Отчёт по лабораторной работе №11

*дисциплина: Операционные системы*

Максим Александрович Мишонков

Содержание

# 1 Цель работы

Целью выполнения данной лабораторной работы является изучение основ программирования в оболочке ОС UNIX, приобретение навыков написания сложных командных файлов с использованием логических управляющих конструкций и циклов.

# 2 Задание

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

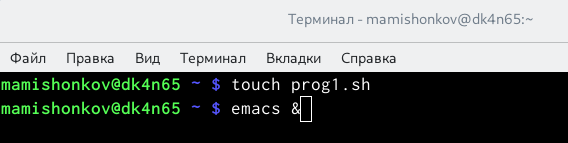
# 3 Теоретическое введение

Команда getopts осуществляет синтаксический анализ командной строки, выделяя флаги, и используется 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 присваивается 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.

# 4 Выполнение лабораторной работы

1. Создал файл для программы 1. (рис. [??])

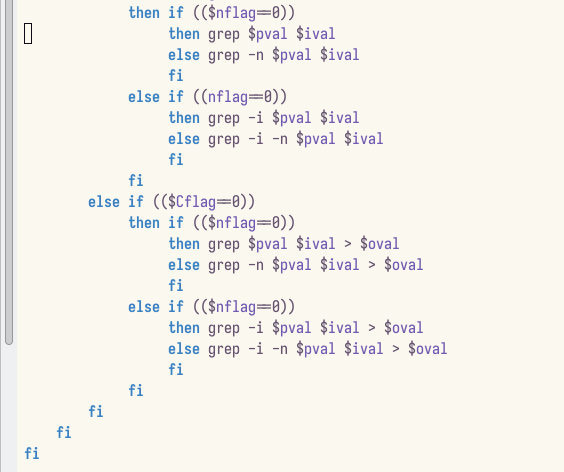


Создание файла

1. Написал текст программы 1. (рис. [??], [??])

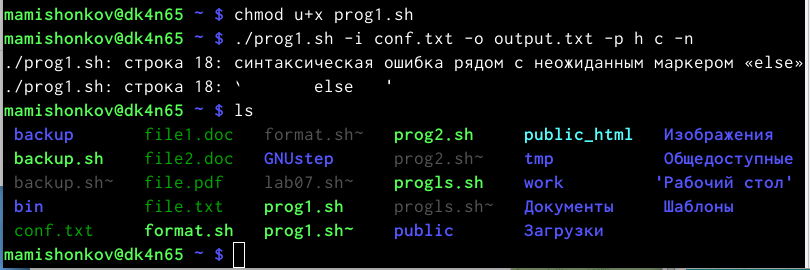


Текст программы



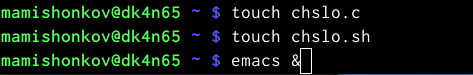
Текст программы

1. Проверил работу написанной программы. (рис. [??])



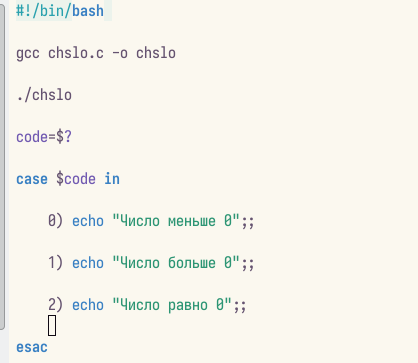
Проверка работы написанной программы

1. Создал файлы для программы 2. (рис. [??])

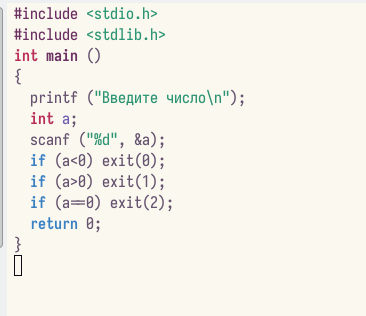


Создание файла

1. Написал текст программы 2. (рис. [??], [??])



Текст программы



Текст программы

1. Создал файлы для программы 3. (рис. [??])

Создание файла

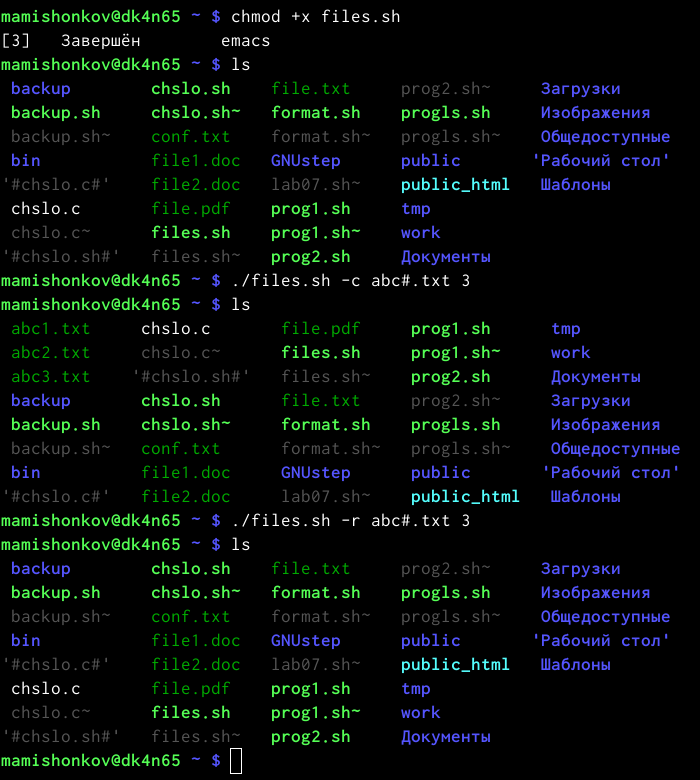
Создание файла

1. Написал текст программы 3. (рис. [??])



Текст программы

1. Проверил работу написанной программы. (рис. [??])



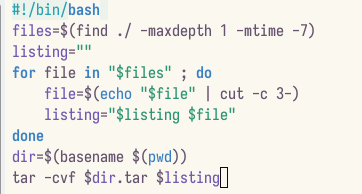
Проверка работы написанной программы

1. Создал файлы для программы 4. (рис. [??])

Создание файла

Создание файла

1. Написал текст программы 4. (рис. [??])



Текст программы

**Контрольные вопросы:**

1. Каково предназначение команды getopts?

Команда getopts осуществляет синтаксический анализ командной строки, выделяя флаги, и используется 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 присваивается 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 имен файлов текущего каталога можно использовать следующие символы:

? — соответствует любому одному символу;  
  
[c1-c1] — соответствует любому символу, лексикографически на ходящемуся между символами c1 и с2;  
  
echo \* — выведет имена всех файлов текущего каталога, что представляет собой простейший аналог команды ls;  
  
ls .c — выведет все файлы с последними двумя символами, равными .c;  
  
echo prog.? — выдаст 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?

* Точка с запятой (;) Вы можете разместить две и более команд в одной и той же строке, разделив эти команды с помощью символа точки с запятой ;.
* Амперсанд (&) В том случае, если строка команды оканчивается символом амперсанда &, командная оболочка не будет ожидать завершения исполнения этой команды. Сразу же после ввода команды будет выведено новое приглашение командной оболочки, а сама команда будет исполняться в фоновом режиме. В момент завершения исполнения команды в фоновом режиме вы получите соответствующее сообщение.
* Cимвол доллара со знаком вопроса
* Двойной амперсанд (&&) Командная оболочка будет интерпретировать последовательность символов && как логический оператор “И”. При использовании оператора && вторая команда будет исполняться только в том случае, если исполнение первой команды успешно завершится (будет возвращен нулевой код завершения).
* Двойная вертикальная черта (||) Оператор || представляет логическую операцию “ИЛИ”. Вторая команда исполняется только тогда, когда исполнение первой команды заканчивается неудачей (возвращается ненулевой код завершения).
* Комбинирование операторов && и || Вы можете использовать описанные логические операторы “И” и “ИЛИ” для создания структур условных переходов в рамках строк команд.
* Знак фунта (#) Все написанное после символа фунта (#) игнорируется командной оболочкой. Это обстоятельство оказывается полезным при возникновении необходимости в написании комментариев в сценариях командной оболочки, причем комментарии ни коим образом не будут влиять на процесс исполнения команд или процесс раскрытия команд командной оболочкой.
* Экранирование специальных символов () Символ обратного слэша  позволяет использовать управляющие символы без их интерпретации командной оболочкой; процедура добавления данного символа перед управляющими символами называется экранированием символов.

1. Какие операторы используются для прерывания цикла?

Для управления ходом выполнения цикла служат команды break и continue [1] и точно соответствуют своим аналогам в других языках программирования. Команда break прерывает исполнение цикла, в то время как continue передает управление в начало цикло, минуя все последующие команды в теле цикла.

1. Для чего нужны команды false и true?

Команда true всегда возвращает ноль в качестве выходного статуса для индикации успеха.

Команда false всегда возвращает не-ноль в качестве выходного статуса для индикации неудачи.

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

Веденная строка означает условие существования файла mani.$s

1. Объясните различия между конструкциями while и until.

Разница между циклом while (пока) и until (пока не) – это условие проверки. Пока ВЫПОЛНЯЕТСЯ условие проверки, цикл while будет продолжать работать. Однако цикл until будет выполняться только пока условие ЛОЖНО.

# 5 Выводы

В ходе выполнения данной лабораторной работы я зучил основы программирования в оболочке ОС UNIX, научился писать сложные командные файлы с использованием логических управляющих конструкций и циклов.