

Лабораторная работа 12

Операционные системы

Богданюк Анна Васильевна

Содержание

1	Цель работы	1
2	Задание.....	1
3	Теоретическое введение	1
4	Выполнение лабораторной работы	2
5	Выводы.....	4

1 Цель работы

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

2 Задание

1. Написать скрипт, который при запуске будет делать резервную копию самого себя
2. Написать пример командного файла, обрабатывающего любое произвольное число аргументов командной строки
3. Написать командный файл — аналог команды ls
4. Написать командный файл, который получает в качестве аргумента командной строки формат файла

3 Теоретическое введение

Командный процессор (командная оболочка, интерпретатор команд shell) — это программа, позволяющая пользователю взаимодействовать с операционной системой компьютера. В операционных системах типа UNIX/Linux наиболее часто используются следующие реализации командных оболочек: – оболочка Борна (Bourne shell или sh) — стандартная командная оболочка UNIX/Linux, содержащая базовый, но при этом полный набор функций; – C-оболочка (или csh) — надстройка на оболочкой Борна, использующая C-подобный синтаксис команд с возможностью сохранения истории выполнения команд; – оболочка Корна (или ksh) — напоминает

оболочку C, но операторы управления программой совместимы с операторами оболочки Борна; – BASH — сокращение от Bourne Again Shell (опять оболочка Борна), в основе своей совмещает свойства оболочек C и Корна (разработка компании Free Software Foundation). POSIX (Portable Operating System Interface for Computer Environments) — набор стандартов описания интерфейсов взаимодействия операционной системы и прикладных программ. Стандарты POSIX разработаны комитетом IEEE (Institute of Electrical and Electronics Engineers) для обеспечения совместимости различных UNIX/Linux-подобных операционных систем и переносимости прикладных программ на уровне исходного кода. POSIX-совместимые оболочки разработаны на базе оболочки Корна. Рассмотрим основные элементы программирования в оболочке bash. В других оболочках большинство команд будет совпадать с описанными ниже.

4 Выполнение лабораторной работы

Для начала создаю файл p1.sh и делаю его исполняемым. Выполняю его (рис. 1).

```
avbogdanyuk@Bogdanyuk:~$ touch p1.sh
avbogdanyuk@Bogdanyuk:~$ nano p1.sh
avbogdanyuk@Bogdanyuk:~$ mkdir backup
avbogdanyuk@Bogdanyuk:~$ bash p1.sh
p1.sh
```

Создание файла и вывод

Пишу скрипт, который при запуске будет делать резервную копию самого себя в другую директорию backup в вашем домашнем каталоге. При этом файл должен архивироваться одним из архиваторов на выбор zip, bzip2 или tar.(рис. 2).

```
GNU nano 6.2 p1.sh
#!/bin/bash
tar -cvf ~/backup/backup.tar p1.sh_
```

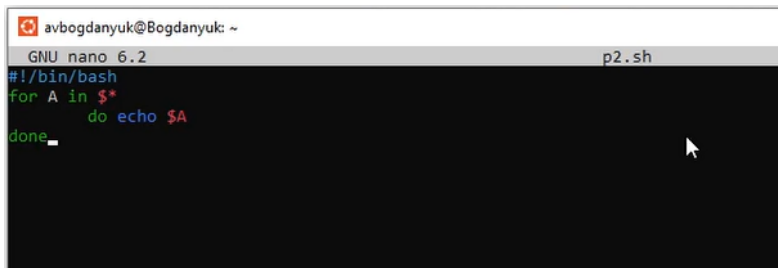
Скрипт

Для начала создаю файл p2.sh и делаю его исполняемым. Выполняю его (рис. 3).

```
avbogdanyuk@Bogdanyuk:~$ ls
CHANGELOG.md  bin                lab9_alg.ipynb
LICENSE       go1.22.0.linux-amd64.tar.gz  newdir
README.md     hugo_extended_0.123.7_linux-amd64.deb  p1.sh
backup        lab8_alg.ipynb      p2.sh
avbogdanyuk@Bogdanyuk:~$ chmod +x p2.sh
avbogdanyuk@Bogdanyuk:~$ nano p2.sh
avbogdanyuk@Bogdanyuk:~$ bash p2.sh 1221 toxic britney hahah 1232
1221
toxic
britney
hahah
1232
avbogdanyuk@Bogdanyuk:~$
```

Создание файла и вывод

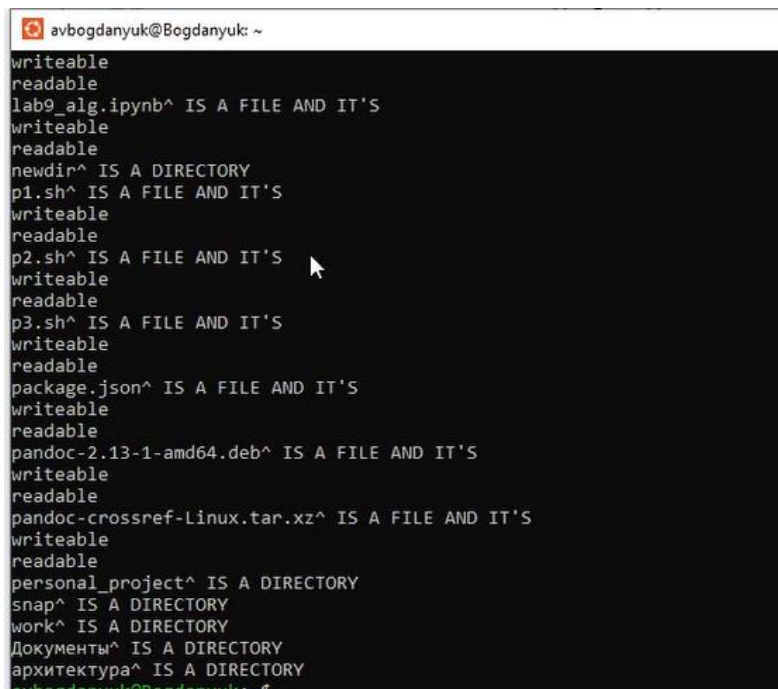
Пишу пример командного файла, обрабатывающего любое произвольное число аргументов командной строки, в том числе превышающее десять. (рис. 4).



```
avbogdanyuk@Bogdanyuk: ~  
GNU nano 6.2 p2.sh  
#!/bin/bash  
for A in $*  
do echo $A  
done
```

Скрипт

Для начала создаю файл p3.sh и делаю его исполняемым. Выполняю его (рис. 5).



```
avbogdanyuk@Bogdanyuk: ~  
writeable  
readable  
lab9_alg.ipynb^ IS A FILE AND IT'S  
writeable  
readable  
newdir^ IS A DIRECTORY  
p1.sh^ IS A FILE AND IT'S  
writeable  
readable  
p2.sh^ IS A FILE AND IT'S  
writeable  
readable  
p3.sh^ IS A FILE AND IT'S  
writeable  
readable  
package.json^ IS A FILE AND IT'S  
writeable  
readable  
pandoc-2.13-1-amd64.deb^ IS A FILE AND IT'S  
writeable  
readable  
pandoc-crossref-Linux.tar.xz^ IS A FILE AND IT'S  
writeable  
readable  
personal_project^ IS A DIRECTORY  
snap^ IS A DIRECTORY  
work^ IS A DIRECTORY  
Документы^ IS A DIRECTORY  
архитектура^ IS A DIRECTORY  
avbogdanyuk@Bogdanyuk: ~$
```

Создание файла и вывод

Пишу командный файл — аналог команды ls (без использования самой этой команды и команды dir). Требуется, чтобы он выдавал информацию о нужном каталоге и выводил информацию о возможностях доступа к файлам этого каталога. (рис. 6).

```
avbogdanyuk@Bogdanyuk: ~
GNU nano 6.2 p3.sh
for A in *
do
    if test -d "$A"
    then
        echo "$A^ IS A DIRECTORY"
    else
        echo "$A^ IS A FILE AND IT'S"
        if test -w $A
        then
            echo "writeable"
            if test -r $A
            then
                echo "readable"
            else
                echo "neither readable nor writeable"
            fi
        fi
    fi
done
```

Скрипт

Для начала создаю файл p4.sh и делаю его исполняемым. Выполняю его (рис. 7).

```
avbogdanyuk@Bogdanyuk:~$ touch p4.sh
avbogdanyuk@Bogdanyuk:~$ chmod +x p4.sh
avbogdanyuk@Bogdanyuk:~$ nano p4.sh
avbogdanyuk@Bogdanyuk:~$ bash p4.sh
```

Создание файла и вывод

Пишу командный файл, который получает в качестве аргумента командной строки формат файла (.txt, .doc, .jpg, .pdf и т.д.) и вычисляет количество таких файлов в указанной директории. Путь к директории также передаётся в виде аргумента командной строки. (рис. 8).

```
avbogdanyuk@Bogdanyuk: ~
GNU nano 6.2 p4.sh *
#!/bin/bash
format = ""
directory = ""
echo "Write down a file format"
read format
echo "Write down name of directory"
read directory
find "${directory}" -name "*.${format}" -type f | wc -l
ls
```

Скрипт

5 Выводы

В ходе выполнения лабораторной работы были изучены основы программирования в оболочке ОС UNIX/Linux. Научилась писать небольшие командные файлы.