

# рах - архиватор POSIX

рах может делать много необычных вещей, не стесняйтесь добавлять больше потрясающих трюков рах!

## Введение

Архиватор POSIX, `рах`, представляет собой попытку создания стандартизированного архиватора с лучшими функциями `tar` и `cpio`, способного обрабатывать все распространенные типы архивов.

Однако это **не справочная** страница, на ней **не** будут перечислены все возможные варианты, о них **не** будет подробной информации `рах`. Это только введение.

Эта статья основана на `debianized Berkeley` реализации `рах`, но специфичные для реализации вещи должны быть помечены как таковые. К сожалению, пакет `Debian`, похоже, больше не поддерживается.

## Обзор

### Режимы работы

Существует четыре основных режима работы для *перечисления, чтения, записи и копирования* архивов. Они переключаются с помощью комбинаций `-r` `-w` параметров командной строки и:

Режим	RW-Опции
Список	<i>нет RW-опций</i>
Читать	<code>-r</code>
Напишите	<code>-w</code>
Копировать	<code>-r -w</code>

### Список

В *режиме списка* `рах` записывает список членов архива в стандартный вывод (оглавление). Если в командной строке указано соответствие шаблону, печатаются только совпадающие имена файлов.

Читать

*Прочитайте* архив. `pax` будет считывать данные архива и извлекать элементы в текущий каталог. Если в командной строке указано соответствие шаблону, извлекаются только совпадающие имена файлов.

При чтении архива тип архива определяется из архивных данных.

Напишите

*Напишите* архив, что означает создание нового или добавление к существующему. Все файлы и каталоги, указанные в командной строке, вставляются в архив. По умолчанию архив записывается в стандартный вывод.

Если в командной строке не указаны файлы, считываются имена `STDIN` файлов.

Режим записи - это единственный режим, в котором вам нужно указать тип архива с `-x` `<TYPE>` помощью, например `-x ustar`.

Копировать

*Режим копирования* аналогичен `cpio` режиму сквозной передачи. Он предоставляет способ реплицировать полную или частичную файловую иерархию (со всеми `pax` опциями, например, группы перезаписи) в другое место.

Архивировать данные

Когда вы не укажете ничего особенного, `pax` попытается прочитать архивные данные из стандартного ввода (режимы чтения / списка) и записать архивные данные в стандартный вывод (режим записи). Это гарантирует `pax`, что его можно легко использовать как часть конструкции канала оболочки, например, для чтения сжатого архива, который распакован в канале.

Возможность указать путь к архивируемому файлу заключается `-f` в том, что этот файл будет использоваться как ввод или вывод, в зависимости от операции (чтение / запись / список).

Когда `pax` считывает архив, он пытается угадать тип архива. Однако в режиме *записи* необходимо указать, какой тип архива добавлять с помощью `-x` `<TYPE>` переключателя. Если вы опустите этот параметр, будет создан архив по умолчанию (POSIX говорит, что он определен реализацией, Berkeley `pax` создает `ustar`, если параметры не указаны).

Поддерживаются следующие форматы архивирования (реализация Berkeley):

ustar	Формат POSIX TAR (по умолчанию)
cpio	Формат POSIX CPIO
tar	классический формат TAR BSD
bcpio	старый двоичный формат CPIO
sv4cpio	Формат CPIO SVR4
sv4crc	Формат CPIO SVR4 с CRC

Berkeley `pax` поддерживает опции `-z` и `-j`, подобно GNU `tar`, для фильтрации архивных файлов через GZIP / BZIP2.

## Соответствие участников архива

В режимах *чтения* и *списка* вы можете указать шаблоны, чтобы определить, какие файлы перечислять или извлекать.

- нотация шаблона - это та, которая известна POSIX-оболочке, то есть та, которая известна Bash без `extglob`
- если указанный шаблон соответствует полному каталогу, он влияет на все файлы и подкаталоги указанного каталога
- если вы укажете `-s` параметр, `pax` он инвертирует совпадения, т.е. Он соответствует всем именам файлов, **кроме** тех, которые соответствуют указанным шаблонам
- если шаблоны не заданы, `pax` будет "сопоставлять" (перечислять или извлекать) все файлы из архива
- **Чтобы избежать конфликтов с расширением пути к оболочке, разумно указывать шаблоны!**

### Несколько различных примеров шаблонов

```
pax -r
```

```
pax -r
# должно быть эквивалентно
pax -r
```

## Использование `pax`

Это краткое описание использования `pax` в качестве обычной системы архиватора, как вы бы использовали `tar`.

## Создание архива

Эта задача выполняется с помощью базового синтаксиса

```
# архивировать содержимое в стандартный
pax -w >archive.tar README.txt *.данные в формате png/

# эквивалент, извлеките содержимое архива непосредственно в файл
pax -w -x ustar -f archive.tar README.txt *.данные в формате png/
```

`pax` находится в режиме *записи*, указанные имена файлов упаковываются в архив:

- `README.txt` это обычный файл, он будет упакован
- `*.png` является глобусом пути для вашей оболочки, оболочка заменит все совпадающие имена файлов перед выполнением. `pax` Результатом является

список имен файлов, которые будут упакованы, как в `README.txt` примере выше

- `data/` это каталог. **Все**, что находится в этом каталоге, будет упаковано в архив, т.е. Не Просто пустой каталог

Когда вы укажете `-v` опцию, `pax` будут записаны пути к файлам, вставленным в архив `STDERR`.

Когда и только когда аргументы имени файла не указаны, `pax` пытается прочитать имена файлов из `STDIN`, разделенные символами новой строки. Таким образом, вы можете легко комбинировать `find` с `pax`:

```
Найти. -имя '*.txt' | pax -wf textfiles.tar -x ustar
```

## Список содержимого архива

Стандартный формат вывода списка членов архива - просто вывести каждое имя файла в отдельную строку. Но формат вывода можно настроить для включения разрешений, временных меток и т. Д. со `-o listopt=<FORMAT>` спецификацией. Синтаксис спецификации формата сильно зависит от `printf(3)` спецификации формата.

**К сожалению** `pax`, утилита, поставляемая с Debian, похоже, не поддерживает эти форматы расширенных списков.

Тем не менее, `pax` список членов архива отображается в формате а `ls -l -like`, когда вы указываете `-v` опцию:

```
pax -v
# или, конечно
, pax -vf myarchive.tar
```

## Извлечение из архива

Вы можете извлечь все файлы или файлы (не), соответствующие определенным шаблонам, из архива, используя такие конструкции, как:

```
# "нормальное" извлечение
pax -rf myarchive.tar '*.txt'
```

```
# с перевернутым шаблоном
pax -rf myarchive.tar -c '*.txt'
```

## Копирование файлов

Чтобы скопировать содержимое каталога в другой каталог, аналогично `cp -a` команде, используйте:

```
mkdir destdir
pax -rw dir destdir #создает копию dir в destdir/, то есть destdir/dir
```

## Копирование файлов через ssh

Чтобы скопировать содержимое каталога в другой каталог в удаленной системе, используйте:

```
pax -w localdir | ssh user@host "самый удаленный cd && pax -r -v"
пользователь pax -w localdir | gzip | ssh@host "cd distantdir && gunzip | pax -r -v" #сжимает отправленные данные
```

Эти команды создают копию localdir в distantdir (distantdir/dir) на удаленной машине.

## Расширенное использование

### Резервное копирование вашей повседневной работы

**Примечание:** -T является расширением и не определяется POSIX.

Допустим, у вас есть доступ на запись к файловому серверу, установленному в дереве вашей файловой системы. В режиме *копирования* вы можете указать pax копировать только файлы, которые были изменены сегодня:

```
mkdir /n/mybackups/$(дата +%A)/
данные pax -rw -T 0000/ /n/mybackups/$(дата +%A)/
```

Это делается с помощью -T переключателя, который обычно позволяет указать временное окно, но в данном случае только время начала, что означает "сегодня в полночь".

Когда вы выполняете эту "очень простую резервную копию" после повседневной работы, у вас будет копия измененных файлов.

**Примечание:** %A формат from date расширяется до названия текущего дня, локализованного, например, "Пятница" (en) или "Mittwoch" (de).

То же самое, но с архивом, может быть выполнено с помощью:

```
pax -w -T 0000 -f /n/mybackups/$(дата +%A)
```

В этом случае название дня является архивным файлом (вам не нужно расширение имени файла, например .tar , но вы можете добавить его, если хотите).

## Изменение имен файлов при архивировании

pax способен переписывать имена файлов во время архивирования или при извлечении из архива. В этом примере создается архив tar, содержащий holiday\_2007/ каталог, но имя каталога внутри архива будет holiday\_pics/ :

```
pax -x ustar -w -f holiday_pictures.tar -s '/^holiday_2007/holiday_pics/' holiday_2007/
```

Параметр, отвечающий за обработку строк, является `-s <REWRITE-SPECIFICATION>` . Он принимает спецификацию перезаписи строки в качестве аргумента в форме `/OLD/NEW/[gp]` , которая является `ed(1)` аналогом регулярного выражения (BRE) для `old` и, как правило, может использоваться как популярная конструкция `sed s/from/to/` . В качестве разделителя можно использовать любой ненулевой символ, поэтому для искажения имен путей (содержащих косые черты) вы можете использовать `#/old/path#/new/path#` .

Необязательные флаги `g` и `p` используются для применения подстановки (**g**) по долям к строке или для (**p**) преобразования исходной и перезаписанной строк `STDERR` .

`-s` В командной строке можно указать несколько параметров. Они применяются к строкам имен файлов или элементов архива. Это происходит в том порядке, в котором они указаны.

## Исключение файлов из архива

Команда `-s`, показанная выше, может использоваться для исключения файла. Результатом замены должна быть нулевая строка: например, предположим, что вы хотите исключить все каталоги CVS для создания архива исходного кода. Мы собираемся заменить имена, содержащие `/CVS/`, на `nothing` , обратите внимание на `.*` они необходимы, потому что нам нужно сопоставить весь путь.

```
pax -w -x ustar -f release.tar -s',.*CVS/.*,, ' мое приложение
```

Вы можете использовать несколько вариантов `-s`, например, допустим, вы также хотите удалить файлы, оканчивающиеся на `~`:

```
pax -w -x ustar -f release.tar -s',.*CVS/.*,, ' -'s/.*~// ' мое приложение
```

Это также можно сделать во время чтения архива, например, предположим, что у вас есть архив, содержащий каталог `"usr"` и `"etc"`, но вы хотите извлечь только каталог `"usr"`:

```
архив pax -r -f.tar -s',^etc/.*,, ' # каталог etc/ dir не извлекается
```

## Получение имен файлов архива из стандартного интерфейса

Например `cpio`, `pax` может считывать имена файлов из стандартного ввода (`stdin`). Это обеспечивает большую гибкость - например, `find(1)` команда может выбирать файлы / каталоги способами, которые `pax` не может сделать сам. В режиме **записи** (создание архива) или в режиме **копирования**, когда имена файлов не указаны, `pax` ожидает считывания имен файлов из стандартного ввода. Например:

```
# Резервное копирование конфигурационных файлов, измененных менее 3 д
ней назад
, найдите /etc -введите f -mtime -3 | pax -x ustar -w -f /backups/et
c.tar

# Скопируйте только каталоги, а не файлы
mkdir / target
find . -введите d -print | pax -r -w -d /target

# Создайте резервную копию всего, что изменилось с момента последней
находки резервной копии . -новее /var/run/mylastbackup -print0 |
pax -0 -x ustar -w -d -f /резервные копии/mybackup.tar
touch /var/run/mylastbackup
```

`-d` Опция указывает `pax` `not` выполнять рекурсию в каталоги, которые он читает (`cpio -style`). Без `-d` него `pax` выполняет рекурсию во все каталоги (`tar -style`).

**Примечание:** `-0` опция не является стандартной, но присутствует в некоторых реализациях.

## От tar до pax

`pax` может обрабатывать формат `tar` архива, если вы хотите переключиться на стандартный инструмент с псевдонимом типа:

```
псевдоним tar='echo ИСПОЛЬЗУЙ PAX, идиот. pax - стандартный архивато
р!; # '
```

в вашем `~/.bashrc` может быть полезно 😊.

Вот краткое сравнение таблиц (GNU) `tar`, `pax` которое поможет вам переключиться:

TAR	PAX	Примечания
<code>tar xzvf</code> <code>file.tar.gz</code>	<code>pax -rvz -f</code> <code>file.tar.gz</code>	<code>-z</code> является расширением, POSIXly: <code>gunzip &lt;file.tar.gz   pax -rv</code>
<code>tar czvf</code> <code>archive.tar.gz</code> <code>path ...</code>	<code>pax -wvz -f</code> <code>archive.tar.gz</code> <code>path ...</code>	<code>-z</code> является расширением, POSIXly: <code>pax -wv path   gzip &gt;</code> <code>archive.tar.gz</code>
<code>tar xjvf</code> <code>file.tar.bz2</code>	<code>bunzip2</code> <code>&lt;file.tar.bz2  </code> <code>pax -rv</code>	

TAR	PAX	Примечания
tar cjvf archive.tar.bz2 path ...	pax -wv path   bzip2 > archive.tar.bz2	
tar tzvf file.tar.gz	pax -vz -f file.tar.gz	-z является расширением, POSIXly: gunzip <file.tar.gz   pax -v

pax возможно, ustar ( tar ) не создает архивы по умолчанию, но имеет собственный формат pax, добавьте -x ustar , если хотите, чтобы pax создавал архивы tar!

## Реализации

- AT & T AST toolkit (<http://www2.research.att.com/sw/download/>) | страница руководства (<http://www2.research.att.com/~gsf/man/man1/pax.html>)
- Набор инструментов семейной реликвии (<http://heirloom.sourceforge.net/index.html>) | страница руководства (<http://heirloom.sourceforge.net/man/pax.1.html>)
- OpenBSD pax (<http://www.openbsd.org/cgi-bin/cvsweb/src/bin/pax/>) | страница руководства (<http://www.openbsd.org/cgi-bin/man.cgi?query=pax&apropos=0&sektion=0&manpath=OpenBSD+Current&arch=i386&format=>
- MirBSD pax (<https://launchpad.net/paxmirabilis>) | manpage (<https://www.mirbsd.org/htman/i386/man1/pax.htm>) - Debian основывает свой пакет на этом.
- Спецификация SUS pax (<http://pubs.opengroup.org/onlinepubs/9699919799/utilities/pax.html>)



## Обсуждение

📄 [howto/pax.txt](#) 📅 Last modified: 2015/08/09 04:55 by [bill\\_thomson](#)

This site is supported by Performing Databases - your experts for database administration

Bash Hackers Wiki





Except where otherwise noted, content on this wiki is licensed under the following license:  
GNU Free Documentation License 1.3