

Лабораторная работа №10

Назирова М.Т.

8 апреля 2023

Российский университет дружбы народов, Москва, Россия

Информация

- Назирова Малика Темржоновна
- студентка группы НКабд-04-22
- Факультет физико-математических и естественных наук
- Российский университет дружбы народов
- 1032225197@pfur.ru
- https://github.com/sweetliiikk/study_2022-2023_os-intro.git



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 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.

Теоретическое введение

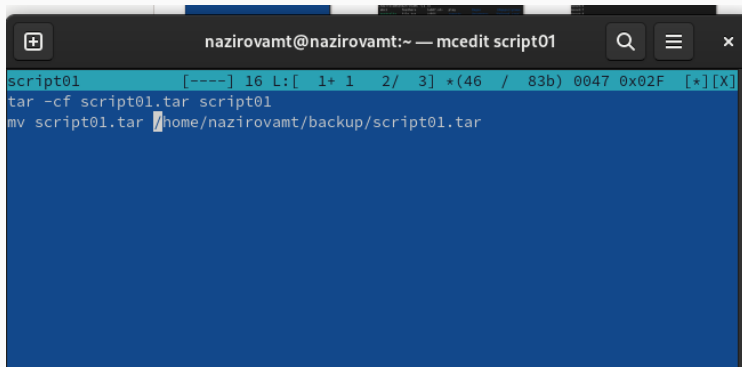
Командный процессор (командная оболочка, интерпретатор команд `shell`) — это программа, позволяющая пользователю взаимодействовать с операционной системой компьютера. В операционных системах типа UNIX/Linux наиболее часто используются следующие реализации командных оболочек: – оболочка Борна (Bourne shell или `sh`) — стандартная командная оболочка UNIX/Linux, содержащая базовый, но при этом полный набор функций; – С-оболочка (или `csh`) — надстройка на оболочке Борна, использующая С-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 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

Выполнение лабораторной работы

1. Написать скрипт (рис. (fig:001?)), 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 (рис. (fig:002?)).



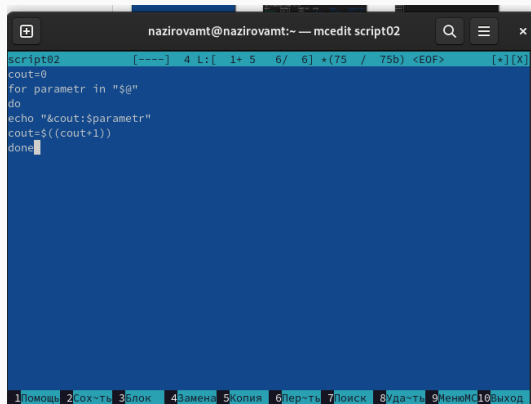
```
nazirovamt@nazirovamt:~ — mcedit script01
script01  [----] 16 L:[ 1+ 1  2/ 3] *(46 / 83b) 0047 0x02F [*] [X]
tar -cf script01.tar script01
mv script01.tar /home/nazirovamt/backup/script01.tar
```

Выполнение лабораторной работы

```
nazirovamt@nazirovamt:~  
tar: script01.tar: Функция open завершилась с ошибкой: Нет такого файла или каталога  
tar: Error is not recoverable: exiting now  
mv: не удалось выполнить stat для 'script01.tar': Нет такого файла или каталога  
[nazirovamt@nazirovamt ~]$ mcedit script01  
  
[nazirovamt@nazirovamt ~]$ chmod +x script01  
[nazirovamt@nazirovamt ~]$ ls  
abcl      feathers  lab07.sh~  play      Видео      Общедоступные  
australia file.txt  LABSS      reports   Документы  'Рабочий стол'  
backup    games     may        script01  Загрузки   Шаблоны  
bin       '#LAB#'   monthly    ski.places  Изображения  
conf.txt  '#lab07.sh#' my_os      work      Музыка  
[nazirovamt@nazirovamt ~]$ mcedit script01  
  
[nazirovamt@nazirovamt ~]$ chmod +x script01  
[nazirovamt@nazirovamt ~]$ ls  
abcl      feathers  lab07.sh~  play      Видео      Общедоступные  
australia file.txt  LABSS      reports   Документы  'Рабочий стол'  
backup    games     may        script01  Загрузки   Шаблоны  
bin       '#LAB#'   monthly    ski.places  Изображения  
conf.txt  '#lab07.sh#' my_os      work      Музыка  
[nazirovamt@nazirovamt ~]$ ./script01  
[nazirovamt@nazirovamt ~]$
```

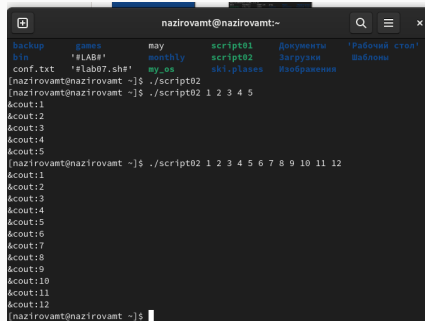
Рис. 2: Запуск скрипта

2. Написать пример командного файла (рис. (fig:003?)), обрабатывающего любое произвольное число аргументов командной строки, в том числе превышающее десять. Например, скрипт может последовательно распечатывать значения всех переданных аргументов(рис. (fig:004?)).



```
script02  [-----]  4 L: [ 1+ 5  6/  6] *(75 / 75b) <EOF>  [*][X]
cout=0
for parametr in "$@"
do
echo "&cout:$parametr"
cout=$((cout+1))
done
```

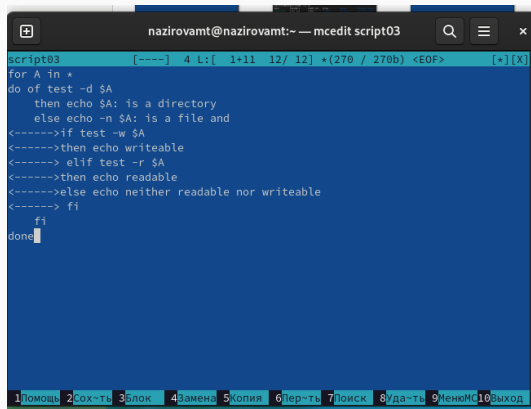
1Помощь 2Сохранить 3Блок 4Замена 5Копия 6Перейти 7Поиск 8Удалить 9Меню 10Выход

A terminal window titled 'nazirovamt@nazirovamt:~' with search, menu, and close icons. It shows a file manager view at the top with folders like 'backup', 'games', 'may', 'script01', 'Документы', and 'Рабочий стол'. Below, it shows the execution of './script02' with arguments '1 2 3 4 5' and then '1 2 3 4 5 6 7 8 9 10 11 12', resulting in output lines from '&cout:1' to '&cout:12'.

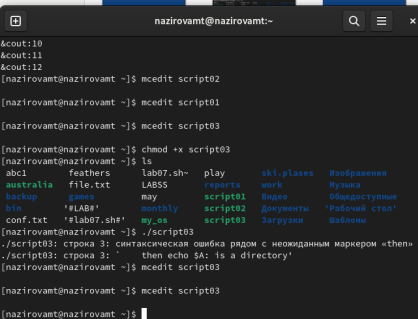
```
nazirovamt@nazirovamt:~  
backup    games      may        script01   Документы 'Рабочий стол'  
bin        '#LAB#'    monthly    script02   Загрузки   Шаблоны  
conf.txt   '#lab07.sh#' my_os      skl.plases Изображения  
[nazirovamt@nazirovamt ~]$ ./script02  
[nazirovamt@nazirovamt ~]$ ./script02 1 2 3 4 5  
&cout:1  
&cout:2  
&cout:3  
&cout:4  
&cout:5  
[nazirovamt@nazirovamt ~]$ ./script02 1 2 3 4 5 6 7 8 9 10 11 12  
&cout:1  
&cout:2  
&cout:3  
&cout:4  
&cout:5  
&cout:6  
&cout:7  
&cout:8  
&cout:9  
&cout:10  
&cout:11  
&cout:12  
[nazirovamt@nazirovamt ~]$
```

Рис. 4: Запуск скрипта

3. Написать командный файл — аналог команды `ls` (рис. (fig:005?))(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 (рис. (fig:006?)).



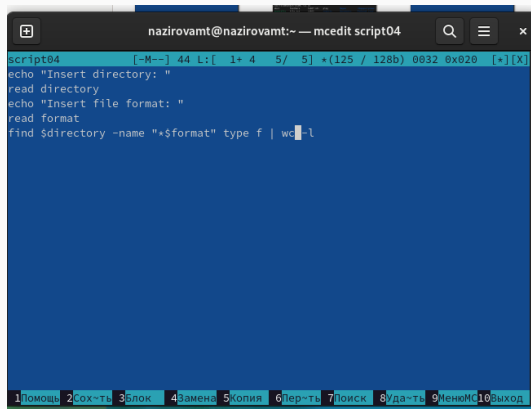
```
script03      [----]  4 L: [ 1+11 12/ 12] *(270 / 270b) <EOF>      [*][X]
for A in *
do of test -d $A
    then echo $A: is a directory
    else echo -n $A: is a file and
<-----> if test -w $A
<-----> then echo writeable
<-----> elif test -r $A
<-----> then echo readable
<-----> else echo neither readable nor writeable
<-----> fi
    fi
done
```



```
nazirovamt@nazirovamt:~  
&cout:10  
&cout:11  
&cout:12  
[nazirovamt@nazirovamt ~]$ mcedit script02  
[nazirovamt@nazirovamt ~]$ mcedit script01  
[nazirovamt@nazirovamt ~]$ mcedit script03  
[nazirovamt@nazirovamt ~]$ chmod +x script03  
[nazirovamt@nazirovamt ~]$ ls  
abc1      feathers  lab07.sh~  play      ski.places  Изображения  
australia file.txt  LAB55      reports   work        Музыка  
backup    games    may        script01  Видео       Общедоступные  
bin       '#LAB#'  monthly    script02  Документы  'Рабочий стол'  
conf.txt  '#lab07.sh~' my_os      script03  Загрузки   Шаблоны  
[nazirovamt@nazirovamt ~]$ ./script03  
./script03: строка 3: синтаксическая ошибка рядом с неожиданным маркером «then»  
./script03: строка 3: '      then echo $A: is a directory'  
[nazirovamt@nazirovamt ~]$ mcedit script03  
[nazirovamt@nazirovamt ~]$ mcedit script03  
[nazirovamt@nazirovamt ~]$
```

Рис. 6: Запуск скрипта

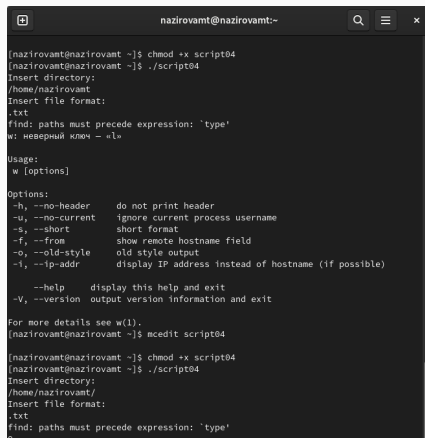
4. Написать командный файл (рис. (fig:007?)), который получает в качестве аргумента командной строки формат файла (.txt, .doc, .jpg, .pdf и т.д.) и вычисляет количество таких файлов в указанной директории. Путь к директории также передаётся в виде аргумента командной строки (рис. (fig:008?)).



```
script04 [-M--] 44 L: [ 1+ 4 5/ 5] *(125 / 128b) 0032 0x020 [*][X]
echo "Insert directory: "
read directory
echo "Insert file format: "
read format
find $directory -name "*$format" type f | wc -l
```

1Помощь 2Сох-ть 3Блок 4Замена 5Копия 6Пер-ть 7Поиск 8Уда-ть 9Меню 10Выход

Выполнение лабораторной работы



```
nazirovamt@nazirovamt:~  
[nazirovamt@nazirovamt ~]$ chmod +x script04  
[nazirovamt@nazirovamt ~]$ ./script04  
Insert directory:  
/home/nazirovamt  
Insert file format:  
.txt  
find: paths must precede expression: `type'  
w: неверный ключ - «l»  
  
Usage:  
w [options]  
  
Options:  
-h, --no-header      do not print header  
-u, --no-current     ignore current process username  
-s, --short          short format  
-f, --from           show remote hostname field  
-o, --old-style      old style output  
-i, --ip-addr        display IP address instead of hostname (if possible)  
  
--help              display this help and exit  
-V, --version        output version information and exit  
  
For more details see w(1).  
[nazirovamt@nazirovamt ~]$ mcedit script04  
  
[nazirovamt@nazirovamt ~]$ chmod +x script04  
[nazirovamt@nazirovamt ~]$ ./script04  
Insert directory:  
/home/nazirovamt/  
Insert file format:  
.txt  
find: paths must precede expression: `type'
```

Рис. 8: Запуск скрипта

Выводы

Изучили основы программирования в оболочке ОС UNIX/Linux. Научились писать небольшие командные файлы. :::