1. \*\*Создание директории и файла:\*\*

Создайте директорию с именем "MyDirectory" и в этой директории создайте файл "MyFile.txt". Затем выведите список файлов и директорий в текущем каталоге.

#!/bin/bash

mkdir MyDirectory

cd MyDirectory

touch MyFile.txt

cd ..

ls

2. \*\*Копирование файлов:\*\*

Скопируйте все файлы с расширением ".txt" из одной директории в другую. Затем выведите список скопированных файлов.

#!/bin/bash

SOURCE\_DIR="/source\_directory"

TARGET\_DIR="/ target\_directory"

cp "$SOURCE\_DIR"/\*.txt "$TARGET\_DIR"

ls "$TARGET\_DIR"/\*.txt

3. \*\*Поиск слова:\*\*

Напишите скрипт, который будет искать все файлы в текущей директории и ее поддиректориях, содержащие слово "ключевое\_слово". Выведите список найденных файлов.

#!/bin/bash

grep -rl "ключевое\_слово" .

4. \*\*Архивирование и распаковка:\*\*

Создайте архив (tar) из нескольких файлов и директорий, а затем распакуйте его. Убедитесь, что файлы восстановлены корректно.

#!/bin/bash

SOURCE\_DIR="/home/user/project"

FILES\_TO\_ARCHIVE="file1.txt file2.txt dir1 dir2"

ARCHIVE\_NAME="my\_archive.tar"

TARGET\_DIR="/home/user/unpacked\_project"

cd $SOURCE\_DIR

tar -cvf $ARCHIVE\_NAME $FILES\_TO\_ARCHIVE

mkdir -p $TARGET\_DIR

tar -xvf $ARCHIVE\_NAME -C $TARGET\_DIR

if [ $? -eq 0 ]; then

echo "Архив $ARCHIVE\_NAME успешно распакован в $TARGET\_DIR."

else

echo "Ошибка при распаковке архива."

exit 1

fi

echo "Содержимое директории $TARGET\_DIR:"

ls -R $TARGET\_DIR

5. \*\*Обработка текстового файла:\*\*

Создайте текстовый файл с несколькими строками текста. Напишите скрипт, который будет читать файл и выдавать каждую строку в обратном порядке.

#!/bin/bash

FILE="example.txt"

while IFS= read -r line; do

reversed\_line=$(echo "$line" | rev)

echo "$reversed\_line"

done < "$FILE"

6. \*\*Автоматизация резервного копирования:\*\*

Напишите скрипт, который будет регулярно (например, каждую неделю) создавать резервные копии определенных директорий и сохранять их с датой в имени файла.

#!/bin/bash

SOURCE\_DIRECTORIES="/home/user/documents /home/user/pictures /var/www/html"

BACKUP\_DIRECTORY="/home/user/backups"

CURRENT\_DATE=$(date +%Y-%m-%d)

ARCHIVE\_NAME="backup-$CURRENT\_DATE.tar.gz"

ARCHIVE\_PATH="$BACKUP\_DIRECTORY/$ARCHIVE\_NAME"

mkdir -p "$BACKUP\_DIRECTORY"

tar -czvf "$ARCHIVE\_PATH" $SOURCE\_DIRECTORIES

Добавляем в cron

crontab –e

0 2 \* \* 0 /backup\_script.sh

7. \*\*Подсчет количества слов:\*\*

Напишите скрипт, который будет принимать текстовый файл в качестве аргумента и подсчитывать количество слов в этом файле.

#!/bin/bash

if [ $# -eq 0 ]; then

echo "Использование: $0 <имя\_файла>"

exit 1

fi

FILE="$1"

if [ ! -f "$FILE" ]; then

echo "Файл $FILE не существует."

exit 1

fi

WORD\_COUNT=$(wc -w < "$FILE")

echo "Количество слов в файле '$FILE': $WORD\_COUNT"

8. \*\*Создание случайных паролей:\*\*

Напишите скрипт, который будет генерировать случайные пароли заданной длины и сохранять их в файл.

#!/bin/bash

read -p "Длина пароля:" length

read -p "Имя файла:" filename

read -p "Количество паролей:" number

# Функция для генерации случайного пароля

generate\_password() {

local length="$1"

local password=$(openssl rand -base64 48 | tr -dc 'A-Za-z0-9@#$%&\*()-\_=+' | head -c "$length")

echo "$password"

}

# Проверка на наличие аргументов

if [ -z "$length" ] || [ -z "$filename" ]; then

echo "Необходимо указать длину пароля и имя файла."

exit 1

fi

# Проверка на число в длине пароля и количестве паролей

if ! [[ "$length" =~ ^[0-9]+$ ]] || ! [[ "$number" =~ ^[0-9]+$ ]]; then

echo "Длина пароля и количество паролей должны быть целыми числами."

exit 1

fi

# Генерация паролей и сохранение их в файл

echo "Генерация $number паролей длиной $length символов в файл $filename..."

for ((i=1; i<=number; i++)); do

password=$(generate\_password "$length")

echo "$password" >> "$filename"

done

echo "Пароли успешно сгенерированы и сохранены в $filename."

9.Подсчет файлов:

Напишите скрипт, который будет использовать цикл for для подсчета количества файлов и директорий в текущей директории.

#!/bin/bash

file\_count=0

directory\_count=0

for item in \*; do

if [ -f "$item" ]; then

((file\_count++))

elif [ -d "$item" ]; then

((directory\_count++))

fi

done

echo "В текущей директории:"

echo "Файлов: $file\_count"

echo "Директорий: $directory\_count"

10. \*\*Автоматизация задачи обновления системы:\*\*

Напишите скрипт, который будет проверять наличие обновлений системы и, если они доступны, автоматически устанавливать их.

#!/bin/bash

echo "Проверка наличия обновлений..."

sudo apt update > /dev/null 2>&1

updates=$(apt list --upgradable 2>/dev/null | grep -c upgradable)

if [ "$updates" -eq 0 ]; then

echo "Нет доступных обновлений."

else

echo "Доступно $updates обновлений. Установка..."

sudo apt upgrade -y > /dev/null 2>&1

echo "Обновления успешно установлены."

fi