

[ИМЯ](#)[командная строка](#)[описание](#)[ключи запуска](#)[подводные камни](#)[примеры](#)[возвращаемые значения](#)[смотри также](#)[предупреждение](#)[версия](#)[автор](#)[focal \(1\) xxd.1.gz](#)Provided by: [xxd_8.1.2269-1ubuntu5.21_amd64](#) 

ИМЯ

xxd - создаёт представление файла в виде шестнадцатеричных кодов или выполняет преобразование.

КОМАНДНАЯ СТРОКА

```
xxd -h[elp]
xxd [ключи] [входной_файл [выходной_файл]]
xxd -r[everse] [ключи] [входной_файл [выходной_файл]]
```

ОПИСАНИЕ

xxd создаёт представление указанного файла или данных, прочитанных из потока ввода, в виде шестнадцатеричных кодов. Эта команда также может выполнять преобразование заданных шестнадцатеричными кодами данных в исходный бинарный формат. Подобно командам [uuencode\(1\)](#) и [uudecode\(1\)](#), она позволяет выполнять преобразование бинарных данных в ASCII-код, который можно передавать по электронной почте,

этого, программа xxd позволяет выполнять декодирование в поток стандарта. Также может применяться для создания заплаток для бинарных файлов.

КЛЮЧИ ЗАПУСКА

Если входной файл не задан, то соответствующие данные читаются из потока ввода. В случае, если в качестве входного файла используется символ '-', данных также выступает поток стандартного ввода. В том случае, если задан выходной файл (или вместо него используется символ '-'), результат направляется в поток стандартного вывода.

Обратите внимание, что используется "ленивый" алгоритм разбора ключей: проверяет более одной буквы ключа, если в этом ключе не используется параметр, между единственным символом ключа и соответствующим параметром не являются обязательными. Параметры ключей могут быть заданы с использованием десятичного, шестнадцатеричного формата. Таким образом, ключи `-c8`, `-c 8`, `-c 010` и `-cols` равнозначными.

-a | -autoskip

Включает автоматический пропуск: вместо последовательности символов используется одиночный символ '*'. По умолчанию не применяется.

-b | -bits

Вместо шестнадцатеричного кода используются биты (двоичные символы). В использовании этого ключа вместо обычного шестнадцатеричного представления используются наборы из восьми символов "1" и "0". Каждая строка начинается с номера строки в шестнадцатеричном виде, а завершается символьным представлением (в виде ascii или ebcdic). Ключи `-r`, `-p`, `-i` в этом режиме не работают.

-c кол | -cols кол

Задаёт количество октетов <кол>, которое выводится на каждой строке. По умолчанию используется значение 16 (`-i`: 12, `-ps`: 30, `-b`: 6). Максимальное значение: 256.

-E | -EBCDIC

Изменяет способ кодирования символов в правой колонке с ASCII на EBCDIC, но не изменяет шестнадцатеричное представление. Данный ключ не имеет параметров. Используются ключи `-r`, `-p` или `-i`.

-g байт | -groupsize байт

Позволяет выполнять группировку указанного количества символов (от одного до шестнадцатеричных цифр или восьми битов), отделяя группы друг от друга.

Значение -g 0 применяется для отказа от использования группировки. используется значение <байт> равное 2 в обычном режиме и 1 в (Группировка не применяется в режимах postscript и include.

-h | -help

Выводит справку по доступным ключам командной строки и завершает работу. Создание шестнадцатеричного представления не выполняется.

-i | -include

Позволяет создавать вывод в стиле подключаемых заголовочных файлов. содержит полноценное определение статического массива данных, соответствует имени входного файла, если xxd не считывает данные из стандартного ввода.

-l длина | -len длина

Завершает работу после записи заданного в параметре <длина> количества байтов.

-p | -ps | -postscript | -plain

Использует непрерывный формат вывода шестнадцатеричного кода, "простой" стиль или стиль "postscript".

-r | -revert

Изменяет смысл операции на противоположный: позволяет выполнять шестнадцатеричного представления в бинарный код (или применять результаты заплоты). Если вывод происходит не в поток стандартного вывода, то добавление кода к соответствующему файлу. При использовании комбинации происходит чтение "простого" шестнадцатеричного представления без информации о номерах строк и какого-либо специального раскрытия колонок. символы новой строки могут встречаться в любом месте исходных данных.

-seek смещение

При использовании после ключа -r : добавлять указанное <смещение> к позициям, обнаруженным в исходных данных.

-s [±][-]seek

Начинает работу с указанного абсолютного (или относительного) <смещения> во входном файле. ± указывает, что смещение является относительным текущей файловой позиции в потоке стандартного ввода (бессмысленно, если происходит не из потока стандартного ввода). - указывает, что должно быть добавлено указанное количество символов от конца ввода (либо, если сочетается с ±, текущей позиции файла в потоке стандартного ввода). Если ключ -s не используется, то xxd начинает работу от текущей позиции в файле.

-u Использует шестнадцатеричные цифры в верхнем регистре. По умолчанию цифры в нижнем регистре символов.

-v | -version

Отображает информацию о версии программы.

ПОДВОДНЫЕ КАМНИ

xxd -r обладает встроенным интеллектом для распознавания информации о номерах строк. Возможен поиск по входному файлу, то номера строк в начале шестнадцатеричного представления могут быть неупорядоченными, некоторые строки пропущены или пересекаются друг с другом. В этих случаях **xxd** использует поиск по следующей позиции. Если поиск по входному файлу невозможен, то **xxd** пропускает строки, которые заполняются нулевыми байтами.

xxd -r никогда не выводит сообщений об ошибках. Мусор пропускается молча.

При редактировании шестнадцатеричных представлений бинарных файлов обращайтесь к **xxd -r** пропускает в строке ввода любые данные после прочтения достаточного количества шестнадцатеричных данных (см. ключ **-c**). Это означает, что изменения в колонках с печатными символами `ascii` (или `ebcdic`), всегда игнорируются при преобразовании шестнадцатеричного представления в стиле `postscript` с помощью **-t** - **r** количество колонок не учитывается. В этом случае распознаются все символы, похожие на пары шестнадцатеричных цифр.

Обратите внимание на различие между командами

% xxd -i файл

и

% xxd -i < файл

Команда **xxd -s +seek** может отличаться от **xxd -s seek**, поскольку для того, чтобы вернуть данные на входе назад, используется вызов `lseek(2)`. При использовании `+` команда отличается, если входные данные поступают с потока стандартного ввода, а поток стандартного ввода не находится в начале файла к тому моменту, когда команда запущена и приступает к чтению ввода. Нижеследующие примеры помогут прояснить (и больше запутать!) ситуацию...

Отмотка назад потока стандартного ввода; необходимо, поскольку `'cat'` уже ввел данные до конца потока стандартного ввода:

```
% sh -c 'cat > plain copy; xxd -s 0 > hex copy' < file
```

Вывод шестнадцатеричного представления от позиции в файле 0x480 (= 1024+128) означает "относительно текущей позиции", таким образом `128' добавляет килобайту, где завершает работу dd:

```
% sh -c 'dd of=plain snippet bs=1k count=1; xxd -s +128 > hex snippet' < file
```

Вывод шестнадцатеричного представления от позиции в файле 0x100 (= 1024-768)

```
% sh -c 'dd of=plain snippet bs=1k count=1; xxd -s +-768 > hex snippet' < file
```

В то же время, следует заметить, что подобные ситуации встречаются довольно редко. Символ `+' обычно не используется. Автор предпочитает наблюдать за работой `strace(1)` или `truss(1)` в тех случаях, когда применяется ключ `-s`.

ПРИМЕРЫ

Вывести всё, кроме первых трёх строк (0x30 байтов) файла **file** :

```
% xxd -s 0x30 file
```

Вывести три строки (0x30 байтов) от конца файла **file** :

```
% xxd -s -0x30 file
```

Вывести 120 байтов в виде непрерывного шестнадцатеричного представления по строке:

```
% xxd -l 120 -ps -c 20 xxd.1
2e544820585844203120224d616e75616c207061
676520666f7220787864220a2e5c220a2e5c2220
32317374204d617920313939360a2e5c22204d61
6e207061676520617574686f723a0a2e5c222020
2020546f6e79204e7567656e74203c746f6e7940
7363746e7567656e2e7070702e67752e6564752e
```

Вывести первые 120 байтов этой страницы справочника по 12 октетов в строке:

```
% xxd -l 120 -c 12 xxd.1
0000000: 2e54 4820 5858 4420 3120 224d .TH XXD 1 "M
```

```

000000c: 616e 7561 6c20 7061 6765 2066  anual page f
0000018: 6f72 2078 7864 220a 2e5c 220a  or xxd"..\".
0000024: 2e5c 2220 3231 7374 204d 6179  .\" 21st May
0000030: 2031 3939 360a 2e5c 2220 4d61  1996..\" Ma
000003c: 6e20 7061 6765 2061 7574 686f  n page autho
0000048: 723a 0a2e 5c22 2020 2020 546f  r:..\" To
0000054: 6e79 204e 7567 656e 7420 3c74  ny Nugent <t
0000060: 6f6e 7940 7363 746e 7567 656e  ony@sctnugen
000006c: 2e70 7070 2e67 752e 6564 752e  .ppp.gu.edu.

```

Показать дату из файла xxd.1:

```
% xxd -s 0x28 -l 12 -c 12 xxd.1
```

```
0000028: 3231 7374 204d 6179 2031 3939  21st May 199
```

Скопировать **входной_файл** в **выходной_файл** с добавлением 100 байтов со знач
начало файла:

```
% xxd входной_файл |. xxd -r -s 100 > выходной_файл
```

Заменить дату в файле xxd.1:

```
% echo '0000029: 3574 68' |. xxd -r - xxd.1
```

```
% xxd -s 0x28 -l 12 -c 12 xxd.1
```

```
0000028: 3235 7468 204d 6179 2031 3939  25th May 199
```

Создать 65537-байтный файл, все байты которого имеют значение 0x00, кр
байта, который должен иметь значение 'A' (0x41):

```
% echo '010000: 41' |. xxd -r > file
```

Создать шестнадцатеричное представление этого файла с использованием
пропуска:

```
% xxd -a -c 12 file
```

```
00000000: 0000 0000 0000 0000 0000 0000  ....
```

```
*
```

```
000ffffc: 0000 0000 40 .....A
```

Создать 1-байтный файл, содержащий символ 'A'. Число после '-r -s' добавля
строк, найденным в файле; иначе говоря, предшествующие байты пропускаются:

```
% echo '010000: 41' |. xxd -r -s -0x10000 > file
```

xxd можно использовать в качестве фильтра в редакторе, например в [vim\(1\)](#), шестнадцатеричное представление области между метками `a` и `z`:

```
:a,z!xxd
```

Вы можете использовать xxd в качестве фильтра в редакторе, например восстановления данных из шестнадцатеричного представления между метками `a` и `z`:

```
:a,z!xxd -r
```

Вы можете использовать xxd в качестве фильтра в редакторе, например восстановления данных из единственной строки шестнадцатеричного представления курсор в соответствующую строку и наберите

```
!!xxd -r
```

Чтобы прочитать единственный символ из канала связи:

```
% xxd -c1 < /dev/term/b &  
% stty < /dev/term/b -echo -opost -isig -icanon min 1  
% echo -n foo > /dev/term/b
```

ВОЗВРАЩАЕМЫЕ ЗНАЧЕНИЯ

Программа xxd завершает работу со следующими значениями:

- | | |
|-----|---|
| 0 | Ошибки не обнаружены. |
| -1 | Операция не поддерживается (выполнение <code>xxd -r -i</code> пока невозможно). |
| 1 | Ошибка при разборе ключей командной строки. |
| 2 | Проблемы во входном файле. |
| 3 | Проблемы в выходном файле. |
| 4,5 | Желательная позиция поиска недостижима. |

СМОТРИ ТАКЖЕ

```
uuencode(1), uudecode(1), patch(1)
```

ПРЕДУПРЕЖДЕНИЕ

Странность этой программы соответствует особенностям мозга её создателя. на свой страх и риск. Копируйте файлы, отслеживайте вызовы, становитесь во

ВЕРСИЯ

Эта страница справочника документирует xxd версии 1.7.

АВТОР

(с) 1990-1997 Юрген Вайгерт (Juergen Weigert)

<jnweiger@informatik.uni-erlangen.de>

Вы можете свободно распространять программу со ссылкой на меня.

Если использование этой программы принесло вам какой-то доход, поделитесь с

Если вы потеряли деньги, то я тут не причём.

Первый вариант страницы справочника написан Тони Наджентом (Tony Nugent)

<tony@sctnugen.ppp.gu.edu.au> <T.Nugent@sct.gu.edu.au>

Небольшие изменения внесены Брамом Мооленааром (Bram Moolenaar). Страница Юргеном Вайгертом (Juergen Weigert).

Powered by the [Ubuntu Manpage Repository](#), file bugs in [Launchpad](#)

© 2019 Canonical Ltd. Ubuntu and Canonical are registered trademarks of Canonical Ltd.