Лабораторная работа №10

Программирование в командной процессоре ОС UNIX. Командные файлы

Никифоров Г.

2

Российский университет дружбы народов, Москва, Россия, планета Земля, Солнечная система, галактика Млечный путь

Информация

Докладчик

- Никифоров Георгий Сергеевич
- студентов
- студент
- студентович
- Математический институт им. Никольского
- https://github.com/gsnikiforov

Вводная часть

Актуальность

• Умение программировать в ОС Unix бывает полезным навыком при работе с компьютером.

Ход работы

Цель работы

Цель работы— познакомиться с основами программирования в командной оболочке ОС Unix.

Выполнение лабораторной работы

Необходимо выполнить задания:

1. Написать скрипт, который при запуске будет делать резервную копию самого себя (то есть файла, в котором содержится его исходный код) в другую директорию backup в вашем домашнем каталоге. При этом файл должен архивироваться одним из архиваторов на выбор zip, bzip2 или tar. Способ использования команд архивации необходимо узнать, изучив справку.

Данная задача была решена в файле vadim.sh:

```
#!/bin/bash
vadim=$0
outDir=~/backup/
outFile=${outDir}${vadim:2}.tar
mkdir -p $outDir
tar -cf $outFile $vadim
echo Created backup file $outFile successfully!
```

2. Написать пример командного файла, обрабатывающего любое произвольное число аргументов командной строки, в том числе превышающее десять. Например, скрипт может последовательно распечатывать значения всех переданных аргументов.

for naramotor in \$+

```
artem.sh:
#!/bin/bash
let N=0
function PrintArgument () { # two arguments have to be passed
    echo $1. $2
echo This program prints out all the arguments you have just passed:
PrintArgument $N $0
let N++
```

```
treska.sh:
#!/bin/bash
dir=$1
if [ ! $dir ]
then
    # If dir specified is empty
    dir=./
fi
cd $dir
for file in $(echo *)
do
    if [[ -r $file ]]
```

thon ocho -n n

```
whatever sh
#!/bin/bash
# First parameter to be passed is an extention and
# second is a path like this
# ./whatever.sh .tex ~/Documents
N=$#
if (($N != 2))
then
    echo There must be 2 parameters, not $#, go fak yourself
    exit
fi
```

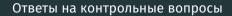


1. Объясните понятие командной оболочки. Приведите примеры командных оболочек. Чем они отличаются?

Ответ: командная оболочка позволяет исполнять команды.

2. Что такое POSIX?

Om bem: POSIX — набор стандартов описания интерфейсов взаимодействия операционной системы и прикладных программ.



3. Как определяются переменные и массивы в языке программирования bash?

Ответ: xthtp hfdyj.



4. Каково назначение операторов let и read?

Omвem: let позволяет выполнять арифметические операции при задании переменных, read считывает стандартный поток вывода.

Ответы	на контрольные вопросы
5.	Какие арифметические операции можно применять в языке про

граммирования bash?

Ответ: стандартные.

6. Что означает операция (())?

Ответ: (()) вычисляют логические условные выражения.

7. Какие стандартные имена переменных Вам известны?

Omвет: PATH, ENV, TERM.

8. Что такое метасимволы?

Ответ: специальные символы.

9. Как экранировать метасимволы?

Ответ: как угодно, но можно через .

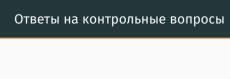
10. Как создавать и запускать командные файлы?

Omsem: для создания файла применить команду touch <file> && chmod +x <file>. Для запуска ввести ./<file>.



11. Как определяются функции в языке программирования bash?

Omeem: при помощи ключевого слова function.



12. Каким образом можно выяснить, является файл каталогом или обычным файлом?

Omeem: ls -l выведет дополнительную информацию.

13. Каково назначение команд set, typeset и unset?

Ответ: таково.

14. Как передаются параметры в командные файлы?

Ответ: через пробел при запуске программы.

15. Назовите специальные переменные языка bash и их назначение.

Ответ: см. вопрос 7.

Результаты

Заключение

В ходе выполнения лабораторной работы были изучены основы программирования в командной оболочке OS Unix. Цель работы была достигнута.