

# Лабораторная работа № 10

---

Алади Принц Ч.

14 Апрель 2023

Российский университет дружбы народов, Москва, Россия

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.

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:

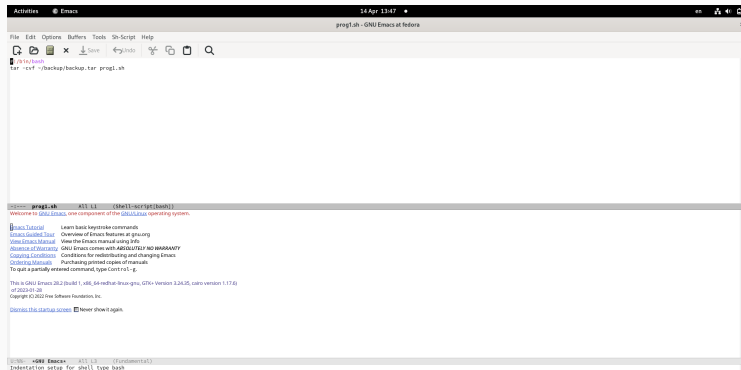
- оболочка Борна (Bourne shell или sh) — стандартная командная оболочка UNIX/Linux, содержащая базовый, но при этом полный набор функций;
- C-оболочка (или csh) — надстройка на оболочкой Борна, использующая C-подобный синтаксис команд с возможностью сохранения истории выполнения команд;
- оболочка Корна (или ksh) — напоминает оболочку 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).

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.

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. рис. ((fig:001?; fig:002?; fig:003?))



# Выполнение лабораторной работы



The screenshot shows the Emacs editor interface. The title bar indicates the file is 'prog1.sh - GNU Emacs at fedora'. The menu bar includes File, Edit, Options, Buffers, Tools, Sh-Script, and Help. The toolbar contains icons for opening files, saving, undo, redo, and search. The main text area contains the following code:

```
#!/bin/bash
tar -xvf ~/backup/backup.tar prog1.sh
```

Below the code, the Emacs status bar shows 'prog1.sh' and '(shell-script) bash'. A scroll bar is visible on the right side of the text area.

===== prog1.sh All L3 (shell-script) bash

Welcome to [GNU Emacs](#), one component of the [GNU/Linux](#) operating system.

[Emacs Tutorial](#) Learn basic keystroke commands  
 [Emacs Guide for TeX](#) Overview of Emacs features at gnu.org  
 [View Emacs Manual](#) View the Emacs manual using Info  
 [About Emacs](#) GNU Emacs comes with [ABSOLUTE TOP SECRET](#)  
 [Copyright Conditions](#) Conditions for redistributing and changing Emacs  
 [Ordering Manuals](#) Purchasing printed copies of manuals  
To quit a partially entered command, type `Ctrl-G`.

This is GNU Emacs 26.2 (build 1, x86\_64-redhat-linux-gnu, GTK+ Version 3.24.25, cairo version 1.17.4)  
of 2024-01-08  
Copyright (C) 2022 Free Software Foundation, Inc.

[Display this startup screen](#) ☒ Never show it again.

===== GNU Emacs All L3 (fundamental)

Indentation setup for shell type bash

Рис. 1: Текст первой программы

# Выполнение лабораторной работы

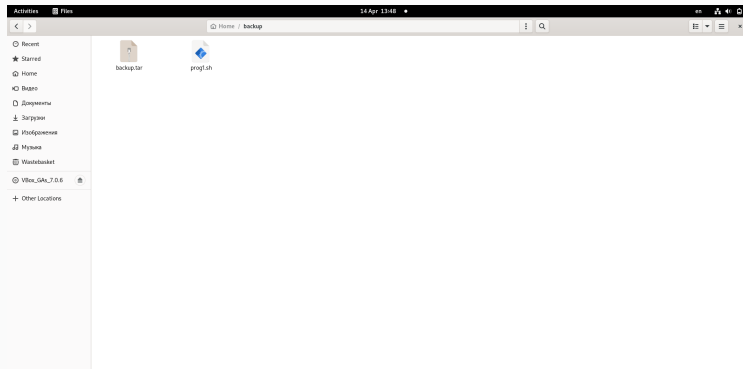
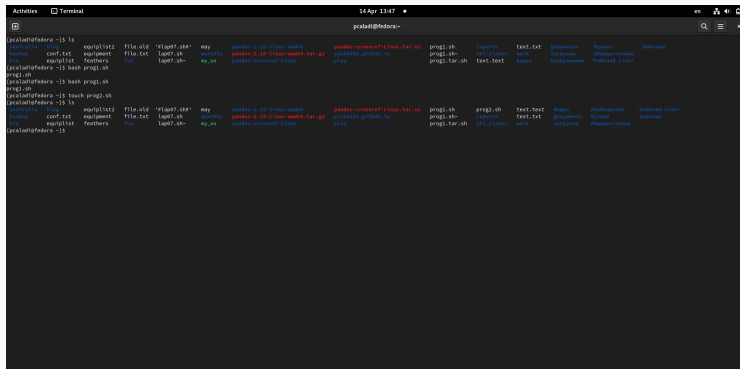


Рис. 2: Проверка работы программы

# Выполнение лабораторной работы



A terminal window titled 'pcaladi@fedora' with a date and time of '14 Apr 13:47'. The terminal shows a series of commands and their outputs. The first command is 'ls', which lists files in the current directory. The second command is 'touch prog2.sh', which creates a new file. The third command is 'ls', which lists the files again, showing the new file has been created. The terminal output is as follows:

```
pcaladi@fedora ~$ ls
anaconda  files  equiplist2  file.sld  'flag07.sh'  may  pandas-2.9.1-linux-64bit  pandas-crossref-linux.tar.xz  prog1.sh  reports  text.txt  xorgx11  xorgx11-devel  xorgx11-fonts
conf.txt  equiplist  file.txt  lap07.sh  monthly  pandas-crossref-linux  play  prog1.sh  reports  text.txt  xorgx11  xorgx11-devel  xorgx11-fonts
$ touch prog2.sh
pcaladi@fedora ~$ ls
anaconda  files  equiplist2  file.sld  'flag07.sh'  may  pandas-2.9.1-linux-64bit  pandas-crossref-linux.tar.xz  prog1.sh  prog2.sh  reports  text.txt  xorgx11  xorgx11-devel  xorgx11-fonts
conf.txt  equiplist  file.txt  lap07.sh  monthly  pandas-crossref-linux  play  prog1.sh  prog2.sh  reports  text.txt  xorgx11  xorgx11-devel  xorgx11-fonts
$
```

Рис. 3: Создание файла для второй программы, проверка содержимого домашнего каталога

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.

((fig:004?; fig:005?))

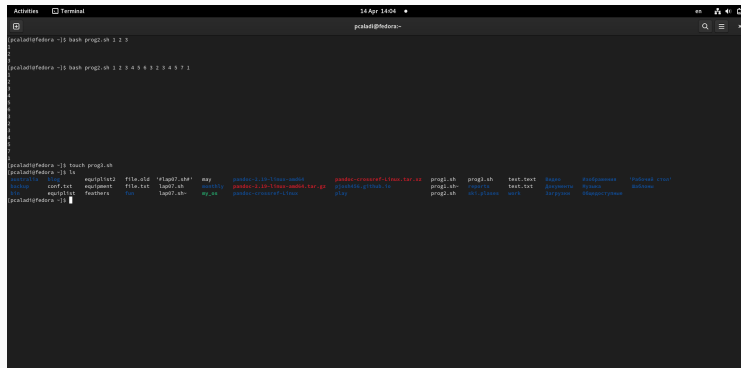


The image shows a screenshot of a text editor window titled "Text Editor" with a timestamp of "14 Apr 13:55". The editor is open to a file named "prog2.sh". The code inside the file is a shell script with the following lines:

```
1 #!/bin/bash
2 # echo "password" >/dev/null
3 # read n
4 for A in $(
5 do echo $A
6 done
```

Рис. 4: Текст второй программы

# Выполнение лабораторной работы



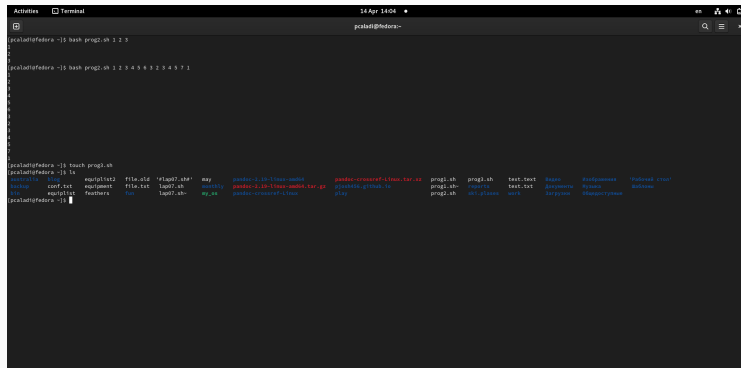
A terminal window titled 'Terminal' with a date indicator '14 Apr 14:04'. The prompt is 'pcaladi@fedora:~'. The user enters the command 'bash prog2.sh 1 2 3', followed by 'bash prog2.sh 1 2 3 4 5 6 7 2 3 4 5 7 1'. The output shows a series of numbers: 1, 2, 3, 4, 5, 6, 7, 2, 3, 4, 5, 7, 1. Then the user enters 'touch prog3.sh'. Finally, the user enters 'ls', which displays a long listing of files and directories in the current directory.

```
pcaladi@fedora:~$ bash prog2.sh 1 2 3
1
2
3
pcaladi@fedora:~$ bash prog2.sh 1 2 3 4 5 6 7 2 3 4 5 7 1
1
2
3
4
5
6
7
2
3
4
5
7
1
pcaladi@fedora:~$ touch prog3.sh
pcaladi@fedora:~$ ls
desktops  blog      equipment  file.sld  'lap07.sh'  may      pandoc-2.10-1.14ex-ubuntu  pandoc-crossref-1.14ex.tar.gz  prog1.sh  prog1.sh  test.txt  topics  xapptemplates  'xapptemplates'  'xapptemplates'
backup    conf.txt  equipment  file.txt  lap07.sh    monthly  pandoc-2.10-1.14ex-ubuntu  pandoc-crossref-1.14ex.tar.gz  prog1.sh  prog1.sh  test.txt  topics  xapptemplates  'xapptemplates'  'xapptemplates'
bin       equipment  features   foo       lap07.sh    my_08    pandoc-crossref-1.14ex    play                                prog1.sh  prog1.sh  test.txt  topics  xapptemplates  'xapptemplates'  'xapptemplates'
```

Рис. 5: Проверка работы второй программы

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. ((fig:006?; fig:007?; fig:008?))

# Выполнение лабораторной работы



```
pcaladi@fedora:~$ bash prog2.sh 1 2 3
1
2
3
pcaladi@fedora:~$ bash prog2.sh 1 2 3 4 5 6 7 2 3 4 5 7 1
1
2
3
4
5
6
7
2
3
4
5
6
7
pcaladi@fedora:~$ touch prog3.sh
pcaladi@fedora:~$ ls
desktops  blog  equip1.txt  file.sld  'lap07.sh'  may  pandoc-2.10.1-latest-manual  pandoc-crossref-1.10.0.tar.gz  prog1.sh  prog1.sh  test.txt  topics  workspace  'workspace.ctm'
backup  conf.txt  equipment  file.txt  lap07.sh  monthly  pandoc-2.10.1-latest-manual.tar.gz  pandoc-crossref-1.10.0.tar.gz  prog1.sh  prog1.sh  test.txt  topics  workspace  'workspace.ctm'
bin  equip1.txt  features  foo  lap07.sh  my_se  pandoc-crossref-1.10.0  play  prog1.sh  prog1.sh  test.txt  topics  workspace  'workspace.ctm'
```

Рис. 6: Создание файла для третьей программы



# Выполнение лабораторной работы

A screenshot of a text editor window titled 'Text Editor' with a timestamp of '14 Apr 14:03'. The editor shows a shell script file named 'prog3.sh' with the following content:

```
1 #!/bin/bash
2 for A in *
3 do
4   if test -d $A
5   then
6     echo $A: is a directory
7   else
8     echo -n $A: is a file and
9     if test -w $A
10    then
11      echo writable
12    else
13      echo readable
14    else
15      echo neither readable nor writable
16    fi
17  fi
18 done
```

Рис. 7: Текст третьей программы

## Выполнение лабораторной работы

[illegible]

Рис. 8: Проверка работы третьей программы

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. ((fig:009?; fig:010?))



The screenshot shows a window titled 'Text Editor' with a file named 'prog3.sh'. The script contains a loop that iterates over the variable 'A' and checks if 'test' is a directory, a file, or neither. The code is as follows:

```
1 #!/bin/bash
2 for A in *
3 do if test -d $A
4 then echo $A: is a directory
5 else echo -n $A: is a file and
6 if test -w $A
7 then echo writable
8 elif test -r $A
9 then echo readable
10 else echo neither readable nor writable
11 fi
12 fi
13 done
```

Рис. 9: Текст четвертой программы



В процессе выполнения лабораторной работы я изучила основы программирования в оболочке ОС UNIX/Linux. Научилась писать небольшие командные файлы.

1. Лабораторная работа № 10. Программирование в командном процессоре ОС UNIX. Командные файлы [Электронный ресурс]. URL: <https://esystem.rudn.ru/>.

Спасибо за внимание!