

Лабораторная работа №12

Отчёт

Андрюшин Никита Сергеевич

Содержание

1	Цель работы	5
2	Задание	6
3	Выполнение лабораторной работы	7
4	Выводы	11
	Список литературы	12

Список иллюстраций

3.1	Код первой программы	7
3.2	Проверка работы первой программы	7
3.3	Код второй программы	8
3.4	Проверка работы второй программы	8
3.5	Код третьей программы	8
3.6	Проверка работы третьей программы	9
3.7	Код четвёртой программы	9
3.8	Проверка работы четвёртой программы	10

Список таблиц

1 Цель работы

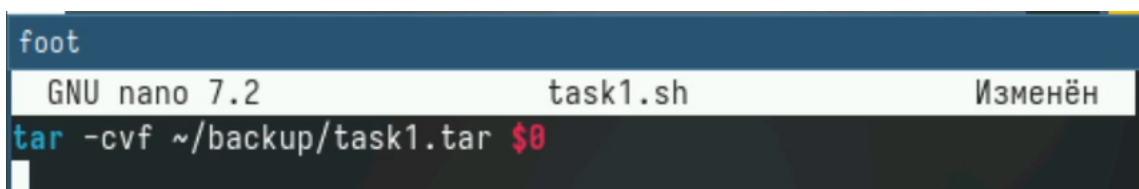
Изучить основы программирования в оболочке ОС UNIX/Linux. Научиться писать небольшие командные файлы [1]

2 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.

3 Выполнение лабораторной работы

Напишем код первой программы (рис. 3.1).



```
foot
GNU nano 7.2 task1.sh Изменён
tar -cvf ~/backup/task1.tar $0
```

Рис. 3.1: Код первой программы

И проверим работу (рис. 3.2).

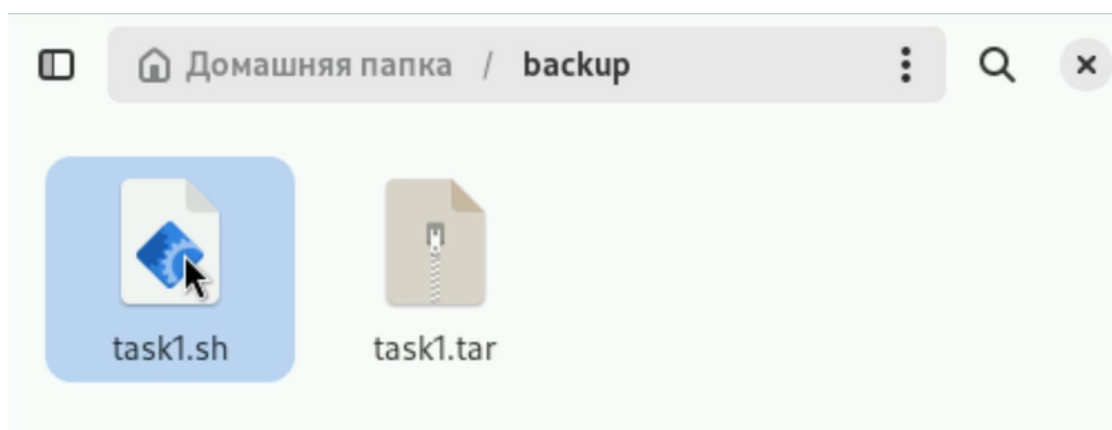


Рис. 3.2: Проверка работы первой программы

Напишем код второй программы (рис. 3.3).

```
foot
GNU nano 7.2 task2.sh
for i in "$@"
do echo ${i}
done
```

Рис. 3.3: Код второй программы

И проверим работу (рис. 3.4).

```
[nsandryushin@nsandryushin ~]$ ./task2.sh gdf thdfg dgfsg fdgdf
dfgfdg "fdgfdg fdg d drg d"
gdf
thdfg
dgfsg
fdgdf
dfgfdg
fdgfdg fdg d drg d
[nsandryushin@nsandryushin ~]$
```

Рис. 3.4: Проверка работы второй программы

Напишем код третьей программы (рис. 3.5).

```
foot
GNU nano 7.2 task3.sh
echo "$1/ " | tr -d "\n";
stat --printf "%A" "$1/";
echo
for i in $1/*
do echo "${i} " | tr -d "\n";
stat --printf "%A" "${i}";
echo
done
```

Рис. 3.5: Код третьей программы

И проверим работу (рис. 3.6).

```
[nsandryushin@nsandryushin ~]$ nano task3.sh
[nsandryushin@nsandryushin ~]$ ./task3.sh ~
/home/nsandryushin/ drwx-----
/home/nsandryushin/#1# -rw-r--r--
/home/nsandryushin/#2# -rw-r--r--
/home/nsandryushin/#3# -rw-r--r--
/home/nsandryushin/#4# -rw-r--r--
/home/nsandryushin/abc1 -rw-rw-r--
/home/nsandryushin/australia drwxr--r--
/home/nsandryushin/backup drwxr-xr-x
/home/nsandryushin/conf.txt -rw-r--r--
/home/nsandryushin/Downloads drwxr-xr-x
/home/nsandryushin/feathers -rw-rw-r--
/home/nsandryushin/file.txt -rw-r--r--
/home/nsandryushin/git-extended drwxr-xr-x
/home/nsandryushin/kconf.txt drwxr-xr-x
/home/nsandryushin/#lab07.sh# -rw-r--r--
/home/nsandryushin/lab07.sh -rw-r--r--
/home/nsandryushin/Lab1 drwxrwx---
```

Рис. 3.6: Проверка работы третьей программы

Напишем код четвёртой программы (рис. 3.7).

```
foot
GNU nano 7.2 task4.sh
let COUNT=0
for i in $2/*. $1
do let COUNT++
done
echo $COUNT
```

Рис. 3.7: Код четвёртой программы

И проверим работу (рис. 3.8).

```
[nsandryushin@nsandryushin ~]$ ./task4.sh txt ~  
5
```

Рис. 3.8: Проверка работы четвёртой программы

4 Выводы

В результате выполнения лабораторной работы были получены навыки написания скриптовых файлов

Список литературы

1. Kulyabov. Лабораторная работа № 12. Программирование в командном процессоре ОС UNIX. Командные файлы. RUDN.