Лабораторная работа 1

2. Авторизуйтесь в системе, запустите эмулятор терминала и измените

пароль пользователя.

passwd user1 12345

5. Авторизуйтесь под одним или несколькими пользователями в

различных виртуальных консолях.

adduser user2

6. Подготовьте файл с информацией об авторизованных в системе

пользователях и текущем времени. Для этого воспользуйтесь

перенаправлением вывода в файл.

who; date > usersanddate.log

Лабораторная работа 2

4. Используя команды оболочки, создайте в домашнем каталоге дерево

каталогов согласно схеме, приведённой ниже: в домашнем каталоге –

cat1, содержащий каталоги cat2 и cat3. Каталог cat1/cat2 содержит

каталог cat3. Каталог cat1/cat3 содержит каталог cat4. Каталог cat1/cat2

содержит каталог cat5. Каталог cat1/cat2/cat3 содержит cat6 и cat7.

Каталог cat1/cat8 содержит символическую ссылку на каталог

cat1/cat2/cat3/cat6. Каталог cat1 содержит каталог cat8.

mkdir cat1

cd cat1

mkdir cat2/cat3/cat6

cd cat3

mkdir cat7

cd cat2

mkdir cat5

cd cat1

mkdir cat3/cat4

mkdir cat8

cd cat8

$ ln -s matreshka.zhmih cat1/cat2/cat3/cat6

Чё за матрёшка???? Я запутолси…

5. Удалить каталоги с дублирующиеися именами.

rm -rf / \*это точно удалит все дублирующиеся каталоги.

7. Создать каталог dir. В файл dir/date поместить текущую дату.

mkdir dir

cd dir

touch failik; date > failik

Лабораторная работа 3

3. Отсортируйте вывод команды ls -l по дате изменения содержимого по

месяцам.

ls -l -tr

4. Из файла passwd\_example получите имена всех пользователей,

содержащихся в первом поле каждой строки, и поместите

отсортированный в обратном порядке результат в файл cut\_result.

cut -d: -f1 /etc/passwd\_example | sort -r

5. С помощью текстового редактора (gedit, nano) измените имена

нескольких пользователей в этом файле и сохраните результат в новый

файл cut\_result2.

nano /etc/passwd\_example

KurwaEzhik

\*неистово стирает\*

KurwaUzhik

ctr+s

ctr+x

6. Сравните содержимое файлов с именами пользователей с помощью

программы diff.

\*Яростно сравнил\*

7. Добавьте к содержимому файла cut\_result содержимое файла

cut\_result2.cat cut\_result >> cut\_result2

8. С помощью команды uniq избавьтесь от дубликатов в файле cut\_result. $ uniq cut\_result

11.Определите тип используемого терминала.

who am i

Лабораторная работа 4

4. Получите имена всех файлов, не являющихся символическими

ссылками или каталогами, и поместите их в файл filelist1.txt.

find -not -type d -and -not -type l >> filelist1.txt.

5. С помощью команд find, xargs и ls получите полную информацию об

атрибутах файлов домашнего каталога размер которых превышает 5

Килобайт и поместите результат в файл filelist2.txt

find -type f -and -size +5k | xargs ls -l >> filelist2.txt

6. С помощью команды locate получите список имен файлов содержащих

в названии строку “bash”.locate -b bash

7. Для команд, используемых в предыдущих подпунктах, получите

расположения файлов справочных руководств.

whereis -m find xargs ls locate wc

8. Из файла passwd\_example c помощью утилиты grep, получите записи

пользователей с домашними каталогами в папке home, c указанием

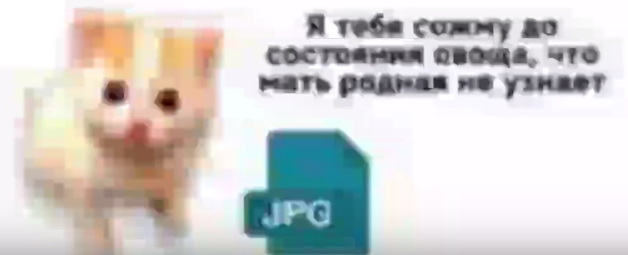
номеров строк. Поместите результат в файл filelist3.txt.

cat passwd\_example | grep -n /home/ >> filelist3.txt

9. Сожмите файл filelist1.txt с сохранением исходного файла, утилитой

gzip c различными степенями сжатия. Для получившихся файлов

узнайте процент коэффициента сжатия.



Лабораторная работа 5

ЗНАКОМСТВО С ПРОЦЕССАМИ

3. Создайте файл proc1, содержащий список процессов пользователя

root, отсортированный по идентификатору родительского процесса.

Используйте команду ps и изученные ранее утилиты.

nano script.sh

#!/bin/bash

ps -fu root --sort ppid > proc1

4. Получите информацию о процессах вашего пользователя, имеющих

статус работоспособный.

ps -U root

5. Добавьте к файлу proc1 сведения о процессе в данный момент

потребляющий больший процесс ресурсов центрального процессора.

top -c -b > proc1

Лабораторная работа 7

ОСНОВЫ НАПИСАНИЯ СКРИПТОВ НА BASH

2. Создайте скрипт, находящий сумму двух переданных ему аргементов.

Выведите результат сложения и сообщение о том, меньше ли результат

нуля, либо больше, либо равен нулю.

nano scripit.sh

#!/bin/bash

if [ "$#" -ne 2 ]; then

echo "Введите 2 числа"

exit 1

fi

sum=$(($1 + $2))

if [ "$sum" -lt 0 ]; then

echo "Сумма меньше 0."

elif [ "$sum" -gt 0 ]; then

echo "Сумма больше 0."

else

echo "Сумма равна 0."

fi

3. Создайте скрипт, принимающий в качестве агрумента строку, и

создающий архив, включающий в себя все файлы из домашней

директории пользователя с расширением .txt. Имя итогового архива

должно состоять из переданного аргумента и отметки времени

строка\_час\_минута\_секунда.

nano scripit2.sh

#!/bin/bash

if [ "$#" -ne 1 ]; then

echo "Использование: $0 имя\_для\_архива"

exit 1

fi

name\_prefix="$1"

timestamp=$(date +"%H\_%M\_%S")

archive\_name="${name\_prefix}\_${timestamp}.tar.gz"

home\_dir="$HOME"

tar -czf "$archive\_name" —directory="$home\_dir" $(find "$home\_dir" -name "\*.txt" -type f -print)

if [ $? -eq 0 ]; then

echo "Архив $archive\_name успешно создан."

else

echo "Произошла ошибка при создании архива."

exit 1

fi