Лабораторная работа № 10

Программирование в командном процессоре ОС UNIX. Командные файлы

Юсупова Ксения Равилевна

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

Напишем код для первой программы (Написать скрипт, который при запуске будет делать резервную копию самого себя в другую директорию backup в вашем домашнем каталоге)(рис. 1).

код для первой программы

Рис. 1: код для первой программы

Проверили код на работу (рис. 2).

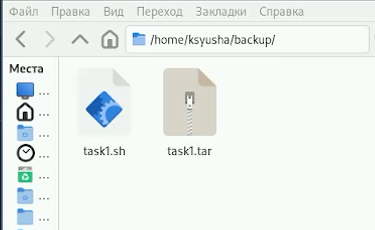


Рис. 2: проверили первый код

Напишем код для второй программы (пример командного файла, обрабатывающего любое произвольное число аргументов командной строки, в том числе превышающее десять)(рис. 3).

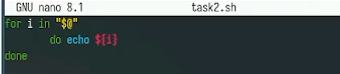


Рис. 3: код для второй программы

Проверили код на работу (рис. 4).

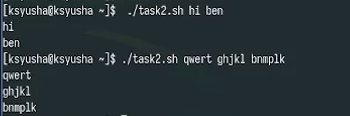


Рис. 4: Проверили код на работу

Написали код для третьей программы (Написать командный файл — аналог команды ls (без использования самой этой команды и команды dir).)(рис. 5).

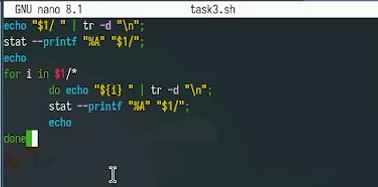


Рис. 5: код для третьей программы

Проверили код на работу (рис. 6).

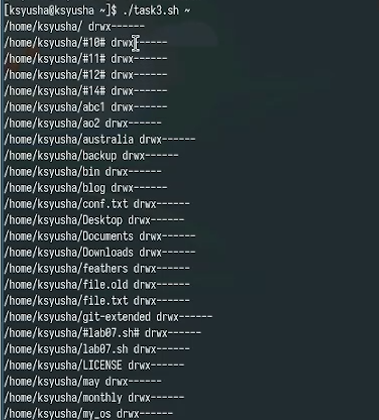


Рис. 6: Проверили код на работу

Написали код для четвертой программы (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) (рис. 7).

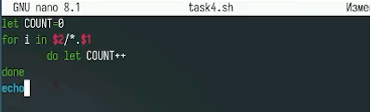


Рис. 7: код для четвертой программы

Проверили код на работу (рис. 8).

Проверили код на работу

Рис. 8: Проверили код на работу

# 4 Ответы на контрольные вопросы

1. Командная оболочка (shell) - программа-интерпретатор для взаимодействия пользователя с ядром ОС. Примеры: bash (стандартная), zsh (с автодополнением), fish (интерактивная), dash (легковесная). Отличаются синтаксисом, функционалом и скоростью работы.
2. POSIX - стандарт для совместимости UNIX-систем, унификации API и командных интерфейсов.
3. В bash переменные: var=“значение”. Массивы: arr=(“эл1” “эл2”) - индексированный, declare -A dict=([“кл”]=“зн”) - ассоциативный.
4. let - для арифметики: let “sum=5+5”. read - чтение ввода: read -p “Имя:” name.
5. Арифметические операции: + - \* / % (остаток), сравнения == != > <, битовые & | ^ << >>.
6. (( )) - для арифметики: res=$((x+y)) и сравнений: if ((x>y)).
7. Стандартные переменные: $HOME (домашний каталог), $PATH (пути программ), $USER (пользователь), $SHELL (оболочка).
8. Метасимволы - спецсимволы: \* (любые символы), ? (один символ), > (перенаправление).
9. Экранирование: \* (обратный слэш), ’*’ (полное), “*” (частичное).
10. Создать файл script.sh с #!/bin/bash, дать права chmod +x script.sh, запустить ./script.sh.
11. Функции: func(){ команды; return; }. Вызов: func.
12. Проверка типа: [ -f file ] - файл, [ -d dir ] - каталог.
13. set - управление shell, typeset - тип переменной, unset - удаление.
14. Параметры передаются при вызове ./script.sh p1 p2. В скрипте: $1, $2 - параметры, $@ - все, $# - количество.
15. Спецпеременные: $0 - имя скрипта, $? - код возврата, $$ - PID, $! - PID фонового.

# 5 Вывод

В ходе лабораторной работы мы изучили основы программирования в оболочке ОС UNIX/Linux и научились писать небольшие командные файлы