Отчет

Лабораторная работа №10

Павлова Варвара Юрьевна НПМбд-02-21

Содержание

Список иллюстраций

Список таблиц

Цель работы

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

Задание

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 архивациинеобходимо узнать, изучив справку.
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.
3. Написать 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.
4. Написать 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.

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.

Выполнение лабораторной работы

1. Изучаю информацию о командах архивации zip(рис. [-@fig:002]), bzip2 (рис. [-@fig:003]) и tar (рис. [-@fig:004]) с помощью команды man.(рис. [-@fig:001])

Рис. 0.1: вызов справки

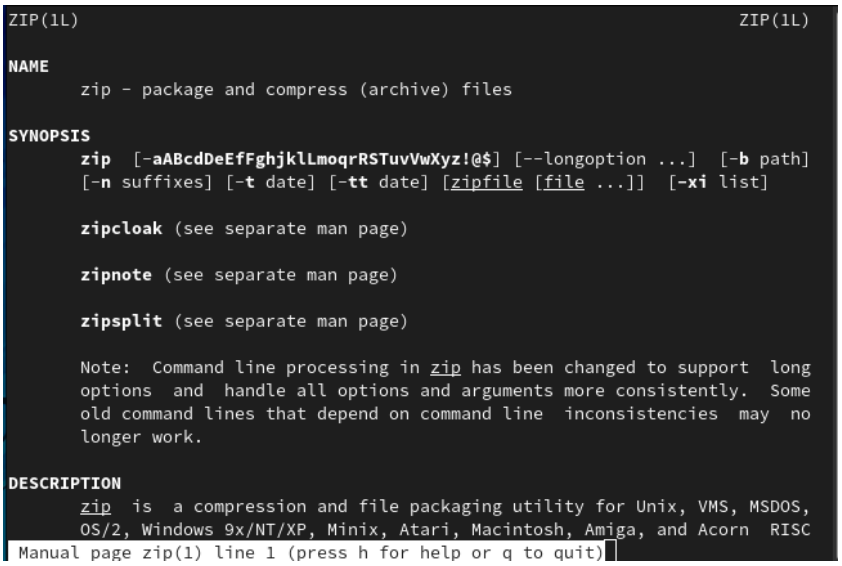

Рис. 0.2: man zip

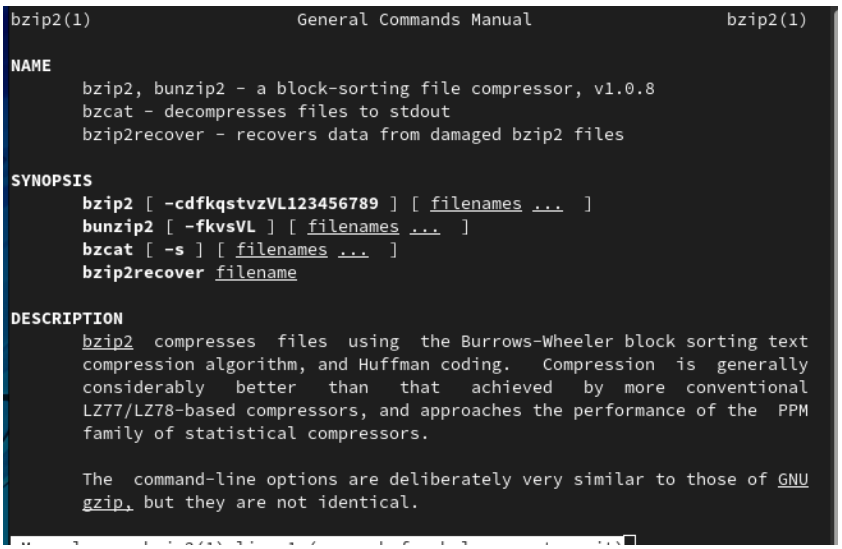

Рис. 0.3: man bzip2

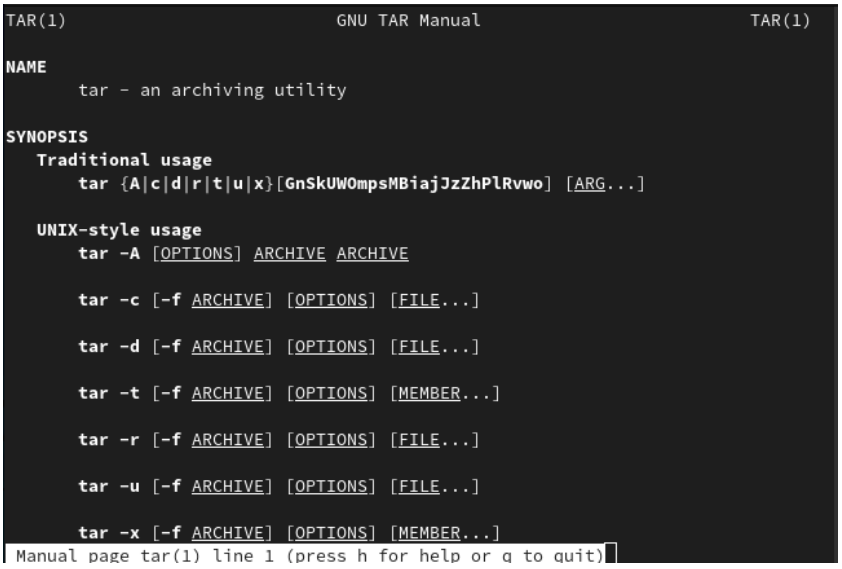

Рис. 0.4: man tar

2. Создаю первый исполняемый файл backup.sh и открываю редактор emacs. (рис. [-@fig:005])

Рис. 0.5: создание файла

3. Пишу скрипт, который при запуске будет делать резервную копию самого себя (то есть файла, в котором содержится его исходный код) в другую директорию backup в домашнем каталоге. При этом файл архивируется архиватором bzip2. (рис. [-@fig:006])

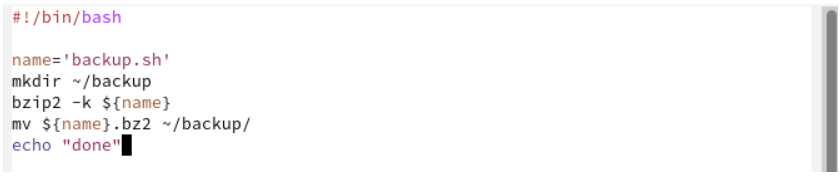

Рис. 0.6: написание скрипта

4. Добавляю право на выполнение файла и проверяю его работу. (рис. [-@fig:007])

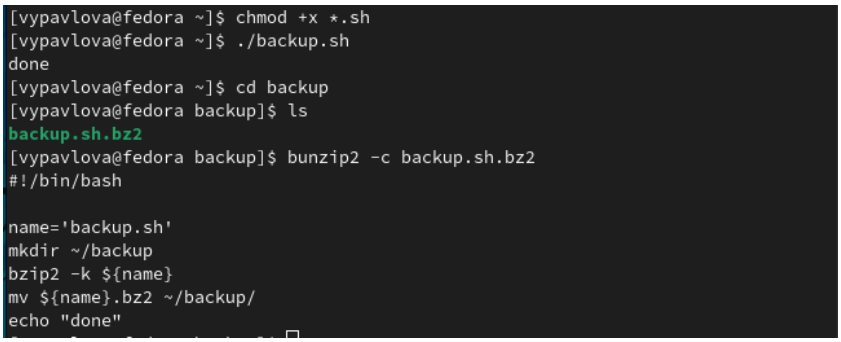

Рис. 0.7: проверка первого файла

5. Создаю второй исполняемый файл second.sh и открываю редактор emacs. (рис. [-@fig:008])

Рис. 0.8: создание файла

6. Пишу командный файл, обрабатывающий любое произвольное число аргументов командной строки, в том числе превышающее десять. Скрипт последовательно распечатывает значения всех переданных аргументов. (рис. [-@fig:009])

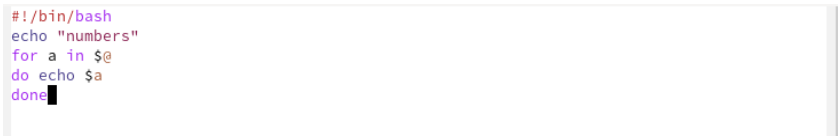

Рис. 0.9: написание скрипта

7. Добавляю право на выполнение файла и проверяю его работу. (рис. [-@fig:010])

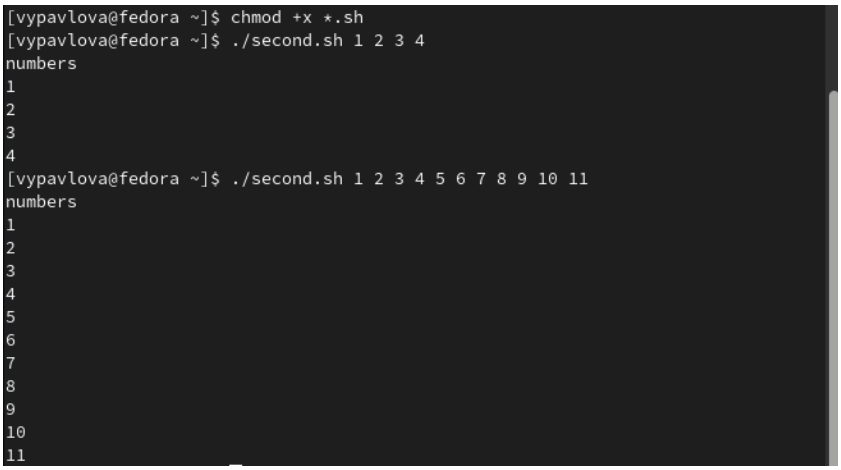

Рис. 0.10: проверка второго файла

8. Создаю третий исполняемый файл third.sh и открываю редактор emacs. (рис. [-@fig:011])

Рис. 0.11: создание файла

9. Пишу командный файл — аналог команды ls (без использования самой этой команды и команды dir). Скрипт выдает информацию о нужном каталоге и

выводит информацию о возможностях доступа к файлам этого каталога. (рис. [-@fig:012])

```
#!/bin/bash
a="$1"
for i in ${a}/*
do
    echo "$i"

    if test -f $i
    then echo "regular file"
    fi

    if test -d $i
    then echo "directory"
    fi

    if test -r $i
    then echo "reading is allowed"
    fi

    if test -w $i
    then echo "writing is allowed"
    fi
done
```

Рис. 0.12: написание скрипта

10. Добавляю право на выполнение файла и проверяю его работу. (рис. [-@fig:013])

```
[vypavlova@fedora ~]$ chmod +x *.sh
[vypavlova@fedora ~]$ ./third.sh ~
/home/vypavlova/backup
directory
reading is allowed
writing is allowed
/home/vypavlova/backup.sh
regular file
reading is allowed
writing is allowed
/home/vypavlova/backup.sh~
regular file
reading is allowed
writing is allowed
/home/vypavlova/bin
directory
reading is allowed
writing is allowed
/home/vypavlova/Desktop
directory
reading is allowed
writing is allowed
/home/vypavlova/Documents
```

Рис. 0.13: проверка третьего файла

11. Создаю четвертый исполняемый файл fourth.sh и открываю редактор emacs. (рис. [-@fig:014])

Рис. 0.14: создание файла

12. 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. [-@fig:015])

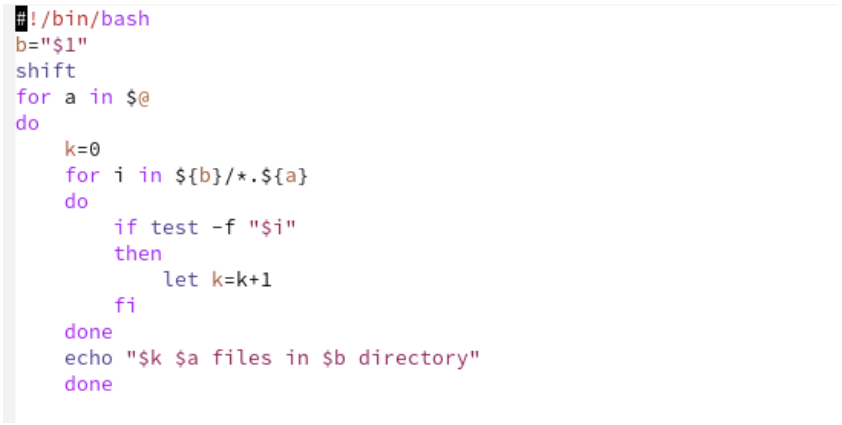

Рис. 0.15: написание скрипта

13. Добавляю право на выполнение файла и проверяю его работу. (рис. [-@fig:016])

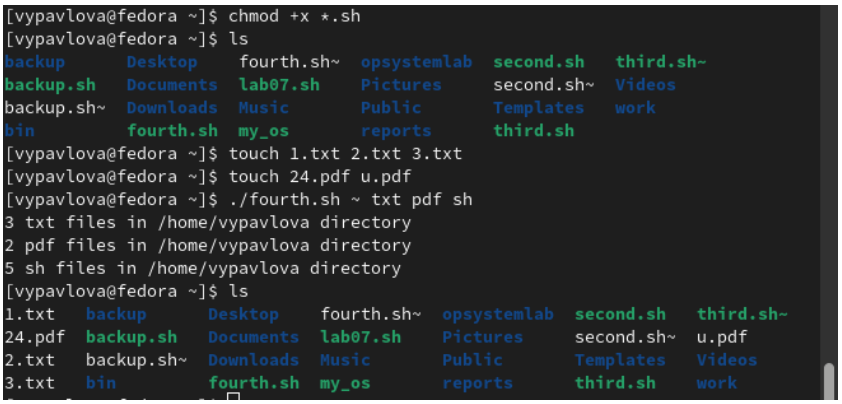

Рис. 0.16: проверка четвертого файла

Выводы

Выполняя данную лабораторную работу я изучила основы программирования в оболочке ОС UNIX/Linux и научилась писать небольшие командные файлы.

Список литературы