Отчет по лабораторной работе №12

Программирование в командном процессоре ОС UNIX. Командные файлы

Разанацуа Сара Естэлл

Содержание

# 1 Цель работы

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

# 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

Создаю файл prog1.sh в котором буду пистаь программу с расширением sh (shell) с помощью утилиты touch, далее делаю его исполняемым с помощью chmod +x, открываю файл в любом текстовом редакторе, пишу в нем код, и после того как я написала программу в файле, я могу его запустить bash (рис. fig. 1).

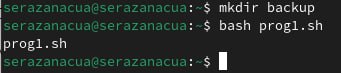


Рис. 1: Выполнение программы 1

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 (рис. fig. 2).



Рис. 2: Написанная программа 1

#!/bin/bash tar -cvf ~/backup/backup.tar prog1.sh Создаю файл prog2.sh в котором буду пистаь программу с расширением sh (shell) с помощью утилиты touch, далее делаю его исполняемым с помощью chmod +x, открываю файл в любом текстовом редакторе, пишу в нем код, и после того как я написала программу в файле, я могу его запустить bash (рис. fig. 3).

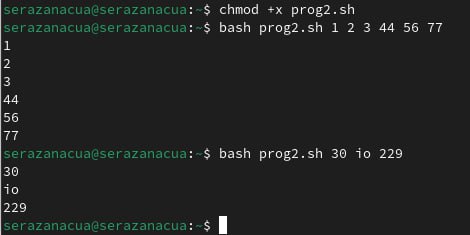


Рис. 3: Выполнение программы 2

Пример командного файла, обрабатывающего любое произвольное число аргументов командной строки, в том числе превышающее десять. Например, скрипт может последовательно распечатывать значения всех переданных аргументов (рис. fig. 4).



Рис. 4: Написання программа 2

#!/bin/bash for A in $\* do echo $A done

Создаю файл prog3.sh в котором буду пистаь программу с расширением sh (shell) с помощью утилиты touch, далее делаю его исполняемым с помощью chmod +x, открываю файл в любом текстовом редакторе, пишу в нем код, и после того как я написала программу в файле, я могу его запустить bash (рис. fig. 5).

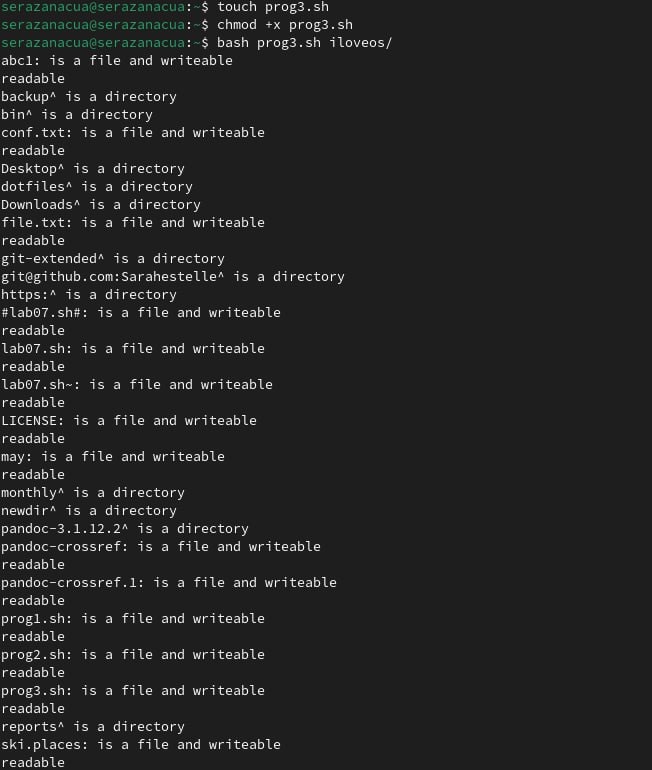


Рис. 5: Выполнение программы 3

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. (рис. fig. 6).

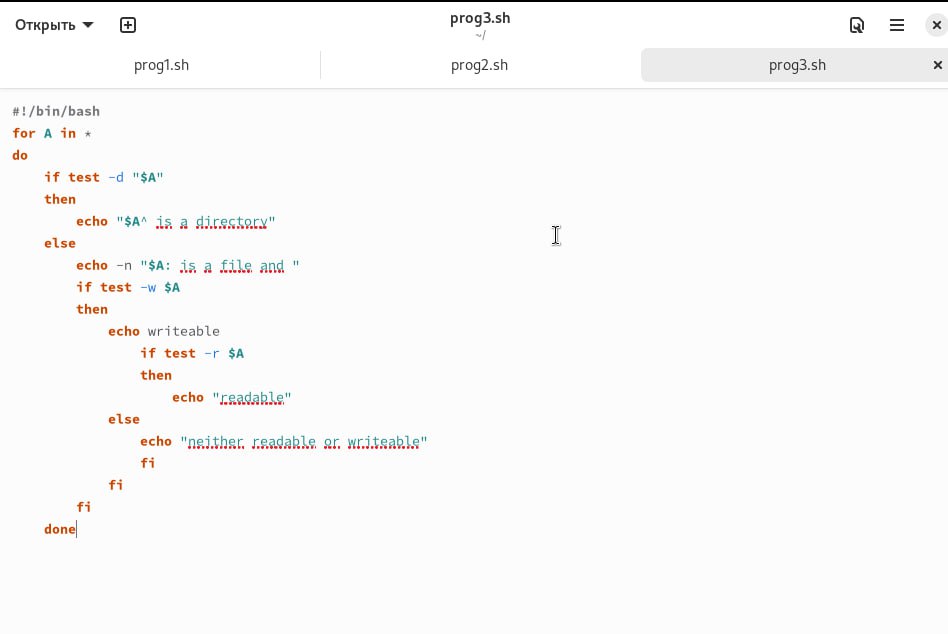


Рис. 6: Написанная программа программа 3

#!/bin/bash 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 if test -r $A then echo “readable” else echo “neither readable or writeable” fi fi fi done

Создаю файл prog4.sh в котором буду пистаь программу с расширением sh (shell) с помощью утилиты touch, далее делаю его исполняемым с помощью chmod +x, открываю файл в любом текстовом редакторе, пишу в нем код (рис. fig. 7).

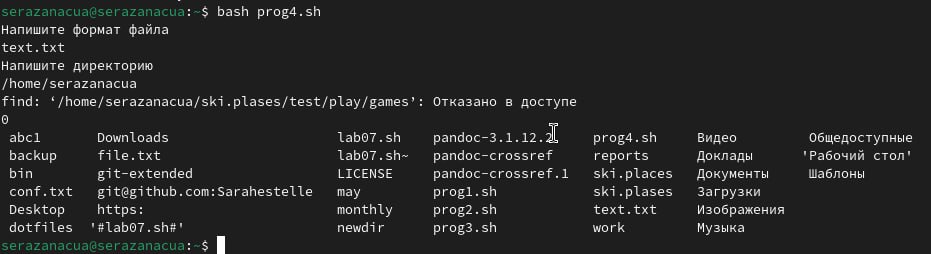


Рис. 7: Создание файла с программой 4

* 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 (рис. fig. 8).

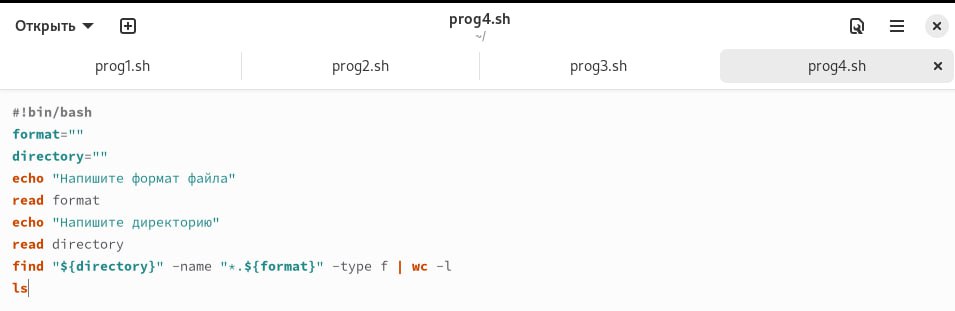


Рис. 8: Написанная программа 4

# 4 Выводы

* При выполнении данной лабораторной работы я изучила основы программирования в оболочке ОС UNIX/Linux, научилась писать небольшие командные файлы.