**Файловые команды**  
  
**ls** – список файлов и каталогов  
**ls -la** – форматированный список со скрытыми каталогами и файлами  
**cd dir** – сменить директорию на dir  
**cd** – сменить на домашний каталог  
**pwd** – показать текущий каталог  
**mkdir dir** – создать каталог dir  
**rm file** – удалить file  
**rm -r dir** – удалить каталог dir  
**rm -f file** – удалить форсированно file  
**rm -rf dir** – удалить форсированно каталог dir  
**cp file1 file2** – скопировать file1 в file2  
**cp -r dir1 dir2** – скопировать dir1 в dir2; создаст каталог dir2, если он не существует  
**mv file1 file2** – переименовать или переместить file1 в file2. если file2 существующий каталог - переместить file1 в каталог file2  
**ln -s file link** – создать символическую ссылку link к файлу file  
**touch file** – создать file  
**cat > file** – направить стандартный ввод в file  
**less file** – вывести содержимое file  
**head file** – вывести первые 10 строк file  
**tail file** – вывести последние 10 строк file  
**tail -f file** – вывести содержимое file по мере роста, начинает с последних 10 строк  
  
**Управление процессами**  
  
**ps** – вывести ваши текущие активные процессы  
**ps aux** - вывести все процессы  
**ps aux | grep -v grep | grep -i %name** - найти процесс %name (можно использовать частичное название)  
**top** – показать все запущенные процессы  
**kill pid** – убить процесс с id pid  
**killall proc** – убить все процессы с именем proc  
**bg** – список остановленных и фоновых задач; продолжить выполнение остановленной задачи в фоне  
**fg** – выносит на передний план последние задачи  
**fg n** – вынести задачу n на передний план  
Права доступа на файлы  
**chmod octal file** – сменить права file на octal, раздельно для пользователя, группы и для всех добавлением:   
4 – чтение (r)  
2 – запись (w)  
1 – исполнение (x)  
  
Примеры:  
**chmod 777** – чтение, запись, исполнение для всех  
**chmod 755** – rwx для владельца, rx для группы и остальных.  
Дополнительные опции: **man chmod**.  
  
**SSH**  
  
**ssh user@host** – подключится к host как user  
**ssh -p port user@hos**t – подключится к host на порт port как user  
**ssh-copy-id user@host** – добавить ваш ключ на host для user чтобы включить логин без пароля и по ключам  
  
**Поиск**  
  
**grep pattern files** – искать pattern в files  
**grep -r pattern dir** – искать рекурсивно pattern в dir  
**command | grep pattern** – искать pattern в выводе command  
**locate file** – найти все файлы с именем file  
  
**Системная информация**  
  
**date** – вывести текущую дату и время  
**cal** – вывести календарь на текущий месяц  
**uptime** – показать текущий аптайм  
**w** – показать пользователей онлайн  
**whoami** – имя, под которым вы залогинены  
**finger user** – показать информацию о user  
**uname -a** – показать информацию о ядре  
**cat /proc/cpuinfo** – информация ЦПУ  
**cat /proc/meminfo** – информация о памяти  
**man command** – показать мануал для command  
**df** – показать инф. о использовании дисков  
**du** – вывести “вес” текущего каталога  
**du -sh каталог** - вывести "вес" определенного каталога в human readable виде  
**free** – использование памяти и swap  
**whereis app** – возможное расположение программы app  
**which app** – какая app будет запущена по умолчанию  
  
**Архивация**  
  
**tar cf file.tar files** – создать tar-архив с именем file.tar содержащий files  
**tar xf file.tar** – распаковать file.tar  
**tar czf file.tar.gz files** – создать архив tar с сжатием Gzip  
**tar xzf file.tar.gz** – распаковать tar с Gzip  
**tar cjf file.tar.bz2** – создать архив tar с сжатием Bzip2  
**tar xjf file.tar.bz2** – распаковать tar с Bzip2  
**gzip file** – сжать file и переименовать в file.gz  
**gzip -d file.gz** – разжать file.gz в file  
  
**Сеть**  
  
**ping host** – пропинговать host и вывести результат  
**whois domain** – получить информацию whois для domain  
**dig domain** – получить DNS информацию domain  
**dig -x host** – реверсивно искать host  
**wget file** – скачать file  
**wget -c file** – продолжить остановленную закачку  
**nmap -v -A scanme.nmap.org** - сканировать узел  
  
**Установка пакетов**  
  
Установка из исходников:  
**./configure**  
**make**  
**make install**  
**dpkg -i pkg.deb** – установить пакет (Debian)

**Привилегированный запуск приложений**

**sudo комманда** - запуск команды под именем привилегированного пользователя  
**gksu комманда** - тоже самое, разница в том что появляется графическое окно с просьбой ввести пароль  
в обоих случаях вводится пароль **вашего текущего пользователя**.

**Остальные команды**  
  
**crontab -e** – редактировать список задач cron

**Система**

**Ctrl+Alt+Delete** - перезагрузить компьютер  
**Ctrl+Alt+Backspace** - перезагрузить X-сервер

**Ctrl + Alt + Стрелка вправо (влево)** - Переключить на следующий (предыдущий) рабочий стол. Примечание: если у вас включен compiz и стоит "Куб рабочего стола", то для вращения куба держите нажатыми **Ctrl+Alt+Левая кнопка мыши** и вращайте куб движениями мыши.

**Alt+F1** - вызов главного меню (аналог нажатия кнопки Win в Windows)  
**Alt+F2** - окно запуска приложений (аналог Win+R в Windows)

**Ctrl + Alt + F1** = Переключиться на первую виртуальную консоль   
**Ctrl + Alt + F2**(F3)(F4)(F5)(F6) = Переключиться на виртуальную консоль номер ....  
**Ctrl + Alt + F7** = Переключиться в графический режим

**Консоль / Терминал**

**Ctrl+A** = Передвинуть курсор в начало строки  
**Ctrl+E** = Передвинуть курсор в конец строки  
**Tab** = показать возможные команды (Например: пишем iw и тыкаем tab, консоль нам показывает варианты = iwconfig iwevent iwgetid iwlist iwpriv iwspy) а также полезно для подстановки имени файла в команды  
**Ctrl+U** = Удалить строку  
**Ctrl+K** = Удалить строку от курсора  
**Ctrl+W** = Удалить строку до курсора  
  
**Стрелки "Вверх" "Вниз**" =  История команд пользователя  
**Ctrl+R** = Поиск по истории команд пользователя  
  
**Shift+PageUp / PageDown** = Пролистывание вывода терминала  
**Ctrl+L** = Очистить терминал **Ctrl+D**=Выход (аналог команды "exit")  
**Shift+insert** = Вставить

**Nautilus (Файловый менеджер)**

**Ctrl+H** : Показать скрытые файлы **Ctrl+T** : Создать новую вкладку **Ctrl+W:** Закрыть текущую вкладку или окно если вкладка одна **Ctrl+PgUp (PgDown) :** Переместиться на предыдущую (следующую) вкладку **Alt+1 (2, 3, 4..) :** Переместиться на вкладу 1 (2, 3, 4 ..), если существует **Alt+Home** : Перейти в домашнюю директорию  
**Alt+вверх** : Перейти в родительскую директорию (на одну выше), если такая существует  
**Alt+влево (вправо)** : Вперед (назад) по истории переходов  
  
**Alt+Enter** : свойства файла\папки  
  
**Ctrl+1 (2, 3)** : Переключение в вид значки (список, компактный)  
  
**Ctrl+Shift+N** : Создать новую директорию  
**Ctrl+M** : Создать ссылку на текущий файл (директорию)  
  
**Del** : Удалить в корзину  
**Shift+Del** : Удалить (без корзины)

**Firefox**  
  
**Ctrl+F:** Поиск текста  
**Ctrl+L:** Активировать адресную строку, выделив текущий урл.  
**Ctrl+PgUp:** Следующая вкладка  
**Ctrl+PgDwn:** Предыдущая  вкладка  
**Ctrl+Tab/Ctrl-Shift+Tab:** Тоже самое  
**Ctrl-Shift-T** :  Открыть последнюю закрытую вкладку  
  
  
**Ctrl+T** : Новая вкладка  
**Ctrl+R или F5:** перезагрузить страницу  
**Ctrl + U** :Просмотр HTML-кода страницы

**Как правильно редактировать файл /etc/fstab**

Оригинал: [How to edit and understand /etc/fstab](http://www.tuxfiles.org/)  
Автор: Nana Langstedt  
Свободный перевод: Алексей Дмитриев  
Дата перевода: 11 декабря 2007

**Что такое файл /etc/fstab и для чего он нужен**

Один из конфигурационных файлов в Линукс-системах носит имя fstab. Он содержит информацию обо всех разделах жесткого диска и других носителях информации в компьютере. Этот файл находится в каталоге /etc, вот почему полный путь к нему выглядит как /etc/fstab.

В /etc/fstab прописано, куда и как разделы винчестера и другие носители должны быть примонтированы. Если вы не имеете доступа к Windows разделу, не можете примонтировать CD, не в состоянии записать, как рядовой пользователь, файл на дискету, или испытываете трудности с CD-RW, то, скорее всего, у вас неверно сконфигурирован /etc/fstab. Редактируя этот файл, обычно решают все проблемы с монтированием.

Файл /etc/fstab это обычный текстовый файл, поэтому его можно редактировать в любом текстовом редакторе. Единственное требование - наличие прав суперпользователя. Так что, прежде чем приступать, войдите в систему как root или используйте команду su, чтобы получить права root.

**Как выглядит файл /etc/fstab**

В каждой конкретной системе файл /etc/fstab выглядит не так, как в другой, ведь разделы, устройства, и их свойства, различаются в разных системах. Но скелет структуры файла всегда одинаков. Вот пример содержимого файла /etc/fstab:

/dev/hda2 / ext2 defaults 1 1

/dev/hdb1 /home ext2 defaults 1 2

/dev/cdrom /media/cdrom auto ro,noauto,user,exec 0 0

/dev/fd0 /media/floppy auto rw,noauto,user,sync 0 0

proc /proc proc defaults 0 0

/dev/hda1 swap swap pri=42 0 0

Что же означает вся эта тарабарщина? Как легко заметить, каждая строка содержит информацию об одном разделе или устройстве. Первый столбец содержит имя устройства, второй - точку его монтирования, третий - тип файловой системы, четвертый - опции монтирования, пятый (число) - опции дампа, шестой (число) опции проверки файловой системы. Давайте подробно рассмотрим всю эту информацию.

**Первый и второй столбцы: Устройство и точка монтирования**

Первый и второй столбцы просты и понятны. Они содержат ровно то же самое, что вы пишете в командной строке, когда даете команду mount, то есть имя устройства (раздела) и точку его монтирования. Точка монтирования, указанная в /etc/fstab, является точкой монтирования по умолчанию. Эта та директория, куда будет примонтировано устройство, если вы не указали другой, когда давали команду mount.

Большинство дистрибутивов Линукса создают специальные директории для точек монтирования. Большинство дистрибутивов создают их в каталоге /mnt, некоторые (в том числе и SuSE), в каталоге /media. Как вы возможно догадались, глядя на распечатку fstab, я привела в качестве примера именно точки монтирования SuSE.

Что все это означает практически? Если я дам команду:

$ mount /dev/fd0/

...то моя дискета будет смонтирована в /media/floppy, потому что эта точка монтирования указана в /etc/fstab и поэтому используется по умолчанию. Вот если строчки /dev/fd0 в моем файле /etc/fstab не окажется, то команда mount будет сильно обескуражена, так как не будет знать, куда следует монтировать дискету.

Точки монтирования по умолчанию легко изменить, если они вас почему-либо не устраивают. Для этого нужно заменить директории в файле /etc/fstab на любые другие, реально существующие директории. Если подходящих не существует, то просто создайте их.

Некоторые разделы и устройства монтируются автоматически, в процессе загрузки системы. Взгляните на приведенный выше пример. Видите две строчки:

/dev/hda2 / ext2 defaults 1 1

/dev/hdb1 /home ext2 defaults 1 2

Они означают, что /dev/hda2 будет примонтирован в директорию /, а /dev/hdb1 - в директорию /home. Это произойдет автоматически, когда система загружается. Если этого не произойдет, то система не сможет работать, так как все программы находятся именно в директории /, и, если она не смонтирована, то и доступа к программам нет! Откуда система узнает, куда вы хотите примонтировать /dev/hda2, а куда /dev/hdb1? Посмотрев файл /etc/fstab, конечно.

**Третий столбец: Файловая система**

Третий столбец файла /etc/fstab указывает тип файловой системы раздела или устройства. Поддерживается множество различных файловых систем, но мы рассмотрим только наиболее употребительные.

**ext2 и ext3** С большой вероятностью ваши Линукс-разделы отформатированы в Ext3. Раньше стандартом была система Ext2, но в наши дни почти все дистрибутивы используют по умолчанию Ext3 или ReiserFS. Ext3 более современная система, чем Ext2 и отличается от нее своей журналируемостью. Это, в практическом плане, означает, что, если вы обесточите ваш компьютер, вместо того, чтобы выключить его по всем правилам, то вы не потеряете информацию, и не будете долго ждать при следующем включении, пока ваш компьютер проверяет файловую систему.

**reiserfs** Вполне возможно, что ваши Линукс-разделы отформатированы в ReiserFS. Подобно Ext3, ReiserFS тоже журналируемая файловая система, но она является гораздо более "продвинутой". Многие дистрибутивы Линукс (включая SuSE) используют ReiserFS по умолчанию.

**swap** Своп значит подкачка. Файловая система типа "swap" используется в разделах подкачки.

**vfat и ntfs** Windows разделы используют либо Vfat, либо NTFS. В 9х сериях (95, 98, МЕ) применялась Vfat, более известная как FAT32, в сериях NT (NT, 2000, XP) используется NTFS. В 2000 и XP можно применять и Vfat тоже. Если вы хотите иметь возможность писать в свои Windows-разделы из Линукса, советую отформатировать их в Vfat, потому что в Линуксе запись в NTFS-разделы до сих пор может причинить головную боль.

**auto** Нет-нет, это не тип файловой системы :-) Опция "auto" просто означает, что тип файловой системы определяется автоматически. Если снова взглянете на пример файла /etc/fstab, приведенный выше, то увидите, что и floppy и CD-ROM - оба - имеют вместо типа файловой системы опцию "auto". Почему? - Дело в том, что в этих устройствах могут применяться различные типы файловых систем. Одна дискета может быть отформатирована для Windows, другая - для Линукс (Ext2). Довольно разумно позволить системе самой определить тип файловой системы на носителях вроде дискет и оптических дисков.

**Четвертый столбец: Опции монтирования**

В четвертом столбце перечислены все опции, с которыми устройство или раздел будут смонтированы. По совместительству, это еще и самый сложный для заполнения столбец, но, зная некоторые самые употребительные опции, вы избежите большинства недоразумений. Я рассмотрю только наиболее широко распространенные опции, а по поводу остальных - смотрите ман-страницу mount.

**auto и noauto** Если задана опция auto, то устройство будет смонтировано автоматически во время запуска компьютера (или по команде mount -a ). Эта опция включена по умолчанию. Если вам не нужно, чтобы устройство монтировалось автоматически, вы должны прописать опцию noauto в /etc/fstab. С опцией noauto, устройство или раздел могут быть смонтированы только явно.

**exec и noexec** Если хотите запускать двоичные программы, которые находятся в данном разделе, то применяйте опцию exec, а если не хотите - то noexec. Последнее может быть полезно, если на разделе содержатся программы, которые не могут работать в вашей системе, например Windows- приложения, либо программы, нежелательные к запуску по той или иной причине.

Опция exec включена по умолчанию, и очень хорошо, что включена. Представьте, что было бы, если бы вы по ошибке использовали для монтирования своего корневого раздела опцию noexec...

**ro** Монтирует файловую систему в режиме "только чтение".

**rw** Монтирует файловую систему в режиме "чтение и запись". Применение данной опции вылечит головную боль многих новых пользователей Линукс, рвущих волосы оттого, что не могут записывать: на дискету, в Windows-разделы или куда-либо еще.

**sync and async** Эти опции определяют как осуществляется ввод/вывод в данную файловую систему: синхронно или асинхронно. Обратите внимание, что в примере опция sync применена с дискетой. Попросту говоря, когда вы копируете файл на дискету, то запись физически происходит в тот самый момент, когда дана команда копировать. Если же применяется опция async, ввод и вывод происходят неодновременно (асинхронно). В случае с дискетой это означает, что физически запись может произойти много позже команды. В этом нет ничего плохого, и во многих случаях даже предпочтительно, но может иметь неприятные побочные следствия: если вытащить дискету из дисковода, не отмонтировав ее, скопированного файла на ней может не оказаться.

По умолчанию применяется опция async. Но, может быть, стоит для дискеты прописать sync, особенно если вы привыкли вытаскивать неотмонтированные дискеты, подобно тому, как это делается в Windows.

**defaults** По умолчанию включены следующие опции: rw, suid, dev, exec, auto, nouser и async.

**Пятый и шестой столбцы: Опции dump и fsck**

Дамп - это опция резервного копирования, а fsck - опция проверки файловой системы. Я не стану слишком много о них распространяться, так как для этого может понадобиться отдельная статья, но скажу несколько слов, чтобы вы не гадали, что они могут означать.

Пятый столбец файла /etc/fstab - это опция дампа, выраженная числом. От значения этого числа зависит, будет ли создаваться резервная копия данной файловой системы. Если это ноль, программа dump проигнорирует такую файловую систему. Как видно из примера, в большинстве строк в пятом столбце нули.

В шестой колонке опция программы fsck (filesystem check- проверка файловой системы). Программа fsck использует значение чисел в этом столбце, чтобы определить, в каком порядке проверять файловые системы. Если там ноль, то файловая система вообще не будет проверяться.

**Примеры записей в файл /etc/fstab**

Для примера мы разберем два случая, которые чаще прочих расстраивают новых пользователей Линукса: дискета и CD-ROM (хотя дискеты в последнее время употребляются все реже).

/dev/fd0 /media/floppy auto rw,noauto,user,sync 0 0

Эта строка означает, что дискета монтируется по умолчанию с директорию /media/floppy и что тип файловой системы при этом определяется автоматически. Это полезно, так как тип файловой системы на дискетах может быть различным. Особое внимание обратите на опции **rw** и **user**: они обязательно должны быть прописаны, если вы хотите монтировать дискету и записывать на нее, будучи рядовым пользователем. Если это не получается, проверьте файл /etc/fstab на предмет наличия этих опций. Еще обратите внимание на опцию **sync**. С таким же успехом может быть и async, по причинам, которые мы уже обсудили.

/dev/cdrom /media/cdrom auto ro,noauto,user,exec 0 0

Снова отметьте опцию **user**, позволяющую рядовому пользователю монтировать компакт диски. Опция ro установлена потому, что нет смысла монтировать CD-ROM в режиме "чтение-запись", ведь на него все равно ничего не запишешь. А вот опция **exec** очень кстати, если надо запустить какую-либо программу с компакт-диска.

Обратите также внимание на применение опции **noauto** как с дискетой, так и с CD-ROM, это означает, что они не будут автоматически смонтированы при запуске системы. Это очень разумно для съемных носителей, которых при запуске может просто не быть в дисководах, ведь нет смысла пытаться монтировать то, чего нет.

**Перечень и краткое описание основных команд при работе операционной системе Linux и список рекомендуемой литературы**

C:\Program Files\Sun\VirtualBox\linux\lin_os_files\smb_yel.gif[Простейшие действия](file:///C:\Program%20Files\Sun\VirtualBox\linux\lin_os.html#A1)   
C:\Program Files\Sun\VirtualBox\linux\lin_os_files\smb_yel.gif[Стандартные команды и команды, дающие информацию по системе команд](file:///C:\Program%20Files\Sun\VirtualBox\linux\lin_os.html#A2)   
C:\Program Files\Sun\VirtualBox\linux\lin_os_files\smb_yel.gif[Работа с сетью](file:///C:\Program%20Files\Sun\VirtualBox\linux\lin_os.html#A3)   
C:\Program Files\Sun\VirtualBox\linux\lin_os_files\smb_yel.gif[Некоторые команды администрирования](file:///C:\Program%20Files\Sun\VirtualBox\linux\lin_os.html#A4)   
C:\Program Files\Sun\VirtualBox\linux\lin_os_files\smb_yel.gif[Контроль процессов](file:///C:\Program%20Files\Sun\VirtualBox\linux\lin_os.html#A5)   
C:\Program Files\Sun\VirtualBox\linux\lin_os_files\smb_yel.gif[Встроенные в Linux программные утилиты и языки](file:///C:\Program%20Files\Sun\VirtualBox\linux\lin_os.html#A6)   
C:\Program Files\Sun\VirtualBox\linux\lin_os_files\smb_yel.gif[Литература о linux на русском языке](file:///C:\Program%20Files\Sun\VirtualBox\linux\lin_os.html#A7)   
C:\Program Files\Sun\VirtualBox\linux\lin_os_files\smb_yel.gif[Документация про Linux в Интернете](file:///C:\Program%20Files\Sun\VirtualBox\linux\lin_os.html#A8)

При работе в ОС Linux можно использовать большое количество разнообразных команд.    
Ниже приведено очень краткое описание некоторого минимального набора, ориентированного на начинающего пользователя, не обладающего правами системного администратора. Про большинство этих команд можно узнать подробнее в интерактивном режиме, обратившись к справочной системе с помощью команды:

**man <имя изучаемой команды>**

Еще полезно:

**man -k <ключевое слово>**

В конце описания дан список литературы на русском языке, в которой содержится более подробная информация про ОС linux, особенно необходимая тем пользователям, которые хотят установить Linux на своих компьютерах и иметь права системного администратора.

**1. Простейшие действия**   
**ls**    -выдать список файлов в текущем каталоге.

**cd   [каталог]**    -сменить текущий каталог. Если имя каталога не указывается, то текущим становится домашний каталог пользователя.

**cp** <что\_копировать>  <куда\_копировать> -копировать файлы.

**mv**   <что\_перемещать>  <куда\_перемещать> -переместить или переименовать файл.

**ln -s**   <на\_что\_сделать\_ссылку>  <имя\_ссылки> -создать символическую ссылку.

**rm**   <файлы>     -удалить файлы.

**mkdir**  <каталог>    -создать новый каталог.

**rmdir**   <каталог>    -удалить пустой каталог.

**rm -r**   <файлы и/или каталоги> (рекурсивное удаление) -удалить файлы или каталоги и их подкаталоги. **ОСТОРОЖНЕЙ** с этой командой, поскольку пока у Linux нет системы полного восстановления удаленных файлов (если вы не пользуетесь специальными программами для помещения удаленных файлов в специальный каталог, что-то вроде "корзинки" в ОС Windows).

**cat**  <имя\_файла>    -вывод содержимого файла на стандартный вывод (по умолчанию - на экран).

Можно записать вводимый на экран текст с помощью следующей последовательности действий:    
**cat >** <имя\_файла>    
**.**   
**.**   
**.**   
**CTRL/d**

**more** <имя\_файла>    -просмотр содержимого длинного текстового файла по страницам.

**less**   <имя\_файла>    -просмотр содержимого текстового файла с возможностью вернуться к предыдущим страницам. Нажмите q, когда захотите выйти из программы. "less" - аналог команды DOS "more", хотя очень часто "less" бывает более удобной чем "more".

**pico**   <имя\_файла>    -редактировать текстовый файл с помощью текстового редактора pico.

**lynx**   <html\_файл или ссылка>    -просмотр файла html или WWW ссылки с помощью текстового браузера Lynx.

**tar -zxvf**   <файл>    -распаковать архив tgz или tar.gz

**find** <каталог> **-name** имя\_файла     -найти файл с именем "имя файла" и отобразить результат поиска на экране. Поиск начинается с каталога <каталог>; "имя\_файла" может содержать маску для поиска.

**pine**     -хорошая текстово - ориентированная программа для чтения электронной почты.

**mc**     -запустить программу управления файлами "Midnight Commander" (Выглядит как "Norton Commander", но по своим возможностям ближе к far).

**./Имя\_Программы**     -запустить на исполнение исполняемый файл в текущем каталоге, если текущего каталога нет в списке каталогов, указанных в переменной окружения PATH.

**xterm** (в X терминале)     -запустить простой терминал в графической оболочке X-windows. Для того чтобы выйти из него, наберите **"exit"**.

''   [К оглавлению](file:///C:\Program%20Files\Sun\VirtualBox\linux\lin_os.html#ogl)

**2. Стандартные команды и команды, дающие информацию по системе команд** **(всегда набираются в одной строке). Описание:**   
**pwd**   -вывести имя текущего каталога.

**whoami**   -вывести имя под которым Вы зарегистрированы.

**date**   -вывести дату и время.

**time**  <имя программы>    -выполнить программу и получить информацию о времени, нужном для ее выполнения. Не путайте эту команду с **date**. Например: Я могу определить выполнить команду **ls** и узнать, как много времени требуется для вывода списка файлов в каталоге, набрав последовательность: **time ls**

**who**   -определить кто из пользователей работает на машине.

**rwho -a**    -определение всех пользователей, подключившихся к вашей сети. Для выполнения этой команды требуется, чтобы был запущен процесс **rwho**.

**ruptime**   -какие машины работают в сети, и какие остановлены.

**finger**  <имя\_пользователя>   -системная информация о зарегистрированном пользователе. Попробуйте: **finger**  <ваш login-name>

**uptime**   -количество времени, прошедшего с последней перезагрузки операционной системы.    
**ps a**   -вывести список текущих процессов в Вашем сеансе работы.

**top**   -интерактивный список текущих процессов, отсортированных по использованию центрального процессора.

**uname -a**    -вывести информацию о версии операционной системы.

**free**    -вывести информацию по использованию памяти.

**df -h**   -вывести информацию о свободном и используемом месте на дисках.

**du . -bh | more**   -вывод на экран информации о размере файлов и каталогов, начиная с текущего каталога.

**set|more**   -вывести текущие значения переменных окружения. (Не для всех shell. Для csh/tcsh - printenv | more, хотя set тоже покажет полезную информацию.)

**echo $PATH**   -вывести значение переменной окружения "PATH" Команда echo может использоваться для вывода значений любых переменных окружения. Воспользуйтесь командами set или printenv для получения полного списка.

''   [К оглавлению](file:///C:\Program%20Files\Sun\VirtualBox\linux\lin_os.html#ogl)

**3. Работа с сетью**   
**ssh**    -обеспечивает безопасное вхождение в удаленный сеанс работы с другой машиной, а также позволяет выполнить заданную команду на удаленной машине без вхождения в сеанс работы:

**ssh [-l** Ваше\_имя\_пользователя\_на\_удаленной\_машине**]** <имя\_удаленной\_машины>   -вхождение в сеанс на удаленной машине. Используйте имя машины или ее IP адрес. (Вы должны быть зарегистрированы на этой удаленной машине).

        (если Ваше\_имя\_пользователя одинаково на локальной и удаленной машинах, то его можно не набирать, т.е.:    
        **ssh** <имя\_удаленной\_машины>    -позволит Вам войти в сеанс на удаленной машине)

**ssh**  <Ваше\_имя\_пользователя\_на удаленной машине@><имя\_удаленной\_машины>   <команда>    -выполнит на удаленном компьютере заданную команду и передаст Вам на экран результат ее выполнения (При установлении соединения по **ssh** Вы должны будете ввести пароль, который Вы имеете на удаленном компьютере; при этом пароль будет передан по сети в зашифрованном виде, т.е. безопасным образом.)

**scp**    -обеспечивает безопасное копирование файлов в сети:

**scp**    <имя\_файла\_на\_локальном\_компьютере>  <Ваше\_имя\_пользователя\_на удаленной машине>@<имя\_удаленной\_машины>:    -cкопирует файл с локального компьютера в Вашу корневую директорию на удаленном компьютере (наличие ":" в конце команды обязательно).

**telnet**  <имя\_удаленной\_машины>    -связаться по telnet с другой машиной. Войдите в сеанс работы после установления связи с помощью вашего пароля.

**ftp**  <имя\_удаленной\_машины>    -связаться по ftp с удаленным компьютером. Этот тип связи хорош для копирования файлов с/на удаленную машину.

**Предпочтительно не пользоваться командами telnet и ftp, а использовать только ssh и scp, поскольку они обеспечивают безопасность сетевого соединения!!!**

**hostname -i**   -показывает IP адрес компьютера, на котором Вы работаете.

''   [К оглавлению](file:///C:\Program%20Files\Sun\VirtualBox\linux\lin_os.html#ogl)

**4. Некоторые команды администрирования**   
**alias ls="ls -Fskb --color"**   -создать alias - псевдоним для того чтобы одной командой можно было запустить более сложную комбинацию команд. Поместите создание alias в файл **/etc/bashrc** если вы хотите, чтобы эти псевдонимы были доступны всем пользователям вашей системы.

Для **tcsh**    -формат определения alias другой:

**alias la 'ls -AF --color=none'**

**kapasswd**   -команда для изменения пароля для доступа к файловой системе AFS. При работе на базовом Linux-кластере ЛИТ следует пользоваться только этой командой (**а не командой passwd!**) для изменения пароля для вхождения в кластер.

**passwd**   -изменить свой пароль на каком-либо локальном компьютере.

**chmod**  <права доступа> <файл>     -изменить права доступа к файлу, владельцем которого вы являетесь.   
Есть три способа доступа к файлу:   
чтение - **read (r)**, запись - **write (w)**, исполнение - **execute (x)** и три типа пользователей:   
владелец файла - **owner (u)**, члены той же группы, что и владелец файла **(g)** и все остальные **(o)**.   
Поверить текущие права доступа можно следующим способом:

**ls -l имя\_файла**

Если файл доступен всеми способами всем пользователям, то напротив имени файла будет следующая комбинация букв: **rwxrwxrwx**   
Первые три буквы - это права доступа для владельца файла, второй триплет - права доступа для его группы, следующая тройка - права доступа для остальных. Отсутствие права доступа показывается как "-".; Например: Эта команда позволит вам установить права доступа на чтение для файла "junk" для всех (all=user+group+others):

**chmod a+r junk**

Эта команда отнимет право доступа на исполнение файла у всех кроме пользователя и группы:

**chmod o-x junk**

Для получения дополнительной информации наберите **chmod --help** или **man chmod** или почитайте любое руководство по Linux. Вы можете установить права доступа по умолчанию для создаваемых вами файлов с помощью команды **"umask"** (наберите man umask).

**chown**  <новый\_владелец> <файлы>     -изменить владельца файлов.

**chgrp**  <новая\_группа> <файлы>     -изменить группу для файла.

Вы можете использовать две последние команды, после того как сделали копию файла для кого-либо.

''   [К оглавлению](file:///C:\Program%20Files\Sun\VirtualBox\linux\lin_os.html#ogl)

**5. Контроль процессов**   
**ps axu | grep** <Ваше\_имя\_пользователя>    -отобразить все процессы, запущенные в системе от Вашего имени пользователя.

**kill <PID>**    -"убить" процесс. Для начала определите PID Вашего "убиваемого" процесса при помощи **ps**.

**killall**  <имя\_программы>     -"убить" все процессы по имени программы.

**xkill** (в терминале X window)     -"убить" процесс, на окно которого укажите курсором.

''   [К оглавлению](file:///C:\Program%20Files\Sun\VirtualBox\linux\lin_os.html#ogl)

**6. Встроенные в Linux программные утилиты и языки**   
**emacs** (в X терминале)    -редактор emacs. Очень многофукционален, но весьма сложен для неопытных пользователей.

**gcc**  <c\_исходник>    -GNU C компилятор. В сети есть очень хорошие руководства по использованию.

**g++**  <cpp\_исходник>     -GNU C++ компилятор.

**perl**    -очень мощный скриптовый язык. Чрезвычайно гибкий, но с довольно сложным синтаксисом. Очень популярен среди продвинутых пользователей.

**python**     -современный и довольно элегантный объектно-ориентированный интерпретатор. Выглядит таким же мощным и немного проще, чем perl.

**g77**     -GNU FORTRAN компилятор.

**f2c**    -перекодировщик из FORTRAN в C.

**fort77**     -компилятор FORTRAN. Выполняет f2c, а затем использует gcc или g++.

**grep**    -поиск фрагмента текста в файлах, удовлетворяющего набранной маске. Маска определяется с помощью стандартной системы обозначений, называемой "регулярные выражения".

**tr**    -translation utility (другими словами - замена букв в текстовом файле).

**gawk**    -GNU awk (используется для обработки форматированных текстовых файлов). Лучше вызывать просто **awk**.

**sed**    -утилита для обработки текстовых файлов.