Отчёт по прохождению курса третий этап

Абронина Алиса Кирилловна

Содержание

Список иллюстраций

# 1 Продвинутые темы

## 1.1 3.1 Текстовый редактор vim

: переводит в командный режим; q команда вывода; Enter выволнение команды

![Figure 1: Задание 1](data:application/octet-stream;base64,)

Figure 1: Задание 1

word - разбивает по буквам, цифрам, знакам; WORD - разбивает только по пробелам -> Strange\_ TEXT is\_here. 2=2 YES! -> 5 WORD

![Figure 2: Задание 2](data:application/octet-stream;base64,)

Figure 2: Задание 2

d2w - удалить два слова; ww - перейти на слово three; yw - yank (копировать) four; P,p - вставка до и после курсора; i - вставка текста вручную

![Figure 3: Задание 3](data:application/octet-stream;base64,)

Figure 3: Задание 3

делала все по заданию

![Figure 4: Задание 4](data:application/octet-stream;base64,)

Figure 4: Задание 4

v - войти в Visual mode; d- удалоить выделенное; y - копировать

![Figure 5: Задание 5](data:application/octet-stream;base64,)

Figure 5: Задание 5

## 1.2 3.2 Скрипты на bash: основы

каждая оболочка имеет свою локальную историю; последний активный bash сохраняет тольео свои команды С1,С2,С3

![Figure 6: Задание 1](data:application/octet-stream;base64,)

Figure 6: Задание 1

сначала переход /home/bi, там создается файл, cd на Desktop уже после создания

![Figure 7: Задание 2](data:application/octet-stream;base64,)

Figure 7: Задание 2

переменные начинаются с буквы или \_, далее цифры и буквы; остальные варинты содержат недопустимые символы

![Figure 8: Задание 3](data:application/octet-stream;base64,)

Figure 8: Задание 3

Скрипт выводит аргументы в нужном формате, экранируя $1 и $2, чтобы они тотбражались как текст

![Figure 9: Задание 4](data:application/octet-stream;base64,)

Figure 9: Задание 4

## 1.3 3.3 Скрипты на bash: ветвления и циклы

$# - кол-во аргументов(всегда>=0), 5 -ge 5 - истина -s $0 - файл скирпта существует и не пуст

![Figure 10: Задание 1](data:application/octet-stream;base64,)

Figure 10: Задание 1

первое уловие - ложь, второе - ложь, третье - ложь, выполняется echo four потом var =5 все условие ложные и снова four

![Figure 11: Задание 2](data:application/octet-stream;base64,)

Figure 11: Задание 2

если аргумент равен 1, выводится 1 student, если от 2 до 4 N students, если 5 или больше, выводится A lof of students. Скрипт реализует это через case и переменную res, как требует задание

![Figure 12: Задание 3](data:application/octet-stream;base64,)

Figure 12: Задание 3

перебираются строки a,b c\_d всего 5 итераций: каждая итерация выводит start, если str > “c” выполняется continue, пропуская echo “finish”, только c\_d > c поэтому continue срабатывает 1 раз

![Figure 13: Задание 4](data:application/octet-stream;base64,)

Figure 13: Задание 4

проверка на пуской ввод и неккоректный возраст есть, группы определяются правильно по условиям, программа работае до тех пор, пока пользователь не выводет пустое имя или некорректный возраст, вывод соотвествует требованиям

![Figure 14: Задание 5](data:application/octet-stream;base64,)

Figure 14: Задание 5

## 1.4 3.4 Скрипты на bash: разное

let “a+=b” правильно, но не отмечено, возможно баг; a=b записывать строку, ане сумму; let a = a + b синтаксическая оошибка; a+=$b пробел делает команду некорректной

![Figure 15: Задание 1](data:application/octet-stream;base64,)

Figure 15: Задание 1

после cd /home/bi pwd покажет эту директорию

![Figure 16: Задание 2](data:application/octet-stream;base64,)

Figure 16: Задание 2

program запускает команду $? содержит код невозврата, перенаправляет вывод

![Figure 17: Задание 3](data:application/octet-stream;base64,)

Figure 17: Задание 3

делала все по заданию

![Figure 18: Задание 4](data:application/octet-stream;base64,)

Figure 18: Задание 4

алгоритм Евклида реализован корректно(с вычитание, ане остатком), условия выхода из функции - проверка равенста, бесконечный цикл с возможность выхода реальзован правильно, проверка на завершение работы реализована через пустой ввод

![Figure 19: Задание 5](data:application/octet-stream;base64,)

Figure 19: Задание 5

есть коректный выход по команде exit, есть проверка на валидность чисел через регулярные выражения, есть базовые арифметические операции реальзованы через case скрипт завершает работу при любой ошибке

![Figure 20: Задание 6](data:application/octet-stream;base64,)

Figure 20: Задание 6

## 1.5 3.5 Продвинутый поиск и редактирование

name чувствителен к регистру, iname - игнорирует регистр

![Figure 21: Задание 1](data:application/octet-stream;base64,)

Figure 21: Задание 1

path проверяет весь путь , name только имя файла проверяет

![Figure 22: Задание 2](data:application/octet-stream;base64,)

Figure 22: Задание 2

mindpath 2 - ищет только начиная со второго уровня( то есть первый исключается), а есть третьий у него глубина 4

![Figure 23: Задание 3](data:application/octet-stream;base64,)

Figure 23: Задание 3

-А 1, -В 1, -С 1 добавляют дополнительные строки вокруг совпадений. Тк каждая строка совпадение, результат будет одинаково полноым в любом случае

![Figure 24: Задание 4](data:application/octet-stream;base64,)

Figure 24: Задание 4

в начале команды - 1 символ или ничего дальше ubuntu с любым регитсром $ конец строки

![Figure 25: Задание 5](data:application/octet-stream;base64,)

Figure 25: Задание 5

sed по умолчанию выводит всю строку, р - означает повторно печатать строки

![Figure 26: Задание 6](data:application/octet-stream;base64,)

Figure 26: Задание 6

делала все по заданию

![Figure 27: Задание 7](data:application/octet-stream;base64,)

Figure 27: Задание 7

## 1.6 3.6 Строим графики в gnuplot

опция р сохраняет окно графика открытым после завершения gnuplot

![Figure 28: Задание 1](data:application/octet-stream;base64,)

Figure 28: Задание 1

команда set key autotitle columnhead говорит gnuplot использовать газоловки столбцов как названия но в data/csv нет заголовков

![Figure 29: Задание 2](data:application/octet-stream;base64,)

Figure 29: Задание 2

делала все по заданию

![Figure 30: Задание 3](data:application/octet-stream;base64,)

Figure 30: Задание 3

делала все по заданию

![Figure 31: Задание 4](data:application/octet-stream;base64,)

Figure 31: Задание 4

## 1.7 3.7 Разное

rwxrw-r– =764 ( в восьмеричной СС)

![Figure 32: Задание 1](data:application/octet-stream;base64,)

Figure 32: Задание 1

chmod +w dir - дает право на изменение содержимого, chown - передает права владельца, что дает полные возможности в каталоге

![Figure 33: Задание 2](data:application/octet-stream;base64,)

Figure 33: Задание 2

wc считает: строки, слова, байсты, длину самой блинной строки

![Figure 34: Задание 3](data:application/octet-stream;base64,)

Figure 34: Задание 3

делала по заданию

![Figure 35: Задание 4](data:application/octet-stream;base64,)

Figure 35: Задание 4

делалал по заданию

![Figure 36: Задание 5](data:application/octet-stream;base64,)

Figure 36: Задание 5