



Интерактивная система просмотра системных руководств (ман-ов)

Тема

Набор

Категория

[\[Список руководств | Печать\]](#)

ed (1)

- [ed](#) (1) (Solaris ман: Команды и прикладные программы пользовательского уровня)
- [ed](#) (1) (FreeBSD ман: Команды и прикладные программы пользовательского уровня)
- >> [ed](#) (1) (Русские ман: Команды и прикладные программы пользовательского уровня)
- [ed](#) (1) (Linux ман: Команды и прикладные программы пользовательского уровня)
- [ed](#) (1) (POSIX ман: Команды и прикладные программы пользовательского уровня)
- [ed](#) (4) (FreeBSD ман: Специальные файлы /dev/*)
- Ключ [ed](#) обнаружен в базе ключевых слов.

ed(1)

ed(1)

НАЗВАНИЕ

ed, **red** - текстовый редактор

СИНТАКСИС

```
ed [-s] [-p строка] [-x] [-C] [файл]
red [-s] [-p строка] [-x] [-C] [файл]
```

ОПИСАНИЕ

ed - это стандартный текстовый редактор. Если указан аргумент **файл**, **ed** применяет к нему команду **e** (см. [ниже](#)); т.е. файл считывается в буфер **ed**, чтобы его можно было редактировать. Как **ed**, так и **red** обрабатывают *символы дополнительного набора* (supplementary code set characters) в файле и распознают их в строке приглашения, указанной в опции **-p** (см. [ниже](#)), в соответствии с локалью, указанной в переменной среды **LC_STYPE** (см. **LANG** в [environ\(5\)](#)). В регулярных выражениях поиск шаблонов осуществляется над символами, а не над байтами, как описано ниже.

- s** Подавляет выдачу количества байтов командами **e**, **E**, **g** и **w**, выдачу диагностики командами **e** и **q**, а также выдачу приглашения **!** после команды **!shell**.
- p** Позволяет пользователю указывать строку приглашения. Строка может содержать символы дополнительного набора.
- x** Опция шифрования; при ее использовании, **ed** имитирует команду **X** и предлагает пользователю ввести пароль. Этот ключ используется для шифрования и дешифрования текста с помощью алгоритма **crypt(1)**. Команда **X** сама делает предположение, является ли считываемый текст зашифрованным или нет. Файл временного буфера также шифруется с помощью преобразованной версии ключа, введенного для опции **-x**. См. **crypt(1)**. См. также раздел **"ПРИМЕЧАНИЯ"** в конце этой страницы справочного руководства.
- C** Опция шифрования; то же, что и опция **-x**, но **ed** имитирует команду **C**. Команда **C** действует аналогично команде **X**, но все считываемые тексты считаются зашифрованными.

ed работает с копией редактируемого файла; сделанные в копии изменения не влияют на файл, пока не будет дана команда **w** (запись). Копия редактируемого текста находится во временном файле, который называется *буфером*. Имеется только один буфер.

red - это ограниченная версия **ed**. Он позволяет редактировать файлы только в текущем каталоге. Он запрещает выполнение команд интерпретатора с помощью команды **!shell**. Попытки обойти эти ограничения ведут к выдаче сообщения об ошибке (UX:ed: ERROR: Restricted shell).

ed и **red** поддерживают возможность форматирования **fspec**. Если включить строку спецификации формата в качестве первой строки файла и вызвать **ed** с терминала, работающего в режиме **stty -tabs** или **stty tab3**, заданные шаги табуляции будут автоматически использоваться при сканировании файла. Например, если первая строка файла содержала:

```
<:t5,10,15 s72:>
```

табуляции будут установлены в столбцах 5, 10 и 15, и максимальная длина строки будет предполагаться равной 72 символам. **ПРИМЕЧАНИЕ:** при вводе текста в файл этот формат не действует; вместо этого, поскольку терминал работает в режиме **stty -tabs** или **stty tab3**, табуляции устанавливаются в каждом восьмом столбце.

Команды редактора **ed** имеют простую и регулярную структуру: ноль, один или два адреса, за которыми идет односимвольная команда, после которой могут быть указаны параметры команды. Эти адреса указывают одну или несколько строк буфера. Каждая команда, требующая указания адреса, имеет адреса по умолчанию, поэтому очень часто адреса можно не указывать.

В общем случае, в строке можно задавать только одну команду. Некоторые команды разрешают вводить текст. Этот текст помещается в соответствующее место в буфере. Когда **ed** воспринимает текст, говорят, что он находится в *режиме ввода*. В этом режиме команды не распознаются; все введенные символы просто накапливаются. Чтобы выйти из режима ввода, введите точку (.) в начале строки и сразу же нажмите клавишу Enter.

ed поддерживает ограниченную форму записи *регулярных выражений* (regular expressions); регулярные выражения используются в адресах для указания строк и в некоторых командах (например, **s**) для задания частей строки, подлежащих замене. Регулярное выражение задает множество символьных строк. Говорят, что элемент этого множества строк *сопоставляется* с регулярным выражением. (Полная информация о регулярных выражениях представлена на странице справочного руководства [grep\(1\)](#).)

Чтобы понять адресацию в **ed**, необходимо помнить, что в любой момент имеется *текущая строка*. В общем случае, текущая строка - это последняя строка, затронутая командой; конкретное влияние на текущую строку указывается в описании каждой команды. Адреса строятся следующим образом:

1. Символ **.** указывает на текущую строку.
2. Символ **\$** указывает на последнюю строку в буфере.
3. Десятичное число **n** указывает на **n**-ную строку в буфере.
4. **'x** указывает на строку, помеченную символом **x**, который должен быть символом нижнего регистра (**a-z**). Строки помечаются с помощью команды **k**, которая описана ниже.
5. Регулярное выражение в косых чертах (**/**) указывает на первую строку, найденную при поиске вперед от текущей строки до конца буфера и сопоставляющуюся с этим регулярным выражением. При необходимости, поиск продолжается с начала буфера включительно до текущей строки, так что просматривается весь буфер. См. также [последний абзац](#) раздела **"ОПИСАНИЕ"** ниже.
6. Регулярное выражение в знаках вопроса (**?**) указывает на первую строку, найденную при поиске назад от текущей строки до конца буфера и сопоставляющуюся с этим регулярным выражением. При необходимости, поиск продолжается с конца буфера включительно до текущей строки, так что просматривается весь буфер. См. также [последний абзац](#) раздела **"ОПИСАНИЕ"** ниже.
7. Адрес, за которым идет знак плюс (**+**) или знак минус (**-**), за которым идет десятичное число, указывает на этот адрес плюс (соответственно, минус) указанное число строк. В качестве сокращения для **+.5** можно использовать **.5**.
8. Если адрес начинается с **+** или **-**, добавление или вычитание осуществляется по отношению к данной строке; например, **, -5** означает **.-5**.
9. Если адрес заканчивается знаком **+** или **-**, то к адресу добавляется 1 или от адреса отнимается 1, соответственно. Вследствие этого правила и Правила 8 выше, адрес - указывает на строку, предшествующую текущей. (Чтобы сохранить совместимость с более ранними версиями редактора, символ **^** в контексте адреса полностью эквивалентен **-**.) Более того, хвостовые символы **+** и **-** имеют кумулятивный эффект, так что **--** указывает на текущую строку минус 2.
10. Для удобства, запятая (**,**) соответствует паре адресов **1,\$**, а точка с запятой (**;**) - паре **.,\$**.

Команды могут требовать ноль, один или два адреса. Для команд, не требующих адреса, указание адреса считается ошибкой. Команды, воспринимающие один или два адреса, используют адреса по умолчанию вместо недостающих адресов; если задано больше адресов, чем требуется такой команде, используются последние из указанных.

Обычно, адреса разделяются запятой (,). Они также могут разделяться точкой с запятой (;). В последнем случае вычисляется первый адрес, текущая строка (.) устанавливается в это значение, а затем вычисляется второй адрес. Эту возможность можно использовать для определения начальной строки при поисках вперед и назад (см. Правила 5 и 6 выше). Вторым адрес в любой паре адресов должен соответствовать строке в буфере, идущей **после** строки, соответствующей первому адресу.

В следующем списке команд **ed** скобки перед командой не являются частью адреса; они просто показывают адрес(а) по умолчанию для этой команды.

Аргументы **файл** команд **e**, **E**, **f**, **g**, **w** и **W** подлежат расширению как шаблоны имен файлов в **sh(1)**. Они должны отделяться от буквы, задающей команду, одним или несколькими пробелами или символами табуляции.

В общем случае недопустимо указание в строке более одной команды. Однако, после любой команды (кроме **e**, **f**, **g** и **w**) можно указать суффикс **l**, **n** или **p**, при этом, соответственно, будет выдана, выдана с номером или напечатана текущая строка, как описано ниже для команд **l**, **n** и **p**.

(.)a

<текст>

▪

Команда добавления (append) воспринимает ноль или более строк и добавляет их после указанной строки в буфер. Текущей строкой (.) остается последняя вставленная строка или, если ни одна строка не вставлена, строка, указанная в адресе. Для этой команды допустим адрес 0: он вызывает помещение "добавляемого" текста в начало буфера. Максимально в одной строке с терминала можно ввести **LINE_MAX** байтов (включая символ новой строки). Параметр **LINE_MAX** определен в файле **limits.h**.

(.)c

<текст>

▪

Команда изменения (change) удаляет указанные строки из буфера, а затем воспринимает ноль или более строк текста, которые заменяют в буфере удаленные строки. Текущей строкой (.) становится последняя введенная строка или, если ни одна строка не была введена, первая не удаленная строка.

c

То же, что и команда **X**, описанная ниже, но **ed** предполагает, что весь текст, считываемый командами **e** и **g**, является зашифрованным, если только он не пустой.

(.,.)d

Команда удаления (delete) удаляет из буфера все указанные строки. Текущей становится строка, следующая за последней удаленной; если удаленные строки находились в конце буфера, текущей становится новая последняя строка.

e файл

Команда редактирования (edit) удаляет все содержимое буфера, а затем считывает в буфер содержимое файла. Текущей строкой (.) становится последняя строка буфера. Если **файл** не указан, используется последнее запомненное имя файла, если оно есть (см. команду **f**). Выдается количество прочитанных символов; **файл** запоминается для возможного использования в качестве имени файла по

умолчанию в последующих командах **e**, **g** и **w**. Если вместо **файла** указать **!**, остаток строки рассматривается как команда интерпретатора shell (**sh(1)**), выходной поток которой должен быть считан. Такая команда интерпретатора shell не запоминается в качестве текущего имени файла. См. также раздел "**ДИАГНОСТИКА**" ниже.

E файл

Команда **E** аналогична **e**, но редактор не проверяет, были ли сделаны в буфере какие-либо изменения после последней команды **w**.

f файл

Если **файл** указан, команда запоминания имени файла (file-name) изменяет текущее запомненное имя файла на указанное; в противном случае, она выдает текущее запомненное имя файла.

(1,\$)g/регулярное_выражение/список_команд

В глобальной (global) команде первым шагом является пометка каждой строки, сопоставляющейся с указанным **регулярным_выражением**. Затем, для каждой такой строки выполняется заданный **список_команд**, причем текущая строка (.) предварительно устанавливается в данную. Единственная команда или первая из **списка_команд** в строке выполняется как глобальная. Все строки многострочного **списка_команд**, кроме последней, должны заканчиваться обратной косой (\); команды **a**, **i** и **c**, а также соответствующий ввод, допустимы. Точку (.), завершающую режим ввода, можно не указывать, если она будет последней строкой **списка_команд**. Пустой **список_команд** эквивалентен команде **p**. Команды **g**, **G**, **v** и **V** в **списке_команд** использовать нельзя. См. раздел "**ПРИМЕЧАНИЯ**" и **последний абзац** раздела "**ОПИСАНИЕ**" ниже.

(1,\$)G/регулярное_выражение/

В интерактивной глобальной команде (Global), первым шагом является пометка каждой строки, сопоставляющейся с указанным **регулярным_выражением**. Затем, каждая такая строка печатается, текущая строка (.) устанавливается в данную и можно ввести и выполнить любую одиночную команду (кроме команд **a**, **c**, **i**, **g**, **G**, **v** и **V**). После выполнения этой команды печатается следующая помеченная строка и т.д.; новая строка действует как пустая команда; ввод **&** вызывает повторное выполнение последней команды, выполненной в рамках текущего вызова **G**. Учтите, что команды, вводимые в ходе выполнения команды **G**, могут обращаться и влиять на любые строки в буфере. Команду **G** можно прекратить, послав сигнал прерывания (ASCII **DEL** или **BREAK**).

h

Команда получения справки (help) выдает краткое сообщение об ошибке, объясняющее причину выдачи последнего диагностического сообщения **?**.

H

Команда **H** заставляет **ed** войти в режим, в котором сообщения об ошибках печатаются для всех последующих диагностических сообщений **?**. Она также даст объяснения по предыдущему **?**, если он был. Команда **H** включает и отключает этот режим; первоначально он отключен.

(.)i

<текст>

▪

Команда вставки (insert) воспринимает ноль или более строк **текста** и вставляет их в буфер перед указанной строкой. Текущей строкой (.) становится последняя вставленная строка или, если ни одна строка не вставлена, строка, указанная в адресе. Эта команда отличается от команды **a** только тем, куда помещается введенный текст. Адрес 0 для этой команды недопустим. Максимально в одной строке

с терминала можно ввести **LINE_MAX** байтов (включая символ новой строки). Параметр **LINE_MAX** определен в файле **limits.h**.

(.,.+1)j

Команда соединения (**join**) соединяет подряд идущие строки путем удаления соответствующих символов новой строки. Если указан один адрес, эта команда ничего не делает.

(.)kx

Команда пометки (**mark**) помечает указанную строку символом **x**, который должен быть буквой нижнего регистра (**a-z**). Адрес **'x** после этого будет указывать на данную строку; текущая строка (**.**) не изменяется.

(.,.)l

Команда выдачи (**list**) выдает указанные строки в недвусмысленном виде: некоторые непечатные символы (например, символы табуляции или забоя) представляются визуальными мнемоничными эквивалентами. Все остальные непечатные символы печатаются в виде восьмеричных кодов, длинные строки переносятся и конец каждой строки помечается символом **\$**. Команду **l** можно добавлять к любой команде, кроме **e**, **f**, **g** и **w**.

(.,.)ma

Команда перемещения (**move**) переставляет указанную строку или строки после строки с адресом **a**. При использовании адреса **0** в качестве **a** указанная строка или строки перемещаются в начало файла. Если адреса попадает в диапазон перемещаемых строк, это является ошибкой; текущей строкой (**.**) становится последняя из перенесенных строк.

(.,.)n

Команда нумерации (**number**) выдает указанные строки, предвеля каждую номером строки и символом табуляции; текущей строкой (**.**) становится последняя выданная строка. Команду **n** можно добавлять к любой команде, кроме **e**, **f**, **g** и **w**.

(.,.)p

Команда печати (**print**) выдает (печатает) указанные строки; текущей строкой (**.**) становится последняя выданная строка. Команду **p** можно добавлять к любой команде, кроме **e**, **f**, **g** и **w**. Например, **dp** удаляет текущую строку и печатает новую текущую строку.

P

Редактор будет выдавать приглашение ***** для ввода всех последующих команд. Команда **P** переключает этот режим; первоначально он отключен.

q

Команда выхода (**quit**) заставляет **ed** прекратить работу. Автоматическая запись изменений в файл не выполняется; см. однако раздел **"ДИАГНОСТИКА"** ниже.

Q

Происходит выход из редактора без проверки, были ли сделаны изменения в буфере с момента выполнения последней команды **w**.

(\$)g файл

Команда чтения (**read**) читает содержимое **файла** в буфер. Если **файл** не указан, используется текущее запомненное имя файла, если оно есть (см. команды **e** и **f**). Текущее запомненное имя файла не изменяется, если это только не самое первое имя файла, упомянутое с момента вызова **ed**. Адрес **0** допустим для **g** и вызывает считывание файла в начало буфера. Если чтение выполнено успешно,

выдается количество прочитанных символов; текущей строкой (.) становится последняя прочитанная строка. Если вместо файла указан !, остаток строки рассматривается как команда интерпретатора shell (см. [sh\(1\)](#)), стандартный выходной поток которой должен быть считан.

Например, `$r !ls` добавляет текущий каталог к концу редактируемого файла. Такая команда shell не запоминается в качестве текущего имени файла.

(.,.)s/регулярное_выражение/замена/

или **(.,.)s/регулярное_выражение/замена/g**

или **(.,.)s/регулярное_выражение/замена/n** **n = 1-512**

Команда подстановки (substitute) проверяет каждую указанную строку на вхождение указанного **регулярного_выражения**. В каждой строке, в которой найдено сопоставление, все (непересекающиеся) вхождения сопоставившихся подстрок заменяются на указанную **замену**, если после команды указан индикатор глобальной замены **g**. Если глобальный индикатор не указан, заменяется только первое вхождение сопоставившейся подстроки. Если после команды указано число **n**, только **n**-ное вхождение сопоставившейся подстроки заменяется в каждой указанной строке. Если ни в одной из указанных строк замену осуществить не удалось, это рассматривается как ошибка. Вместо / для ограничения **регулярного_выражения** и **замены** можно использовать любой символ, кроме пробела и новой строки; текущей строкой (.) становится последняя строка, в которой произошла замена. См. также [последний абзац](#) в разделе "ОПИСАНИЕ" ниже.

Амперсанд (&) в строке **замены** заменяется подстрокой, сопоставившейся с регулярным выражением в текущей строке. Специальную интерпретацию символа & в этом контексте можно отменить, предварив его символом \. В качестве более общего свойства, символы \n, где **n** - цифра, заменяются на текст, сопоставившийся с **n**-ым подвыражением указанного регулярного_выражения, находящимся между \ (и \). В случае вложенных подвыражений в круглых скобках, **n** определяется путем подсчета вхождений \ (, начиная слева. Если символ % является единственным символом в **замене**, в качестве замены в текущей команде подстановки используется **замена** из последней выполненной команды подстановки. Символ % теряет свое специальное значение, если в строке замены есть и другие символы или когда перед ним стоит \.

Строку можно разбить на несколько, подставляя в нее символы новой строки. Символ новой строки в **замене** должен быть замаскирован путем ввода перед ним обратной косой (\). Такая подстановка недопустима как часть **списка_команд** в командах **g** и **v**.

(.,.)ta

Эта команда работает так же, как команда **m**, но после адреса **a** (который может быть **0**) помещается копия указанных строк; текущей строкой (.) становится последняя скопированная строка.

u

Команда отмены (undo) отменяет действие последней команды, изменившей что-либо в буфере, а именно последней из команд **a**, **c**, **d**, **g**, **i**, **j**, **m**, **r**, **s**, **t**, **v**, **G** или **V**.

(1,\$)v/регулярное_выражение/список_команд

Эта команда аналогична глобальной команде **g**, но на первом шаге помечаются строки, не сопоставившиеся с **регулярным_выражением**.

(1,\$)V/регулярное_выражение/

Эта команда аналогична интерактивной глобальной команде **G**, но на первом шаге помечаются строки, не сопоставившиеся с **регулярным_выражением**.

(1,\$)w файл

Команда записи (**w**rite) записывает указанные строки в **файл**. Если **файл** не существует, он создается с режимом **666** (доступен на чтение и запись для всех), если только маска создания файла не требует другого режима; см. описание специальной команды **umask** в **sh(1)**. Текущее запомненное имя файла не изменяется, если только **файл** - не первое имя файла, указанное с момента вызова **ed**. Если **файл** не указан, используется текущее запомненное имя файла, если оно есть (см. команды **e** и **f**); текущая строка (**.**) не изменяется. Если команда выполнена успешно, выдается количество записанных символов. Если вместо **файла** указать **!**, остаток строки рассматривается как команда интерпретатора **shell**, стандартным входным потоком для которой будут указанные строки. Такая команда **shell** не запоминается в качестве текущего имени файла.

(1,\$)w файл

Эта команда аналогична описанной выше команде записи, за исключением того, что добавляет указанные строки в конец **файла**, если он существует. Если **файл** не существует, он создается, как описано выше для команды **w**.

X

Выдается приглашение ввести ключ, который используется в последующих командах **e**, **r** и **w** для дешифрования и шифрования текста с помощью алгоритма **crypt(1)**. Редактор сам делает предположение о том, является ли текст, считанный командами **e** и **r**, зашифрованным. Пустой ключ отключает шифрование. См. также опцию **-x** командной строки **ed**.

(\$)=

Выдается номер указанной строки; эта команда не изменяет текущую строку (**.**).

!команда_shell

Остаток строки после **!** посылается командному интерпретатору системы UNIX для интерпретации в качестве команды. См. **sh(1)**. В тексте этой команды незамаскированный символ **%** заменяется запомненным именем файла; если в качестве первого символа **команды_shell** указан **!**, он заменяется текстом предыдущей команды **shell**. Так что **!!** повторит последнюю выполненную команду **shell**. Если выполнялось любое расширение командной строки, выдается строка после расширения; текущая строка (**.**) не изменяется.

(.+1)<новая_строка>

Адрес в строке без команды вызывает выдачу указанной строки. Просто новая строка, без адреса, эквивалентна команде **+.1p**; таким образом, нажатие клавиши **Enter** удобно использовать для прохода по буферу.

Если послан сигнал прерывания (**ASCII DEL** или **BREAK**), **ed** выдает **?** и возвращается в режим ввода команд.

Некоторые ограничения размеров: не более **LINE_MAX** байтов в строке, 256 байтов в списке команд глобальной команды и **PATH_MAX** байтов в полных именах файлов (включая символы косой черты). Параметры **LINE_MAX** и **PATH_MAX** определены в файле **limits.h**. Ограничение на количество строк зависит от объема памяти: каждая строка занимает одно слово.

При чтении файла **ed** игнорирует символы **ASCII NUL**.

Если файл не заканчивается символом перевода строки, **ed** добавляет его и выдает сообщение, объясняющее, что было сделано.

Если закрывающий разделитель регулярного выражения или строки замены (например, /) будет последним символом перед новой строкой, этот ограничитель можно не вводить. При этом будет напечатана указанная строка. Следующие пары команд эквивалентны:

```
s/s1/s2 s/s1/s2/p
g/s1      g/s1/p
?s1      ?s1?
```

Ошибки

? Ошибка команды.

?файл Недопустимый файл. (Для получения детальных разъяснений используйте команды **h** и **H**).

Если в буфере были сделаны изменения после выполнения последней команды **w**, записавшей весь буфер, **ed** предупреждает пользователя при попытке уничтожить буфер **ed** с помощью команд **e** или **q**. Он выдает **?** и позволяет продолжить редактирование. Повторная команда **e** или **q** после этого будет выполнена. Опция командной строки **-s** подавляет эту возможность.

Файлы

\$TMPDIR

если эта переменная среды имеет не пустое значение, оно используется вместо **/var/tmp** в качестве имени каталога для хранения временного рабочего файла.

/var/tmp

если каталог **/var/tmp** существует, он используется в качестве каталога для хранения временного рабочего файла.

/tmp

если переменная среды **TMPDIR** не существует или имеет пустое значение и если не существует каталог **/var/tmp**, то каталог **/tmp** используется для хранения временного рабочего файла.

ed.hup

в этом файле сохраняются результаты редактирования, если отключается терминал.

/usr/lib/locale/локаль/LC_MESSAGES/uxcore.abi

файл сообщений для текущего языка (См. **LANG** в [environ\(5\)](#).)

ССЫЛКИ

[edit\(1\)](#), [ex\(1\)](#), [fspec\(4\)](#), [grep\(1\)](#), [regex\(5\)](#), [sed\(1\)](#), [sh\(1\)](#), [stty\(1\)](#), [umask\(1\)](#), [vi\(1\)](#)

ПРИМЕЧАНИЯ

Опция `-`, хотя и продолжает поддерживаться, заменена в документации опцией `-s`, соответствующей стандарту синтаксиса команд (Command Syntax Standard - см. [intro\(1\)](#)).

Опции и команды шифрования поставляются в пакете утилит шифрования (Encryption Utilities), который доступен только в Соединенных Штатах.

Команда `!` не может использоваться в глобальных командах `g` и `v`.

Команду `!` и `!` в качестве имени файла в командах `e`, `r` и `w` нельзя использовать, если редактор вызван из ограниченного командного интерпретатора (см. [sh\(1\)](#)).

Последовательность `\n` в регулярных выражениях не сопоставляется с символом новой строки.

Если входной поток редактора получается из командной строки (например, `ed файл < файл_команд_ed`), при первой же ошибке происходит выход из редактора.

На работу `ed` влияют следующие переменные среды: `LANG`, `LC_ALL`, `LC_CTYPE`, `LC_MESSAGES`, `LC_TIME`, `TZ` (см. `LANG` в [environ\(5\)](#)).

Copyright 1994 Novell, Inc.

Copyright 2000 В. Кравчук, [OpenXS Initiative](#), перевод на русский язык

Поиск по тексту MAN-ов:

Партнёры:



При поддержке
inferno solutions*

Хостинг:



Hoster.ru
хостинг провайдер

[Закладки на сайте](#)
[Проследить за страницей](#)

Created 1996-2024 by [Maxim Chirkov](#)
[Добавить](#), [Поддержать](#), [Вебмастеру](#)