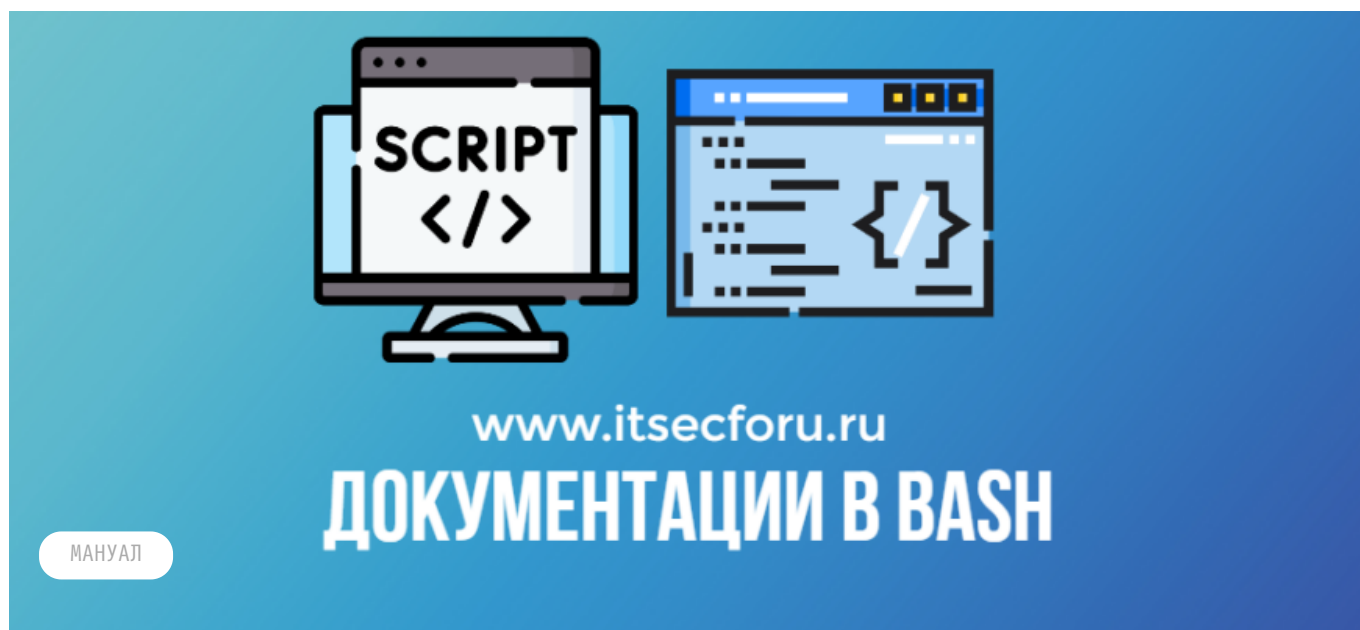


[ГЛАВНАЯ](#) » [МАНУАЛ](#)

## Как встраивать документацию в скрипты Bash



АВТОР

cryptoparty



НА ЧТЕНИЕ

5 мин



ОПУБЛИКОВАНО

09.08.2023

Документирование работы приложения, его назначения и предполагаемого использования действительно важно, даже если речь идет о простом shell-скрипте.

Для облегчения сопровождения кода в самых простых случаях документация может быть встроена непосредственно в скрипты.

В этом руководстве мы узнаем, как включать синтаксис Pearl's Plain Old Documentation (POD) в скриптах bash и как преобразовывать его в различные форматы с помощью утилит pod2, таких как pod2man и pod2html.

Конструкции Heredoc и do-nothing



Для встраивания документации в Bash-скрипты можно использовать две конструкции: “Heredoc” и “do-nothing”.

Первая позволяет передать многострочный ввод определенной команде.

С ее помощью можно, например, вывести многострочное сообщение на стандартный вывод, не повторяя несколько раз команду echo.

Вместо того чтобы написать:

```
echo "Script usage:"
echo "  myscript [options] argument"
echo ""
echo "Options:"
echo "  -h, --help Show this message"
echo "  -f, --foo This option does foo"
echo "  -b, --bar This option does bar"
```

Мы бы написали:

```
cat << EOF
Script usage:
  myscript [options] argument

Options:
  -h, --help Show this message
  -f, --foo This option does foo
  -b, --bar This option does bar
EOF
```

Второй фрагмент, как видите, гораздо понятнее (и быстрее в наборе).

Синтаксис конструкции “Heredoc” довольно прост: слева от оператора << пишется команда, которой мы хотим передать многострочный ввод; справа от оператора вместо него ставится разделитель (часто используется EOF, но это просто стандарт), который отмечает конец Heredoc.

Если разделитель не заключен в кавычки, то переменные внутри документа раскрываются.



Мы можем использовать конструкцию “Here doc” для встраивания документации внутрь скрипта Bash.

В такой ситуации, однако, мы не хотим выполнять никакую команду: здесь на помощь приходит конструкция “do-nothing”.

В Bash команда : – это Null-команда: она ничего не делает и всегда успешно завершается (ее статус exist всегда равен 0).

Используя эти две конструкции вместе, мы можем встраивать документацию в shell-скрипты следующим образом:

```
#!/bin/bash
#
# Bash code goes here
#
exit 0

: << EOF
Script usage:
  myscript [options] argument

Options:
  -h, --help Show this message
  -f, --foo This option does foo
  -b, --bar This option does bar

EOF
```

Обратите внимание, что я явно добавил инструкцию exit 0 непосредственно перед разделом документации: это сделано для того, чтобы оболочка не обрабатывала эту часть при выполнении скрипта.

## Использование формата документации Perl Plain Old Documentation

В приведенных выше примерах документация к скрипту состояла из обычного текста.

Однако для создания хорошо отформатированных документов можно использовать формат Perl Plain Old Documentation (POD).

Приведем пример, в котором используется минимальный набор команд:

```
: << EOF
=pod

=head1 NAME

Here is a brief description of what are script does.

=head1 SYNOPSIS

scriptname [options] argument

=head1 OPTIONS

-h, --help
    Show this message

-f, --foo
    This option does foo

-b, --bar
    This option does bar

=cut

EOF
```

Первый “командный абзац”, который мы использовали в примере, – `=pod`.

Он просто сигнализирует о начале документа POD.



В данном случае он излишен, так как любая команда начинает блок документа; однако он может быть полезен, когда документация должна начинаться с обычного текста.

Затем мы использовали команду `=head1`, которая создает заголовок первого уровня (уровни идут от 1 до 6).

Наконец, мы использовали `=cut`, который сигнализирует о завершении блока POD.

Перед командой должна быть пустая строка, а после нее – еще одна.

## Извлечение и генерация документации с помощью программ `pod2`

После того как мы закончили описывать функциональные возможности нашего скрипта, мы можем использовать утилиты `pod2` для генерации документов различного типа, таких как `roff` или HTML-страницы, которые можно читать, соответственно, с помощью утилиты `man` и любого веб-браузера.

На ArchLinux и Debian утилиты `pod2` входят в пакет `“perl”`, который должен быть уже установлен в вашей системе.

Однако для явной установки пакета в первом дистрибутиве мы можем выполнить команду:

Терминал

```
$ sudo pacman -S perl
```

В последнем случае мы используем `apt`:

Терминал

```
$ sudo apt install perl
```



В Fedora и других дистрибутивах, основанных на ней, утилиты pod2 поставляются в виде отдельных пакетов.

Пакет “perl-podlators” включает pod2man, а “perl-Pod-Html” – pod2html.

Мы можем установить их с помощью dnf:

Терминал

```
$ sudo dnf install perl-podlators perl
```

Для генерации roff-страницы мы вызываем pod2man и передаем в качестве аргумента путь к скрипту, после чего перенаправляем stdout утилиты в файл:

Терминал

```
$ pod2man testscript.sh > testscript.1
```

Мы можем прочитать документ, запустив его:

Терминал

```
$ man ./testscript.1
```

Чтобы сделать страницу доступной в масштабах всей системы, чтобы мы могли читать ее без указания полного пути, мы можем скопировать ее в каталог /usr/share/man/man1.




Для генерации HTML-версии документа вместо этого мы используем pod2html:



### Терминал

```
$ pod2html testscript.sh > testscript.html
```

См. также:

-  [Как установить Perl-модули на Linux](#)
-  [Как преобразовать текстовые файлы в верхний или нижний регистр](#)
-  [Как найти версию пакета в системах Linux](#)

[#bash](#) [#linux](#)

Пожалуйста, не спамьте и никого не оскорбляйте. Это поле для комментариев, а не спамбокс. Рекламные ссылки не индексируются!

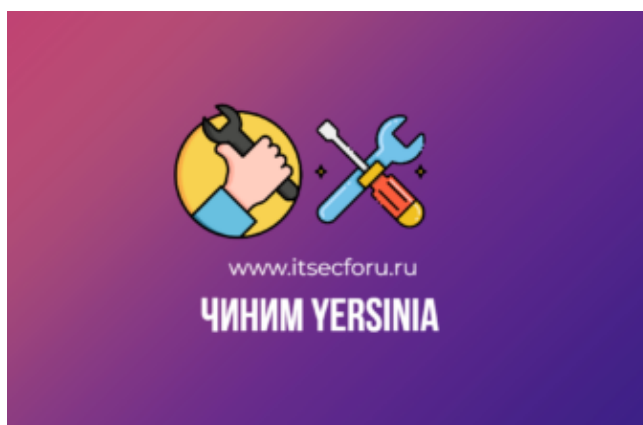
## Добавить комментарий



Комментарий

Отправить комментарий

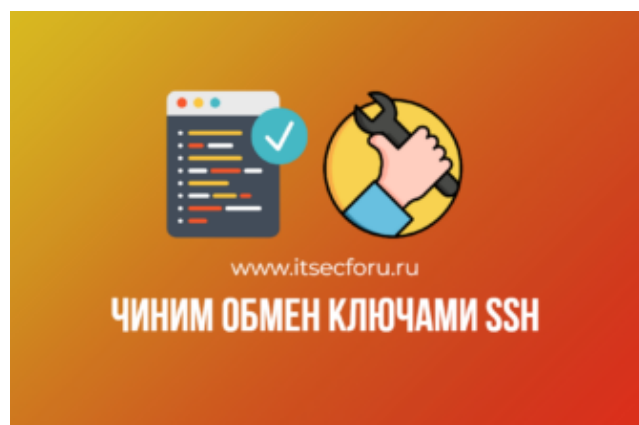
## Вам также может понравиться



### 🔧 Исправление ошибки Yersinia has been configured with -disable-gtk option

При работе в Kali Linux можно столкнуться с этой ошибкой

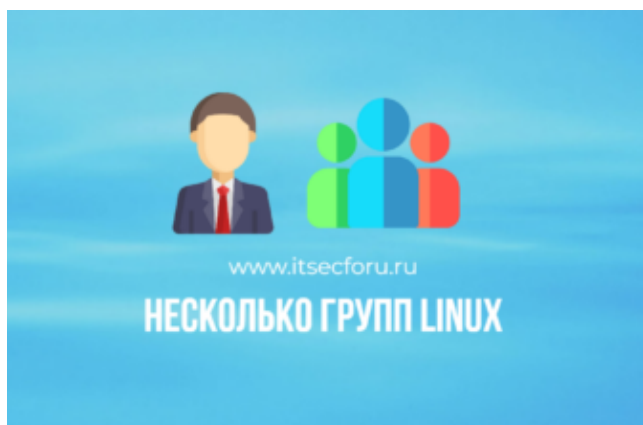
💬 0



### 🔑 Унаследованный алгоритм обмена ключами SSH

Когда клиент SSH подключается к серверу, каждая сторона

💬 0



### 🐧 Как добавить пользователя в несколько групп на Linux



### 🍓 Как использовать Raspberry Pi для мониторинга сетевого трафика



Эффективное управление пользователями и их членством

🗨 0



## 🌐 Как использовать базу данных GeoIP с nginx

Используйте базу данных GeoIP с nginx для регистрации

🗨 0

Для Raspberry Pi существует множество инструментов

🗨 0



## 🚀 Ядро Linux в режиме tainted

Ядро можно модифицировать с помощью загружаемых модулей

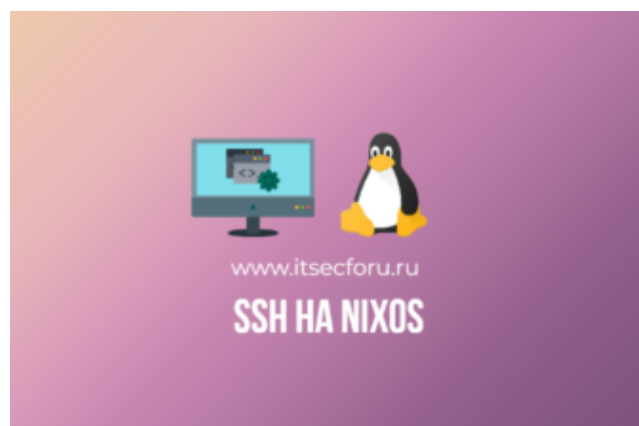
🗨 0



## 🐧 Различия между /bin, /sbin, /usr/bin, /usr/sbin, /usr/local/bin, /usr/local/sbin

Обзор Все разновидности Linux следуют определённому

🗨 0



## 🔑 Как установить openSSH на NixOS

В этом кратком руководстве мы рассмотрим шаги по установке

🗨 0




Свежие комментарии

Сер к записи [Как увеличить звук выше 100% в Kali Linux](#)


Артем к записи [Как настроить аутентификацию GitLab FreeIPA](#)


[сгугторарту](#) к записи [Как настроить аутентификацию GitLab FreeIPA](#)

Артем к записи [Как настроить аутентификацию GitLab FreeIPA](#)


[сгугторарту](#) к записи  [Как использовать Nmap для сканирования уязвимостей?](#)


## Свежие записи

 [Исправление ошибки Yersinia has been configured with -disable-gtk option](#)

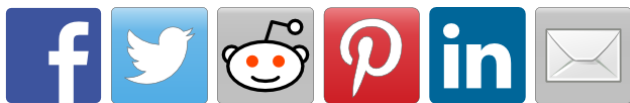
 [Как управлять и устранять неполадки SELinux](#)

[Лучшие биржи игровых ценностей 2023: ваш гид по продаже и покупке игровых аккаунтов](#)

 [3 метода проведения разведки DNS](#)

 [Активная и пассивная разведка: объяснение](#)

## Поделиться



© 2024 \$ information Security Squad

