Задания по работе с процессами.

1.Вывести список процессов по конкретному терминалу.

*ps* выведет по-умолчанию процессы терминала пользователя, чтобы вывести все – *ps -aux*

2. зайти обычным пользователем и вывести список своих процессов в виде дерева.

*pstree USERname*

3.Определить 10 процессов, отсортированных по убыванию использования процессора.

*top, n 10* (количество строк)

4.Определить 10 процессов, отсортированных по убыванию использования памяти.

*top, shift+m, n 10*

5.Написать скрипт, который будет запускать пустой бесконечный цикл, и запустить его в фоновом режиме.

*/home/empty\_infiniti.sh*

*#!/bin/bash*

*while true*

*do*

*echo “”*

*sleep 999*

*done*

*sh empty\_infiniti.sh &*

6.Найти процесс из предыдущего задания и завершить его.

*ps -hf*, вывод с родителями **ppid** (f показывает с ppid), (h показывает имя файла команды)

***kill (PID).***

Удалить по имени. *pkill sh\** (все что начинается с sh)

7.Запустить команду date через 2 минуты от текущего времени.

date | at now + 1 minute – НЕ РАБОТАЕТ

echo “date” | at -m now + 1 minute – НЕ РАБОТАЕТ

8.Запустить команду “watch ifconfig” через screen.

*Screen,*

*ifconfig*

9.Выполнить задание с помощью программы ttyrec и прислать на проверку.

10. Написать скрипт для автоматизации задания по пользователям. Между отдельными пунктами скрипт должен выводить на чистый экран результат выполнения и спрашивать: продолжить или завершить.

**while true**

**do**

**echo –n “Do you want to create new user? (yes /no)”**

**read answer**

**if [ $answer == no ]**

**then**

**break**

**elif [ $answer == yes ]**

**then**

**echo –n “Input username:”**

**read uname**

**useradd $uname**

**clear**

**echo “Username $uname successfully created”**

**else**

**echo “input only yes or no”**

**fi**

**done**

11. Написать скрипт для автоматизации задания по правам доступа отдельными пунктами скрипт должен выводить на чистый экран результат выполнения и спрашивать: продолжить или завершить.

**while true**

**do**

**while true**

**do**

**echo –n “Do you want to create a new directory? (yes /no)”**

**read aswer**

**if [ $answer == no ]**

**then**

**break 2**

**elif [ $answer == yes ]**

**then**

**echo –n “Input directory name:”**

**read dirname**

**mkdir $dirname**

**clear**

**echo “Directory $dirname successfully created”**

**break**

**else**

**echo “Input yes or no”**

**fi**

**done**

**while true**

**do**

**echo –n “Choose access rights for $dirname directory (use XXX format)”**

**read rightsdir**

**if [ $rightsdir –le 777 ]**

**then**

**chmod $rightsdir $dirname**

**break**

**else**

**echo “wrong number, input in XXX format”**

**fi**

**done**

**done**

12. Заархивируйте все файлы из одной из папок и разархивируйте в другой.

Создание выполняется с ключом **c**. Синтаксис следующий:

tar -c<опции> <новый архивный файл> <что сжимаем>

tar -czvf archive.tar.gz /home/dmosk

***z*** — сжать архив в gzip (без этого параметра, tar не сжимает, а создает так называемый тарбол); ***c*** — ключ на создание архива; ***v*** — verbose режим, то есть с выводом на экран процесса (очень удобно для наблюдением за ходом работы, но в скриптах можно упустить); ***f*** — использовать файл (обязательно указываем, так как в большей степени работаем именно с файлами)

Распаковка выполняется с ключом **x** с синтаксисом:

tar -x<опции> <архивный файл>

Пример распаковки:

tar -xvf archive.tar.gz

**Примеры**

**gz**

Распаковка .gz файла:

tar -xvf archive.tar.gz

*\* при возникновении ошибки****This does not look like a tar archive****, можно воспользоваться командой****gzip -d archive.tar.gz****.*

tar -cf filename.tar file1 file2 fileN – создаем архив без сжатия

**tar -cf arhiv.tar test1 test2 - запаковываем**

**tar -xvf arhiv.tar -C /home/dzcat1 - распаковываем**

13. Сделайте еще раз тоже самое задание из предыдущего пункта, только архив сделайте сжатый.

**tar -czvf arhsz.tar.gz test1 test2**

**tar -xvf arhsz.tar.gz -C /home/dzcat2**

14. Напишите скрипт, который будет генерировать текстовый файл (с текстом) размером 1Мб.

**#!/bin/bash**

**echo -n “Input filename:”**

**read filename**

**echo -n “Input filesize in bytes”**

**read filesize**

**re=’^[0-9]+$’**

**if ! [[ $filesize =~ $re ]]**

**then**

**echo “error: Not a number” >&2; exit 1**

**else**

**tr -dc A-Za-z </dev/urandom |head -c $filesize > $filename.txt**

**echo “$filename.txt successfully created”**

**fi**

15. Сожмите его с помощью gzip с разной степенью сжатия, Сравните размеры. И потом разожмите.

**gzip -1 -c 1mfile.txt > 1mfileC1.txt.gz**

**gzip -9 -c 1mfile.txt > 1mfileC9.txt.gz**

16. Скрипты, приложить в сжатом архиве.

17. Настройте, чтобы при входе в систему, система приветствовала Вас так, как Вам было бы приятно.

**~~В /etc/motd добавить~~**

**B root/bashrc**

**echo “Welcome back, Master”**

**echo “Date, Elapsed, Users, CPU:”**

**uptime**

**echo “Free RAM:”**

**free -m**

**echo “Space on HDD”**

**df -h**

18. Добавьте в приветствие краткую сводку о состоянии системы. (время работы, загрузка проца, памяти, дисков).

19. Выполните скрипты в программе ttyrec и пришлите файл записи для проверки