

Admins SU

Админские штучки

Home -> Linux -> Linux для начинающих -> Примеры использования grep в Linux.

01
МАРТА



Примеры использования grep в Linux.

5

Max Isaev Linux для начинающих

Сегодня я расскажу о возможностях утилиты grep. Изначально она появилась еще в UNIX и на данный момент является частью любого дистрибутива BSD, Mac OS, Linux. Предназначение ее — поиск строк согласно условию, описанному регулярным выражением.

Существуют модификации классического grep — egrep, fgrep, rgrep. Все они заточены под конкретные цели, при этом возможности grep перекрывают весь функционал.

Итак, посмотрим на утилиту поближе.

Самым простым примером использования команды является вывод строки, удовлетворяющей шаблону, из файла.

Например мы хотим найти строку, содержащую 'user' в файле /etc/mysql/my.cnf. Для этого воспользуемся следующей командой:

```
1 grep user /etc/mysql/my.cnf
```

Мы нашли нужную нам строку, но как потом найти ее в самом файле? Для этого мы добавим указание номера строки с помощью ключа -b.

```
1 grep -bi 'user' /etc/mysql/my.cnf
```

Ключ -i в примере выше делает наш поиск регистронезависимым.

Теперь давайте попробуем вывести содержимое этого конфига без комментариев:

```
1 grep -v '^#' /etc/mysql/my.cnf
```



Рубрики

Linux
Linux для начинающих
Mac
Mac Tricks
Безопасность
Переводы
Разное
Разработка
Системы мониторинга
Управление инфраструктурой

Свежие записи

Решаем проблему роста Windows. Что делать с разросшимся WINSXS.

Как добавить постоянные статические маршруты (persistent static routes) в Mac OS X.

Инструкция по использованию Fortinet SSL VPN Client.

Знакомство с CMake. Часть 3. CMakeCache, модули CMake, зависимости сборки.



Admins.su

Благодаря ключу `-v` мы исключаем строки с символом `#`. «`^`» означает, что «`#`» должна находиться в начале строки.

Ниже я привел пример команды, обрезающей комментарии всех видов, а также пустые строки:

```
1 | grep -vE '^[[:blank:]]*#|^$'.
```

grep часто используется в связке с программными каналами (pipes) для обработки вывода других утилит.

Например команда

```
1 | find /dev | grep sd
```

вернет список устройств, содержащих `sd` в названии.

Команда

```
1 | find / | grep lala
```

вернет все файлы, имеющиеся в системе, содержащие в названии `'lala'`.

Раскраска grep.

Иногда удобно, когда искомое нами слово подсвечивается цветом.

Все это уже есть в grep, остается только включить. Для этого в команде можно использовать ключ `—color=auto`.

Пример:

```
1 | grep --color=auto a /etc/mysql/my.cnf
```

Искомая буква «а» подсветится цветом.

Для того, чтобы не писать каждый раз такой длинный ключ — создадим псевдоним grep в файле `~/.bashrc` (Применимо только к bash):

```
1 | echo export GREP_OPTIONS='--color=auto' &gt;
```

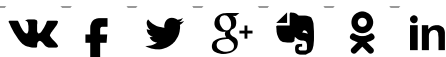
Теперь надо перечитать конфигурацию bash:

```
1 | reset
```

maybe? Интересная песочница для отладки операций с файлами в скриптах Linux.

Оцените материал!

★★★★★ (Нет оценок)



Все, теперь подсветка искомого слова будет включена по умолчанию.

Регулярные выражения.

Регулярные выражения — мощный инструмент, используемый во многих утилитах Linux. grep — не исключение. С помощью этих выражений можно описать любую искомую строку.

С помощью «|» можно указать условие «или». В примере ниже grep выдаст все строки с вхождениями user или db. Для обработки подобных выражений необходимо использовать ключ -E или использовать egrep.

```
1 | grep -E -i -w 'user|db' /etc/mysql/my.cnf
```

Символами ^ и \$ мы можем указать местоположение искомого слова. ^ обозначает начало строки, \$ — его конец.

Найдем все строки, начинающиеся на #

```
1 | grep -i '^#' /etc/mysql/my.cnf
```

Теперь найдем все строки, заканчивающиеся на «!»

```
1 | grep -i '!$' /etc/mysql/my.cnf
```

А теперь отобразим номера пустых строк:

```
1 | grep -bi '^$' /etc/mysql/my.cnf
```

Поиск строк, содержащих цифру:

```
1 | grep -i '[0-9]' /etc/mysql/my.cnf
```

Поиск строк с двумя цифрами подряд:

```
1 | grep -i '[0-9][0-9]' /etc/mysql/my.cnf
```

Найдем все строки из двух символов:

```
1 | grep '^..$' /etc/mysql/my.cnf
```

Как вы заметили, точка является спецсимволом. Для того, чтобы



знаки в выражении, нам потребуется их экранировать символом \.

Смотрите также

Например:



```
T
Г 1 | grep '\Собаки/Никто и подумать не
```

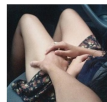
МОГ, ЧТО ОНИ...

Экранировать знак \ можно самим собой:

Реклама

```
1 | grep '\\ ' /etc/mysql/my.cnf
```

Муж хотел проверить свою жену и ...



Теперь попробуем найти все ip адреса, указанные
в формате XXX.XXX.XXX.XXX



```
L
1 | grep -E '[[[:digit:]]{1,3}\. [[[:digit:]]{1,3}\.
```

Напоследок приведу маленькую таблицу с
универсальными выражениями.

Как закрыть доступ в интернет
определенному приложению в Linux |



```
А
1 | [[[:alnum:]]] - символы букв и цифр.
2 | [[[:alpha:]]] - символы букв.
3 | [[[:blank:]]] - space и tab.
4 | [[[:digit:]]] - цифры: '0 1 2 3 4 5 6 7 8 9'.
```

Relap

[Комментарии](#) [Сообщество](#) [1 Войти ▾](#)[❤️ Рекомендовать](#) [🔗 Поделиться](#)[Новое в начале ▾](#)

ВОЙТИ С ПОМОЩЬЮ

ИЛИ ЧЕРЕЗ DISQUS [?](#)**Valdemar Fahrenheit** • 3 года назад

Не очень понятно `wmctrl -m | grep -i
"^mode: $"`

Результат пустой

^ | ▾ • Ответить • Поделиться ›

**Александр Коновалов** • 3 года назад

Все круто, только `grep -b` не показывает номер строки вхождения, для этого нужен `grep -n`
`echo -e "mama\npapa\ndeda" | grep -b
"de"`

10:deda

`echo -e "mama\npapa\ndeda" | grep -n
"de"`

3:deda

^ | ▾ • Ответить • Поделиться ›

**Георгий** • 3 года назад

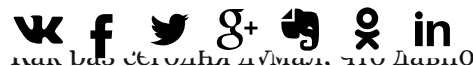
Классно так всё объяснили! Надеюсь, ресурс не умрёт. Очень всё доступно изложено, с наглядными примерами...в избранное!


^ | ▾ • Ответить • Поделиться ›


**Max Isaev** → Георгий

• 3 года назад

Георгий, спасибо.



MySQL 5.6 с поддержкой NoSQL на Debian Squeeze. 

 Массовое переименование файлов в Linux.