Задача 1. Используя команду who, определите пользователей, работающих в

системе. Вывод команды who направьте в файл users.txt. Просмотрите содержимое

файла users.txt.

#!/bin/bash

who >> users.txt

Задача 2. Напишите скрипт, который выполняет следующие действия:

? Выводит на терминал меню, которое предлагает выбор из следующих

действий:

0 удалить файл;

0 переименовать файл;

0 переместить файл;

0 создать файл.

? запрашивает имена файлов, для выбранного действия;

выполняет выбранное действие.

#!/bin/bash

echo "1- delete file"

echo "2- rename file"

echo "3 -move file"

echo "4 - create file"

echo "Enter the file name "

read filee

echo -n "Enter the command number "

read number

if [[ $number = '1' ]]

then

rm $filee

echo "Deleting the file was successful"

fi

if [[ $number = '2' ]]

then

echo "Enter the new file name "

read name

mv $filee $name

echo "Rename the file was successful"

fi

if [[ $number = '3' ]]

then

echo "Enter the catalog "

read catalog

mv $filee $catalog

echo "Move the file was successful"

fi

if [[ $number = '4' ]]

then

echo "Enter the name of the new file "

read name\_file

touch $name\_file

echo "File created successuful"

fi

Задача 3. Написать скрипт поиска одинаковых по их содержимому файлов в двух

каталогах, например Dir1 и Dir2. Пользователь задаёт имена Dir1 и Dir2 в качестве

первого и второго аргумента командной строки. В результате работы программы

файлы, имеющиеся в Dir1, сравниваются с файлами в Dir2 по их содержимому. На

экран выводятся число просмотренных файлов и результаты сравнения. Проверить

работу программы для каталога /log череusr (tty1, tty2) под вашейDir1)и любого

каталога в каталоге /log череhome (tty1, tty2) под вашейDir2).

Задача 2. Напишите сценарий, проверяющий имя текущего каталога и выводящий

сообщение об ошибке, если оно короче пяти символов. Если больше или равно пяти

символам, то выводящий посекундно в цикле имена файлов текущего каталога и их

порядковый номер.

#!/bin/bash

dir1=$1

dir2=$2

num=$(find $dir1 $dir2 -type f 2>/dev/null | wc -l)

(

#count number of equall files

#checked=0

IFS=$'\n'

#-type f -ищем имя файла

set1=$(find $dir1 -type f)

set2=$(find $dir2 -type f)

#num=${set1}${set2}

for var1 in $set1

do

if [ -f "$var1" ]

then

#echo $var1

for var2 in $set2

do

#if [ "$var1" != "$var2" ]

#then

if [ -f "$var2" ]

then

if (diff -q $var1 $var2)

then

echo $var1 = $var2

#let "checked=checked + 1"

fi

fi

#fi

done

fi

done

) 2>/tmp/errors.txt | grep =

sed -i "s/^/$(basename $0:" ")/g" /tmp/errors.txt

cat /tmp/errors.txt >&2

echo $num

Задача 3.

Написать скрипт, находящий в заданном каталоге и во всех его подкаталогах все

файлы, владельцем которых является заданный пользователь. Имя владельца и

каталог задаются пользователем в качестве первого и второго аргумента командной

строки. Скрипт выводит результаты в файл третий аргумент командной строки) в

виде полный путь, имя файла, его размер. На консоль выводится общее число

просмотренных файлов.

#! /bin/bash

name=${PWD##\*/}

if [ ${#name} -lt 5 ]; then

>&2 echo "$0 : The directory name is less then 5"

else

counter=1

for file in \*

do

echo "$counter: $file"

counter=$(($counter+1))

sleep 1

done

fi

#!/bin/bash

if [ $# -ne 3 ]; then

>&2 echo "$0 :3 params(user,dir,file) are expected"

exit 1

fi

user=$1

dir=$2

file=$3

n=0

OLD\_IFS=$IFS

IFS=$'\x0a'

for path in $(find "$dir" -type f -user "$user" -printf "$(pwd) %f | %s bytes\n")

do

((n++))

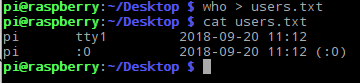
echo $path

done > $file

echo "$n files"

IFS=$OLD\_IFS

Задача 1. Используя команду who, определите пользователей, работающих в системе. Вывод команды who направьте в файл users.txt. Просмотрите содержимое файла users.txt.



Задача 2. Напишите скрипт, который выполняет следующие действия:

● Выводит на терминал меню, которое предлагает выбор из следующих

действий:

○ удалить файл;

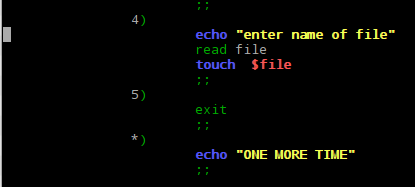
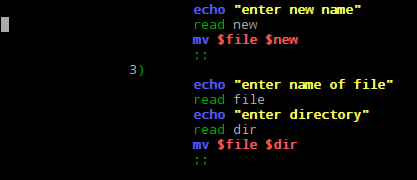
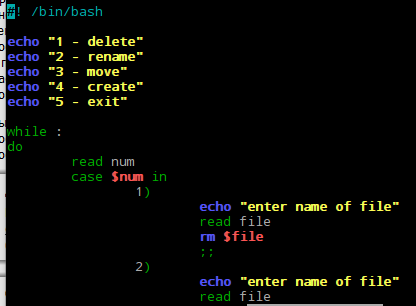
○ переименовать файл;

○ переместить файл;

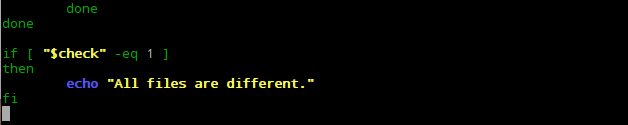
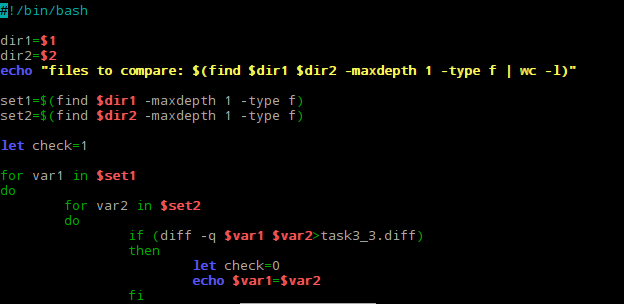
○ создать файл.

● запрашивает имена файлов, для выбранного действия;

● выполняет выбранное действие.





Задача 3. Написать скрипт поиска одинаковых по их содержимому файлов в двух каталогах, например Dir1 и Dir2. Пользователь задаёт имена Dir1 и Dir2 в качестве первого и второго аргумента командной строки. В результате работы программы файлы, имеющиеся в Dir1, сравниваются с файлами в Dir2 по их содержимому. На экран выводятся число просмотренных файлов и результаты сравнения. Проверить работу программы для каталога /usr (Dir1) и любого каталога в каталоге /home (Dir2). 

Задача 2. Требуется проверить, является ли файл обычным или он является

каталогом. Если это обычный файл, то сценарий должен выводить имя файла и его

размер. В случае, если размер файла превышает килобайт, то размер должен

выводиться в килобайтах. Если размер превышает мегабайт — в мегабайтах.

|  |  |
| --- | --- |
| #! /bin/bash |  |
|  |  |  |
|  |  | file=$1 |
|  |  | declare -i siz |
|  |  | if $(test -f $file) |
|  |  | then |
|  |  | siz=$(du -b $file | cut -f1) |
|  |  |  |
|  |  | if (( "$siz" > 1024 && "$siz" < 1024\*1024 )) |
|  |  | then |
|  |  | echo "$(du -k $1) in kb" |
|  |  | fi |
|  |  |  |
|  |  | if (( "$siz" > 1024\*1024 )) |
|  |  | then |
|  |  | echo "$(du -m)in mb" |
|  |  | fi |
|  |  |  |
|  |  | if (( "$siz" < 1024 )) |
|  |  | then |
|  |  | echo "$1: $siz b" |
|  |  | fi |
|  |  |  |
|  |  | fi |

Задача 3. Написать скрипт, находящий в заданном каталоге и во всех его

подкаталогах все файлы, заданного размера в заданном каталоге (tty1, tty2) под вашейимя каталога

задаётся пользователем в качестве первого аргумента командной строки). Диапазон

(tty1, tty2) под вашеймин.– макс.) размеров файлов задаётся пользователем в качестве второго и

третьего аргументов командной строки. Cкрипт выводит результаты поиска в файл

(tty1, tty2) под вашейчетвертый аргумент командной строки) в виде: полный путь, имя файла, его размер.

На консоль выводится общее число просмотренных файлов. Проверить работу

программы для каталога /log череusr и диапазона (tty1, tty2) под вашеймин.– макс.) 1000 1010.

|  |
| --- |
| #! /bin/bash |
|  | find $1 -size +$2 -size -$3 -printf "%p %s bytes\n" > $4 |
|  | echo "$(find $1 -type f | wc -l) files were checked" |