

Лабораторная работа №10

ДИОН ГОНССАН СЕДРИК МИШЕЛ

¹RUDN 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.

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

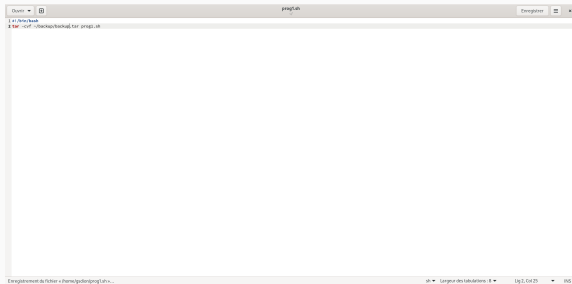


Рис. 1: Текст первой программы

Выполнение лабораторной работы

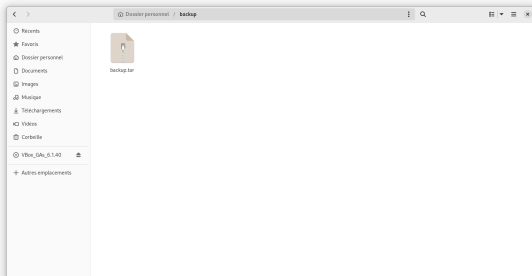
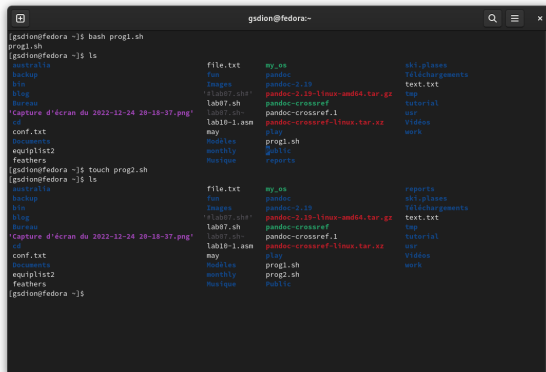


Рис. 2: Проверка работы программы

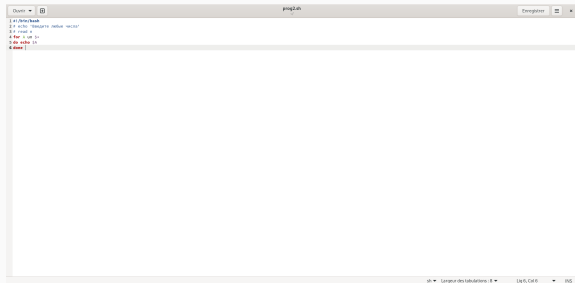
Выполнение лабораторной работы



```
gsdion@fedora:~  
[gsdion@fedora ~]$ bash prog1.sh  
prog1.sh  
[gsdion@fedora ~]$ ls  
australia      file.txt        my_os           skt.planes  
backup         fun             pandoc          téléchargements  
bin            Images          pandoc-2.19     text.txt  
blog           'Lab07.sh'     'pandoc-2.19-linux-umd64.tar.gz'  
Bureau         Lab07.sh       pandoc-crossref tap  
conf.txt       Lab07.sh       pandoc-crossref.1 tutorial  
Documents      Lab10-1.asm    pandoc-crossref-linux.tar.xz use  
equiplist2     may            play            Videos  
feathers       Modules        prog1.sh        work  
               monthly       public  
               Musique       reports  
[gsdion@fedora ~]$ touch prog2.sh  
[gsdion@fedora ~]$ ls  
australia      file.txt        my_os           reports  
backup         fun             pandoc          skt.planes  
bin            Images          pandoc-2.19     téléchargements  
blog           'Lab07.sh'     'pandoc-2.19-linux-umd64.tar.gz'  
Bureau         Lab07.sh       pandoc-crossref text.txt  
conf.txt       Lab07.sh       pandoc-crossref.1 tap  
Documents      Lab10-1.asm    pandoc-crossref-linux.tar.xz tutorial  
equiplist2     may            play            use  
feathers       Modules        prog1.sh        Videos  
               monthly       prog2.sh        work  
               Musique       public
```

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?))



```
1 #!/bin/bash
2 # echo "Введите время часа"
3 # read h
4 # for i in $(seq 1 5)
5 # do echo $i
6 done
```

Рис. 4: Текст второй программы

```
[gsdion@fedora ~]$ bash prog2.sh
prog2.sh: ligne 4: erreur de syntaxe près du symbole inattendu « un »
prog2.sh: ligne 4: `for A un $*`
[gsdion@fedora ~]$ bash prog2.sh
[gsdion@fedora ~]$ bash prog2.sh 1 2 3 4 5 6 7 8 9
1
2
3
4
5
6
7
8
9
[gsdion@fedora ~]$ bash prog2.sh 1 5 8 9
1
5
8
9
[gsdion@fedora ~]$ bash prog2.sh 9 0 12 3 4
9
0
12
3
4
[gsdion@fedora ~]$
```

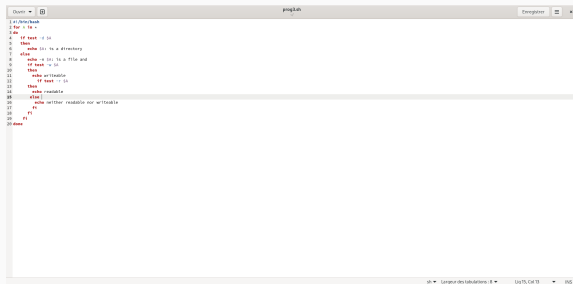
Рис. 5: Проверка работы второй программы

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

```
[gsdion@fedora ~]$ touch prog3.sh
[gsdion@fedora ~]$ ls
australia      file.txt       my_os          Public
backup         fun           pandoc         reports
bin            Images        pandoc-2.19    ski.places
blog           'lab007.sh'   pandoc-2.19-linux-amd64.tar.gz  Téléchargements
Bureau        lab07.sh      pandoc-crossref  text.txt
'Capture d'écran du 2022-12-24 20-18-37.png' lab07.sh      pandoc-crossref.1  tmp
cd             lab10-1.asm   pandoc-crossref-linux.tar.xz  tutorial
conf.txt       may           play           usr
Documents     Modèles       prog1.sh       Vidéos
equiplist2    monthly      prog2.sh       work
feathers      Musique       prog3.sh
```

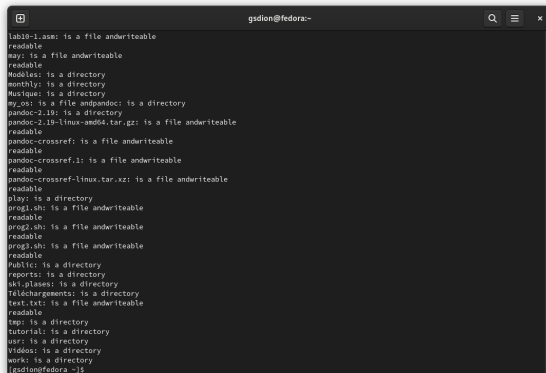
Рис. 6: Создание файла для третьей программы



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

Рис. 7: Текст третьей программы

Выполнение лабораторной работы

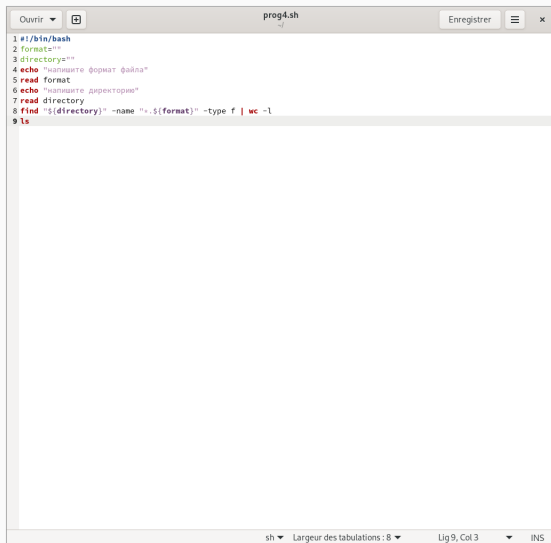


```
gsdion@fedora:~  
lab10-1.asm: is a file and writeable  
readable  
may: is a file and writeable  
readable  
Modèles: is a directory  
monthly: is a directory  
Musique: is a directory  
my_091 is a file and pandoc: is a directory  
pandoc-2.19: is a directory  
pandoc-2.19-linux-amd64.tar.gz: is a file and writeable  
readable  
pandoc-crossref: is a file and writeable  
readable  
pandoc-crossref.1: is a file and writeable  
readable  
pandoc-crossref-linux.tar.xz: is a file and writeable  
readable  
play: is a directory  
prog1.sh: is a file and writeable  
readable  
prog2.sh: is a file and writeable  
readable  
prog3.sh: is a file and writeable  
readable  
Public: is a directory  
reports: is a directory  
ski_places: is a directory  
Téléchargements: is a directory  
text.txt: is a file and writeable  
readable  
tsp: is a directory  
tutorials: is a directory  
usr: is a directory  
Vidéos: is a directory  
work: is a directory  
gsdion@fedora ~$
```

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?))

Выполнение лабораторной работы



```
1 #!/bin/bash
2 format=""
3 directory=""
4 echo "напишите формат файла"
5 read format
6 echo "напишите директорию"
7 read directory
8 find "${directory}" -name ".*${format}" -type f | wc -l
9 ls
```

Рис. 9: Текст четвертой программы

Выполнение лабораторной работы

```
lgsdion@fedora ~]$ touch prog4.sh
lgsdion@fedora ~]$ bash prog4.sh
напишите формат файла
txt
напишите директорию
/home/lgsdion
62
australia      file.txt      my_os        prog4.sh
backup         fun          pandoc       Public
bin            Images       pandoc-2.19  reports
blog          'filelab07.sh' pandoc-2.19-linux-and64.tar.gz ski_places
Bureau        lab07.sh     pandoc-crossref Téléchargements
'capture d'écran du 2022-12-24 20-18-37.png' lab07.sh     pandoc-crossref.1 text.txt
cd            lab10-1.asm  pandoc-crossref-linux.tar.xz tmp
conf.txt      may          play         tutorial
Documents    Modèles     prog1.sh     usr
equiplist2   monthly     prog2.sh     Vidéos
feathers      Musique     prog3.sh     work
lgsdion@fedora ~]$
```

Рис. 10: Проверка работы четвертой программы

В рамках лабораторной работы я изучил основы программирования командного интерпретатора UNIX/Linux. Я научился писать небольшие командные файлы. ## Список литературы

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

Спасибо за внимание!