#bash #genisoimage #mkisofs #AcetoneISO #7z #QemuImg #VHD #raw #NTFS #EXT4 #VeraCrypt #UDF #HFS #ISO9660 #FAT32

История одного ISO или как не потерять данные. Часть №2. Практика.

Сегодня мы поговорим о создании образа из папки в Linux.

Казалось бы - тривиальная задача, что может быть проще?

Но, даже в таких простых вопросах утилиты с графическим интерфейском (GUI) могут серьёзно подвезти.

А консольные утилиты (CLI) и вовсе могут завести в тупик при решении подобных задач.

Давайте разбираться во всем вместе, и решать все возникающие по ходу дела вопросы.

Для примера возьмем Linux-игру - Don't Starve Together. Да, она под Linux.

Она идеально подходит для наших тестов!

И не пишите, что я мог бы взять что-нибудь поменьше или попроще.

Так было бы - а) не интересно, б) мы не увидели бы всех возможных ошибок и проблем, и в) Учатся ведь на ошибках, что мы и сделаем в этой статье.

В следующей статье мы поговорим о блочных устройствах разных ОС - Linux и Windows. Их особенностях, возможностях и ограничениях.

Ну а с вами как всегда был Shadow.

Подписывайтесь на канал, ставьте лайки, комментируйте.

Всем Добра и Удачи!

Создание образа из папки.

# Подготовка.

Чуть не забыл самое главное. По умолчанию все утилиты - что консольные, что графические создают образ из папки, т.е. если вам нужно упаковать не только файлы из папки, но и саму папку - создайте ещё одну папку с любым именем и переместите всё в неё.

При указании папки укажите только что созданную.

Например у меня: DontStarveTogether - внутри файлы игры.

$ mkdir -p Dont_Starve_Together

$ mv DontStarveTogehter Dont_Starve_Together/

$ ls Dont_Starve_Together/

# DontStarveTogehter

# AcetoneISO, genisoimage.

Первое что каждый из нас, скорее всего сделает (включая меня), особенно если лень - попробует создать образ из папки при помощи утилиты AcetoneISO.

Но при попытке создать образ возникает ошибка 255.

Почему же она возникает? Об этом немного позже.

Теперь попробуем сделать тоже самое, но с помощью консольных утилит (mkisofs или genisoimage никакой разницы).

$ genisoimage -o DontStarveTogether_Linux.iso Dont_Starve_Together/

Примонтируем наш новенький iso образ или откроем его через архиватор и посмотрим, что у нас получилось внутри образа.

О УЖАС! Почему папки какие-то обрезанные и переименованные? Более того, при детальном изучении образа окажется, что внутри него нет папок и файлов с вложенностью более 8-ми!

Неужели всё так плохо и срочно надо пытаться запускать UltraISO под wine-ом?

**Давайте разбираться!**

В предыдущей статье мы говорили об ограничениях различных файловых систем. Вы спросите а при чем тут **ISO-образ**?

А при том, что у него тоже имеется файловая система, только именуется она **форматом образа** и тоже **имеет свои ограничения**.

В некоторые форматы можно впихнуть вплоть до внутренней разбивки разделов блочных устройств, а в некоторых лишь небольшую информацию. Первая - это комбинированный образ диска, который может состоять из нескольких форматов - например, HFS+UDF и ISO-9660. Второй нам уже известен - ISO-9660, например.

По умолчанию все графические утилиты создают образ из папки по строго заданным предустановленным командам тех же самых консольных утилит.

Да вы можете взять исходный код AcetoneISO и перекомпилировать. Но тогда в следующий раз для новой задачи вам придется взять его и вновь перекомпилировать под новую задачу.

А ведь так с любыми графическими утилитами. Не только с AcetoneISO. Это был лишь пример!

Создадим регулярное выражение, с помощью которого и определим максимальную вложенность наших директорий на скорую руку. Под каждую задачу выражение надо будет переписывать.

```
$ find ./Dont_Starve_Together/ -type d | tr -d '[:alnum:]' | tr -d '_' | tr -d '-' | tr -d '.' | tr -d '+' | tr -d '@' | awk '{print length}' | sort -ud | sort -nr | head -n1

# 9
```

И также посмотрим только на совпадающие кодировки файлов.

$ find Dont_Starve_Together/ -type f -exec enca -i {} \; | sort -ud

# ... UTF-7 ...

Вот тут и встречается те самые ограничения. для ISO-9660, HFS и UDF.

Если помните, то мы говорили, что у формата ISO-9660 должен быть уровень соответствия по именам файлов, регистрам, вложенности, длине имени и другим параметрам.

Вернёмся к консольной утилите genisoimage. Что же с ней не так?

А с ней всё так.

Просто, наши файлы не совсем правильные, из которых мы пытаемся создать образ диска.

А именно - кодировка, вложенности, имена файлов, длина имени, вложенность директорий и другие.

Что же тогда делать?

Для решения этой проблемы не хватает нескольких ключей и параметров.

Разберем все ключи и параметры по порядку.

# Ключи и параметры genisoimage.

**Обратите внимание, что некоторые параметры надо использовать с осторожностью, т.к. некорректное чтение файлов и последующая их упаковка в образ может грозить ошибкой чтения данных и/или монтирования образа диска!**

-R

Расширения Rock Ridge записываются поверх файловой системы ISO 9660. Так, что оптический диск с Rock Ridge может быть прочитан программным обеспечением, рассчитанным на работу с ISO 9660.

-UDF

Уходим от ISO стандарта, особенно если у нас большие файлы или много файлов, где итоговый объем получается большим.

**Хорошо сочетается с предыдущим ключом.**

-no-iso-translate

Не переводить символы '#' и '~', являющие некорректными для имен файлов iso9660. Эти символы, хотя и некорректны, но тем не менее часто используются в системах Microsoft. Это является нарушением стандарта ISO9660, но работает на многих системах.

**Использовать осторожно.**

-r

Редко используется. **Чаще всего не нужен, но знать о нём необходимо.** Владельцы файлов и режимы переопределяются в более логичные значения. Т.е. права доступа перезаписываются для всех файлов и каталогов только-для-чтения и выполнения. При использовании на Win32, бит исполнения устанавливается для всех файлов. Это является следствием недостатка прав файлов в системе Win32 и на уровне эмуляции Cygwin POSIX.

-iso-level 4

На уровне 3 нет никаких ограничений по уровням соответствий. + соответствие уровням согласно нововведениям. Уровень 4 — это тот же ISO-9660, но с ключом -R формат UDF более корректно, как бы послойно, накладывается на последний с помощью Rock Ridge.

-U

Разрешает "непереводимые" имена файлов, совершенно нарушая форматы iso9660, описанные выше. Принудительно активирует флаги -d, -l, -L, -N, -relaxed-filenames, -allow-lowercase, -allow-multidot и -no-iso-translate.

Позволяет иметь одну точку в имени файла, а также символы в верхнем регистре. Это полезно только на системе HP-UX, где встроенная файловая система CDFS не распознает НИКАКИЕ расширения.

**Использовать максимально осторожно.**

-D

Не использовать глубокое перераспределение каталогов, в вместо этого просто упаковать их в том виде, что мы их видим. Это является нарушением стандарта ISO9660, но работает на многих системах.

**Использовать осторожно.**

В принципе это вполне достаточно, чтобы корректно создать образ диска из сложных входных данных.

## Итоговая команда:

$ genisoimage -UDF -D -U -no-iso-translate -iso-level 4 \

-o DontStarveTogether_Linux.iso Dont_Starve_Together/

Без опций -D -U и -no-iso-translate — genisoimage всё равно перемещает некоторые файлы и папки и именует их по согласно стандарту UDF.

А с ними, грубо говоря создается образ 1 к 1, без изменений. Это не есть хорошо, но иногда выхода нет и приходится чем-то жертвовать. **Стоит использовать их осторожно и всегда тщательно перепроверять результат.**

Можете отказаться от опций -iso-level, или установить её в уровень 3. Но тогда вам придётся мириться с тем, что у вас будет не только формат UDF, но и не все файлы и папки могут корректно прочитаться из-за различия кодировок. Может понадобится дополнительная опция -input-charset или output-charset. И даже в это случае первый вариант выигрывает и по скорости и по корректности чтения данных. Но это не значит, что работать не будет — Будет, ещё как! Но гораздо хуже.

Тоже самое касается параметра -R. Обычно его не используют в случаях создания комбинированных образов, хотя бывают и исключения. В обычных бытовых ситуациях параметр самый полезный из всех опций. Однако, **при его использовании обязательно** стоит **указывать** одну из 2 опций: **-UDF или -udf. Регистр важен!** Первый обычный формат UDF, благодаря -R накладывается на ISO с помощью Rock Ridge, второй рационализированный UDF, который также накладывается на ISO с помощью Rock Ridge.

Образ диска прекрасно монтируется, открывается и работает - и в Linux и в Windows.

**Теперь всё точно под вашим контролем!**

А ещё можно по-колдовать с форматом HFS и другими, создавая комбинированные образы. Но об этом мы поговорим с вами как-нибудь в другой раз.

**P.S.:** Конечно же есть ещё один способ создания образов, который гарантирует целостность ваших данных, а также не требует каких-то дополнительных и сложных утилит или навыков. Простой, но слегка замороченный. О нём мы поговорим в следующей статье про блочные устройства!

man genisoimage, man mkisofs.

Надеюсь, я, хоть немного, вас заинтересовал. Всем до встречи. Пока-Пока.