

ФГБОУ ВО

Уфимский Государственный Авиационный Технический Университет

Кафедра АСУ

Отчет по

лабораторной работе №3

«СОЗДАНИЕ И ВЫПОЛНЕНИЕ КОМАНДНЫХ ФАЙЛОВ В СРЕДЕ ОС
LINUX»

по дисциплине «Операционные системы и сети»

Выполнил:
ст. гр. МКН-216
Биганяков А.А.

Проверил:
Старший преподаватель
Ямилева А.М.

Уфа 2022

Цель работы: Целью работы является изучение методов создания и выполнения командных файлов на языке bash - интерпретатора.

Ход работы:

```
1 if (test -d DIR)
2     then
3         echo "Такой каталог существует"
4     else
5         mkdir DIR
6 fi
7 >DIR/MyFile.txt
8 echo "Введите фамилию и имя"
9 read init
10 echo $init >> DIR/MyFile.txt
11 date >> MyFile.txt
```

```
sumashedsh1y@sumashedsh1y-VirtualBox:~$ sh 1.sh
Введите фамилию и имя
Biganiakov Arslan
sumashedsh1y@sumashedsh1y-VirtualBox:~$ sh 1.sh
Такой каталог существует
Введите фамилию и имя
Biganiakov Arslan
sumashedsh1y@sumashedsh1y-VirtualBox:~$ █
```

Задание 1. Создание каталога DIR с текстовым файлом включающим свою фамилию и имя

```
sumashedsh1y@sumashedsh1y-VirtualBox:~$ >2.sh
sumashedsh1y@sumashedsh1y-VirtualBox:~$ sh 2.sh
sumashedsh1y@sumashedsh1y-VirtualBox:~$ sh 2.sh
sumashedsh1y@sumashedsh1y-VirtualBox:~$ █
```

```
1 cd DIR
2 mv MyFile.txt Old_MyFile.txt
3 chmod 744 Old_MyFile.txt
```

cd – привязка к директории

Задание 2. Переименование файла в каталоге и установка к нему атрибутов

```

1 echo -n "Введите первое число: "
2 read l
3 echo -n "Введите второе число: "
4 read r
5 pr=`expr $l '*' $r`
6 sum=`expr $l + $r`
7 expr $pr / $sum

```

```

sumashedsh1y@sumashedsh1y-VirtualBox:~$ sh 3.sh
Введите первое число: 3
Введите второе число: 4
1
sumashedsh1y@sumashedsh1y-VirtualBox:~$ 

```

Задание 3. Вычисление омического сопротивления двух параллельно соединенных резисторов

```

1 echo -n "Введите число: "
2 read z
3 s=""
4 while [ $z -gt 0 ]
5 do
6     s=`expr $z % 8`$s
7     z=`expr $z / 8`
8 done
9 echo $s

```

```

sumashedsh1y@sumashedsh1y-VirtualBox:~$ sh 4.sh
Введите число: 12
14
sumashedsh1y@sumashedsh1y-VirtualBox:~$ sh 4.sh
Введите число: 100
144
sumashedsh1y@sumashedsh1y-VirtualBox:~$ 

```

Задание 4. Перевод заданного десятичного числа в восьмеричную форму
gt оператор сравнения – больше

```

1 for word in "lesson1" "lesson2" "lab 1" "lab 2" "lab 3" "lab 4" "lab 5"
2 do
3     echo -n $word
4     case $word in
5         "lab 1") echo -n " date: 2022-03-03";;
6         "lab 2") echo -n " date: 2022-03-06";;
7         "lab 3") echo -n " date: 2022-03-09";;
8         "lab 4") echo -n " date: 2022-03-12";;
9         "lab 5") echo -n " date: 2022-03-15";;
10    esac
11    echo
12 done

```

```
sumashedsh1y@sumashedsh1y-VirtualBox:~$ sh 5.sh
lesson1
lesson2
lab 1 date: 2022-03-03
lab 2 date: 2022-03-06
lab 3 date: 2022-03-09
lab 4 date: 2022-03-12
lab 5 date: 2022-03-15
sumashedsh1y@sumashedsh1y-VirtualBox:~$ █
```

Задание 5. Перебор сложных значений и вывод каждого значения с новой строки

```
1 echo -n "Что тебе нравится в УГАТУ?"
2 echo "Варианты ответа: "
3 for var in "Лекции" "Практики" "Лабораторные"
4 do
5     echo $var
6 done
7 echo ""
8 read otvet
9 echo $otvet > anketa
10 case $otvet in
11     Лекции) echo $otvet ", да слушать их одно удовольствие";;
12     Практики) echo $otvet ", согласен очень интересно решать задачки";;
13     Лабораторные) echo $otvet ", отличный выбор, шанс проверить знания на практике");;
14 esac
```

```
sumashedsh1y@sumashedsh1y-VirtualBox:~$ sh 6.sh
Что тебе нравится в УГАТУ? Варианты ответа:
Лекции
Практики
Лабораторные

Лабораторные
Лабораторные , отличный выбор, шанс проверить знания на практике)
sumashedsh1y@sumashedsh1y-VirtualBox:~$ █
```

Задание 6. Опросник

Вывод: В ходе данной лабораторной работы было изучено методы создания и выполнения командных файлов на языке bash — интерпретатора.

Ответы на контрольные вопросы:

1. **Какое назначение имеют shell - файлы?**

Выполнимыми могут быть не только файлы, полученные в результате компиляции и сборки, но и файлы, написанные на языке shell. Их выполнение происходит в режиме интерпретации с помощью shell-интерпретатора.

2. Как создать shell - файл и сделать его выполняемым?

Если вы подготовите в редакторе файл "f1", то это будет обычный текстовый файл, содержащий команду "echo", которая при выполнении выдает все написанное правее ее на экран. Можно сделать файл "f1" выполняемым с помощью команды "chmod 755 f1". Но его можно выполнить, вызвав явно команду (!) "sh" ("shell"): sh f1 или sh < f1

Файл можно выполнить и в текущем экземпляре "shell". Для этого существует специфическая команда "." (точка), т.е.. f1.

3. Какие типы переменных используются в shell - файлах?

Имя shell-переменной - это начинающаяся с буквы последовательность букв, цифр и подчеркиваний.

Значение shell-переменной - строка символов.

То, что в shell всего два типа данных: строка символов и текстовый файл, с одной стороны, позволяет легко вовлекать в программирование конечных пользователей, никогда ранее программированием не занимавшихся, а с другой стороны, вызывает некий внутренний протест у многих программистов, привыкших к существенно большему разнообразию и большей гибкости языковых средств.

Примеры типов переменных:

-date
-read, echo -n
-var echo

4. В чем заключается анализ цепочки символов?

Алфавит – это конечное множество I элементов, называемых символами. Цепочка или слово в алфавите I – это конечная последовательность элементов (символов) из алфавита I. Например, если алфавит языка состоит только из заглавных и строчных букв латинского алфавита, то любые последовательности этих букв являются цепочками (словами), в том числе цепочка может состоять из одного символа.

С цепочками (словами) могут быть проделаны действия, которые имеют следующие обозначения:

- - x^n . Цепочка символов x повторяется (x пишется без пробелов одна за другой) n раз. Например, $abba^2$ – это $abbaabba$.
- - x^k . Цепочка символов x записывается в обратной последовательности. Например, $portR$ – это $trop$.
- - xy . За цепочкой символов x без пробела помещается цепочка символов y .
- - x^* . Цепочка символов x в цикле может повторяться нуль и более раз.
- - x^+ . Цепочка символов x должна повторяться один и больше раз. В алгоритмическом языке Pascal это реализуется оператором repeat, а в языке Си – оператором цикла do while.
- - $|x|$. Определение длины цепочки символов x (количество символов в цепочке).
- - $\{\}$ или l , или e – обозначение пустой цепочки символов.
- - $[x]$. Так обозначается необязательная цепочка символов. Например, такая запись нужна для того, чтобы обозначить, что перед числом знак может быть, а может и отсутствовать.

Кроме алфавита и цепочки символов (слов), важным понятием является язык.

Язык в алфавите I – это произвольное множество цепочек (слов).

5. Какие встроенные команды используются в shell - файлах?

Командный интерпретатор в среде UNIX выполняет две основные функции:

представляет интерактивный интерфейс с пользователем, т.е. выдает приглашение, и обрабатывает вводимые пользователем команды;

обрабатывает и исполняет текстовые файлы, содержащие команды интерпретатора (командные файлы);

В последнем случае, операционная система позволяет рассматривать командные файлы как разновидность исполняемых файлов. Соответственно различают два режима работы интерпретатора: интерактивный и командный.

В среде UNIX (в отличие, скажем, от DOS) имеются несколько различных командных интерпретаторов. Перечислим наиболее популярные:

- $/bin/sh$ - Bourne shell. Исторически это первая командная оболочка, разработанная для первой версии ОС UNIX. В настоящее время эта оболочка является основной в версиях UNIX System V.

- `/bin/csh` - C-shell. Оболочка, синтаксис командного языка которой приближен к языку С. Является основной оболочкой для Берклеевской разновидности ОС UNIX.
- `/bin/ksh` - k-shell.
- `/bin/rsh` - Restricted shell. Представляет собой sh с ограниченными возможностями (прежде всего для защиты ОС от несанкционированных действий пользователя).

6. Как производится управление программами?

Когда пользователи входят в систему UNIX, они общаются с одним из интерпретаторов команд. В этой главе изучается интерпретатор команд Shell, sh. Этот интерпретатор является программой UNIX, которая поддерживает обладающий большими возможностями командный язык. Каждый вызов этого интерпретатора называется оболочкой Shell; каждая оболочка выполняет единственную функцию - считывает со стандартного устройства ввода команду и выполняет ее.

Поскольку Shell дает пользователю возможность общаться с операционной системой на языке высокого уровня, UNIX может выполнять задачи, недоступные менее сложным операционным системам. Команды, которые обычно должны писаться на традиционных языках программирования, здесь могут быть записаны в нескольких строчках процедуры Shell. В других операционных системах команды выполняются в прямой последовательности. В UNIX благодаря Shell команды могут:

- объединяться для образования новых команд;
- передавать позиционные параметры;
- добавляться или переименоваться пользователем;
- выполняться внутри циклов или по определенному условию;
- создаваться для локального выполнения без риска вступления в конфликт с командами других пользователей;
- выполняться в фоновом режиме.

Более того, команды могут перенаправлять ввод исходных данных для команды от одного источника к другому и перенаправлять вывод в файл, на терминал, принтер или другой команде. Это обеспечивает необходимую гибкость при выполнении конкретных задач.

7. Назовите операторы создания циклов.

Операторы создания циклов:

- For - оператор цикла с перечислением (for ... do ... done).
- While - оператор цикла с истинным условием (while ... do ... done).
- Case - оператор цикла с ложным условием (until ... do ... done).