**«Использование системы контроля версий Git»**

**Основные команды Bash**

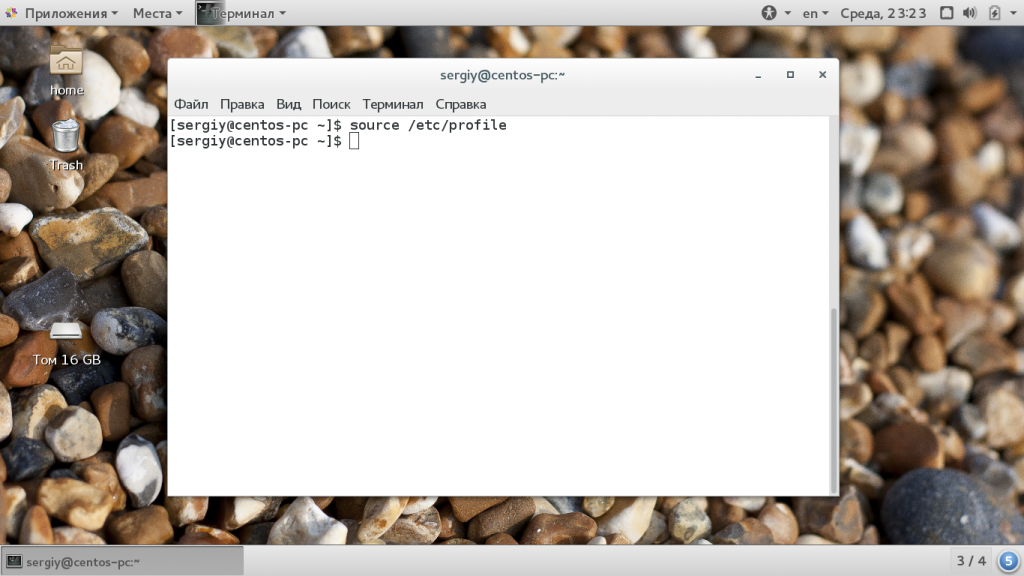
Bash - это командная оболочка или по-другому - интерпретатор команд, который используется по умолчанию в большинстве дистрибутивов Linux. Оболочка Bash обладает огромным количеством возможностей. Это больше чем интерпретатор, в умелых руках Bash может превратиться в полноценный язык программирования.

Основная задача оболочки - выполнять команды, но кроме утилит, расположенных в файловой системе, Bash имеет свой набор команд, многими из которых вы пользуетесь каждый день сами этого не понимая. Эти команды Bash находятся не на диске, а встроены в саму оболочку. В этой статье мы рассмотрим основные команды bash в linux, которые могут быть полезны в некоторых ситуациях.

1. SOURCE

Эта команда позволяет прочитать и выполнить все команды из указанного файла в текущей командной оболочке. Это значит, что все заданные в нем переменные окружения будут видны во всех скриптах или командах, которые вы будете выполнять. Напомню, что если вы запускаете обычный скрипт и объявляете в нем переменную, то она будет доступна только ему потому что для скриптов создается отдельный экземпляр оболочки. Он наследует переменные родителя, но не наоборот. Эта команда часто применяется для обновления окружения после chroot:

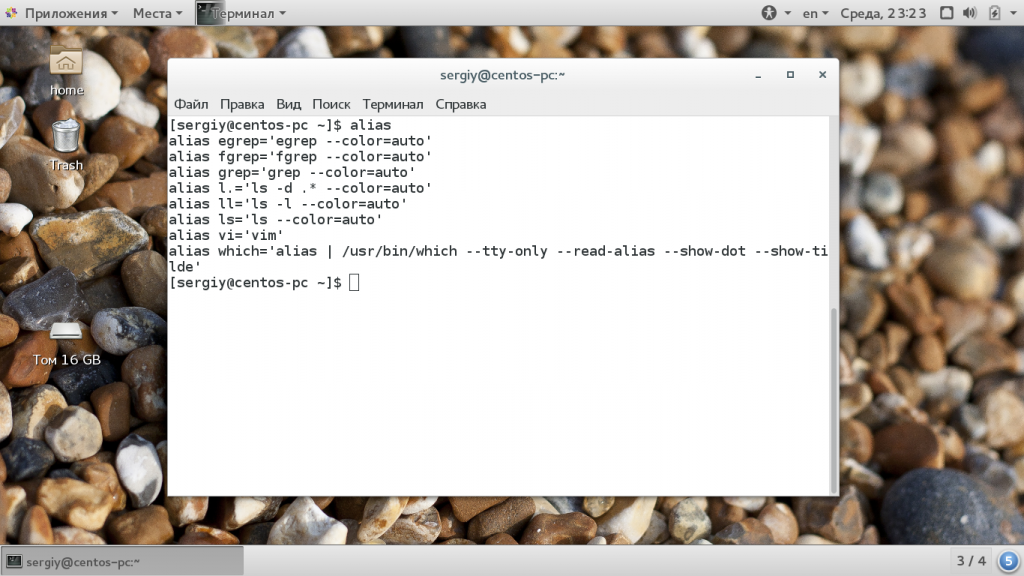
source /etc/profile



2. ALIAS

Команда alias позволяет создавать псевдонимы для других команд. Например, вы можете настроить так, чтобы когда вы вводите update, выполнялась команда sudo apt update && sudo apt upgrade. Если вызвать команду без параметров, то она выведет список уже заданных псевдонимов. Смотрите подробнее о том как создавать алиасы Linux в отдельной статье.

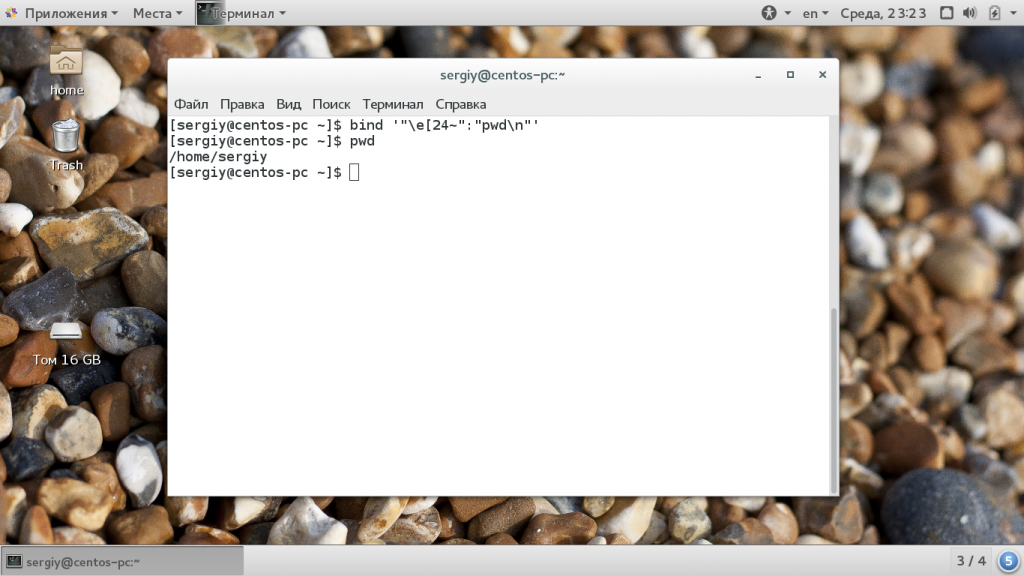
Alias



3. BIND

С помощью этой команды вы можете связать нажатие клавиши с определенной командной или функцией. Например, настроим, чтобы при нажатии F12 в окне терминала выполнялась команда pwd:

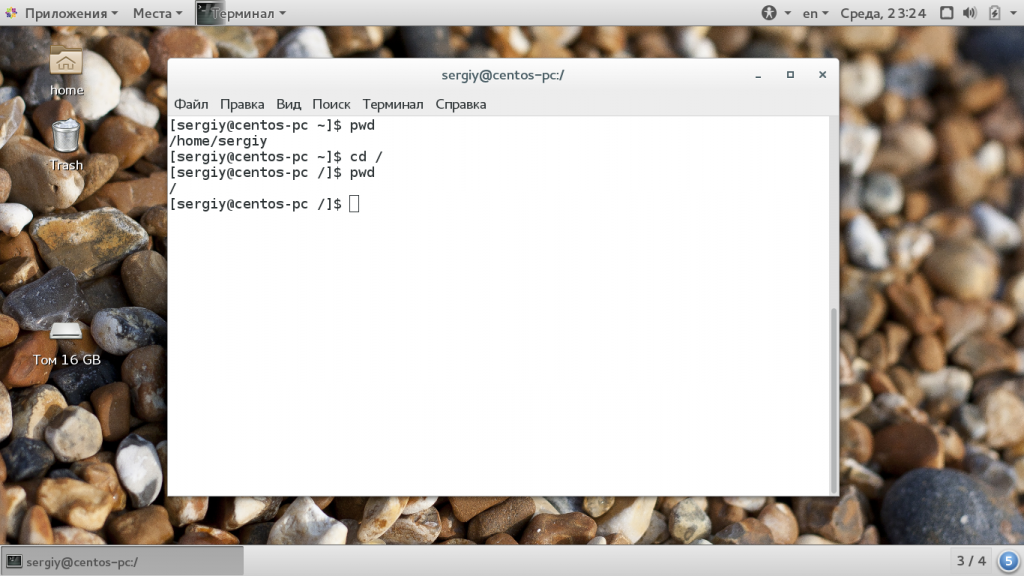
bind '"\e[24~":"pwd\n"'



4. CD

Команда cd используется очень часто при работе с папками в терминале. Она позволяет сменить текущую папку на произвольную. Можно использовать чтобы не набирать длинные пути, также она необходима при компиляции. По умолчанию, текущая папка - домашняя:

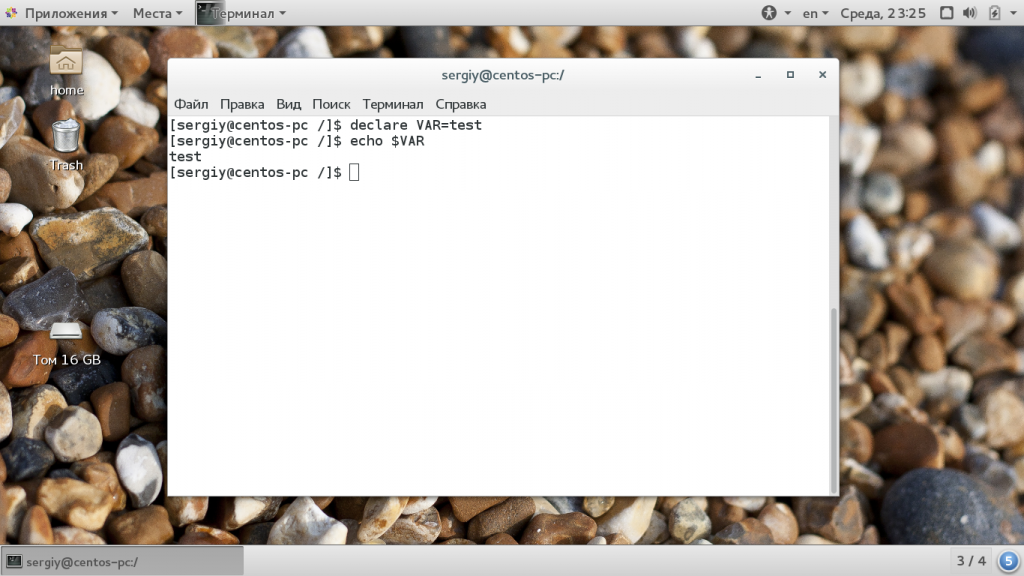
cd /



5. DECLARE

Встроенная команда bash - declare позволяет объявлять переменные. Опция -a позволяет вывести все уже существующие переменные окружения. В отличии от обычного объявления вы можете задать различные параметры, например, сделать переменную только для чтения с помощью опции -r. Также можно объявлять функции.

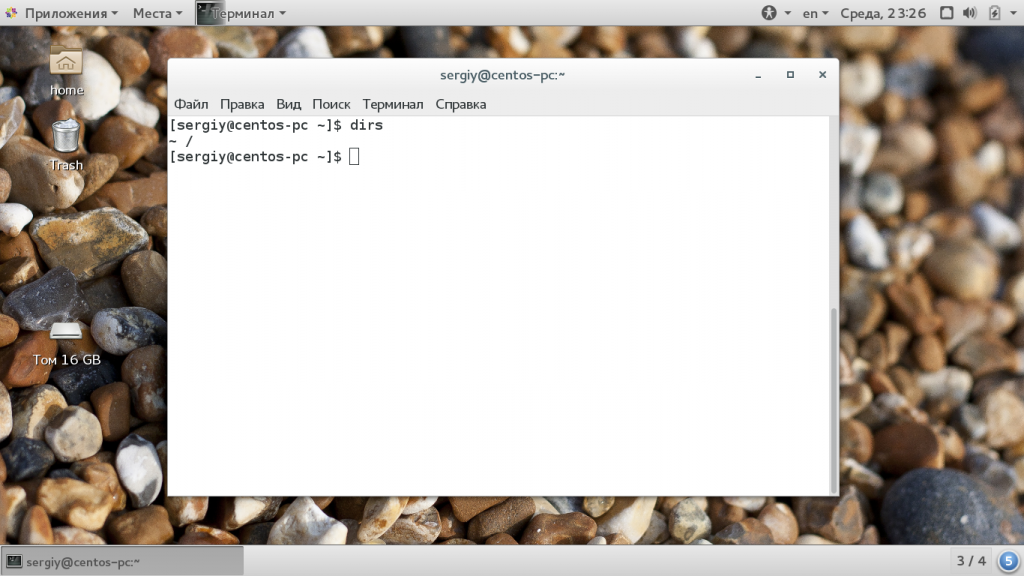
declare VAR=test



6. DIRS

Команда dirs отображает историю смены текущей папки. Каждая предыдущая папка отображается с новой строки. С помощью опции -n можно указать, сколько последних элементов выводить.

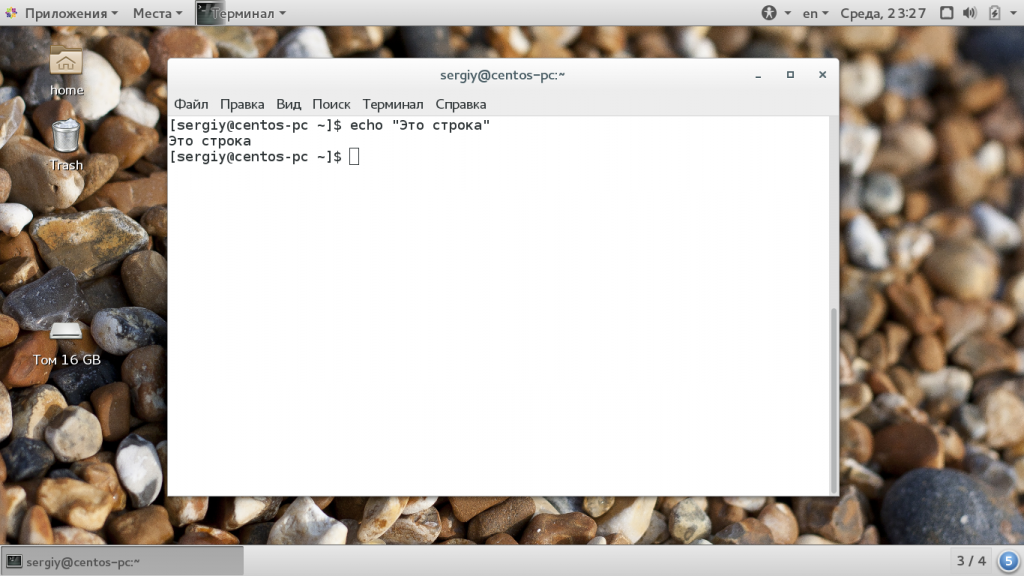
dirs.



7. ECHO

Это одна из самых простых, но в то же время очень полезных команд. Она выводит указанную строку на экран. Часто используется в различных скриптах или для отправки значений в конфигурационные файлы. Подробнее про команду echo смотрите в отдельной статье.

echo "Это строка"



8. ENABLE

Команда enable позволяет включать или отключать встроенные команды bash. Для включения встроенной команды достаточно передать ее в параметрах. Если вы хотите отключить команду, используйте опцию -n, после этого вместо встроенной команды, будет выполняться утилита из файловой системы.

enable -n echo

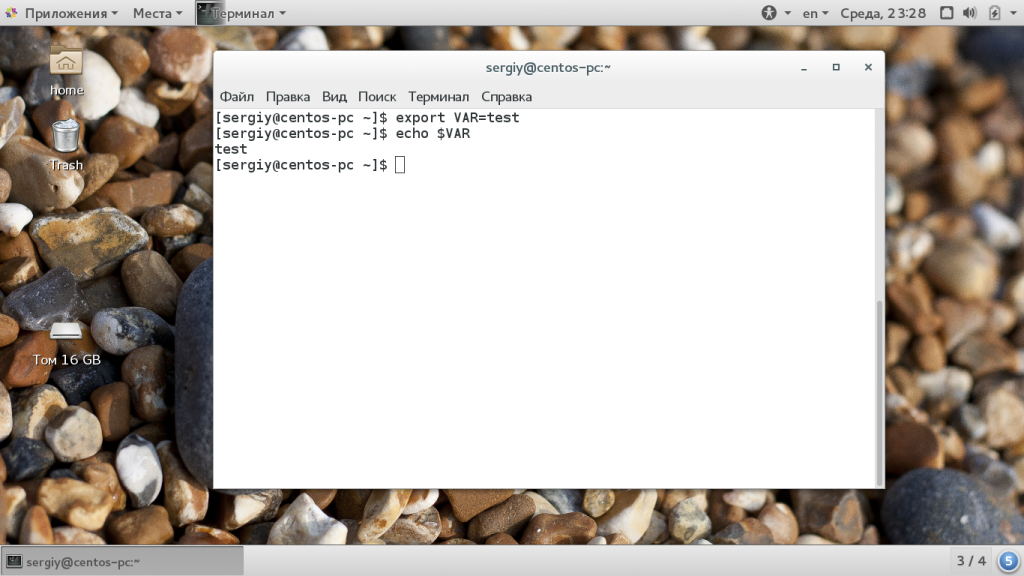
9. EXIT

Универсальная команда, которая позволяет закрыть сеанс оболочки, выйти из терминала или завершить текущую сессию.

10. EXPORT

Еще одна команда для объявления переменных. Все объявленные с помощью нее переменные экспортируются во внешнее окружение среды и будут доступны всем скриптам и программам. С помощью опции -p вы можете посмотреть экспортированные на данный момент переменные.

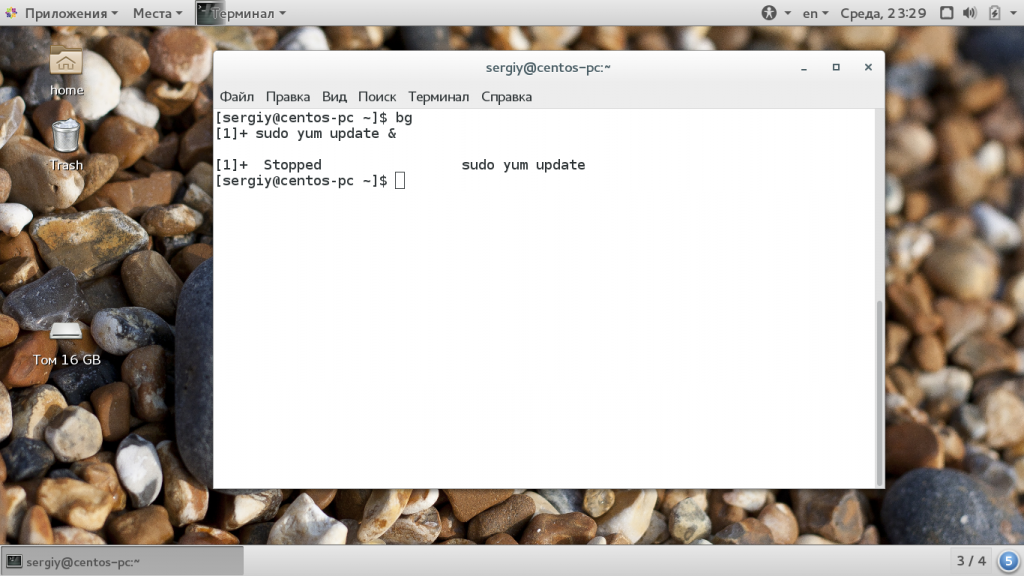
export VAR=test



11. BG

Команда bg позволяет посмотреть команды, которые выполняются в фоне. Если вы нажмете сочетание клавиш Ctrl+Z, то утилита будет свернута в фоновый режим. Этой командой вы можете посмотреть все такие процессы для данной оболочки.

bg



12. FG

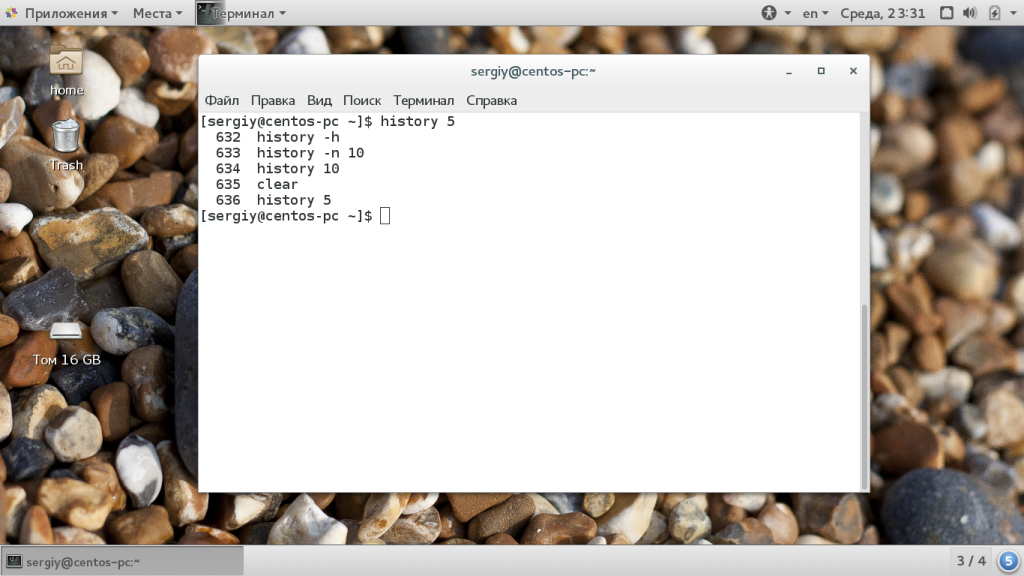
Эта команда позволяет восстановить команду из фона, в параметрах ей нужно передать только идентификатор нужного процесса, который вы можете узнать с помощью bg.

fg 1

13. HISTORY

Команда history linux позволяет посмотреть историю команд, которые вы выполняли ранее. Здесь сохраняется более 1000 последних команд, вы можете все их увидеть, выполнять по них поиск и так далее. Вы можете задать нужное количество записей просто передав нужное значение.

history 10



14. KILL

Команда kill может быть системной утилитой или командой оболочки. Она позволяет отправить указанному процессу сигнал. Например, SIGKILL, SIGTERM и другие. С помощью нее вы можете завершить процесс по его PID или идентификатору Bash.

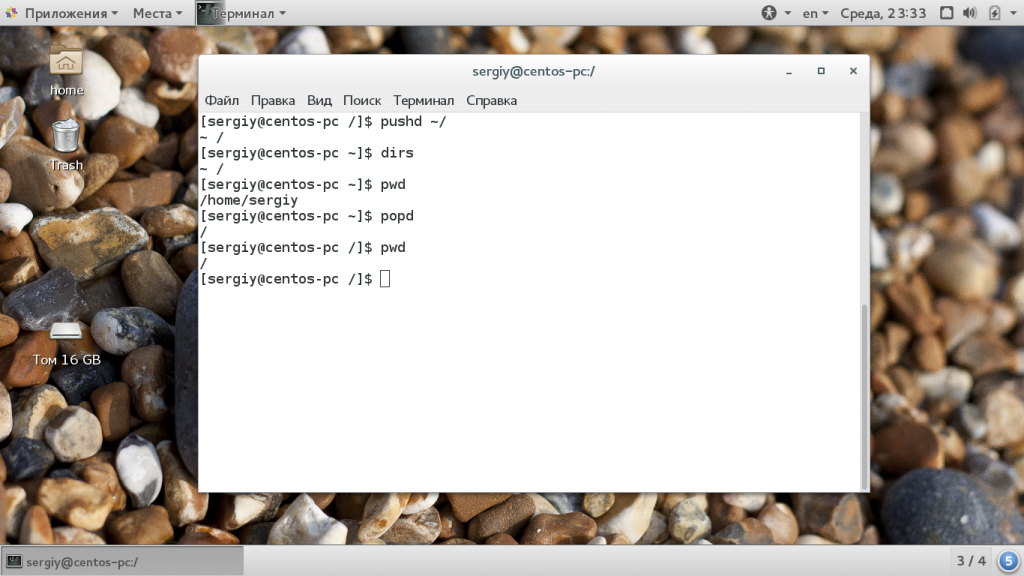
15. LET

Команда let выполняет функцию калькулятора, ей нужно передать математическое выражение, которое она подсчитает. Можно передать несколько таких выражений.

let "2+2"

16. POPD

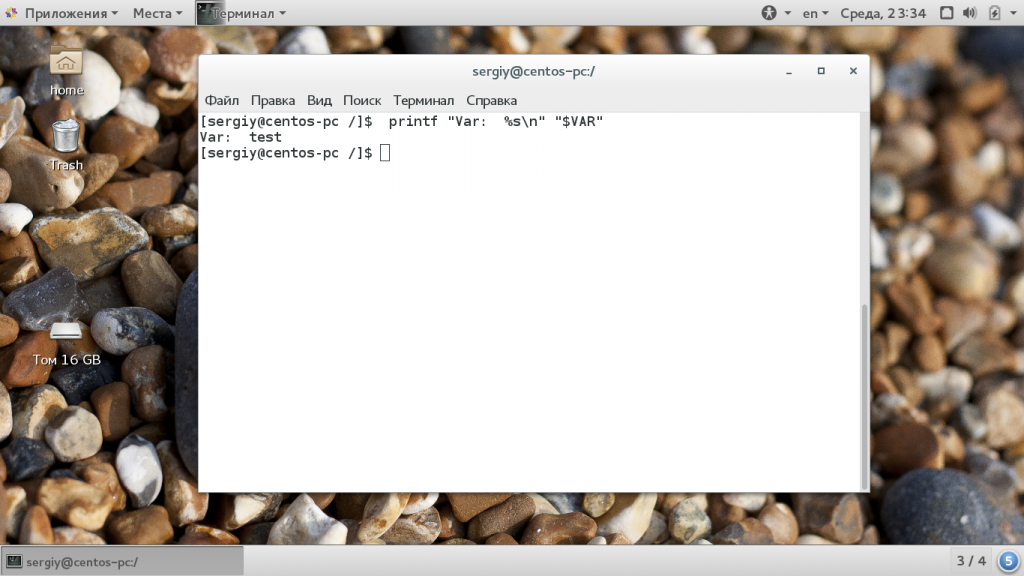
Это еще одна команда, которая используется для навигации по каталогах. Команда удаляет последнюю директорию из стека каталогов. Таким образом, вы перемещаетесь в ранее выбранную текущую папку с помощью cd, на уровень выше. Команде можно передать в параметрах сколько папок нужно удалить. Для этого добавьте плюс, а затем цифру.



17. PRINTF

Эта команда похожа на echo по своему назначению похожа на echo. Она позволяет вывести в терминал строку, но перед выводом вы можете настроить ее форматирование. Синтаксис форматирования очень похож на тот, что используется в популярных языках программирования, таких как Си или PHP. Например, с помощью этой команды bash вы можете подставить значения переменных в произвольные места строки:

printf "Var: %s\n" "$VAR"



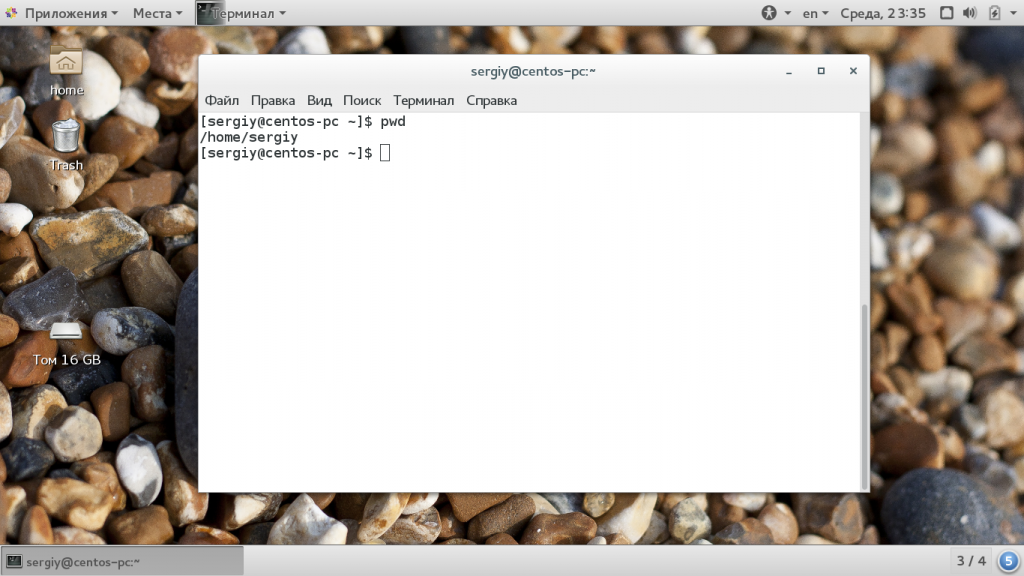
18. PUSHD

Команда pushd выполняет действие, обратное до popd и по назначению аналогична cd. Она устанавливает указанную папку на верхушку стека директорий, тем самым меняет текущую рабочую директорию на указанную вами.

19. PWD

Позволяет отобразить текущую рабочую папку:

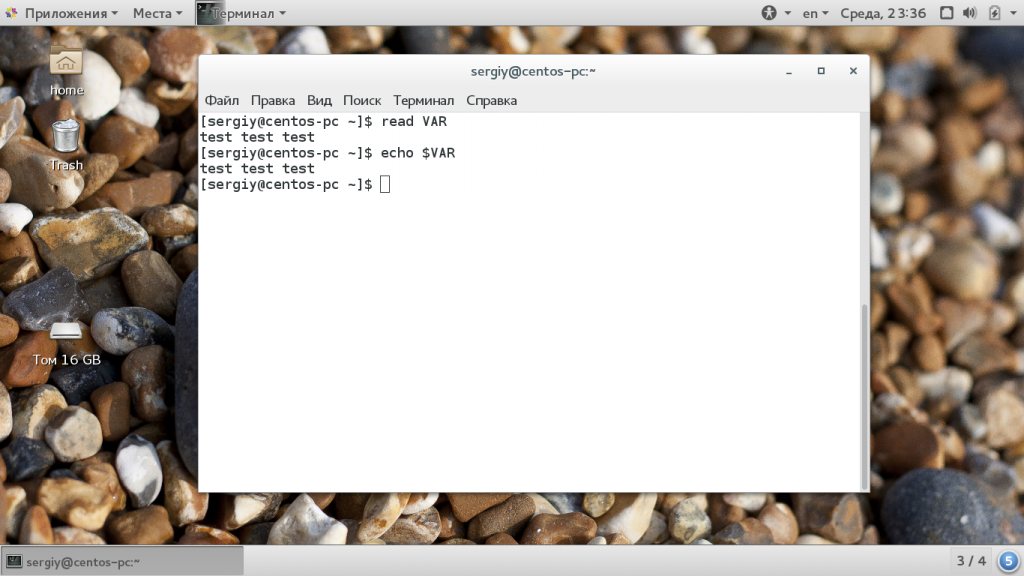
pwd



20. READ

С помощью команды read, вы можете прочитать строку из стандартного ввода или файла в переменную. Есть подобные ей команды оболочки bash, которые позволяют считывать строку в массив. Команде можно передать файл для считывания, а также несколько переменных:

read VAR



21. SUSPEND

Позволяет приостановить выполнение оболочки пока не будет отправлен сигнал SIGCONT. Можно также приостановить выполнение одного из процессов, выполняющихся в фоне.

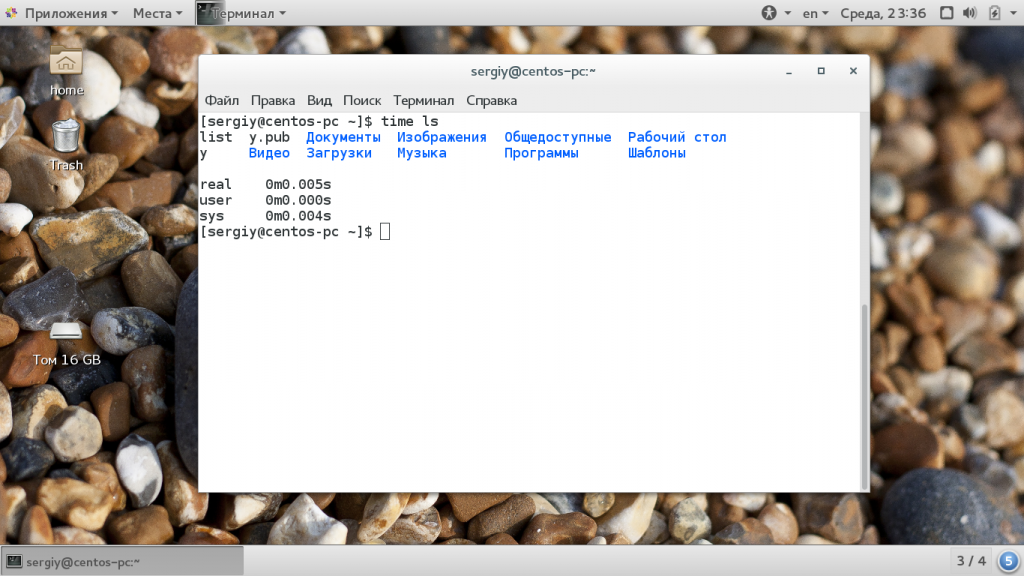
22. UMASK

Во время создания файлов, им присваиваются определенные права доступа. Право на чтение, запись и выполнение для владельца, группы и остальных пользователей. С помощью этой команды вы можете изменить правила, по которым присваиваются эти права.

23. TIME

Встроенная команда time позволяет замерять время выполнения любого скрипта или программы. Измеряется отдельно время выполнения программы и время на ввод пользователя.

time ls



24. WAIT

Наш список команд bash завершает wait. Она позволяет ожидать пока указанный процесс завершит свое выполнение. Вы можете использовать PID или идентификатор Bash для выбора нужного процесса.