Министерство науки и высшего образования Российской Федерации

Федеральное государственное автономное образовательное учреждение

высшего образования

«СЕВЕРО-КАВКАЗСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»

Институт цифрового развития

Кафедра инфокоммуникаций

**«РАЗРАБОТКА СЦЕНАРИЕВ BASH»**

**ОТЧЕТ**

**по лабораторной работе №13**

**дисциплины**

**«Операционные системы»**

|  |  |  |  |
| --- | --- | --- | --- |
|  | | Выполнил:  Мизин Глеб Егорович  2 курс, группа ПИЖ-б-о-21-1,  09.03.04 «Программная инженерия», направленность (профиль) «Разработка и сопровождение программного обеспечения», очная форма обучения  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  (подпись) | |
|  | | Проверил:  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  (подпись) | |
|  | |  | |

Отчет защищен с оценкой \_\_\_\_\_\_\_\_\_\_\_ Дата защиты\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Ставрополь, 2022 г.

**Цель работы**: практическое знакомство с методами создания и использования сценариев ОС Linux.

**Задание №1**: получить полный список ключей команды stat.

Получим справку команды stat при помощи команды men:

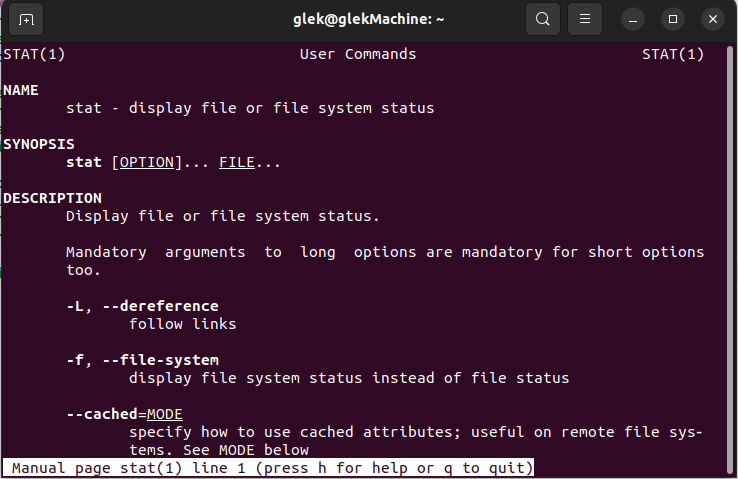


Рисунок 1 – Справка команды stat

**Задание №2**: вычислить факториал целого числа, вводимого с клавиатуры. Предусмотреть проверку правильности ввода аргумента.

Создадим исполняемый файл командой touch и выдадим всем пользователям возможность выполнения этого файла через chmod a+x

В исполняемом файле напишем программу решающую поставленную задачу

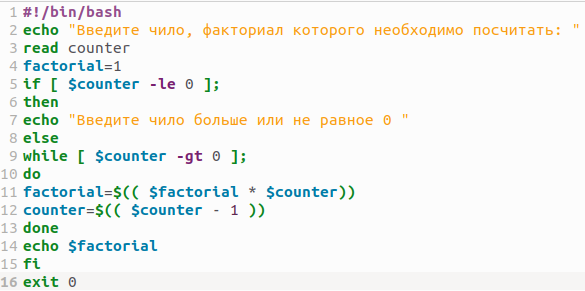


Рисунок 2 – Листинг сценария 1\_task

Далее при помощи команды «. /1\_task» запустим наш исполняемый файл

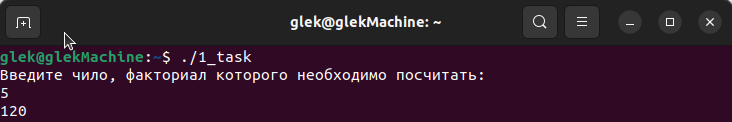


Рисунок 3 – Результат работы сценария 1\_task

**Задание №3**: найти первые N чисел Фибоначчи, используя рекуррентное соотношение Ai+1=Ai+Ai-1 Значения первых двух чисел и необходимое количество чисел N ввести с клавиатуры.

Подготовка файла и запуск будут осуществляется так же, как и в задании №1

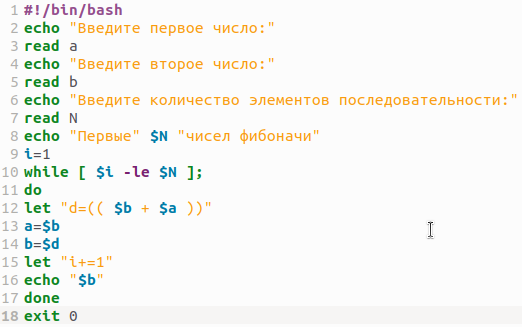


Рисунок 4 – Листинг сценария 2\_task

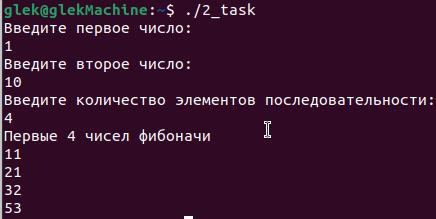


Рисунок 5 – Результат работы сценария 2\_task

**Индивидуальное задание**: В текущем каталоге найти выполняемый файл наибольшего размера.

Для выполнения поставленной задачи была использована команда find с аргументами maxdepth 1 который позволяет производить команду find только в каталоге, в котором находится пользователь. Аргумент type с ключом f выберет только обычные файлы. Аргумент perm выберет фалы с подходящими правами доступа. Команда printf выведет на экран информацию по заданному формату. Команда sort с ключом n отсортирует выданный список по числовым значениям, tail -1 возьмёт последнее значение в списке, а команда cut с ключом -d’ ’ -f 2 разделит выведенную строку по символу « » и выведет 2 значение.

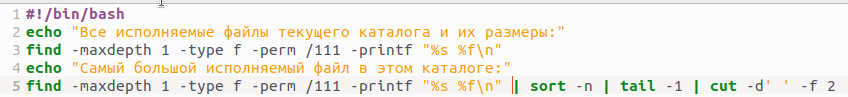


Рисунок 6 – Листинг сценария ind

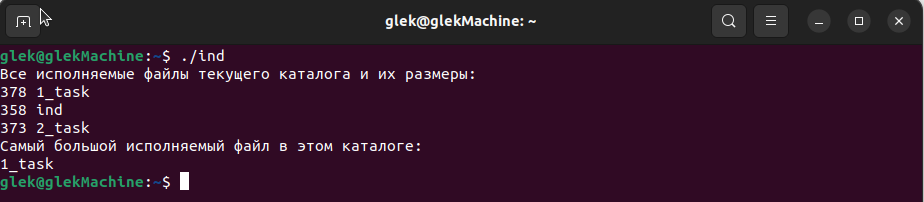


Рисунок 5 – Результат работы сценария ind

**Вывод**: ознакомились с методами создания и использования сценариев ОС Linux.

**Контрольные вопросы**:

1 Назначение, создание и выполнение сценариев.

Создать файл, содержащий команды, можно помощью любого текстового редактора. В данной работе рекомендуется использовать встроенный в mc редактор. Для создания нового файла в mc используйте комбинацию клавиш Shift+F4.

2 Использование кавычек в командной строке.

“” Используются для вывода текста или для проверки пустой переменной

3 Переменные в bash.

В командной оболочке переменные перед применением обычно не объявляются. Вместо этого они создаются (например, когда им присваивается начальное значение). По умолчанию все переменные считаются строками и хранятся как строки, даже когда им присваиваются числовые значения. Командная оболочка и некоторые утилиты преобразуют строки, содержащие числа, в числовые значения, когда с переменными нужно выполнить арифметические операции. Командная оболочка считает fоо и Foo двумя разными переменными, отличающимися от третьей переменной FOO.

4 Перенаправление ввода-вывода и каналы

Подмена стандартного вывода - задача командной оболочки (shell). Shell создает пустой файл, имя которого указано после знака ">", и дескриптор этого файла передается программе под номером 1 (стандартный вывод). Делается это очень просто. При запуске программы из оболочки после выполнения fork() появляется два одинаковых процесса, один из которых - дочерний - должен запустить вместо себя команду (выполнить exec()). Перед этим он закрывает стандартный вывод (дескриптор 1 освобождается) и открывает файл (с ним связывается первый свободный дескриптор, т. е. 1), а запускаемой команде ничего знать и не надо: ее стандартный вывод уже подменен. Эта операция называется перенаправлением стандартного вывода. В том случае, если файл уже существует, shell запишет его заново, полностью уничтожив все, что в нем содержалось до этого. Поэтому, чтобы продолжить записывать данные в textfile, потребуется другая операция - ">>":

5 Превращение сценария в исполняемый файл

Файл сценария можно выполнить двумя способами. Более простой путь — запустить оболочку с именем файла сценария как параметром: $ /bin/sh first Этот вариант будет работать, но лучше запускать сценарий, введя его имя и тем самым присвоив ему статус других команд Linux. Сделать это можно с помощью команды chmod, изменив режим файла (file mode) и сделав его исполняемым для всех пользователей: $ chmod +x first После этого можно выполнять файл с помощью команды $ first При этом может появиться сообщение об ошибке, говорящее о том, что команда не найдена. Исправить ошибку можно введя с клавиатуры в командной строке ./first в каталоге, содержащем сценарий, чтобы задать командной оболочке полный относительный путь к файлу.

6 Команда test или [

На практике в большинстве сценариев широко используется команда [ или test -логическая проверка командной оболочки. В некоторых системах команды [ и test - синонимы, за исключением того, что при использовании команды [ для удобочитаемости применяется и завершающая часть ]. В программном коде команда [ упрощает синтаксис и делает его более похожим на другие языки программирования.

7 Оператор разветвления if

Оператор if очень прост: он проверяет результат выполнения команды и затем в зависимости от условия выполняет ту или иную группу операторов.

8 Проверка выполнения нескольких команд

Проверка выполнения нескольких условий (выполнение нескольких команд): Иногда необходимо выполнить оператор, только если удовлетворяется несколько условий, например

if [ -f this\_file ]; then

if [ -f that\_file ]; then

if [ -f other\_file ]; then

echo "All files present"

fi

fi

fi

9 Оператор выбора case

Конструкция оператора case позволяет сопоставлять содержимое переменной с образцами и затем выполнять разные операторы в зависимости от того, с каким образцом найдено соответствие. Это гораздо проще, чем проверять несколько условий, применяемых во множественных операторах if, elif и else.

10.Операторы цикла

Цикл for предназначен для обработки в цикле ряда значений, которые могут представлять собой любое множество строк. Строки могут быть просто перечислены в программе или, что бывает чаще, представлять собой результат выполненной командной оболочкой подстановки имен файлов.

11 Команды break и continue - назначение, примеры использования

Команда break используется для выхода из циклов for, while и until до того, как будет удовлетворено управляющее условие

Как и одноименный оператор языка С, эта команда заставляет охватывающий ее цикл for, while нли until начать новый проход или следующую итерацию. При этом переменная цикла принимает следующее значение в списке.

12.Команда printf – назначение, отличия от языка С, примеры использования

При использовании команды echo возникает общая проблема - удаление символа перехода на новую строку. В ОС Linux общепринятый метод - echo -n "string to output" printf Команда printf есть только в современных командных оболочках. Группа Х/Ореп полагает, что ее следует применять вместо команды echo для генерации форматированного вывода.

13.Встроенные документы.

Особый способ передачи из сценария командной оьолочки входных данных команде – использование встроенного документа. Такой документ позволяет команде выполняться так, как будто она читает данные из файла или с клавиатуты, в то время как на самом деле она получает их из сценария.

14.Отладка сценариев

При обноружении ошибки при выполнении сценария командная оболочка выводит на экран номер строки, содержащей ошибку. Если ошибка сразу не видна, нужно добавить несколько дополнительных команд echo для вывода значений переменных и протестировать фрагменты программного кода, вводя их в командной оболочке в интерактивном режиме. Основной способ отслеживания наиболее трудно выявляемых ошибок – использование отладочных опций командной оболочки.