

Программирование в командном процессоре ОС UNIX. Командные файлы.

Лабораторная работа №10

Старцева А. С.

15 апреля 2023

Российский университет дружбы народов, Москва, Россия

Информация

- Старцева Алина Сергеевна
- студент 1 курса, группа НММбд-03-22
- Российский университет дружбы народов



Вводная часть

- Командный процессор ОС UNIX
- Командные файлы

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

- Ознакомиться с теоретическим материалом.
- Выполнить упражнения.
- Ответить на контрольные вопросы.

Выполнение лабораторной работы №10

Изучение теории

```
astarceva@asstarceva:~$ man zip
ZIP(1)                                ZIP(1)

NAME
  zip - package and compress (archive) files

SYNOPSIS
  zip [-aABcdDefFghjklmnpqRSTuvVwXyz!0$] [--longoption ...] [-b path]
      [-n suffixes] [-t date] [-tt date] [zipfile [file ...]] [-x! 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)
```

```
astarceva@asstarceva:~$ man bzip2
bzip2(1)                               General Commands Manual      bzip2(1)

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

SYNOPSIS
  bzip2 [-cdfkqtzVL123456789] [filenames ...]
  bunzip2 [-fkvsVL] [filenames ...]
  bzcata [-s] [filenames ...]
  bzip2recover filename

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)
```

```
astarceva@asstarceva:~$ man tar
TAR(1)                                GNU TAR Manual              TAR(1)

NAME
  tar - an archiving utility

SYNOPSIS
  Traditional usage
    tar [A|c|d|r|t|u|x|GnSkvWompsvMBiaJzZhPlRvwO] [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...]
```

Первая программа

```
[astarceva@asstarceva ~]$ touch 10_1.sh
```

```
Открыть 10_1.sh
~/
#!/bin/bash

name="10_1.sh"
mkdir ~/backup/
bzip2 -k ${name}
mv ${name}.bz2 ~/backup/
echo "end"
```

```
[astarceva@asstarceva ~]$ chmod +x *.sh
[astarceva@asstarceva ~]$ ls
10_1.sh  lab09.sh-  Видео  Загрузки  Музыка  'Рабочий стол'
[astarceva@asstarceva ~]$ ./10_1.sh
end
[astarceva@asstarceva ~]$ ls
10_1.sh  lab09.sh-  Видео  Загрузки  Музыка  'Рабочий стол'
[astarceva@asstarceva ~]$ ls backup
10_1.sh.bz2
```

Вторая программа

```
[astarceva@asstarceva ~]$ touch 10_2.sh
```

```
Открыть ▾  10_2.sh    
#!/bin/bash  
echo  
for i in $(  
do echo $@  
done
```

```
[astarceva@asstarceva ~]$ chmod +x *.sh  
[astarceva@asstarceva ~]$ ls  
10_1.sh  lab09.sh  документы  музыка  шаблоны  
10_2.sh  none      загрузки  фотографии  рабочий стол  
backup  Desktop  Избранное  Рабочий стол  
[astarceva@asstarceva ~]$ ./10_2.sh 3 4 5 6  
3 4 5 6  
3 4 5 6  
3 4 5 6  
3 4 5 6  
[astarceva@asstarceva ~]$ ./10_2.sh 3 4 5 6 7 8 9 0 0 0 0 0 0 0 0  
3 4 5 6 7 8 9 0 0 0 0 0 0 0 0  
3 4 5 6 7 8 9 0 0 0 0 0 0 0 0  
3 4 5 6 7 8 9 0 0 0 0 0 0 0 0  
3 4 5 6 7 8 9 0 0 0 0 0 0 0 0  
3 4 5 6 7 8 9 0 0 0 0 0 0 0 0  
3 4 5 6 7 8 9 0 0 0 0 0 0 0 0  
3 4 5 6 7 8 9 0 0 0 0 0 0 0 0  
3 4 5 6 7 8 9 0 0 0 0 0 0 0 0  
3 4 5 6 7 8 9 0 0 0 0 0 0 0 0  
3 4 5 6 7 8 9 0 0 0 0 0 0 0 0  
3 4 5 6 7 8 9 0 0 0 0 0 0 0 0  
3 4 5 6 7 8 9 0 0 0 0 0 0 0 0  
3 4 5 6 7 8 9 0 0 0 0 0 0 0 0  
3 4 5 6 7 8 9 0 0 0 0 0 0 0 0  
3 4 5 6 7 8 9 0 0 0 0 0 0 0 0  
3 4 5 6 7 8 9 0 0 0 0 0 0 0 0  
[astarceva@asstarceva ~]$
```

Третья программа

```
[astarceva@asstarceva ~]$ touch 10_3.sh  
[astarceva@asstarceva ~]$
```

```
Открыть ▾ • 10_3.sh  
~  
  
a = "$1"  
for i in $(a);  
do  
    echo "$i"  
  
    if test -f $i  
    then echo "файл"  
    fi  
  
    if test -d $i  
    then echo "каталог"  
    fi  
  
    if test -r $i  
    then echo "чтение разрешено"  
    fi  
  
    if test -w $i  
    then echo "запись разрешена"  
    fi  
  
    if test -x $i  
    then echo "выполнение разрешено"  
    fi  
done
```

```
[astarceva@asstarceva ~]$ chmod +x *.sh  
[astarceva@asstarceva ~]$ ls  
10_1.sh  backup  video  Изображения  'Рабочий стол'  
10_2.sh  lab09.sh-  документы  Музыка  Видео  
10_3.sh  test  загрузки  Общедоступные  
[astarceva@asstarceva ~]$ ./10_3.sh  
$1  
каталог  
чтение разрешено  
выполнение разрешено  
$1  
каталог  
чтение разрешено  
выполнение разрешено  
$1  
каталог  
чтение разрешено  
выполнение разрешено  
$1  
каталог  
чтение разрешено  
выполнение разрешено  
$1  
каталог  
чтение разрешено  
выполнение разрешено  
$1  
каталог  
чтение разрешено  
выполнение разрешено  
$1  
каталог
```

Четвёртая программа

```
[astarceva@asstarceva ~]$ touch 10_4.sh
```

```
Открыть ▾ 10_4.sh
~/
#!/bin/bash

b="$1"

shift

for j in $(
do
    k=0
    for i in $(b)/*.$(j)
    do
        if test -f "$i"
        then
            let k=k+1
        fi
    done
    echo "$k файлов содержится в директории $b с расширением $j"
done
```

```
[astarceva@asstarceva ~]$ chmod +x *.sh
[astarceva@asstarceva ~]$ ls
10_1.sh  10_4.sh  work      Загрузки  Общедоступные
10_2.sh  backup   Видео     Изображения  'Рабочий стол'
10_3.sh  lab09.sh  Документы  Музыка      Шаблоны
[astarceva@asstarceva ~]$ ./10_4.sh = sh
4 файлов содержится в директории /home/astarceva с расширением sh
```

1. 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 Сподобный 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).
2. 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

12. Команду «set» можно использовать для вывода списка переменных окружения. В системах Ubuntu и Debian команда «set» также выведет список функций командной оболочки после списка переменных командной оболочки. Поэтому для ознакомления со всеми элементами списка переменных окружения при работе с данными системами рекомендуется использовать команду «set | more». Команда «typeset» предназначена для наложения ограничений на переменные. Команду «unset» следует использовать для удаления переменной из окружения командной оболочки.
13. При вызове командного файла на выполнение параметры ему могут быть переданы точно таким же образом, как и выполняемой программе. С точки зрения командного файла эти параметры являются позиционными. Символ \$ является метасимволом командного процессора. Он используется, в частности, для ссылки на параметры, точнее, для получения их значений в командном файле. В командный файл можно передать до девяти параметров. При использовании где-либо в командном файле комбинации символов \$i, где $0 < i < 10$, вместо неё будет осуществлена подстановка значения параметра с порядковым номером i, т. е. аргумента командного файла с порядковым номером i. Использование комбинации символов \$0 приводит к подстановке вместо неё имени данного командного файла.

Результаты

В ходе выполнения лабораторной работы ознакомились с операционной системой Linux. Получили практические навыки работы с редактором Emacs.