Автор: Administrator
01.11.09 16:28 -

HOWTO diff and patch

Содержание
1 Введение
2 Использование diff для создания простого патча
3 Применение простого патча, который мы создали
4 Контекстный патч
5 Получение различий между несколькими файлами
6 Применение патча к нескольким файлам
7 Восстановление оригинального файла из пропатченного
8 Унифицированный формат
9 Сравнение форматов
10 Изменение количества контекстных строк
11 Заключительные слова
12 Источники

[править]
Введение

Команды diff и patch представляют собой мощную комбинацию. Они часто используются для получения разницы между оригинальными файлами и обновленными, чтобы другие люди, у которых есть только оригинальные файлы, могли обновить их, используя файлы, которые содержат только разницу. В этой статье показаны базовые принципы использования этих команд.

В этой статье используются без объяснения некоторые базовые команды Linux, такие как смена директории, копирование файлов и редактирование текстовых файлов.
[править]
Использование diff для создания простого патча

Наиболее простой пример использования команды diff — получение различий между двумя файлами, оригинальным и обновленным. Можете, например, написать насколько слов обычного текста, сделать какие-нибудь изменения, и сохранить изменения во второй файл. Теперь вы можете сравнить эти эти два файла, используя команду diff:
diff originalfile updatedfile

Конечно, надо заменить originalfile и updatedfile соответствующими именами файлов. В результате должно получиться что-то вроде этого: Code: diff originalfile updatedfile

1c1

< These are a few words.

 No newline at end of file

> These still are just a few words.

 No newline at end of file

На заметку: Что бы продемонстрировать создание простого патча, я использовал оригинальный файл, содержащий строку «These are a few words.», и измененный файл, содержащий строку «These still are just a few words.». Вы можете создать эти файлы сами, если хотите запустить команду из статьи и получить тот же результат.

1c1 показывает номер строки и то, что с ней надо сделать. Обратите внимание, что может быть сразу несколько строк(например, 12,15, что означает со строки 12 до строки 15). Символ «c» означает, что патч заменит эту строку. Есть еще два других символа: «a» и «d». Они означают «добавить»(add) и «удалить»(delete) соответственно. Таким образом, синтаксис следующий: (номер строки или диапазон строк)(c, a или d)(номер строки или диапазон строк), хотя когда используются «a» или «d», одна из частей (номер строки или диапазон строк) может содержать только номер одной строки.
Когда используется «c», номера строк слева — это строки в оригинальном файле, которые надо заменить строками, находящимися в патче, а номера строк справа — это строки, которые должны быть в пропатченном файле.
Когда используется «a», номер слева может быть только номером одной строки, который показывает, где надо добавить строку в пропатченном файле, а номера строк справа — это строки, которые должны быть в пропатченном файле.
Когда используется «d», номера строк слева — это строки, которые надо удалить, чтобы получить пропатченную версию файла, а номер строки справа может быть только номером одной строки, который показывает где будут строки в пропатченном файле, если они не будут удалены. Вы можете подумать, что последний номер не нужен, но не забывайте, что патч можно применить для восстановления исходного файла. Это будет объяснено позже.

Знак «<» означает, что патч должен удалить символы после этого знака, а знак ">» означает, что символы после этого знака надо добавить. Когда надо заменить строки («c» между номерами строк), вы увидите оба знака: и «<", и ">». Когда надо добавить строку («a» между номерами строк), вы увидите только знак «>», а когда надо удалить строку («d» между номерами строк), вы увидите только знак «<».

Строка « No newline at end of file» появилась из-за того, что я не не нажал enter после того как набрал слова. Считается хорошим тоном заканчивать текстовый файл пустой строкой. Некоторым программам она необходима для работы. Поэтому эта строка

появилась после работы команды diff. Добавим пустые строки в конец файлов, и получим более короткий вывод команды diff: Code: diff originalfile updatedfile
1c1

< These are a few words.

> These still are just a few words.

Как вы возможно заметили, я не объяснил что означают 3 знака «-». Они означают конец строк, которые надо заменить и начало строк на которые надо заменить. Разделение старых и новых строк. Вы увидите этот знак только при замене («c» между номерами строк).

Если мы хотим создать патч, мы должны поместить вывод команды diff в файл. Конечно это можно сделать, скопировав его из консоли и вставив в вашем любимом текстовом редакторе, а затем сохранив этот файл, но есть способ проще. Мы можем с помощью bash направить вывод команды diff в текстовый файл:
diff originalfile updatedfile > patchfile.patch

Опять же не забудьте заменить originalfile и updatedfile на соответствующие имена файлов. Вы наверное знаете, что опция bash «>» работает со всеми командами. Это очень полезное свойство.
[править]
Применение простого патча, который мы создали

Мы можем использовать патч, который только что создали, чтобы получить из оригинального файла обновленный. Для этого надо скопировать оригинальный файл и патч в одно и тоже место. И затем применить патч:
patch originalfile -i patchfile.patch -o updatedfile

Естественно, и здесь надо изменить имена файлов на необходимые. Если все прошло хорошо, должен получиться файл, идентичный обновленному. Вы можете убедиться в этом, используя команду diff с опцией «-s»:
diff -s updatedfile [/path/to/the/original/updatedfile]/updatefile

Замените текст между [и] на путь к оригинальному файлу. Например, если обновленный файл, который вы использовали при создании патча находится в родительской директории вашей текущей, то «[/path/to/the/original/updatedfile]» надо заменить на «..» (bash понимает это как родительскую директорию от текущей). И конечно надо изменить имена файлов на верные.

Поздравляю! Если diff сообщила, что файлы идентичные, вы только что успешно создали и применили патч! Однако формат патча, который мы только что использовали не

единственный. В следующей главе мы рассмотрим другой формат патча.
[править]
Контекстный патч

В первой главе мы создали патч, используя нормальный формат команды diff. Однако этот формат не обеспечивает контекстной зависимости, а использует строки целиком. Создадим патч для того же файла, но используя контекстный формат:
diff -c originalfile updatedfile

Результат получится следующий: Code: diff -c originalfile updatedfile
*** originalfile 2007-02-03 22:15:48.000000000 0100

--- updatedfile 2007-02-03 22:15:56.000000000 0100

****

*** 1 ****

! These are a few words.

--- 1 ----

! These still are just a few words.

Как вы видите, здесь включено имя файла. Это значит, что нам не придется набирать его во время применения патча. Далее идет дата и время последнего изменения файла. строка с 15 "*" показывает начало изменений. Они показывают, что надо сделать со следующим блоком текста. Два номера 1 — это номера строк (здесь тоже может быть сразу несколько строк), а «!» означает, что строки надо заменить. Строка с «!» перед тремя «-» должна быть заменена второй строкой с «!», которая идет после трех «-»(конечно сам ! не будет включен; это синтаксис контекстного формата).

Как вы можете видеть, здесь нет знаков «c», «a» и «d».Действие, которое нужно сделать, определяется символом в начале строки. «!» означает замену. Другие символы — «+», «-» и " " (пробел). «+» означает добавление, «-» означает удаление, а " " означает ничего не делать: патч использует его чтобы убедиться, что он изменяет правильную часть файла.

Применять этот патч легче: при тех же условиях, что и для предыдущего патча (записываем вывод команды diff в файл, затем копируем патч и оригинал в одно и то же место), надо выполнить следующую команду:
patch -i patchfile.patch -o updatedfile

Вы возможно сейчас думаете: зачем нам надо указывать имя нового файла? Это надо сделать из-за того, что патч старается изменить существующий файл, а не создает

новый. Это удобно при создании патча для нескольких файлов сразу. Это приводит нас к следующей цели: создание патча для дерева файлов. Рассмотрим это в следующей главе.
[править]
Получение различий между несколькими файлами

Наиболее простой способ получить различия между несколькими файлами — это положить их в одну директорию и выполнить команду diff для этой директории целиком. Вы можете просто передать команде diff в качестве параметров имена директорий вместо имен файлов:
diff originaldirectory/ updateddirectory/

Обратите внимание: если в директории есть поддиректории, то надо использовать опцию «-r».

В результате должно получится что-то вроде этого: Code: diff originaldirectory/file1 updateddirectory/file1

1c1

< This is the first original file.

> This is the first updated file.

diff originaldirectory/file2 updateddirectory/file2

1c1

< This is the second original file.

> This is the second updated file.

14d13

< We're going to add something in this file and to delete this line.

26a26

> This is line has been added to this updated file.

Обратите внимание: Я создал несколько файлов для примера. Вы можете скачать

архив, содержащий эти файлы.

Как вы видите, нормальный формат содержит только имена файлов и изменяемые строки.

Теперь используем контекстный формат: Code: diff -c originaldirectory/file1 updateddirectory/file1

*** originaldirectory/file1 2007-02-04 16:17:57.000000000 +0100

--- updateddirectory/file1 2007-02-04 16:18:33.000000000 +0100

****

*** 1 ****

! This is the first original file.

--- 1 ----

! This is the first updated file.

diff -c originaldirectory/file2 updateddirectory/file2

*** originaldirectory/file2 2007-02-04 16:19:37.000000000 +0100

--- updateddirectory/file2 2007-02-04 16:20:08.000000000 +0100

****

*** 1,4 ****

! This is the second original file.

S

O

--- 1,4 ----

! This is the second updated file.

S

O

****

*** 11,17 ****

C

E

- We're going to add something in this file and to delete this line.

S

O

--- 11,16 ----

****

*** 24,28 ****

--- 23,28 ----

C

E

+ This is line has been added to this updated file.

Something will be added above this line.

Первая вещь, которую вы должны были заметить — это увеличение размера; контекстный формат содержит больше информации, чем нормальный. Этого не было заметно в первом премере, так как не было контекста. Однако теперь контекст есть, и

за счет него размер патча увеличился. Кроме того, вы наверное заметили, что имя файла повторяется дважды. Это возможно сделано для того, чтобы легче было понять когда начался патч следующего файла или для обеспечения лучшего восстановления.

Другой способ получить разницу между между несколькими файлами — это написать скрипт, который выполняет команду diff несколько раз и добавляет результат выполнения в один файл. Мы не будем рассматривать этот способ, так как положить все файлы в одну директорию горазда проще.

Создать патч было легко, но использование директорий ставит следующую проблему: будет ли патч изменять только соответствующие файлы в текущей директории, или будет использовать соответствующий путь, указанный в файле? Чтобы узнать это, смотрите следующую главу!
[править]
Применение патча к нескольким файлам

В предыдущей главе мы создали патч для нескольких файлов, воспользовавшись следующей командой:
diff -c originaldirectory/ updateddirectory/ > patchfile.patch

Обратите внимание: мы используем контекстный формат патча, так как это является хорошим тоном.

Теперь надо использовать полученный патч. Скопируйте оригинальную директорию и патч куда-нибудь и примените следующую команду:
patch -i patchfile.patch

Однако возникает ошибка, что невозможно найти файлы для патча. Команда пытается найти файл file1 в текущей директории (по умолчанию патч убирает все пути перед именем файла). И конено файла нет, так как мы пытаемся обновить файлы в директории originaldirectory. Поэтому мы должны заставить патч использовать полный путь. Это делается следующим образом:
patch -p0 -i patchfile.patch

Обратите внимание: Вы может подумать, что можно просто переместиться в originaldirectory и запустить патч. Но это не так! Так делать не стоит: если в в патче содержатся поддиректории, то он будет искать их в рабочей директории, и не найдет, или найдет не те. Используйте опцию «-p», чтобы заставить патч искать файлы в поддиректориях.

Опция «-p» говорит патчу сколько слэшей (включая то, что перед ними, обычно директории) нужно вырезать перед именем файла (обратите внимание, что при использовании опции «-p0», патч будет будет искать файлы и в originaldirectory и в updateddirectory). Когда мы устанавливаем 0, это означает что не надо удалять пути, но можно поставить 1, чтобы удалить первый слэш, или 2, чтобы удалить два слэша, и т. д. Это может быть полезно, если если в патче используется структура каталогов, отличная

от вашей. Например, если в патче используется следующая структура каталогов:
Файл: Пример
(...)

*** /home/username/sources/program/originaldirectory/file1 2007-02-04 16:17:57.000000000 +0100

--- /home/username/sources/program/updateddirectory/file1 2007-02-04 16:18:33.000000000 +0100

(...)

Вам надо просто посчитать количество слэшей (/ (1) home/ (2) username/ (3) sources/ (4) program/ (5)) и передать это число в опции «-p». Если вы используете «-p5», то патч будет искать и в originaldirectory/file1 и в updateddirectory/file1. Не забудьте, что патч рассматривает два слэша друг за другом (как в /home/username//sources) как один. Это вызвано тем, что иногда патч скрипты добавляют дополнительный слэш между директориями.
[править]
Восстановление оригинального файла из пропатченного

Иногда возникает необходимость восстановить оригинальный файл из пропатченного. Например, если в нем содержится ошибка. Для этого надо использовать опцию «-R»:
patch -p0 -R -i patchfile.patch

Обычно эта операция проходит без ошибок, и вы получаете исходный файл. Однако лучше на всякий случай сделать резервную копию перед восстанолением файлов.
[править]
Унифицированный формат

Есть еще один формат вывода различий командой diff: унифицированный формат. Он более компактен, так как содержит уменьшенные контекстные строки. Однако он поддерживается только GNU diff и patch. Если вы его используете, вы должны быть уверены, что у пользователей, для которых патч предназначен, GNU patch. Linux допускает использование этого формата.

Унифицированный формат похож на контекстный, но это не одно и тоже. Патч в унифицированном формате можно создать так:
diff -u originaldirectory/ updateddirectory/

Результат будет седующий: Code: diff -u originaldirectory/file1 updateddirectory/file1

--- originaldirectory/file1 2007-02-04 16:17:57.000000000 +0100

+++ updateddirectory/file1 2007-02-04 16:18:33.000000000 +0100

@@ -1 +1 @@

-This is the first original file.

+This is the first updated file.

diff -u originaldirectory/file2 updateddirectory/file2

--- originaldirectory/file2 2007-02-04 16:19:37.000000000 +0100

+++ updateddirectory/file2 2007-02-04 16:20:08.000000000 +0100

@@ -1,4 +1,4 @@

-This is the second original file.

+This is the second updated file.

S

O

@@ -11,7 +11,6 @@

C

E

-We're going to add something in this file and to delete this line.

S

O

@@ -24,5 +23,6 @@

C

E

+This is line has been added to this updated file.

Something will be added above this line.

Как вы видите, номера строк заключены между «@». Кроме того, есть дополнительный пробел после «+» или «-». Это экономит несколько байт. Другое различие: в унифицированном формате нет специального знака для замены. Он просто удаляет старые строки («-») и добавляет новые («+»). Разница между этими действиями заключается в том, что при замене используется один и тот же номер строки, а при удалении и добавлении разные.
[править]
Сравнение форматов

Читая про три разных формата, вы вероятно задумались: а какой же выбрать? Вот небольшое сравнение:
Нормальный формат наиболее совместимый. Любые команды похожие на diff/patch должны понять его. Его недостаток — это отсутствие контекста.
Контекстный формат широко распространен, но не все команды его понимают. Его преимущество в наличии контекста.
Унифицированный формат тоже включает контекст, и при этом более компактем. Но его поддерживает только GNU diff and patch.

Если вы уверены, что патч будут использовать только пользователи с GNU diff/patch, то лучше всего выбрать унифицированный формат, так как он более компактный. В большинстве других случаев лучший выбор — это контекстный формат. Нормальный формат следует использовать если вы уверены, что пользователь будет применять патч командами, не поддерживающими контекстный формат.
[править]
Изменение количества контекстных строк

Можно заставить команду diff включать в патч меньшее количество строк контекста, чем должно быть. В больших патчах это может сильно уменьшить его размер. Однако если уменьшить количество контекстных строк, это может привести в неработоспособности патча. Цитаты из справки GNU diff: «Для большинства операций в патче должно быть хотя бы две строки контекста.»

Указать количество контестных строк можно несколькими способами:
Если вы хотит использовать контекстный формат, вы можете вы можете совместить эти указания, добавив в опцию «-C». Пример:
diff -C 2 originaldirectory/ updateddirectory/

Предыдущая команда будет использовать контекстный формат с двумя контекстными строками.

Если вы хотит использовать контекстный формат, вы можете вы можете совместить эти указания, добавив в опцию «-U». Пример:
diff -U 2 originaldirectory/ updateddirectory/

Предыдущая команда будет использовать унифицированный формат с двумя контекстными строками.
Если не указывать какой формат вы хотите использовать, то команда будет выглядеть примерно так:
diff -2 originaldirectory/ updateddirectory/

Однако это будет работать только если вы определите формат. Вам необходимо использовать эту опцию или с «-c» или с «u».
[править]
Заключительные слова

Несмотря на то, что эта статья описывает множество особенностей работы команд diff и patch, она не может описать все их возможности. Если вы хотите узнать больше об этих командах, вы можете почитать страницу помощи по этим командам и документацию GNU.
[править]
Источники
Using diff and patch (tutorial)
Alex выполнил оригинальный перевод на Yellow Leaf и любезно разрешил разместить здесь переведенный материал.

Оригинал http://ru.gentoo-wiki.com/wiki/HOWTO_diff_and_patch