## Продвинутая работа с bash

### Цель работы

Научиться работать с контрольными суммами для отслеживания изменений и целостности файлов пакетов.

### Задания для выполнения

1. Создаём пустой файл и вписываем что-нибудь
2. Копируем с сохранением всех прав в другой файл с любым названием и содержимым
3. Сравниваем контрольные суммы
4. Меняем содержимое любого из файлов и смотрим разницу в контрольных суммах
5. Создаём пользователя и задаём ему пароль
6. Разбираемся с /etc/passwd, /etc/shadow, /etc/group
7. Делаем так, чтобы, если нет файла /tmp/passwd.backup, /etc/passwd копировался в /tmp/passwd.backup, а если /tmp/passwd.backup есть, то сравнивался с /etc/passwd
8. Если различаются, то пусть копируется, если нет, пусть завершает работу скрипта

### Методические указания[[1]](#footnote-1)

Контрольная сумма - это цифра или строка, которая вычисляется путем суммирования всех цифр нужных данных. Ее можно использовать в дальнейшем для обнаружения ошибок в проверяемых данных при хранении или передаче. Тогда контрольная сумма пересчитывается еще раз и полученное значение сверяется с предыдущим.

Контрольные суммы Linux с вычисляемые по алгоритму MD5 (Message Digest 5) могут быть использованы для проверки целостности строк или файлов. MD5 сумма - это 128 битная строка, которая состоит из букв и цифр. Суть алгоритма MD5 в том, что для конкретного файла или строки будет генерироваться 128 битный хэш, и он будет одинаковым на всех машинах, если файлы идентичны. Трудно найти два разных файла, которые бы выдали одинаковые хэши.

В Linux для подсчета контрольных сумм по алгоритму md5 используется утилита md5sum. Вы можете применять ее для проверки целостности загруженных из интернета iso образов или других файлов.

Эта утилита позволяет не только подсчитывать контрольные суммы linux, но и проверять соответствие. Она поставляется в качестве стандартной утилиты из набора GNU, поэтому вам не нужно ничего устанавливать.

Синтаксис команды md5sum очень прост:

$ md5sum опции файл

Опций всего несколько и, учитывая задачи утилиты, их вполне хватает:

-c - выполнить проверку по файлу контрольных сумм;

-b - работать в двоичном формате;

-t - работать в текстовом формате;

-w - выводить предупреждения о неверно отформатированном файле сумм;

--quiet - не выводить сообщения об успешных проверках.

Сначала скопируйте файл /etc/group в домашнюю папку чтобы на нем немного поэкспериментировать:

cp /etc/group groups

Например, давайте подсчитаем контрольную сумму для файла /etc/group:

md5sum groups

Или вы можете сохранить сразу эту сумму в файл для последующей проверки:

md5sum groups > groups.md5

Затем каким-либо образом измените этот файл, например, удалите первую строчку и снова подсчитайте контрольные суммы:

md5sum groups

Как видите, теперь значение отличается, а это значит, что содержимое файла тоже изменилось. Дальше верните обратно первую строчку root:x:0: и скопируйте этот файл в groups\_list и

cp groups groups\_list

Затем опять должна быть выполнена проверка контрольной суммы linux:

md5sum groups\_list

Сумма соответствует первому варианту, даже несмотря на то, что файл был переименован. Обратите внимание, что md5sum работает только с содержимым файлов, ее не интересует ни его имя, ни его атрибуты. Вы можете убедиться, что оба файла имеют одинаковые суммы:

md5sum groups groups\_list

Вы можете перенаправить вывод этой команды в файл, чтобы потом иметь возможность проверить контрольные суммы:

md5sum groups groups\_list > groups.md5

Чтобы проверить, не были ли файлы изменены с момента создания контрольной суммы используйте опцию -c или --check. Если все хорошо, то около каждого имени файла появится слово OK или ЦЕЛ:

md5sum -c groups.md5

Но теперь вы не можете переименовывать файлы, потому что при проверке утилита будет пытаться открыть их по имени и, естественно, вы получите ошибку. Точно так же все работает для строк:

$ echo -n "Losst" | md5sum -

$ echo -n "Losst Q&A" | md5sum -

Из этой статьи вы узнали как выполняется получение и проверка контрольной суммы linux для файлов и строк. Хотя в алгоритме MD5 были обнаружены уязвимости, он все еще остается полезным, особенно если вы доверяете инструменту, который будет создавать хэши.

Проверка целостности файлов Linux - это очень важный аспект использования системы. Контрольная сумма файла Linux используется не только вручную при проверке загруженных файлов, но и во множестве системных программ, например, в менеджере пакетов.

### Контрольные вопросы

1. Какие еще существуют алгоритмы вычисления контрольных сумм?
2. Для чего применяются контрольные суммы?

### Дополнительные задания

1. Возьмите скрипт, который был создан на предыдущей лабораторной работе. Создайте установочный скрипт, который проверяет целостность скрипта, сравнивая контрольную сумму с заранее сохраненным значением. При совпадении он должен создать задачу по расписанию, а при несовпадении - выдать ошибку и завершиться.
2. Создайте скрипт, который проверяет целостность основных системных конфигурационных файлов и при обнаружении ошибки восстанавливает их из архивной копии. Добавьте возможность пересчета контрольных сумм при санкционированных изменениях (одновременное сохранение новых копий в архиве). Добавьте возможность регулярной проверки. Реализуйте все функции в одном скрипте, использующем несколько опций.

1. По материалам <https://losst.ru/proverka-kontrolnoj-summy-linux> [↑](#footnote-ref-1)