

Лабораторная работа №10

Программирование в командном процессоре ОС UNIX. Командные файлы

Рогожина Н.А.

13 апреля 2023

Российский университет дружбы народов, Москва, Россия

Информация

- Рогожина Надежда Александровна
- Студентка 1го курса, НКАбд-02-22
- Компьютерные и информационные науки
- Российский университет дружбы народов
- Github

Вводная часть

- Командный процессор (командная оболочка, интерпретатор команд shell) — это программа, позволяющая пользователю взаимодействовать с операционной системой компьютера.

- Командный процессор

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

Содержание лабораторной работы

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 С, но операторы управления программой совместимы с операторами оболочки Борна;
- BASH — сокращение от Bourne Again Shell (опять оболочка Борна), в основе своей совмещает свойства оболочек С и Корна (разработка компании Free Software Foundation).

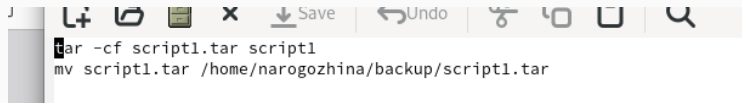
Задание

Задачи:

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.

Задачи:

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.

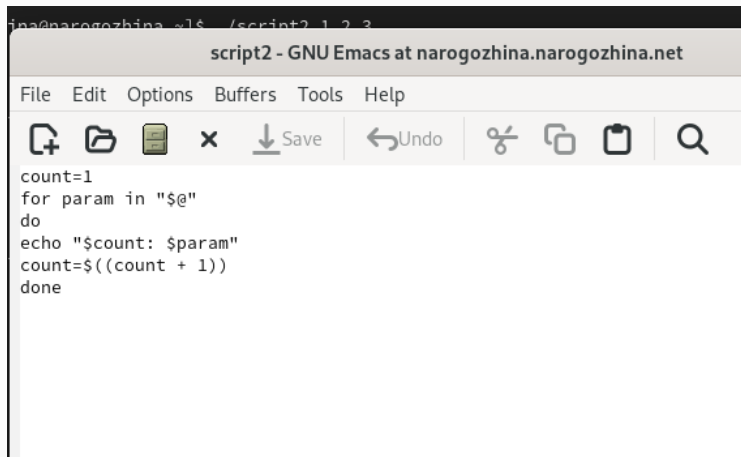
A screenshot of a terminal window. The window has a title bar with standard Linux window controls (minimize, maximize, close) and a menu bar with options like 'File', 'Edit', 'View', 'Window', and 'Help'. Below the menu bar is a toolbar with icons for file operations (new, open, save, print, delete) and editing (undo, redo, find, replace). The terminal area shows two lines of text: 'tar -cf script1.tar script1' and 'mv script1.tar /home/narogozhina/backup/script1.tar'. The cursor is at the end of the second line.

```
tar -cf script1.tar script1  
mv script1.tar /home/narogozhina/backup/script1.tar
```

Рис. 1: Скрипт №1

```
[narogozhina@narogozhina ~]$ chmod +x script1
[narogozhina@narogozhina ~]$ ./script1
[narogozhina@narogozhina ~]$ cd backup
[narogozhina@narogozhina backup]$ ls
script1.tar
[narogozhina@narogozhina backup]$ tar -xf script1.tar
[narogozhina@narogozhina backup]$ ls
script1 script1.tar
```

Рис. 2: Команды



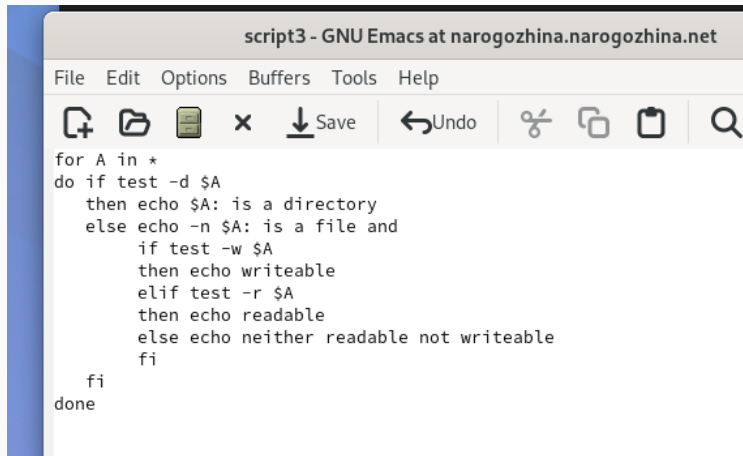
The image shows a terminal window with a GNU Emacs editor. The title bar reads "script2 - GNU Emacs at narogozhina.narogozhina.net". The menu bar includes "File", "Edit", "Options", "Buffers", "Tools", and "Help". The toolbar contains icons for opening a file, saving, undo, redo, copy, paste, and search. The text area displays a shell script with the following content:

```
ina@narogozhina ~$ ./script2 1 2 3  
count=1  
for param in "$@"  
do  
echo "$count: $param"  
count=$((count + 1))  
done
```

Рис. 3: Скрипт №2


```
[narogozhina@narogozhina ~]$ ./script2 1 2 3
1: 1
2: 2
3: 3
[narogozhina@narogozhina ~]$ ./script2 10 20 30 40 50
1: 10
2: 20
3: 30
4: 40
5: 50
[narogozhina@narogozhina ~]$ ./script2 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15
1: 1
2: 2
3: 3
4: 4
5: 5
6: 6
7: 7
8: 8
9: 9
10: 10
11: 11
12: 12
13: 13
14: 14
15: 15
[narogozhina@narogozhina ~]$
```

Рис. 4: Скрипт №2

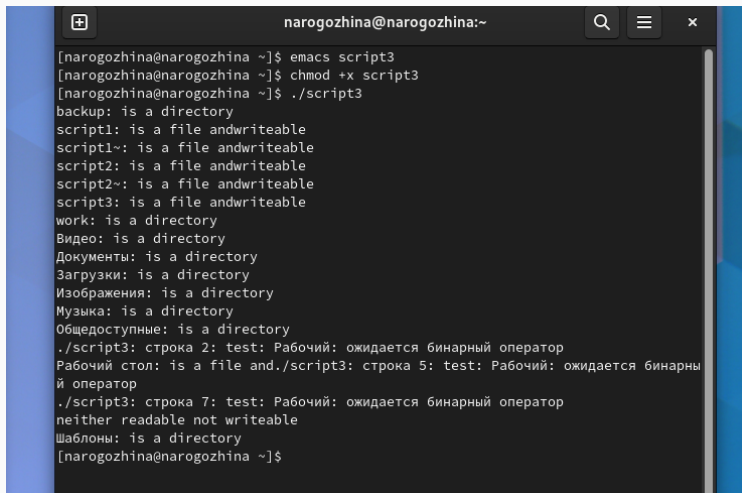


```
script3 - GNU Emacs at narogozhina.narogozhina.net
File Edit Options Buffers Tools Help
[Icons: Open, Save, Undo, Redo, Search, etc.]
Save Undo

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 not writeable
    fi
  fi
done
```

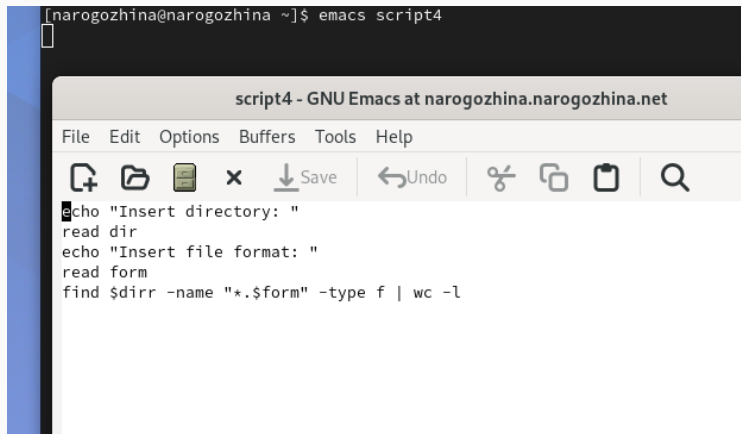
Рис. 5: Скрипт №3

Пример вывода



```
narogozhina@narogozhina:~  
[narogozhina@narogozhina ~]$ emacs script3  
[narogozhina@narogozhina ~]$ chmod +x script3  
[narogozhina@narogozhina ~]$ ./script3  
backup: is a directory  
script1: is a file andwriteable  
script1~: is a file andwriteable  
script2: is a file andwriteable  
script2~: is a file andwriteable  
script3: is a file andwriteable  
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  
./script3: строка 2: test: Рабочий: ожидается бинарный оператор  
Рабочий стол: is a file and./script3: строка 5: test: Рабочий: ожидается бинар  
ый оператор  
./script3: строка 7: test: Рабочий: ожидается бинарный оператор  
neither readable not writeable  
Шаблоны: is a directory  
[narogozhina@narogozhina ~]$
```

Рис. 6: Скрипт №3



```
[narogozhina@narogozhina ~]$ emacs script4
```

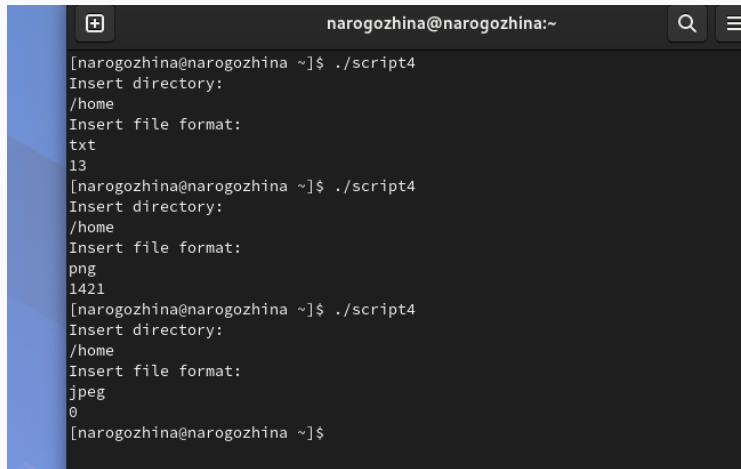
script4 - GNU Emacs at narogozhina.narogozhina.net

File Edit Options Buffers Tools Help

Icons: New, Open, Save, Close, Save, Undo, Cut, Copy, Paste, Find

```
echo "Insert directory: "  
read dir  
echo "Insert file format: "  
read form  
find $dirr -name ".*$form" -type f | wc -l
```

Рис. 7: Скрипт №4



```
narogozhina@narogozhina:~  
[narogozhina@narogozhina ~]$ ./script4  
Insert directory:  
/home  
Insert file format:  
txt  
13  
[narogozhina@narogozhina ~]$ ./script4  
Insert directory:  
/home  
Insert file format:  
png  
1421  
[narogozhina@narogozhina ~]$ ./script4  
Insert directory:  
/home  
Insert file format:  
jpeg  
0  
[narogozhina@narogozhina ~]$
```

Рис. 8: Скрипт №4

Выводы

В ходе лабораторной работы мы познакомились с командными файлами Linux, а также научились писать небольшие командные файлы сами.