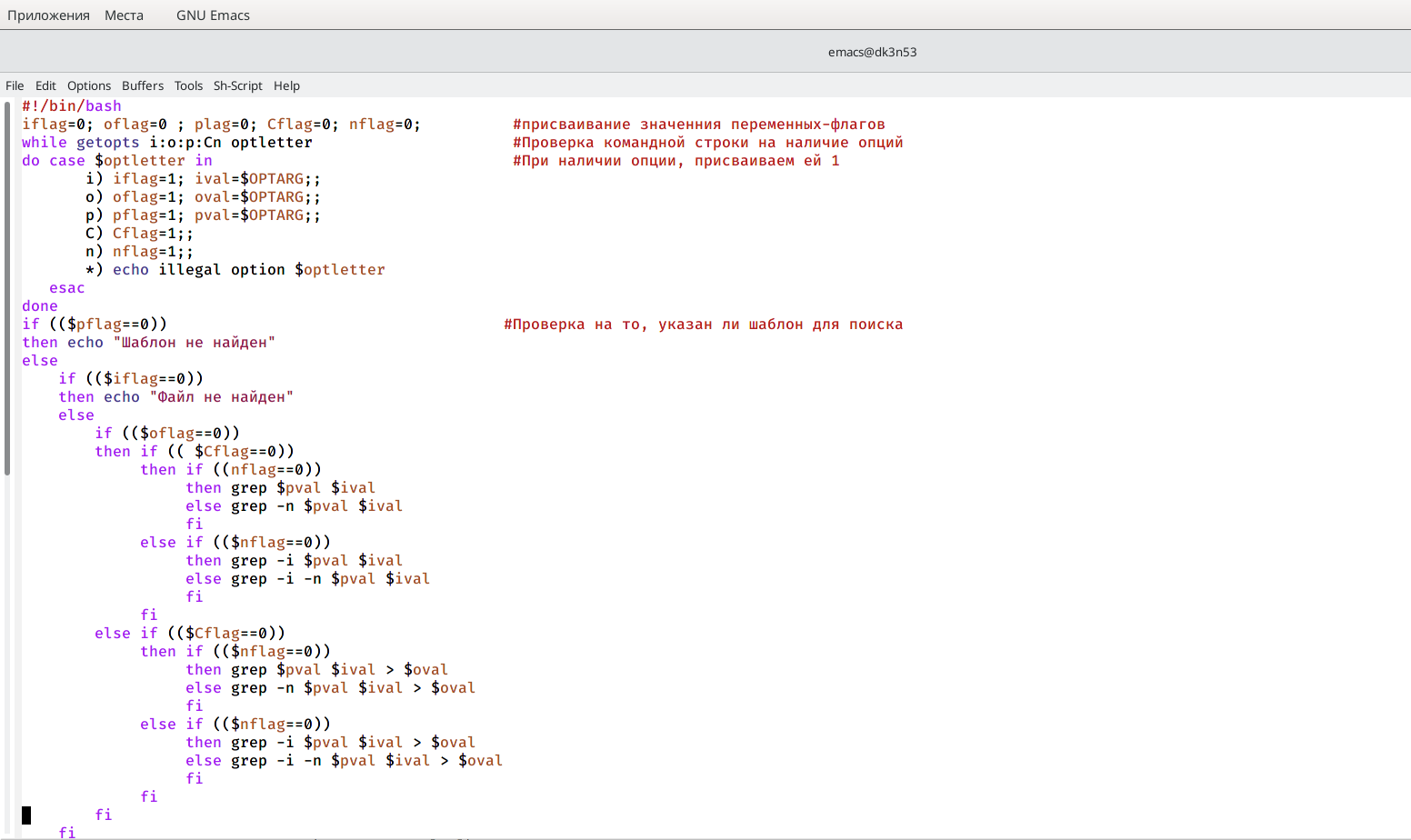


Рис. 2: Текст скрипта

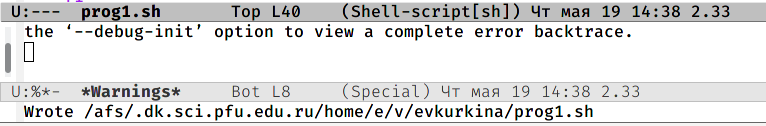


Рис. 3: Сохранение файла

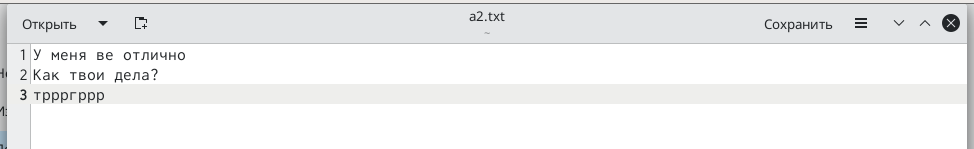


Рис. 4: Текст первого файла

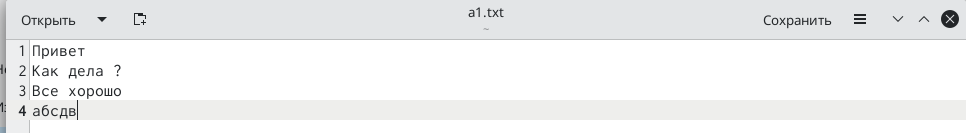


Рис. 5: текст второго файла

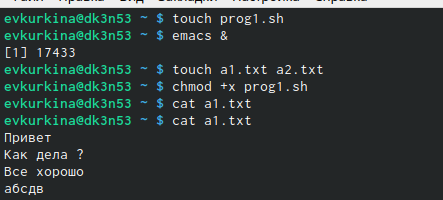


Рис. 6: Права доступа

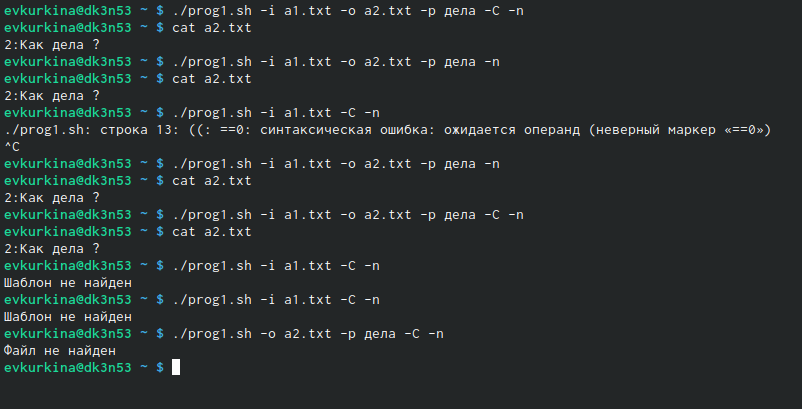


Рис. 7: Проверка работы скрипта

1. Создала два новых файла, для второго скрипта (рис. 8). На языке С написала программу, которая определяет, введенное число меньше, больше или же равно нулю, затем код завершается с помощью функции exit(i)(рис. 9), сохранила его (рис. 10).После командный файл вызывает данную программу и выдает сообщение о том, какое число было введено пользователем(рис. 11). Перешла в консоль, дала права доступа и проверила работу кода (рис. 12).

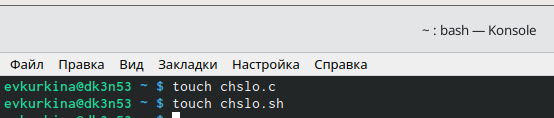


Рис. 8: Создане двух файлов

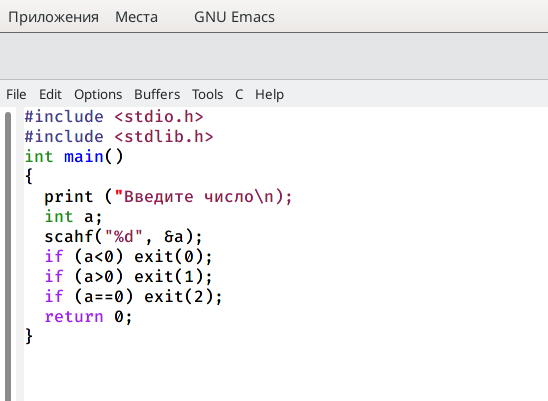


Рис. 9: Код программы на языке С

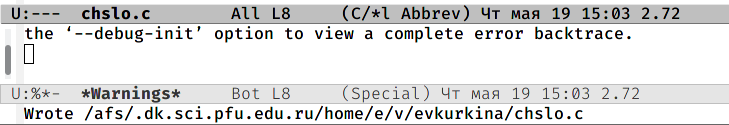


Рис. 10: Сохранение файла

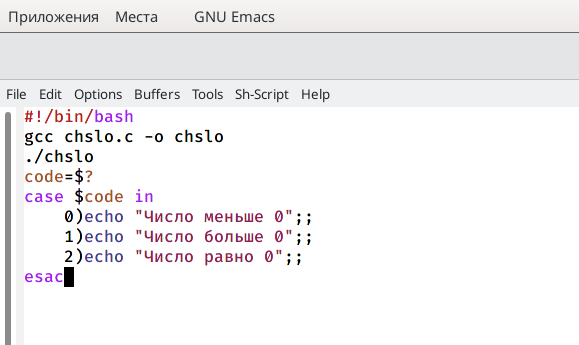


Рис. 11: Код исполняемого файла

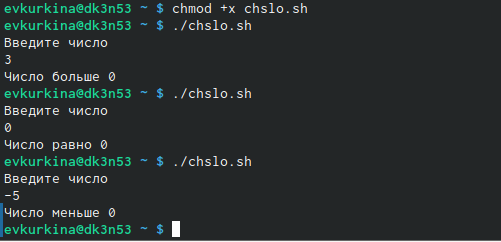


Рис. 12: права доступа и результат работы скрипта

1. Создала новый файл для 3его скрипта (рис. 13). Написала код исполняемого файла, кторый создает указанное пользователем число файлов, которые пронумерованы от 1 до N. Число файлов передается аргументом в командной строке, также этот скрипт может удалить все созданные им файлы(рис. 14). Далее, я предоставила доступ к исполнению, создала три файла, а затем удалила их , с помощью кода.(рис. 15)

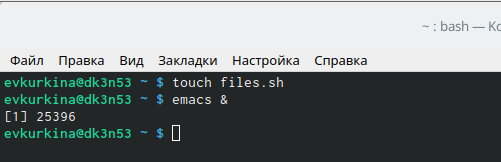


Рис. 13: Создание файлов и переход в emacs



Рис. 14: Код программы

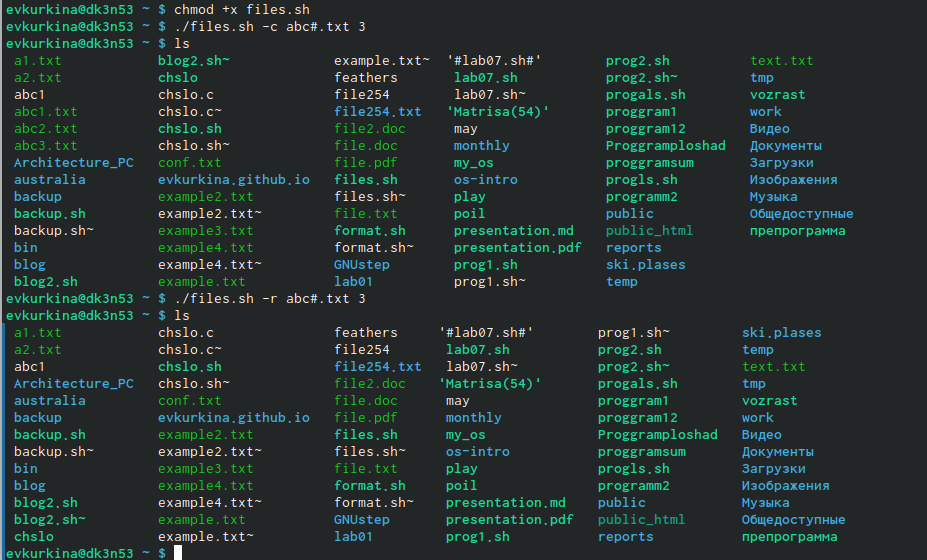


Рис. 15: Права доступа и проверка работы скрипта

1. Создала файл для последнего 4го скрипта(рис. 16). Написала текст исполняемого файла, который с помощью команды tar собирает в архив все файлы, которые находтся в этой директории.(рис. 17). Далее предоставила права доступа на выполнение, создала отдельный каталог, куда поместила файлы, а после проверила работу программы (рис. 18) (рис. 19).

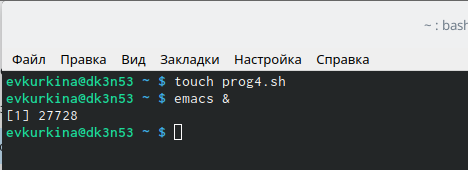


Рис. 16: Создание файла для последнего скрипта

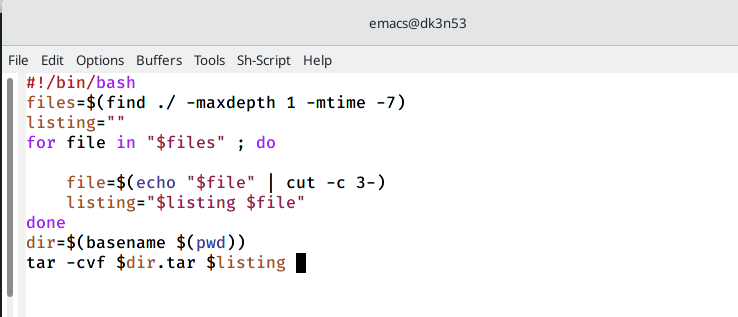


Рис. 17: Текст кода

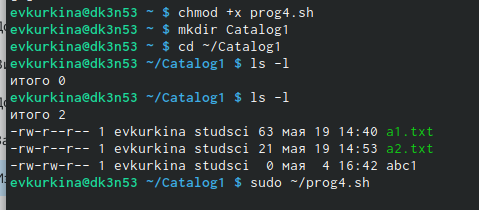


Рис. 18: Создание каталога и проверка работы скрипта

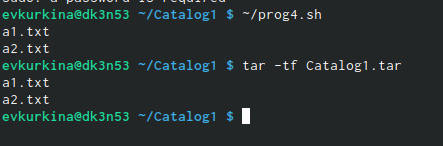


Рис. 19: Результат работы

1. Ответы на контрольные вопросы:

1). Команда getopts осуществляет синтаксический анализ командной строки, выделяя флаги, ииспользуется для объявления переменных. Синтаксис команды следующий: getopts option-string variable [arg…] Флаги − это опции командной строки, обычно помеченные знаком минус; Например,для команды ls флагом может являться -F. Строка опций option-string − эт осписок возможных букв и чисел соответствующего флага. Если ожидается, что некоторый флаг будет сопровождаться некоторым аргументом, то за символом, обозначающим этот флаг, должно следовать двоеточие. Соответствующей переменной присваивается буква данной опции. Еслик оманда getopts может распознать аргумент, то она возвращает истину. Принято включать getopts в цикл while и анализировать введённые данные с помощью оператора case. Функция getopts включает две специальные переменные среды −OPTARG и OPTIND. Если ожидается доплнительное значение,то OPTARG устанавливается в значение этого аргумента. Функция getopts также понимает переменные типа массив, следовательно, можно использовать её в функции не только для синтаксического анализа аргументов функций, но и для анализа введённых пользователем данных.

2). Приперечислении имён файлов текущего каталога можно использовать следующие 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, 1.1 echo\* − выведет имена всех файлов текущего каталога, что представляет собой простейший аналог команды ls; 1.2. ls\*.c−выведет все файлы с последними двумя символами, совпадающими с.c. 1.3. echoprog.?−выведет все файлы, состоящие из пяти или шести символов, первыми пятью символами которых являются prog.. 1.4.[a-z]\*−соответствует произвольному имени файла в текущем каталоге, начинающемуся с любой строчной буквы латинского алфавита.

3). Часто бывает необходимо обеспечить проведение каких-либо действий циклически и управление дальнейшими действиями в зависимости от результатов проверки некоторого условия. Для решения подобных задач язык программирования bash предоставляет возможность использовать такие управляющие конструкции, как for, case, if иwhile. С точки зрения командного процессора эти управляющие 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 программирования bash термин оператор будет использоваться наравне с термином команда. Команды ОСUNIX возвращают код завершения, значение которого может быть использовано для принятия решения о дальнейших действиях. Команда test, например, создана специально для использования в командных файлах. Единственная функция этой команды заключается в выработке кода завершения.

4). Два несложных способа позволяют вам прерывать циклы в оболочке bash. Команда 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 используется в ситуациях, когда больше нет необходимости выполнять блок операторов, но вы можете захотеть продолжить проверять данный блок на других условных выражениях.

5). Следующие две команды ОСUNIX используются только совместно с управляющими 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,неравный нулю(т.е.ложь).Примеры бесконечных циклов:while true do echo hello andy done until false do echo hello mike done.

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

7). Выполнение оператора цикла while сводится к тому,что сначала выполняется последовательность команд(операторов),которую задаёт список-команд в строке,содержащей служебное слово while,а затем,если последняя выполненная 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.Выход 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 завершения(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.

# 4 Выводы

Во время выполнения данной лабораторной работы я изучила основы программирования в оболочке OC UNIX.Научилась писать более сложные командные файлы с использованием логических управляющих конструкций и циклов.

# Список литературы