

# Разбор практического задания

## Первое задание

Опираясь на пример с реализацией скрипта *mvdir*, напишите скрипт *rename*, который будет переименовывать файл.

Открываем текстовый редактор Vim.  
Скрипт начинается с шебанга и пути до shell.

```
#!/bin/bash
```

Затем, для того чтобы переименовать файл, вам требовалось скопировать файл с новым именем, а затем удалить старый файл.

Для того чтобы мы могли брать значения из параметров, переданных скрипту, мы используем позиционные аргументы. Логично предположить, что аргументами нашего скрипта будет текущее имя файла и новое имя файла:

```
cp $1 $2
```

Удаляем старый файл. Лучше использовать *-f*, так как мы не хотим, чтобы скрипт был интерактивным — запрашивал наше подтверждение действия.

```
rm -f $1
```

Готово! Давайте сохраним скрипт, далее нам потребуется добавить ему право на исполнение:

```
chmod +x rename
```

И теперь попробуем его в работе. Не забываем, что нужно указать путь:

```
./rename file image.png  
ls -la
```

Готово, файл с новым именем существует, файл со старым — нет.

## Второе задание

Написать cron-задание для перемещения картинок (jpeg), аудиофайлов (mp3) и документов (docx) из папки Downloads в папки Pictures, Music и Documents соответственно. Задачи должны выполняться в 23:00 каждый день.

Для выполнения нужны однострочники, которые будут перемещать файлы, а также потребуется скопировать их в crontab и написать для них расписание.

Задание для перемещения картинок у вас уже есть.

```
find /home/victoria/Downloads -name "*.jpeg" -exec mv {}  
/home/victoria/Pictures \;
```

По аналогии с ним пишем однострочники для картинок и музыки:

```
find /home/victoria/Downloads -name "*.mp3" -exec mv {}  
/home/victoria/Music \;  
find /home/victoria/Downloads -name "*.docx" -exec mv {}  
/home/victoria/Documents \;
```

Теперь откроем на редактирование конфигурацию cron:

```
crontab -e
```

Скопируем наши однострочники.

Составим расписание: нам потребуется выполнять задачу каждый день каждого месяца каждого года в 23 часа 00 минут:

```
0 23 * * *
```

Теперь закроем редактор. Готово!

Крон-задачи для своего пользователя можно посмотреть при помощи команды:

```
crontab -l
```

## Третье задание

Как будет выглядеть расписание, которое будет что-то делать в ваш день рождения каждый год?

У каждого из вас здесь будет свой ответ, покажем на примере, как это решается.

```
crontab -e
```

Предположим, день рождения 8 марта. То есть нужно сделать так, чтобы скрипт выполнялся в восьмой день третьего месяца каждый год.

Давайте пойдём с конца. Каждый год это \*, далее месяц 3 и порядковый номер дня в месяце 8. В часах и минутах, следуя традиции, проставим нули. Таким образом скрипт сработает ровно в 00:00 в момент наступления 8 марта.