

Разбор практического задания

Первое задание

Доработать скрипт `delete.sh` с учётом того, что файл может быть симлинком или хардлинком. В случае, если это символическая ссылка, необходимо удалять её без архивирования. Если файл — жёсткая ссылка, удалять файл обычным способом и выводить пользователю список ссылок на этот файл.

Добавим переменную, в которой будем хранить `inode` файла. Узнаем её с помощью утилиты `stat` с ключом и шаблоном.

Далее добавим ещё одну, в которой будем хранить имя временного файла, в который будем записывать промежуточные данные.

Теперь нам нужно обрабатывать случай с символической ссылкой. При помощи квадратных скобок, то есть команды `test`, проверим, является ли файл симлинком. Если да, то узнаем, куда она указывает, и далее удалим ссылку командой `unlink` или `rm`, и выведем пользователю сообщение о том, что это была за ссылка и куда она указывала.

Теперь обрабатываем случай с хардлинком.

У нас есть номер `inode`. Давайте теперь узнаем с помощью команды `find`, сколько ещё файлов с такой же `inode` существует.

Отправляем весь `stderr` в `/dev/null`. А `stdout` — во временный файл. Искать файлы мы будем от корня.

Теперь у нас может быть два варианта:

- 1) количество файлов с такой `inode` равно одному, и тогда мы перейдём к архивированию;
- 2) файла будет хотя бы два — значит, мы имеем дело с хардлинком.

Чтобы узнать, сколько у нас есть файлов, посчитаем количество строк в промежуточном файле с выводом команды `find`.

Если у нас таких файлов два или больше, удалим файл и выведем сообщение, что это был хардлинк и на те же данные есть другие ссылки в виде имён файлов.

И удалим временный файл, так как после работы скрипта он нам больше не понадобится.

Давайте создадим файл:

```
touch file
```

Симлинк:

```
ln -s file symlink
```

И несколько хардлинков:

```
ln -s file hardlink1
```

```
ln -s file hardlink2
```

И протестируем наш скрипт.

Второе задание

Написать systemd-юнит который будет запускать очистку файлов в директории ~/TRASH (которые попадают туда после работы скрипта delete для безопасного удаления файлов, реализованного в прошлых модулях) раз в час, удаляя файлы старше дня.

Как мы помним, для этого нам нужно написать юнит service, который будет запускать нужный нам процесс, и юнит timer, который будет делать это по расписанию.

```
sudo vim /etc/systemd/system/trash-cleaner.service
sudo vim /etc/systemd/system/trash-cleaner.timer
systemctl enable trash-cleaner.service
systemctl enable trash-cleaner.timer
```

Третье задание

Написать скрипт, который будет превращать символические ссылки в обычный файл с содержимым, находящимся по символической ссылке.

Нам потребуется утилита readlink, которая возвращает путь, на который указывает символическая ссылка.

Начнём наш скрипт с проверки количества введённых аргументов.

Затем проверим, является ли переданный файл символической ссылкой. (При помощи ключа команды test -L.)

Используем утилиту mv: первым аргументом передадим путь, по которому находится файл, на который указывает символическая ссылка, а вторым — имя симлинка, которую будет превращать в обычный файл.

Давайте протестируем:

```
echo "testfile" > file
ln -s /home/victoria/m11dz/file symlink
```

Для начала давайте попробуем указать обычный файл в качестве аргумента скрипта:

```
./script3.sh file
```

Как и предполагалось, получим ошибку:

```
./script3.sh symlink
ls -la symlink
```

Как мы видим, теперь это обычный файл:

```
cat symlink
```

С тем же содержимым, что и оригинальный файл.

Четвёртое задание

Создадим файл, запишем в него слово Hello,

```
echo "Hello" > file
```

Создадим символическую ссылку на этот файл.

```
ln -s file symlink
```

Теперь скопируем символическую ссылку:

```
cp symlink /tmp/
```

Давайте посмотрим, что у нас получилось:

```
ls -la /tmp/symlink
```

Символическая ссылка превратилась в обычный файл.

Для того чтобы ссылка осталась ссылкой, мы можем использовать ключ -P команды cp. Чтобы ссылка не стала битой, всегда указывайте абсолютный путь.

Например:

```
ln -s /home/victoria/m11dz/file symlink2
```

Тогда при копировании ссылка останется верной:

```
cp -P symlink2 /tmp/symlink2  
ls -la /tmp/symlink2
```