

Презентация лабораторной работы №10

Презентация лабораторной работы №10

Цель

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

Задание

- 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.
- 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.
- 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.
- 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.

Выполнение лабораторной работы

С помощью команды man узнаем информацию про zip, bzip2, tar

```
ZIP(1L) ZIP(1L)

NAME
    zip - package and compress (archive) files

SYNOPSIS
    zip [-aABcdDeEfFghjklLmoqrRSTuvVwXyz!@$] [--longoption ...] [-b path]
    [-n suffixes] [-t date] [-tt date] [zipfile [file ...]] [-xi list]

    zipcloak (see separate man page)

    zipnote (see separate man page)

    zipsplit (see separate man page)

    Note: Command line processing in zip has been changed to support long
    options and handle all options and arguments more consistently. Some
    old command lines that depend on command line inconsistencies may no
    longer work.

DESCRIPTION
    zip is a compression and file packaging utility for Unix, VMS, MSDOS,
    OS/2, Windows 9x/NT/XP, Minix, Atari, Macintosh, Amiga, and Acorn RISC
    Manual page zip(1) line 1 (press h for help or q to quit)
```

zip

```
bzip2(1) General Commands Manual bzip2(1)

NAME
    bzip2, bunzip2 - a block-sorting file compressor, v1.0.8
    bzip2recover - recovers data from damaged bzip2 files

SYNOPSIS
    bzip2 [-cdfkqstvvVL123456789] [filenames ...]
    bunzip2 [-fkvsVL] [filenames ...]
    bzip2recover [-s] [filenames ...]

DESCRIPTION
    bzip2 compresses files using the Burrows-Wheeler block sorting text
    compression algorithm, and Huffman coding. Compression is generally
    considerably better than that achieved by more conventional
    LZ77/LZ78-based compressors, and approaches the performance of the PPM
    family of statistical compressors.

    The command-line options are deliberately very similar to those of GNU
    gzip, but they are not identical.

    Manual page bzip2(1) line 1 (press h for help or q to quit)
```

bzip2

```
TAR(1) GNU TAR Manual TAR(1)

NAME
    tar - an archiving utility

SYNOPSIS
    Traditional usage
        tar {A|c|d|r|t|u|x}[GnSkUW0mpsMBiajJzZhPlRvwo] [ARG...]

    UNIX-style usage
        tar -A [OPTIONS] ARCHIVE ARCHIVE

        tar -c [-f ARCHIVE] [OPTIONS] [FILE...]

        tar -d [-f ARCHIVE] [OPTIONS] [FILE...]

        tar -t [-f ARCHIVE] [OPTIONS] [MEMBER...]

        tar -r [-f ARCHIVE] [OPTIONS] [FILE...]

        tar -u [-f ARCHIVE] [OPTIONS] [FILE...]

        tar -x [-f ARCHIVE] [OPTIONS] [MEMBER...]

Manual page tar(1) line 1 (press h for help or q to quit)
```

tar

Создаем файл `lab01first.sh`, в котором будем писать скрипт, который при запуске будет делать резервную копию самого себя в другую директорию `backup` в нашем домашнем каталоге.

```
[lkkhanina@fedora ~]$ vi lab10first.sh
[lkkhanina@fedora ~]$ cat lab10first.sh
#!/bin/bash
name='lab10first.sh'
mkdir ~/backup
bzip2 -k ${name}
mv ${name}.bz2 ~/backup/
echo "Успех"
[lkkhanina@fedora ~]$ chmod +x lab10first.sh
```

Первый скрипт

```
[lkkhanina@fedora ~]$ ./lab10first.sh
Успех
[lkkhanina@fedora ~]$ ls ~/backup
lab10first.sh.bz2
[lkkhanina@fedora ~]$
```

Запускаем первый скрипт

```
[lkkhanina@fedora ~]$ cd ~/backup
[lkkhanina@fedora backup]$ bunzip2 -c lab10first.sh.bz2
#!/bin/bash
name='lab10first.sh'
mkdir ~/backup
bzip2 -k ${name}
mv ${name}.bz2 ~/backup/
echo "Успех"
[lkkhanina@fedora backup]$
```

Содержание архивированного файла

Далее создаем файл **lab10second**, в котором будет второй скрипт, обрабатывающий любое произвольное число аргументов командной строки, в том числе превышающее десять. Например, скрипт может последовательно распечатывать значения всех переданных аргументов

```
[lkkhanina@fedora backup]$ touch lab10second.sh
[lkkhanina@fedora backup]$ vi lab10second.sh
[lkkhanina@fedora backup]$ cat lab10second.sh
#!/bin/bash
echo "Печатаем аргументы:"
for arg in $@
do echo $arg
done
[lkkhanina@fedora backup]$ chmod +x lab10second.sh
```

Второй скрипт

```
[lkkhanina@fedora backup]$ chmod +x lab10second.sh
[lkkhanina@fedora backup]$ ./lab10second.sh 1 2 3 4 5
Печатаем аргументы:
1
2
3
4
5
[lkkhanina@fedora backup]$
```

Запускаем второй скрипт

Далее создаем файл lab10third, в котором будет третий скрипт, 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

```
[lkkhanina@fedora ~]$ vi lab10third.sh
[lkkhanina@fedora ~]$ chmod +x lab10third.sh
[lkkhanina@fedora ~]$ cat lab10third.sh
#!/bin/bash
a="$1"
for arg in ${a}
do
    echo "$arg"
    if test -f $arg
    then echo "File"
    fi

    if test -d $arg
    then echo "Directory"
    fi

    if test -r $arg
    then echo "Can read"
    fi

    if test -w $arg
    then echo "Can write"
    fi
done
[lkkhanina@fedora ~]$
```

Третий скрипт

```
[lkkhanina@fedora ~]$ ./lab10third.sh ~
/home/lkkhanina
Directory
Can read
Can write
[lkkhanina@fedora ~]$
```

Запускаем третий скрипт

Далее создаем файл lab10fourth, в котором будет четвертый скрипт, 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

```
[lkkhanina@fedora ~]$ vi lab10fourth.sh
[lkkhanina@fedora ~]$ chmod +x lab10fourth.sh
[lkkhanina@fedora ~]$ cat lab10fourth.sh
#!/bin/bash
a="$1"
shift
for arg in $@
do
    k=0
    for i in ${a}/*.${arg}
    do
        if test -f "$i"
        then let k=k+1
        fi
    done
    echo "total $k files that contain in directory $a with format $arg"
done
[lkkhanina@fedora ~]$
```

Четвертый скрипт

```
done
[lkkhanina@fedora ~]$ ./lab10fourth.sh ~ txt
total 4 files that contain in directory /home/lkkhanina with format txt
[lkkhanina@fedora ~]$
```

Запускаем четвертый скрипт

Выводы

Я научилась писать небольшие командный файлы для решения различных вопросов