Сжатие данных.

Немного теории.

Самая популярная для Linux утилита для архивации - tar. Она используется почти везде, для архивации исходников, упаковки пакетов. Для сжатия используются другие утилиты, в зависимости от алгоритма сжатия. Например, zip, bz, xz, lzma и т д. Сначала выполняется архивация, затем сжатие, отдельными программами. Автоматический запуск некоторых утилит сжатия для только что созданного архива поддерживается в tar и других подобных программах с помощью специальных опций.

Дело в том, что сам tar не умеет уменьшать вес архива — эта программа всего лишь упаковывает файлы и папки в несжатом виде.

Можно архивировать данные данные одной лишь утилитой TAR, а можно сначала создать обычный архив с помощью tar, а потом сжать его утилитой специально предназначенной для сжатия. В этом случае мы получаем больше контроля над процессом сжатия.

Сжатие данных без потерь — класс алгоритмов сжатия данных (видео, аудио, графики, документов, представленных в цифровом виде), при использовании которых закодированные данные однозначно могут быть восстановлены с точностью до бита, пикселя, вокселя и т.д. При этом оригинальные данные полностью восстанавливаются из сжатого состояния.

Gzip — один из самых популярных алгоритмов сжатия, который позволяет уменьшить размер файла и сохранить исходный режим файла, право собственности и временную метку.

GZIP обеспечивает сжатие без потерь, исходные данные можно полностью восстановить при распаковке.

GZIP основан на алгоритме DEFLATE, который представляет собой комбинацию кодирования LZ77 и Хаффмана. DEFLATE задумывался как замена LZW и другим запатентованным алгоритмам сжатия данных, которые в то время ограничивали удобство сжатия других архиваторов.

В соответствии с традициями UNIX-программирования, gzip выполняет только две функции: сжатие и распаковку одного файла; упаковка нескольких файлов в один архив

невозможна. При сжатии к оригинальному расширению файла добавляется суффикс .gz. Для упаковки нескольких файлов обычно их сначала архивируют (объединяют) в один файл утилитой tar, а потом этот файл сжимают с помощью gzip. Таким образом, сжатые архивы обычно имеют двойное расширение .tar.gz, либо сокращённое .tgz.

Zip — это наиболее широко используемый формат архивных файлов, поддерживающий сжатие данных без потерь. Наиболее часто в ZIP используется алгоритм сжатия Deflate.

Команда **bzip2** предназначена для сжатия данных без потерь с помощью соответствующей утилиты. Целью использования данной утилиты является экономия дискового пространства. Упомянутый алгоритм позволяет достичь лучшей степени сжатия данных, чем тот, который реализован в рамках утилит gzip и zip, но зачастую худшей степени сжатия данных, чем тот, который реализован в рамках утилиты xz. Кроме того, на уровне декомпресии данных он является более ресурсоемким, чем алгоритм, который реализован в рамках утилиты xz.

LZMA - Алгоритм основан на схеме сжатия данных по словарю, сходной с использованной в LZ77, и обеспечивает высокий коэффициент сжатия (обычно превышающий коэффициент, получаемый при сжатии с использованием bzip2), а также позволяет использовать словари различного размера (до 4 Гб).

xz – программа сжатия без потерь и формат файла, который включает алгоритмы сжатия LZMA / LZMA2. Формат XZ является форматом сжатия одного файла и не предлагает возможности архивирования.

Переходим к практике.

Начнем с утилиты TAR.

Всегда забываю ключи для тех или иных форматов.

$ tar --help

-c создать,

-x извлечение,

-v подробный листинг,

-f - архив,

-r - добавление файлов в конец архива,

-p извлекать информацию о правах доступа к файлу,

--exclude=ШАБЛОН исключать файлы,

--same-owner попытаться извлечь файлы с тем же владельцем, что и в архиве.

$ tar -cvzf Archive.tar.gz Folder/

Просмотр содержимого архива

$ tar -tf Archive.tar.gz

Всё время забываю как распаковать в указанную директорию.

$ tar -C "folder" -xvf Archive.tar.gz

Извлечь одну директорию, один файл или несколько файлов по маске.

$ tar -C "folder" -xvf Archive.tar.gz image/*.png

Извлечение группы файлов

$ tar xvf sedicomm.tar --wildcards '*.txt'

$ tar zxvf sedicomm.tar.gz --wildcards '*.txt'

$ tar jxvf sedicomm.tar.bz2 --wildcards '*.txt'

У tar нет возможности добавлять файлы или каталоги в существующий сжатый файл tar.gz или tar.bz2. Но можно добавить в конец несжатого архива tar.

$ tar -vrf Archive.tar image/

Например Backup системы. Так уже никто не делает. Для этого существует rsync и другие утилиты, в т.ч. графические. Но способ вполне до сих пор рабочий.

$ tar -cvzpf backup.tar.gz --exclude=/dev --exclude=/proc --exclude=/mnt --exclude=/sys / --exclude=/lost+found --exclude=/tmp --exclude=/backup.tar.bz2 /

Восстановление из Backup.

$ tar --same-owner -xvpf backup.tar.gz -C /mnt/

Для всех дальнейших манипуляций со сжатием данных нам понадобится архив без сжатия - tar. Сжимать будем другими утилитами.

$ tar -cvf Archive.tar Folder/

Опции разных утилит.

Во всех утилитах нам важны несколько опций. В некоторых утилитах схожие опции могут отсутствовать. Ключи могут быть разными, поэтому обращаем внимание на такие понятия как:

compress - сжатие,

decompress - распаковка,

keep (don't delete) input files - не удалять исходные файлы,

verbose mode - отображение процесса,

recursive для рекурсивного обхода директорий,

-1 -9 (fast, best) уровень сжатия (чем больше цифра, тем больше сжатие),

force (force overwrite of output file and compress links) - принудительная перезапись выходного файла и сжатие ссылок,

list (list compressed file contents) - список содержимого сжатого файла,

stdout (write on standard output) - запись на стандартный вывод.

При наличие опции stdout, чтобы сохранить исходные файлы у нас есть только 2 способа сделать это. Использовать параметр **-c** который сообщает gzip о необходимости записи в стандартный вывод и перенаправить вывод в файл. Или использовать параметр **-k** (keep).

GZIP, GUNZIP

$ gzip --help

Упаковать, с перезаписью сжатых данных. Без перезаписи - уберите параметр -f.

$ gzip -frkv9 Archive.tar

Обратите внимание, что при использовании опции просмотра списка сжатого содержимого архива, всё что вы увидите - архив.tar

$ gzip -l Archive.tar.gz

Лучше используйте для этого TAR.

Аналог ls.

$ tar -tf Archive.tar.gz

Аналог ls -la.

$ tar -tvf Archive.tar.gz

Аналогично ключ -f. Без него, при наличие файлов для перезаписи, система спросит дальнейшие действия.

$ gunzip -fv Archive.tar.gz

XZ

$ xz --help

Сжать с перезаписью, уровень сжатия максимальный.

$ xz -fzkv9 Archive.tar

Разжать с перезаписью.

$ xz -dvf Archive.tar.xz

LZMA

$ lzma --help

Аналогично сжать.

$ lzma -kvzf9 Archive.tar

Разжать.

$ lzma -dvf Archive.tar.lzma

BZIP2

$ bzip2 --help

Сжать.

$ bzip2 -kvzf9 Archive.tar

Разжать.

$ bzip2 -dvf Archive.tar.bz2

ZIP

Сжать, с выводом процесса каждого файла.

$ zip -rv9 Archive.zip Folder/

Для подавления вывода - q.

$ zip -rq9 Archive.zip Folder/

Указать метод сжатия bzip2

$ zip -rq9 -Z bzip2 Archive.zip Folder/

Добавить файлы в архив.

$ zip Archive.zip -rq9 image/ -Z bzip2

Удаление файла из архива

$ zip -d Archive.zip image/scr.png

Разжать.

$ unzip --help

Содержимое архива. Вот здесь содержимое более-менее адекватное.

$ unzip -l Archive.zip

Извлечь всё.

$ unzip Archive.zip

Извлечь всё в папку.

$ unzip Archive.zip -d tmp

Извлечь одну папку в указанную директорию.

$ unzip Archive.zip image/* -d tmp/

Немного массивных тестов.

Все команды проверялись на папках объёмом *~3 ГБ,* в которых находились медиа-файлы формата mp4 приемлемого качества (*Image 1080p, Video codec H264, Audio codec AAC, 128kbit/s Audio bitrate*). Возможно тест не самый объективный, т.к. не хватает аудио файлов, изображений и документов для полноты картины. Однако, даже при таких скудных данных имеются положительные результаты для небольшого сравнения.

BZIP2 для такого объема работала около *15 минут*.

Для **LZMA** сжатие заняло около *30 минут*.

XZ сжимало данные не более *10 минут*.

GZIP также сжало данные достаточно быстро, примерно *минут 5...10*.

ZIP сжимало папку с данными около *10 минут*.

По итоговому объему архивов самыми выигрышными оказались **GZIP** и **XZ** - *2,5 ГБ*.

Самое быстрое чтение у **GZIP**. Тоже касается и распаковки данных.

Самым медленным оказался **BZIP2** и **LZMA**.

Моё дело маленькое — лишь заинтересовать вас.

Подписывайтесь на канал, комментируйте, ставьте лайки.

Ну а с вами как всегда был Shadow.

Всем Добра и Удачи!