

Презентация по лабораторной работе №10

Программирование в командном процессоре ОС UNIX. Командные файлы

Лушин А.А.

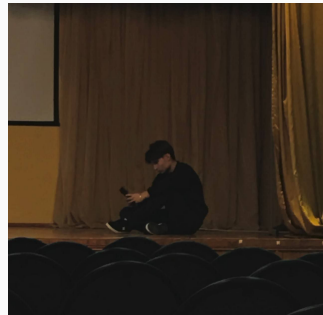
18 февраля 2005

Российский университет дружбы народов, Москва, Россия

Факультет Физико-математических и естественных наук

Информация

- Лушин Артем Андреевич
- бакалавр направления компьютерная и информационные науки
- ученик кафедры информационных технологий
- Российский университет дружбы народов
- lusin5745@gmail.com



Вводная часть

- Изучить основы программирования в оболочке ОС UNIX/Linux. Научиться писать небольшие командные файлы

Ход работы

Первый скрипт при запуске создает свою резервную копию и переносит ее в другой каталог. Так же файл архивирует свою копию в zip архив. Для большего удобства, для начала мы архивируем файл, а затем переносим архив уже в другой каталог. Для архивации используем команды, которые нашли в справочнике.

A terminal window with a dark blue background. At the top, a teal progress bar shows the path `/home/aalushin/script1`, the progress `48/48`, and the completion status `100%`. Below the bar, two lines of white text represent the commands being executed:

```
zip script1.zip script1  
mv script1.zip ~/backup
```

Path	Progress	Status
/home/aalushin/script1	48/48	100%

Второй скрипт обрабатывает числа, которые мы ввели в командной строке. Конкретно в этой программе, результатом работы будет последовательный вывод чисел, которые мы ввели. Можно вводить более 10 чисел в командную строку и все они выведутся в ходе выполнения скрипта.

```
script2      [----] 4 L:[ 1+ 2 3/ 3] *(31 / 31b) <EOF>      [*][X]
for A in $*
do echo $A
done
```


Третий скрипт

Третий скрипт - это аналог команды ls. При его запуске выводятся данные о каталоге и какую-либо информацию о возможностях доступа к файлу или каталогу. Аналогом команды является ls -l, но данный скрипт написан без использования команды ls или dir.

```
scripts [-M--] 4 L:[ 1+11 12/ 13] *(226 / 2270
for A in *
do if test -d $A
    then echo $A: is a directory
    else echo -n $A: is a file and
<----->if test -w $A
<----->then echo writeable
<----->elif test -r $A
<----->then echo readable
<----->else echo neither readable nor writeable
<----->fi
    fi
```

Четвертый скрипт

Этот скрипт очень похож на функцию поиска. При запуске файла, у нас запрашивается название каталога, в котором мы будем искать файлы. Далее запрашивается расширение, которое нам надо найти. После того, как мы ввели расширение файла и директорию, в которой мы ищем эти файлы, у нас выводится число - это количество файлов с таким расширением. Можно доработать программу, чтобы она выводила помимо количества файлов, еще и их названия.

```
script4      [----] 4 L:[ 1+ 4  5/  5] *(68 / 120b) 0032 0x020  [*][X]  
echo "Input directory"  
read dir  
echo "Input format"  
read format  
find ${dir} -maxdepth 1 -name "*${type}" -type f | wc -l
```

Результаты

Контрольные вопросы

1. Объясните понятие командной оболочки. Приведите примеры командных оболочек. Чем они отличаются?
2. Что такое POSIX?
3. Как определяются переменные и массивы в языке программирования bash?
4. Каково назначение операторов `let` и `read`?
5. Какие арифметические операции можно применять в языке программирования bash?
6. Что означает операция `(())`?
7. Какие стандартные имена переменных Вам известны?
8. Что такое метасимволы?
9. Как экранировать метасимволы?

Я изучил основы программирования в оболочке ОС UNIX/Linux. Научился писать командные файлы.

Если однажды сдать ся, это войдет в привычку.