Лабораторная работа 10

Операционные системы

Голощапов Ярослав

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 Теоретическое введение

При перечислении имён файлов текущего каталога можно использовать следующие символы: – \* — соответствует произвольной, в том числе и пустой строке; – ? — соответствует любому одинарному символу; – [c1-c1] — соответствует любому символу, лексикографически находящемуся между символами c1 и с2. Например, – echo \* — выведет имена всех файлов текущего каталога, что представляет собой простейший аналог команды ls; – ls *.c — выведет все файлы с последними двумя символами, совпадающими с .c. – echo prog.? — выведет все файлы, состоящие из пяти или шести символов, первыми пятью символами которых являются prog.. – [a-z]* — соответствует произвольному имени файла в текущем каталоге, начинающемуся с любой строчной буквы латинского алфавита. Такие символы, как ’ < > \* ? |  ” &, являются метасимволами и имеют для командного процессора специальный смысл. Снятие специального смысла с метасимвола называется экранированием метасимвола. Экранирование может быть осуществлено с помощью предшествующего метасимволу символа , который, в свою очередь, является метасимволом. Для экранирования группы метасимволов нужно заключить её в одинарные кавычки. Строка, заключённая в двойные кавычки, экранирует все метасимволы, кроме $, ’ , , “. Например, – echo \* выведет на экран символ *, – echo ab’*|*’cd выведет на экран строку ab*|\*cd.

# 4 Выполнение лабораторной работы

Создаю файл prog1.sh , в котором буду писать программу с расширением sh с помощью утилиты touch (рис. fig. 1).

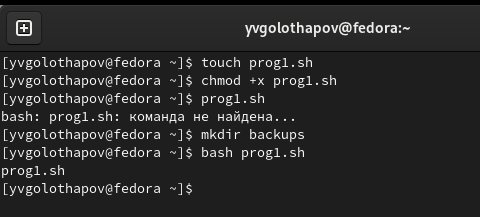


Рис. 1: Создание prog1

Пишу скрипт в файле (рис. fig. **¿fig:002?**) 

Создаю файл prog2.sh (рис. fig. 2)

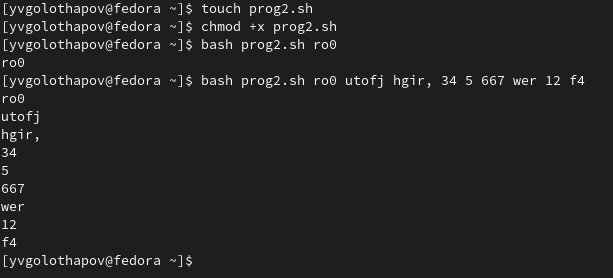


Рис. 2: Создание prog2

Пишу скрипт в файле prog2 (рис. fig. 3)

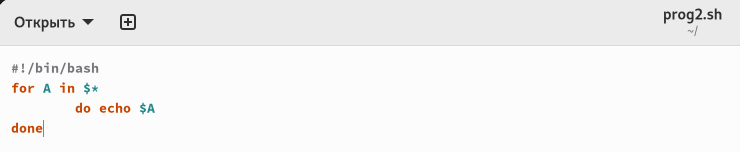


Рис. 3: Скрипт в prog2

Создаю файл prog3.sh (рис. fig. 4)



Рис. 4: Создание prog3

Пишу скрипт в файле prog3 (рис. fig. 5)

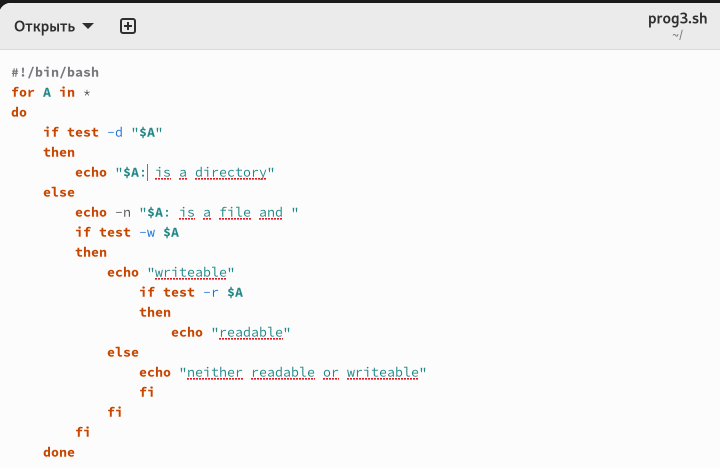


Рис. 5: Скрипт в prog3

Создаю файл prog4.sh (рис. fig. 6)

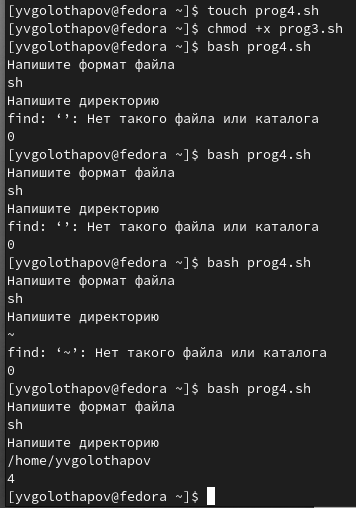


Рис. 6: Создание prog4

Пишу скрипт в файле prog4 (рис. fig. 7)

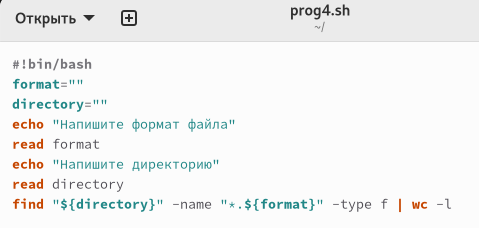


Рис. 7: Скрипт в prog4

# 5 Вывод

При выполнении лабораторной работы я изучил основы программирования в оболочке ОС UNIX/Linux, научился писать небольшие командные файлы.