Отчёт по лабораторной работе №12

Операционные системы

Бекауов Артур Тимурович

Содержание

# 1 Цель работы

Цель данной лабораторной 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 для новой программы меняю права доступа, разрешая его выполнение, таким образом файл становится исполняемым. Открываю файл в редакторе nano и записываю следующий код программы:

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

Сохраняю файл и закрываю редактор nano, далее запускаю исполняемый файл с помощью команды bash. Затем проверяю, что файл создал резервную копию самого себя (в виде архива) и поместил её в директорию backup.(рис. 1).

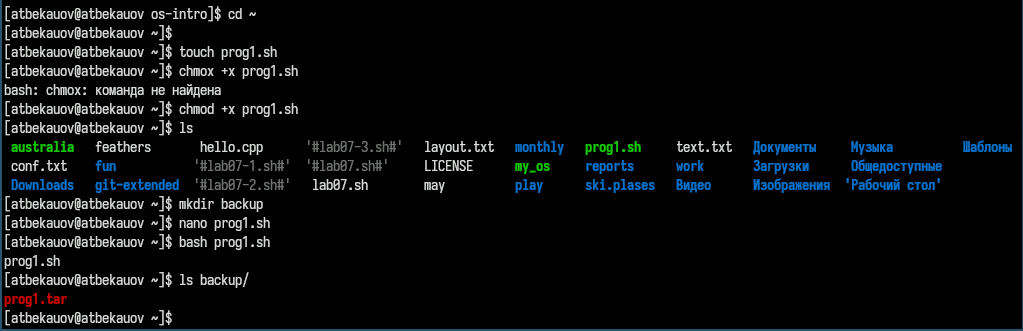


Рис. 1: Программа №1

Создаю файл prog2.sh, меняю права доступа, разрешая его выполнение. Открываю файл в nano (рис. 2).

Программа №2 - создание файла

Рис. 2: Программа №2 - создание файла

Затем ввожу в файл текст программы:

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

Сохраняю файл, выхожу из nano и запускую файл через bash, введя в качестве аргумента разные символы. Как видим символы были продублированны скриптом. (рис. 3).



Рис. 3: Программа №2 - выполнение

Создаю файл prog3.sh, меняю права доступа, разрешая его выполнение. Открываю файл в nano (рис. 4).

Программа №3 - создание файла

Рис. 4: Программа №3 - создание файла

Затем ввожу в файл текст программы:

#!/bin/bash  
echo "Type in dir :"  
read directory  
echo " "  
cd "${directory}"  
for A in \*  
do  
 if test -d "$A"  
 then  
 echo -n "$A is directory"  
 else  
 echo -n "$A is file "  
 if test -r "$A"  
 then  
 echo -n "and readable "  
 fi  
 if test -w "$A"  
 then  
 echo -n " and writable "  
 fi  
 if test -e "$A"  
 then  
 echo -n "and executable "  
 fi  
 fi  
echo  
done

Сохраняю файл, выхожу из nano и запускую файл через bash, скрип просит ввести директорию - ввожу имя директории work. Скрипт выводит информацию о всех объектах этой директории. (рис. 5).

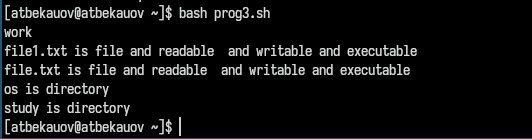


Рис. 5: Программа №3 - выполнение

Создаю файл prog4.sh, меняю права доступа, разрешая его выполнение. Открываю файл в nano (рис. 6).

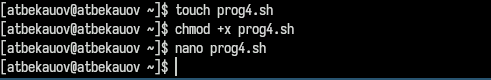


Рис. 6: Программа №4 - создание файла

Затем ввожу в файл текст программы:

#!/bin/bash  
format=""  
directory=""  
echo "Type in form:"  
read format  
echo "Type in dir"  
read directory  
find "${directory}" -name "\*.${format}" -type f |wc -l

Сохраняю файл, выхожу из nano и запускую файл через bash, скрипт просит ввести формат - ввожу txt, скрип просит ввести директорию - ввожу имя директории work. Кол-во txt файлов в директории work - 2.

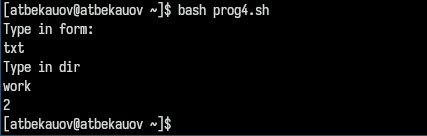


Рис. 7: Программа №4 - выполнение

# 4 Выводы

В ходе данной лаботраторной работы я изучил основы программирования в оболочке ОС UNIX/Linux. Научился писать небольшие командные файлы.

# 5 Ответы на онтрольные вопросы

1. Кратко охарактеризуйте редактор emacs.

Emacs — один из наиболее мощных и широко распространённых редакторов, используемых в мире UNIX. Написан на языке высокого уровня Lisp.

1. Какие особенности данного редактора могут сделать его сложным для освоения новичком?

Большое разнообразие сложных комбинаций клавиш, которые необходимы для редактирования файла и в принципе для работа с Emacs.

1. Своими словами опишите, что такое буфер и окно в терминологии emacs’а.

Буфер - это объект в виде текста. Окно - это прямоугольная область, в которой отображен буфер.

1. Можно ли открыть больше 10 буферов в одном окне?

Да, можно.

1. Какие буферы создаются по умолчанию при запуске emacs?

Emacs использует буферы с именами, начинающимися с пробела, для внутренних целей. Отчасти он обращается с буферами с такими именами особенным образом — например, по умолчанию в них не записывается информация для отмены изменений.

1. Какие клавиши вы нажмёте, чтобы ввести следующую комбинацию C-c | и C-c C-|?

Ctrl + c, а потом | и Ctrl + c Ctrl + |

1. Как поделить текущее окно на две части?

С помощью команды Ctrl + x 3 (по вертикали) и Ctrl + x 2 (по горизонтали).

1. В каком файле хранятся настройки редактора emacs?

Настройки emacs хранятся в файле . emacs, который хранится в домашней дирректории пользователя. Кроме этого файла есть ещё папка . emacs.

1. Какую функцию выполняет клавиша и можно ли её переназначить?

Выполняет функцию стереть, думаю можно переназначить.

1. Какой редактор вам показался удобнее в работе vi или emacs? Поясните почему.

Для меня удобнее был редактор Emacs, так как у него есть командая оболочка. А vi открывается в терминале, и выглядит своеобразно.