

Лабораторная работа №10. Программирование в командном процессоре ОС UNIX. Командные файлы

Парфенова Елизавета Евгеньевна

RUDN University, Moscow, Russian Federation

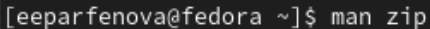
Изучить основы программирования в оболочке ОС UNIX/Linux.
Научиться писать небольшие командные файлы.

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.
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.

Выполнение работы

Ознакомившись с теорией, можно приступить к первому скрипту. В нем было необходимо использовать архив `zip`, поэтому с помощью команды ***man zip*** читаем справку о нем. (рис. 1)

A terminal window with a dark background. The prompt is [eeeparfenova@fedora ~]\$ and the command man zip has been entered.

```
[eeeparfenova@fedora ~]$ man zip
```

Figure 1: Справка о zip

Писать скрипты можно в любом редакторе. Создаем новый файл *script1*. После приступаем непосредственно к коду. Заархивировать файл можно с помощью *zip*, а перенести его с помощью *mv*. (рис. 2)

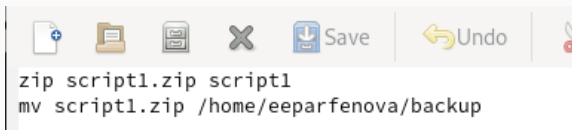


Figure 2: Первый скрипт

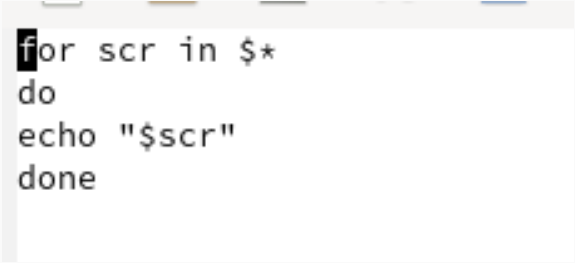
После возвращаемся в консоль, создаем каталог `backup` с помощью **`mkdir`** и делаем файл исполняемым с помощью команды **`chmod +x script1`**. Вызываем скрипт с помощью **`./script1`**. Он сработал успешно. (рис. 3)

```
[eeparfenova@fedora ~]$ mkdir backup
[eeparfenova@fedora ~]$ ls
backup  snap  Бэкеп  Загрузки  Музыка  'Рабочий стол'
script1  work  Документы  Изображения  Общедоступные  Временные
[eeparfenova@fedora ~]$ chmod +x script1
[eeparfenova@fedora ~]$ ./script1
adding: script1 (deflated 29%)
[eeparfenova@fedora ~]$ cd backup/
bash: cd: command not found...
Similar commands are:
'ss'
'cc'
[eeparfenova@fedora ~]$ cd backup/
[eeparfenova@fedora backup]$ ls
script1.zip
```

Figure 3: Работа первого скрипта

Второй скрипт

Переходим ко второму заданию. В нем было необходимо написать командный файл, который бы обрабатывал значения аргументов и печтал их. Создаем файл script2 и осуществляем это с помощью цикла for. (рис. 4)

A screenshot of a terminal window with a light gray background. The terminal shows a shell script with four lines of code: 'for scr in \$*', 'do', 'echo "\$scr"', and 'done'. The first line is highlighted with a black selection box.

```
for scr in $*  
do  
echo "$scr"  
done
```

Figure 4: Второй скрипт

Работа второго скрипта

Снова возвращаемся в терминал и делаем файл исполняемым той же командой, заменив только название. После проверяем скрипт (`./script2`), вписав вначале три аргумента, а затем свыше 10 (это требовалось проверить в задании) (рис. 5)

```
[eeeparfenova@fedora ~]$ ./script2 1 2 3
1
2
3
[eeeparfenova@fedora ~]$ ./script2 1 2 3 4 5 6 7 8 9 10 11
1
2
3
4
5
6
7
8
9
10
11
```

Figure 5: Работа второго скрипта

Переходим к третьему заданию. Создаем файл script3. В первой части кода мы проверяем директория это или файл, а во втором выводим права доступа с помощью циклов. (рис. 6)

```
for A in *
do if test -d $A
  then echo $A: is a directory
  else echo -n $A: is a file and
    if test -w $A
    then echo writeable
    elif test -r $A
    then echo readable
    else echo neither readable nor writeable
    fi
  fi
done
```

Figure 6: Третий скрипт

Сделав файл исполняемым, проверяем его, вызвав с помощью *./script3*. Скрипт сработал успешно и вывел информацию по домашнему каталогу.(рис. 7)

```
[eeparfenova@fedora ~]$ ./script3
backup: is a directory
script1: is a file andwriteable
script2: is a file andwriteable
script2~: is a file andwriteable
script3: is a file andwriteable
snap: is a directory
work: is a directory
Видео: is a directory
Документы: is a directory
Загрузки: is a directory
Изображения: is a directory
Музыка: is a directory
Общедоступные: is a directory
```

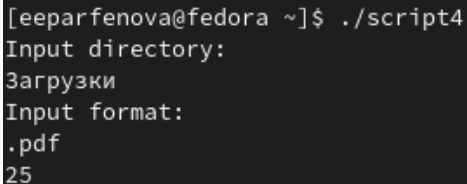
Figure 7: Работа третьего скрипта

Последним заданием было вычислить количество файлов определенного формата в определенной директории. Вначале создаем файл script4. В коде вначале просим пользователя ввести директорию и формат, а затем находим нужное с помощью команды *find*. (рис. 8)

```
echo "Input directory: "  
read directory  
echo "Input format: "  
read format  
  
find ${directory} -maxdepth 1 -name "*${format}" -type f | wc -l
```

Figure 8: Четвертый скрипт

Далее делаем файл исполняемым и вызываем его, используя *./script4*.
Проверяем файл, поискав в “Загрузках” файлы формата pdf. Скрипт сработал успешно. (рис. 9)

A terminal window with a dark background. The prompt is [eeparfenova@fedora ~]\$ and the command ./script4 has been entered. The script prompts for an input directory, and 'Загрузки' (Downloads) is entered. It then prompts for an input format, and '.pdf' is entered. Finally, the number '25' is displayed.

```
[eeparfenova@fedora ~]$ ./script4
Input directory:
Загрузки
Input format:
.pdf
25
```

Figure 9: Работа четвертого скрипта

Мы изучили основы программирования в оболочке ОС UNIX/Linux и научились писать небольшие командные файлы.