РОССИЙСКИЙ УНИВЕРСИТЕТ ДРУЖБЫ НАРОДОВ

Факультет физико-математических и естественных наук

Фундаментальная Информатика и Информационные технологии

ОТЧЕТ

ПО ЛАБОРАТОРНОЙ РАБОТЕ № 10

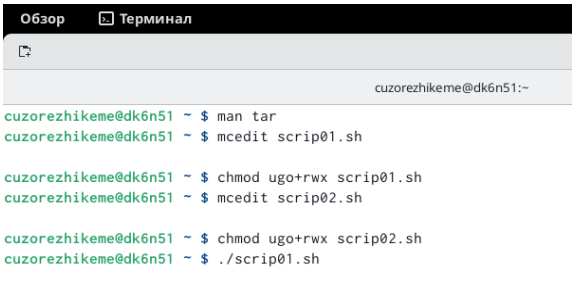
дисциплина: операционные системы

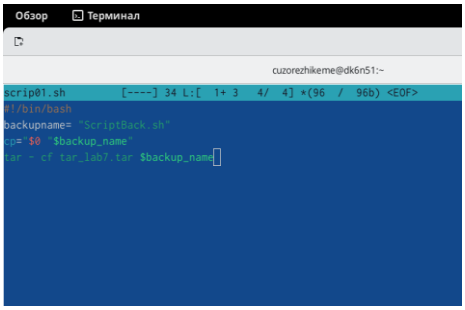
Узор-Ежикеме Чинечелум А.

НФИбд-03-21

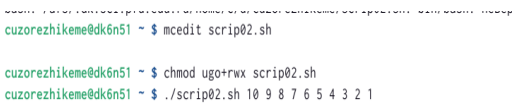
1032215263

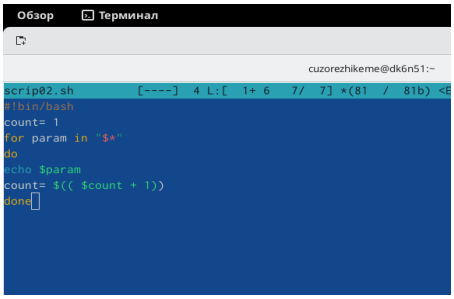
- Написал скрипт, который создает резервную копию самого себя при запуске в другой каталог резервной копии в моем домашнем каталоге, используя архиватор tar.



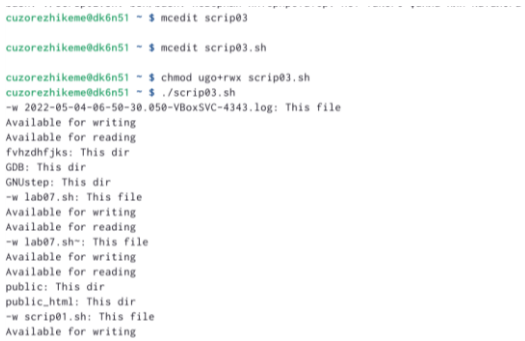


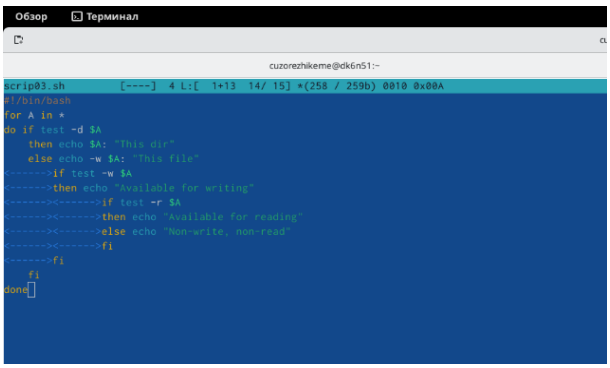
- Написал пример командного файла, который обрабатывает любое произвольное количество аргументов командной строки, включая более десяти.

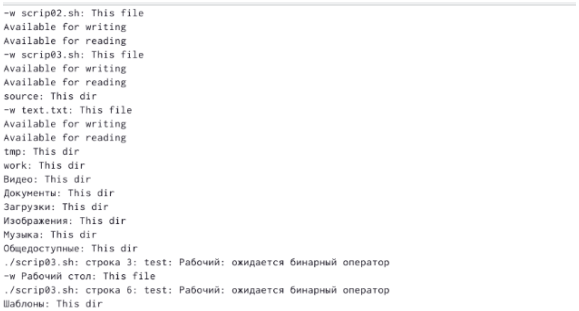




- Написал пакетный файл без использования команды и команды dir, которая предоставляет информацию о нужном каталоге и отображает информацию о возможностях доступа к файлам в каталоге.



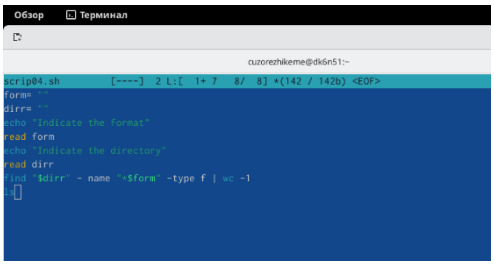




- Написал пакетный файл, который принимает формат файла (.txt, .doc, .jpg, .pdf и т.д.) в качестве аргумента командной строки и вычисляет количество таких файлов в указанном каталоге.

```
cuzorezhikeme@dk6n51 ~ $ mcedit scrip04.sh

cuzorezhikeme@dk6n51 ~ $ chmod ugo+rwx scrip04.sh
cuzorezhikeme@dk6n51 ~ $ ./scrip04.sh
./scrip04.sh: строка 1: : команда не найдена
./scrip04.sh: строка 2: : команда не найдена
Indicate the format
may
Indicate the directory
fvhzdhfjks
find: '~': Нет такого файла или каталога
find: 'name': Нет такого файла или каталога
find: '*may': Нет такого файла или каталога
wc: неверный ключ — «l»
По команде «wc --help» можно получить дополнительную информацию.
 2022-05-04-06-50-30.050-VBoxSVC-4343.log   GDB       lab07.sh    public        scrip01.sh   scrip03.sh   source    tmp   Видео       Загрузки     Музыка          'Рабочий стол'
 fvhzdhfjks                                 GNUstep   lab07.sh~   public_html   scrip02.sh   scrip04.sh   text.txt  work  Документы   Изображения  Общедоступные   Шаблоны
```



Контрольные вопросы

1. Объясните понятие командной оболочки. Приведите примеры командных оболочек.
Чем они отличаются?
- Командная оболочка - это компьютерная программа, представляющая интерфейс командной строки, который позволяет пользователю взаимодействовать с операционной системой компьютера, используя команды, вводимые с клавиатуры, вместо управления графическими пользовательскими интерфейсами (GUI) с помощью комбинации мыши и клавиатуры.

2. Что такое POSIX?
- POSIX расшифровывается как Portable Operating System Interface и является стандартом IEEE, разработанным для облегчения переносимости приложений.

3. Как определяются переменные и массивы в языке программирования bash?
- Массив Bash – Массив представляет собой набор элементов. Переменная - это символьная строка, которой мы присваиваем значение. Присвоенное значение может быть числом, текстом, именем файла, устройства или любым другим типом данных.

4. Каково назначение операторов let и read?

- Команда let является показателем того, что последующие аргументы представляют собой выражение, подлежащее вычислению. Команда read позволяет читать значения переменных со стандартного ввода

5. Какие арифметические операции можно применять в языке программирования bash?
-

Синтаксис	Описание
++x, x++	До и после увеличения.
--x, x--	До и после декремента.
+, -, *, /	Сложение, вычитание, умножение, деление.
%, ** (или ^)	По модулю (остаток) и возведение в степень.
&&, ||, !	Логическое И, ИЛИ и отрицание.
&, |, ^, ~	Побитовое И, ИЛИ, XOR и отрицание
<=, <, >, =>	Меньше или равно, меньше, больше и больше или равно операторам сравнения.
= =, !=	Операторы сравнения равенства и неравенства.
=	Оператор присваивания. Комбинируется с другими арифметическими операторами.

6. Что означает операция (( ))?
- Мы можем записать условия оболочки bash в (( )), чтобы облегчить программирование.

7. Какие стандартные имена переменных Вам известны?
- HOME, IFS, MAIL, TERM, LOGNAME.

8. Что такое метасимволы?
- Метасимволы - это символы, используемые при перечислении имен файлов текущего каталога.

9. Как экранировать метасимволы?
- Заключив его в одинарные кавычки.

10. Как создавать и запускать командные файлы?
- Сначала поместив последовательность команд в текстовый файл, затем файл будет выполнен с помощью команды *bash командный_файл [аргументы]*. Настоятельно рекомендуется изменить код безопасности пакетного файла после создания, чтобы сделать его доступным для вас при выполнении с помощью команды *chmod +x имя_файла.*

11. Как определяются функции в языке программирования bash?
- Для этой функции есть ключевое слово, за которым следует имя функции и список команд, заключенных в фигурные скобки.

12. Каким образом можно выяснить, является файл каталогом или обычным файлом?
- С помощью команды test.

13. Каково назначение команд set, typeset и unset?
- Команда set позволяет вам управлять определенными флагами и характеристиками, чтобы влиять на поведение ваших сценариев bash, typeset устанавливает атрибуты и значения для переменных и функций оболочки, а команда unset используется для удаления переменных во время выполнения программы.

14. Как передаются параметры в командные файлы?
15. Назовите специальные переменные языка bash и их назначение
– $* — отображается вся командная строка или параметры оболочки;
– $? — код завершения последней выполненной команды;
– $$ — уникальный идентификатор процесса, в рамках которого выполняется командный процессор;
– $! — номер процесса, в рамках которого выполняется последняя вызванная на выполнение в командном режиме команда;
– $- — значение флагов командного процессора;
– ${#*} — возвращает целое число — количество слов, которые были результатом $*;
– ${#name} — возвращает целое значение длины строки в переменной name;
– ${name[n]} — обращение к n-му элементу массива;
– ${name[*]} — перечисляет все элементы массива, разделённые пробелом;
– ${name[@]} — то же самое, но позволяет учитывать символы пробелы в самих переменных;
– ${name:-value} — если значение переменной name не определено, то оно будет заменено на указанное value;
– ${name:value} — проверяется факт существования переменной;
– ${name=value} — если name не определено, то ему присваивается значение value;
– ${name?value} — останавливает выполнение, если имя переменной не определено, и выводит value как сообщение об ошибке;
– ${name+value} — это выражение работает противоположно ${name-value}. Если переменная определена, то подставляется value;
– ${name#pattern} — представляет значение переменной name с удалённым самым коротким левым образцом (pattern);
– ${#name[*]} и ${#name[@]} — эти выражения возвращают количество элементов в массиве name.