

# Профессиональное использование консоли с помощью GNU Screen

Дмитрий Васильев

Думаю, никто не будет возражать против того, что наиболее используемым инструментом администратора систем UNIX является текстовая консоль. Консоль позволяет выполнять практически все административные задачи, и тем более без нее нельзя обойтись при управлении серверами, находящимися в другом районе города, или даже в другом городе, или в другой стране. Большинство администраторов могут достаточно лихо пользоваться консольными командами, но они могут с удивлением заметить, что более опытные коллеги каким-то невообразимым образом успевают делать еще больше. В этой статье я рассмотрю один из ключей к эффективному использованию текстовой консоли – менеджер текстовых окон GNU Screen (далее просто Screen).

## Возможности Screen

Одна из основных возможностей Screen – создание и управление несколькими консольными окнами или текстовыми приложениями в одном-единственном окне терминала. В один момент времени на экране терминала может находиться только одно из окон (несколько их может быть только в случае, если они делят экран терминала на части). Screen позволяет быстро переключаться между окнами, сохранять и позднее возвращаться к текстовой сессии и открывать доступ к ней другим пользователям. Плюс к этому Screen позволяет давать окнам заголовки, отслеживать активность в каждом окне и обмениваться содержимым окон с помощью копирования и вставки. И вдобавок Screen позволяет переназначать клавиши, управлять настройкой эмуляции терминала и решать еще много небольших задач, связанных с текстовыми окнами,

в которых может и не быть постоянной необходимости, но которые могут помочь в выполнении административной работы в отдельные моменты времени.

На **рис. 1** изображен Screen с тремя именованными окнами (Work, Configuration, Logs) и открытым в нем текстовым редактором Vim с примером файла конфигурации Screen.

## Первый запуск

Screen может быть запущен с помощью консольной команды screen, но прежде всего проверим, установлен ли Screen в вашей системе, с помощью следующей команды:

```
$ which screen
```

```
/usr/bin/screen
```

Если команда вернула путь, как в примере выше, можно сразу начинать пользоваться, в противном слу-

чае Screen может быть установлен стандартными средствами поддержки пакетов вашей операционной системы или скачан с официального сайта <http://www.gnu.org/software/screen>.

Теперь мы можем запускать Screen, для чего просто набираем:

```
$ screen
```

После запуска на экран может появиться информация о версии и лицензии (при рассмотрении конфигурации мы рассмотрим, как можно отменить вывод данного сообщения), и в этом случае необходимо нажать клавишу «Пробел» или «Ввод», после чего мы попадаем на первый взгляд в обычную текстовую оболочку.

Надо заметить, что в командной строке сразу можно указать команду для запуска определенного приложения, например, для запуска top надо набрать:

```
$ screen top
```

Хорошо, теперь Screen запущен, но, на первый взгляд, нет никаких внешних изменений – мы можем работать в запущенной оболочке или использовать top. Хитрость в том, что во время работы Screen передает все нажатия клавиш непосредственно приложению, но (по умолчанию) оставляет для себя управляющие команды, начинающиеся с комбинации клавиш `<Ctrl> + <a>`.

Эту управляющую комбинацию клавиш можно поменять через файлы конфигурации (которые мы рассмотрим позже), но во всех примерах будет рассматриваться только конфигурация по умолчанию. Таким образом, все клавиатурные команды Screen состоят из двух клавишных комбинаций – управляющей `<Ctrl> + <a>` и непосредственно клавиши или клавишной комбинации, представляющей команду. Практически каждая команда также имеет свой текстовый аналог, который можно выполнить либо через файл конфигурации, либо выполнив следующую команду:

- **Клавишная комбинация:** `<Ctrl> + <a> + <:>`.
- **Текстовая команда:** colon.
- **Описание:** вывести приглашение для ввода текстовых команд в интерактивном режиме.

В данном случае мы должны выполнить две клавиатурные комбинации, сначала нажав `<Ctrl> + <a>`, а затем клавишу двоеточия. При этом Screen выведет в нижней части экрана приглашение для ввода команды в виде двоеточия. Ту же самую команду, которую мы только что выполнили с помощью клавишной комбинации, можно выполнить через конфигурационный файл с помощью текстовой команды colon, что может, например, понадобиться для вывода приглашения о вводе команд при старте или переназначении комбинации клавиш, вызывающих это приглашение. Далее, как и в примере выше, для каждой клавишной комбинации я буду писать соответствующие ей ключевые слова, которые можно использовать в файлах конфигурации или вводить с помощью `<Ctrl> + <a>` и `<:>`.

Здесь может возникнуть закономерный вопрос – а что если приложению нужно непосредственно передать комбинацию клавиш `<Ctrl> + <a>`, которая уже используется в Screen как служебная? Например, такая же комбинация используется библиотекой редактирования командной строки readline для выполнения перехода в начало строки. Для этого случая в Screen предусмотрена следующая команда:

- **Клавишная комбинация:** `<Ctrl> + <a> + <a>`.
- **Текстовая команда:** meta.
- **Описание:** передать текущему предложению клавишную комбинацию `<Ctrl> + <a>`.

## Управление окнами

Пришло время разобраться с командами для управления окнами. При запуске Screen создает одно окно (если

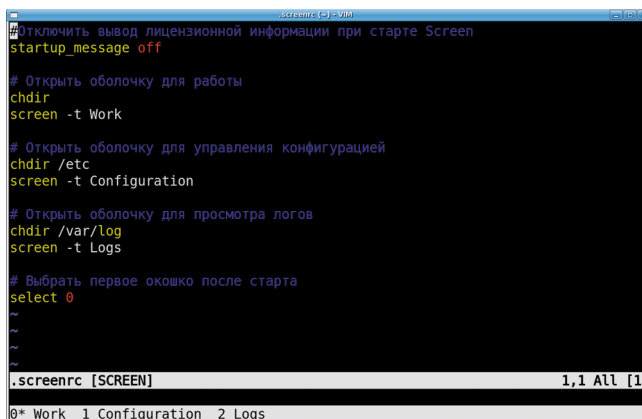


Рисунок 1. Screen с тремя именованными окнами (Work, Configuration, Logs) и открытым в нем текстовым редактором Vim

только в конфигурации не указано создание сразу нескольких окон) и присваивает ему номер 0.

Новое окно с командной оболочкой можно создать с помощью команды ниже:

- **Клавишная комбинация:** `<Ctrl> + <a> + <c>`.
- **Текстовая команда:** screen.
- **Описание:** создать новое окно.

При этом Screen создаст новое окно и присвоит ему следующий по порядку номер. В нашем случае новым окном будет окно с номером 1.

Теперь у нас есть уже два текстовых окна, из которых текущим является только что созданное окно с номером 1, а окно с номером 0 находится в фоновом режиме. Но надо заметить, что даже если окно находится в фоновом режиме, оно может воспринимать вывод запущенного в нем приложения, как если бы оно было текущим. Например, мы могли бы запустить в окне 0 процесс, который выводит дополнительную информацию через определенные промежутки времени (для примера, это может быть команда «tail -f»), и затем переключиться в окно 1 для работы в текстовом редакторе.

Через какое-то время, если мы захотим посмотреть, что происходит в окне 0, мы можем переключиться в это окно и увидеть весь вывод работающего там процесса.

Таблица 1. Наиболее часто используемые команды для управления окнами

Клавишная комбинация	Текстовая команда	Описание
<code>Ctrl + a</code> w	windows	Выводит список номеров и заголовков всех окон в нижней строке экрана
<code>Ctrl + a</code> "	windowlist	Выводит интерактивный список номеров и заголовков всех окон, из которого можно выбрать окошко для переключения
<code>Ctrl + a</code> Номер окна	select	Выбор окна по номеру. Например, команда <code>&lt;Ctrl&gt; + &lt;a&gt;</code> и 0 делает текущим окно с номером 0
<code>Ctrl + a</code> n	next	Переключиться на следующее по порядку окно или на первое окно, если мы находились в последнем окне
<code>Ctrl + a</code> p	prev	Переключиться на предыдущее по порядку окно или на последнее окно, если мы находились в первом окне
<code>Ctrl + a</code> '	select	Вывести приглашение ввода номера окна для переключения
<code>Ctrl + a</code> <code>Ctrl + a</code>	other	Переключиться на последнее используемое окно
<code>Ctrl + a</code> N	number	Показать номер и заголовок текущего окна
<code>Ctrl + a</code> A	title	Установить заголовок текущего окна
<code>Ctrl + a</code> k	kill	Принудительное закрытие окна, например, в случае если приложение не удается закрыть обычным способом

Затем можно переключиться обратно в окно редактирования.

В предыдущем параграфе я описал пример с переключением между окнами, но пока не сказал, какие команды для этого используются. Рассмотрим подробнее наиболее часто используемые команды для управления окнами (см. **таблицу 1**).

При закрытии какого-либо из окон номера других окон не меняются и в нумерации появляются «дырки», которые позднее могут быть заполнены новыми окнами. Хотя в Screen имеется команда для изменения номера окна, она здесь не рассматривается и остается для самостоятельного изучения. В случае если закрыты все окна, Screen завершает свою работу и выводит сообщение:

```
[screen is terminating]
```

## Работа с текстом

Теперь рассмотрим работу с содержимым окон – копирование и вставку текстового содержимого одного окна в другое, мониторинг окон, сохранение текстовой копии экрана и другие похожие задачи (см. **таблицу 2**).

В режиме копирования Screen запускает текстовый редактор, похожий на редактор Vi. Наиболее часто используемые команды редактора:

- клавиши управления курсором перемещают курсор по экранному буферу;
- клавиша «Пробел» отмечает начало и конец региона для копирования. При втором нажатии «Пробел» отмеченный текст копируется в регистр по умолчанию и Screen выходит из режима копирования.

Таблица 2. Команды для работы с содержимым окон

Клавишная комбинация	Текстовая команда	Описание
Ctrl + a	M	monitor
		Включить/выключить мониторинг активности для текущего окна. Когда мониторинг включен и в окне, находящемся в фоновом режиме, производится вывод текста, в служебной строке выводится информация об активности
-	silence	Включить/выключить мониторинг отсутствия активности для текущего окна. Когда мониторинг включен и в окне, находящемся в фоновом режиме, некоторый период времени не производится вывод текста, в служебной строке выводится информация об отсутствии активности
Ctrl + a	h	hardcopy
		Записать текстовое изображение текущего окна в файл hardcopy.номер_окна в каталоге Screen по умолчанию
Ctrl + a	H	log
		Включить/выключить трассировку текста текущего окна в файл screenlog.номер_окна в каталоге Screen по умолчанию
Ctrl + a	[	copy
		Войти в режим копирования. Дополнительные команды этого режима описаны ниже
Ctrl + a	]	paste
		Вставить в окно текст из регистра по умолчанию (куда он предварительно был скопирован с помощью команды выше)

Таблица 3. Команды для работы с регионами экрана

Клавишная комбинация	Текстовая команда	Описание
Ctrl + a	S	split
		Разделить текущее окно или регион на два новых региона. При этом размеры всех регионов меняются для получения дополнительного пространства на экране, и в новом регионе отображается пустое окно
Ctrl + a	Tab	focus
		Переместить текущий фокус отображения на регион ниже или на верхний регион, если текущим является самый нижний регион
Ctrl + a	Q	only
		Закрыть все регионы, кроме текущего
Ctrl + a	X	remove
		Закрыть текущий регион

## Несколько окон на экране

По умолчанию Screen оставляет видимым на экране только одно логическое окно, но при желании есть дополнительная возможность разделить экран терминала на горизонтальные регионы. Таким образом можно отображать сразу несколько окон на одном экране терминала. Рассмотрим подробнее команды для работы с регионами экрана (см. **таблицу 3**).

На **рис. 2** изображен Screen с тремя именованными окнами (Work, Configuration, Logs) и разделенным на два региона экраном. В верхнем регионе открыто окно Work с текстовым редактором Vim и примером файла конфигурации Screen. В нижнем регионе – окно Configuration с частью общего файла конфигурации /etc/screenrc.

## Консольные сессии

И в завершение мы рассмотрим одну из самых популярных возможностей Screen. Представим следующую ситуацию: системный администратор, находясь дома, заходит на удаленный сервер, открывает несколько окон, запускает необходимые задачи, затем закрывает соединение и едет на работу. На работе он опять заходит на тот же сервер и восстанавливает открытые дома окна с результатами выполненных задач. Звучит немного фантастично? Ни в коем случае! Консольные сессии Screen как раз и предназначены для осуществления этой задачи. Они позволяют отсоединить от терминала текущую сессию Screen со всеми окнами и выполняемыми приложениями и затем в любой момент времени присоединиться к ней опять. При этом все выполняемые под Screen приложения будут продолжать работать и использовать свои окна для вывода информации.

Плюс ко всему консольных сессий может быть несколько, но в этой статье мы не будем на этом подробно останавливаться.

Рассмотрим команды отсоединения консольной сессии от терминала (см. **таблицу 4**).

Чтобы присоединиться обратно к отсоединенной сессии или посмотреть список доступных сессий, необходимо запустить Screen с одним из следующих ключей:

- **screen -r** – присоединиться к отсоединенной сессии.
- **screen -R** – присоединиться к первой подходящей отсоединенной сессии. Это может быть полезно в случае, если используется несколько сессий.
- **screen -x** – присоединиться к уже работающей сессии.
- **screen -ls** – вывести список доступных сессий.

## Конфигурация

При запуске Screen по умолчанию выполняются команды из файла /etc/screenrc и файла .screenrc из домаш-

него каталога пользователя. Многие команды, которые можно использовать в файлах конфигурации, были рассмотрены в таблицах выше в поле «Текстовая команда». Ниже представлен пример конфигурации Screen из файла .screenrc:

```
# Отключить вывод лицензионной информации при старте
Screen
startup_message off

# Открыть оболочку для работы
chdir
screen -t Work

# Открыть оболочку для управления конфигурацией
chdir /etc
screen -t Configuration

# Открыть оболочку для просмотра логов
chdir /var/log
screen -t Logs

# Выбрать первое окошко после старта
select 0
```

Этот достаточно простой файл конфигурации открывает три окна с именами Work, Configuration, Logs соответственно в личном каталоге пользователя, в каталоге /etc и каталоге /var/log. После запуска на экране будет находиться первое окно с именем Work. На рис. 1 представлен пример работы Screen с этим файлом конфигурации.

## Что дальше?

Screen имеет огромное количество различных возможностей, и практически невозможно описать их все в одной статье, поэтому некоторая часть функциональности осталась «за кадром».

Вот некоторые возможности/настройки Screen, которые не были рассмотрены выше и остаются для самостоятельного изучения:

- Команды настройки эмуляции терминала.
- Многопользовательские консольные сессии.
- Назначение или переназначение командам клавишных комбинаций.
- Вспомогательные опции текстовых команд. Например, опция -t команды screen, которая используется в примере файла конфигурации.
- Редко используемые команды, например команды, которые можно использовать при настройке запуска Screen, – echo и sleep.
- Переменные среды, которые могут быть полезны при настройке Screen.
- Дополнительные ключи запуска Screen.

Screen – это программа с большой историей, надо заметить, что первая версия появилась в 1987 году. На данный момент к Screen практически не добавляется новых возможностей (одной из причин может быть наличие уже большого их количества на данный момент) и производится в основном только исправление ошибок.

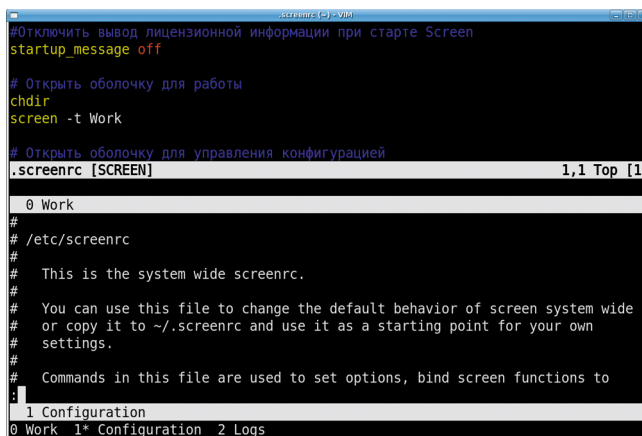


Рисунок 2. Screen с тремя именованными окнами (Work, Configuration, Logs) и разделенным на два региона экраном

Если кому-то не нравится относительная «старость» Screen или не нужно такое большое количество возможностей, то можно обратиться к одной из следующих программ:

- **dtach** (<http://dtach.sourceforge.net>) – основной акцент при разработке этой программы сделан на компактность. По аналогии со Screen она поддерживает возможность работы с консольными сессиями, но не имеет других возможностей Screen.
- **SplitVT** (<http://www.devolution.com/~slouken/projects/splitvt>) – позволяет разделить экран терминала на две половинки и работать сразу с двумя программными оболочками.
- **Tmux** (<http://sourceforge.net/projects/tmux>) – для работы с консольными сессиями Tmux использует модель «клиент-сервер».
- **Twinn** (<http://linuz.sns.it/~max/twinn>) – попытка создать текстовый оконный интерфейс, аналогичный графическим собратьям.

Итак, в этой статье мы рассмотрели возможности и большинство команд менеджера текстовых окон Screen. Я считаю, что изучение Screen или одной из похожих программ может вывести производительность и соответственно профессионализм системного администратора или программиста на качественно новый уровень. В любом случае это не будет пустой тратой времени. 🍷

Таблица 4. Команды отсоединения консольной сессии от терминала

Клавишная комбинация	Текстовая команда	Описание
Ctrl + a d	detach	Отсоединить консольную сессию и оставить ее запущенной в фоновом режиме. При этом Screen пишет: [detached]
Ctrl + a D D	pow_detach	Отсоединить консольную сессию, оставить ее запущенной в фоновом режиме и послать родительскому процессу Screen сигнал HANGUP. При запуске Screen из удаленного терминала это означает отсоединение Screen и затем закрытие соединения. При этом Screen пишет: [power detached]
Ctrl + a z	suspend	Отсоединить консольную сессию и оставить ее запущенной как фоновый процесс родительской командной оболочки. В этом случае для восстановления сессии, кроме команд, описанных ниже, могут быть использованы стандартные команды управления фоновыми процессами командной оболочки
Ctrl + a Ctrl + \	quit	Закрыть все окна сессии и выйти из Screen. Надо заметить, что многие дистрибутивы перепределяют клавишную комбинацию для этой команды, но текстовую команду можно использовать в любом случае