ЛР-01: **Знакомство и работа с ОС Linux**

***ПРЕДВАРИТЕЛЬНО*** *(в самом начале):*

* *в каталоге* **/tmp/g0X** *(где* **0Х** *–номер группы: 05, 06, 07) создать каталог* **LR01** (**cd /**, **cd tmp**, **mkdir LR01** – см. табл. с командами ниже)
* *в каталоге* **/tmp/g0X/LR01***создать файл отчёта* **stYY.txt** *(где* **YY** *– две цифры номера по порядку студента в уч.группе)* (**touch** **stYY.txt** или **echo "Студент <ФИО>" >** **stYY.txt** и т.д. Существует множество способов создания файлов)
* *в конце занятия защитить работу ЛР-01 преподавателю.*

**1. Вход в систему**

***login***: вводим свой «логин» (login) и нажимаем клавишу <***Enter***> (или <***Return***>).

Система выдаст запрос на ввод пароля:

***Password***:

После ввода пароля вы увидите примерно такую надпись:

**[root@localhost /root]# –** такая строка называется приглашением.

Вместо **#** (признак суперпользователя – **root**) может быть символ **$** (обычный пользователь).

В приведенном примере приглашение включает в себя указание имени **пользователя (root),** **имени системы (localhost)** и **текущего каталога (/root).** Вид приглашения тоже можно изменить. Во всех последующих примерах мы будем использовать приглашение, состоящее только из имени пользователя.

**NB:** *Прежде чем предложить вам ввести первую команду, надо сказать, что в любой UNIX-системе учитывается регистр символов, т. е. различаются строчные и прописные буквы. Поэтому вводить все команды и их параметры следует именно так, как указано в примерах, учитывая регистр.*

**ВНИМАНИЕ!!!**

* Прервать выполнение команды: **<Ctrl-c>**
* Приостановить вывод на экран: **<Ctrl-s>**
* Продолжить вывод на экран: **<Ctrl-q>**

**Справочная система Linux**

Одной из первых команда, о которой нужно знать каждому пользователю любой UNIX-системы — это команда **man**. Команда **man** — это система встроенной помощи системы Linux. Вводить ее надо с параметром — именем другой команды или ключевым словом, например,

**[root]# man passwd**

В ответ вы получите описание соответствующей команды или информацию по теме, обозначенной ключевым словом. Поскольку информация обычно не помещается на одном экране, при просмотре можно пользоваться клавишами **<PageUp>** и **<PageDown>**, а также клавишей пробела. Нажатие клавиши **<Q>** в любой момент приводит к выходу из режима просмотра и возврату в режим ввода команд.

Попробуйте просмотреть информацию по рассмотренным уже командам **login** и **passwd**. Заметим, что точно также можно получить информацию по самой команде man. Введите

**[root]# man man**

Также можно использовать для подсказки две следующие команды:

**<command> --help**

**info <command>**

Система организации info-страниц является гипертекстовой. Клавиша <**n**> переместит нас на следующую страницу документа, клавиша <**p**> вернет на предыдущую, а <**u**> - передвинет на один уровень вверх в документации. Выйти из системы **info** можно, нажав <**q**>.

Наберите следующие команды:

| **Команда** | **Краткое описание** | |
| --- | --- | --- |
| **whoami** | Сообщает имя, с которым вы вошли в систему в данном сеансе работы | |
| **w или who** | Сообщает, какие пользователи работают в данный момент в системе | |
| **pwd** | Сообщает имя текущего каталога | |
| **ls** | Выдает список файлов и подкаталогов текущего каталога | |
| **ls –l**  (см. рисунок-таблицу) | Выдает список файлов и подкаталогов текущего каталога в расширенном формате  Обратите внимание на первый символ:  **-** – Отсутствие флага – обычный файл  **l** – Символическая ссылка  **d** – Директория  **b** – Блочное устройство  **c** – Символьное устройство  **p** – Канал, устройство fifo  **s** – Unix-сокет |  |
| **cd <имя\_каталога>** | Осуществляет смену текущего каталога | |
| **ps ax** | Выдает список выполняющихся процессов | |
| **pstree** | “Дерево“ процессов | |
| **cd** | Эта команда служит для перемещения по файловой системе | |
| **cd /** | Переместит вас в **корневой каталог** **/** вне зависимости от вашего прошлого положения | |
| **cd /tmp** | Перенесет вас в **/tmp** | |
| **mkdir g0X** | Создание каталога **g0X** в текущем каталоге | |
| **cd g0X** | Перенесет вас в **/tmp/g0X** | |
| **mkdir LR01** | Создание каталога **LR01**в текущем каталоге **g0X** | |
| **cd LR01** | Перенесет вас в **/tmp/g0X/LR01** | |

**2. Консоль, виртуальные терминалы и оболочка**

Итак, вы приобрели первый опыт работы в текстовом, или "консольном", режиме системы Linux. Понятия "**терминала**" и "**консоли**", которые встретятся нам еще не раз, требуется, вероятно, дополнительно пояснить.

Когда создавалась система UNIX, компьютеры были большими (мэйнфреймами), и пользователи работали на них через множество последовательных интерфейсов для подключения удаленных терминалов.

|  |  |
| --- | --- |
| ***Терминал*** — *это устройство, которое предназначено для взаимодействия пользователя с компьютером и состоит из монитора и клавиатуры*.  **терминал** = среда ввода / вывода текста  **консоль** = физический терминал  **shell** = интерпретатор командной строки (**sh**, **bash** и т.д.) | ***Консоль*** *и* ***терминал*** *тесно связаны. Первоначально они имели в виду оборудование, с помощью которого вы могли взаимодействовать с компьютером: в первые дни существования Unix это означало устройство в стиле* ***телепринтера****, напоминающее пишущую машинку, иногда называемое телетайпом, или сокращенно "****tty****".*  *Название "****терминал****" пришло с электронной точки зрения, а название "****консоль****" с точки зрения мебели.* |

Что делает команда **tty**? Она печатает имя (номер) окна терминала, который вы используете.

**TTY** означает «***телетайп***» (***teletypewriter***). В Linux есть мультиплексор псевдотелетайпа, который обрабатывает соединения от всех псевдотелетайпов окон терминала (PTS). Мультиплексор является ведущим, а **PTS** — ведомыми.

Поскольку в UNIX-системах обычно соблюдаются традиции, клавиатура и монитор персонального компьютера ведут себя так же, как ранее консоль. Преимущество такого решения состоит в том, что все старые программы, создававшиеся для администраторов UNIX, без проблем работают и на новом типе системной консоли.

Но, кроме консоли, Linux позволяет подключать к компьютеру и удаленные терминалы и, более того, обеспечивает возможность работы с несколькими виртуальными терминалами с одной консоли.

Нажмите комбинацию клавиш **Ctrl+Alt+F2**. Вы снова увидите приглашение ***login***:

Однако это не возврат к началу работы с системой — вы просто переключились в другой виртуальный терминал. Здесь вы можете зарегистрироваться под другим именем. Попробуйте войти в систему под именем только что заведенного пользователя.

После этого нажмите комбинацию клавиш **Ctrl+Alt+F1**. Вы вернетесь к первому экрану.

*По умолчанию Linux открывает при запуске 6 параллельных сеансов работы (виртуальных терминалов),* и этим иногда очень удобно пользоваться. Для переключения между виртуальными терминалами используются комбинации **Ctrl+Alt+F1** –- **Ctrl+Alt+F6** (Заметим, что при работе в текстовом режиме тот же результат можно получить, используя комбинации **Alt+F1** – **Alt+F6**, однако в графическом режиме без клавиши **Ctrl** не обойтись, так что лучше сразу привыкать к комбинациям из 3 клавиш). Если в процессе работы вы забыли, в каком терминале находитесь в данный момент, воспользуйтесь командой **tty**, которая выводит имя (номер) терминала в следующем формате: **/dev/tty2**.

Итак, сочетания клавиш **Ctrl+Alt+F1** до **Ctrl+Alt+F6** откроются полноэкранные консоли **TTY**, а **Ctrl+Alt+F7** вернёт вас в графическую среду рабочего стола. (Такое поведение может быть на более старых дистрибутивах Linux)

Сразу же скажем, что, если вы хотите **завершить сеанс работы с системой в одном из терминалов**, вы можете сделать это нажатием комбинации клавиш **Ctrl+D.** Это не приведет ни к остановке работы компьютера, ни к перезагрузке системы. Не забывайте, что Linux — многозадачная и многопользовательская система. Завершение работы одного пользователя не означает, что надо выключать компьютер. Просто завершается сеанс работы одного из пользователей, и система снова выводит в данном терминале приглашение, которое вы уже видели.

Можно **завершить сеанс работы** и введя одну из команд ***logout*** или ***exit***.

Зная теперь как открыть и закрыть сеанс работы в системе, выполните приведенные выше рекомендации, т. е. заведите себя как рядового пользователя (без суперпользовательских прав), завершите все сеансы работы, открытые от имени root, и снова войдите в систему под своим новым именем.

Теперь надо сказать несколько слов об оболочке. Оболочка, или просто ***shell*** (это слово часто не переводят, а оставляют в английском написании), — это программа, которая осуществляет все общение с пользователем. Именно оболочка воспринимает все команды, вводимые пользователем с клавиатуры, и организует исполнение этих команд. Поэтому оболочку можно назвать еще командным процессором (более привычный термин для пользователя DOS, не правда ли?).

Строго говоря, когда выше говорилось, например, "система выводит приглашение", это неправильно, поскольку приглашение выводит именно оболочка, ожидая ввода пользователем очередной команды. Каждый раз, когда очередной пользователь входит в систему, команда login запускает для него командный процессор — оболочку. Если вы логировались со второго терминала под именем пользователя jim (или под другим выбранным вами именем), то обратите теперь внимание на различие в приглашениях у пользователей root и jim.

*У пользователя* ***root*** *приглашение оканчивается символом* ***#****, а* ***у всех остальных*** *пользователей — символом* ***$****.*

Оболочку может запускать не только команда login. Вы можете просто ввести команду **bash** (именно так называется программа-оболочка в системе Linux) и тем самым запустить новый экземпляр оболочки. Выходя из него (по команде **exit** или по комбинации клавиш **Ctrl+D**) вы вернетесь к предыдущему экземпляру оболочки.

***Оболочка bash*** является не только командным процессором, но и **мощным языком программирования**. В ней имеется целый ряд встроенных (внутренних) команд и операторов, а, кроме того, в качестве команды может использоваться любая программа, хранящаяся в виде файла на диске. Список встроенных команд можно получить по команде ***help***.

Рассмотрим теперь еще одну команду, которую вам необходимо знать, поскольку все же компьютер у вас персональный (дома). А это значит, что вы и есть суперпользователь вашего домашнего компьютера. Но, как уже было сказано выше, входить в систему под именем суперпользователя не рекомендуется, поскольку любое неосторожное действие суперпользователя может привести к нежелательным последствиям. Входя под именем простого пользователя, вы, по крайней мере, не можете по неосторожности удалить или испортить системные файлы. В то же время, имеется ряд действий (например, монтирование файловых систем), выполнить которые может только суперпользователь. Не перезагружать же каждый раз компьютер! Именно в таких ситуациях выручает команда **su**. Достаточно ввести команду **su** и текущая оболочка (так и хочется сказать "система") запустит для вас новый экземпляр оболочки, в который вы попадете уже с правами пользователя root. Естественно, что для этого вам придется (в ответ на соответствующий запрос) ввести пароль этого пользователя. Закончив выполнять администраторские действия, выйдите из оболочки, и вы снова станете непривилегированным пользователем с отведенными ему полномочиями.

Если вы вошли в систему под именем **root**, то вы можете аналогичным образом запустить новый экземпляр оболочки от имени любого пользователя, пароль которого вы знаете. Но для этого надо указать имя этого пользователя в командной строке, например:

**[user]$ su jim**

*Когда мы вводим* ***su*** *без указания имени, по умолчанию подставляется имя суперпользователя root.*

Но в ОС Linux есть еще одна возможность временно переключаться в бюджет пользователя root для выполнения административных функций. Вспомните, что Linux — это многопользовательская система, в ней одновременно могут работать несколько пользователей. Поэтому в первом виртуальном терминале можно войти под именем root, а в любом другом терминале — под именем простого пользователя. Основную работу вы можете выполнять как простой пользователь, а когда потребуется выполнить административные функции, вы "*зовете системного администратора*". Для этого достаточно нажать **Ctrl+Alt+F1** — и системный администратор уже тут. По завершении операции, которую может выполнить только суперпользователь, вы немедленно должны вернуться в бюджет простого пользователя. В таком случае вы не рискуете нарушить что-либо в системе, пока еще не набрались необходимого опыта.

**3. Редактирование командной строки. История команд**

| **Клавиша** | **Описание реакции системы** |
| --- | --- |
| Стрелка вправо  или **Ctrl+F** | Перемещение вправо по командной строке в пределах уже набранной цепочки символов плюс один символ справа (место для ввода следующего символа) |
| Стрелка влево  или **Ctrl+B** | Перемещение на один символ влево |
| **Esc+F** | Перемещение на одно слово вправо |
| **Esc+B** | Перемещение на одно слово влево |
| **Home** или **Ctrl+A** | Перемещение в начало набранной цепочки символов |
| **End** или **Ctrl+E** | Перемещение в начало/конец набранной цепочки символов |
| **Del** или **Ctrl+D** | Удаление символа, на который показывает курсор |
| **Backspase** | Удаление символа в позиции, предшествующей курсору |
| **Ctrl+K** | Удалить правую часть строки, начиная с символа, на который указывает курсор |
| **Ctrl+U** | Удалить левую часть строки, включая символ, который находится слева от курсора |
| **Enter** или **Ctrl+M** | Запуск на выполнение команды, определяемой набранной цепочкой символов |
| **Ctrl+L** | Очистить экран и поместить текущую команду в верхней строке экрана |
| **Ctrl+T** | Поменять местами два символа: символ, на который показывает курсор, и символ слева от курсора, затем, курсор переместить на один символ вправо |
| **Esc+T** | Поменять местами два слова: слово, на которое указывает курсор и слово слева от первого |
| **Ctrl+K** | Вырезать часть строки от текущей позиции курсора до конца строки (вырезанная часть строки сохраняется в буфере, ее можно вставить в другое место строки) |
| **Esc+D** | Вырезать часть строки от текущей позиции курсора до конца текущего слова (если курсор указывает на пробел между словами, то вырезается все слово справа от курсора) |
| **Esc+Del** | Вырезать часть строки от текущей позиции курсора до начала текущего слова (если курсор указывает на пробел между словами, то вырезается все слово слева от курсора) |
| **Ctrl+W** | Вырезать часть строки от текущей позиции курсора до предыдущего пробела |
| **Ctrl+Y** | Вставить последний вырезанный текст в позицию курсора |
| **Esc+C** | Символ, на который указывает курсор, заменить на тот же, но заглавный, а курсор переместить на первый пробел справа от текущего слова |
| **Esc+U** | Сделать символы данного слова заглавными, начиная с символа, на который указывает курсор, а курсор установить на пробел справа от слова |
| **Esc+L** | Превратить символы, начиная с символа, на который указывает курсор, до конца данного слова в прописные (маленькие) буквы, а курсор установить на пробел справа от слова |
| **Shift+PgUp Shift+PgDown** | Эти команды позволяют просмотреть несколько страниц экранного вывода (количество зависит от размера видеопамяти); полезны в тех случаях, когда та или иная команда выводит на экран очень много информации, быстро пробегающей по экрану и как бы исчезающей для пользователя; как видите, эта информация не пропадает |
| **Ctrl+C** | Прервать выполнение запущенной команды |
| **Ctrl+D** | Выход из оболочки bash |

Список возможных команд не ограничивается только приведенными в табл. 3, но из-за ограниченности объема книги невозможно привести здесь полный перечень клавиатурных команд. Для получения дополнительной информации воспользуйтесь командой **info bash**.

Linux предоставляет широкие возможности для работы с командной строкой.

Введенные пользователем в данной сессии и в данном окне терминала команды запоминаются в специальном буфере.

С помощью клавиш ↓ и ↑ возможна навигация по списку введенных ранее команд.

Командную строку можно редактировать, перемещаясь по ней с помощью клавиш ← и →.

**Буфер команд**, команда **history**

Команда history выводит на экран содержимое буфера, в котором запоминаются команды, введенные пользователем (вообще говоря, только в текущей сессии и в данном окне терминала), например: **history**

1 13:03 pwd

2 13:03 cd book/posobie

3 13:03 ls –al

4 13:03 iconv –f KOI8R –t CP1251 linux.tex >linux1251.tex

Можно быстро повторить выполнение команды из данного списка, набрав: **!n**

где n – номер команды в списке.

Для дополнительных действий с историей вам могут понадобиться опции. Команда history linux имеет очень простой синтаксис: **$ history опции файл**

В качестве файла можно указать файл истории. По умолчанию история для текущего пользователя хранится в файле **~/.history**, но вы можете задать, например, файл другого пользователя.

А теперь рассмотрим опции:

**-c** - очистить историю;

**-d** - удалить определенную строку из истории;

**-a** - добавить новую команду в историю;

**-n** - скопировать команды из файла истории в текущий список;

**-w** - перезаписать содержимое одного файла истории в другой, заменяя повторяющиеся вхождения.

Так вы можете посмотреть только последние 10 команд: **$ history 10**

А с помощью опции **-d** удалить ненужное, например, удалить команду под номером 1007: **$**  **history -d 1007**

Если вы хотите выполнить поиск по истории **bash**, можно использовать фильтр **grep**.

Например, найдем все команды **ls**: **$** **history | grep ls**

На самом деле работать с историей еще более просто с помощью оболочки, возможно, вы уже используете многие ее функции, но о некоторых точно не знаете. Рассмотрим их:

Чтобы показать предыдущую команду просто нажмите стрелку вверх, так можно просмотреть список раньше выполненных команд.

Вы можете выполнить последнюю команду просто набрав **!!**. Также можно выполнить одну из предыдущих команд указав ее номер **!-2**

Если нужная команда последняя содержала определенное слово, например, tmp, то вы можете ее найти, использовав **!?tmp**

**Настройка истории linux**

Linux - очень настраиваемая и гибкая система, поэтому настроить здесь можно все, в том числе и историю. По умолчанию выводится только номер команды, но вы можете выводить и ее дату. Для этого нужно экспортировать переменную **HISTORYFORMAT** вместе нужным форматом:

**$ export HISTTIMEFORMAT='%F %T '**

**$ history**

Для форматирования можно использовать такие модификаторы:

**%d** – день;

**%m** – месяц;

**%y** – год;

**%T** – штамп времени;

**%F** - штамп даты.

Подробную информацию см. с помощью команд **man**, **info** и **help**.

|  |  |
| --- | --- |
| **$ history >> st01.txt**  **$ ls –l**  **$ cat st01.txt** | **st01.txt** – вместо цифры «1» введите ваш порядковый номер в группе.  **ls –l** –список файлов в текущем каталоге  **cat** – вывод файла на экран |

**4. Завершение работы системы Linux**

В любой момент времени в системе запущено несколько процессов, вы могли видеть это, когда выполняли команду **ps** («*процесс статус*»): **[root]# ps ax**

Если просто выключить питание, эти изменения не будут сохранены и пропадут, что иногда может привести даже к невозможности последующей загрузки системы. Так что надо уметь правильно завершить работу системы перед выключением компьютера. Это делается командой **shutdown**.

Команда **shutdown** может быть выполнена только пользователем root, так что вы либо должны были войти в систему под этим именем, либо должны предварительно выполнить команду **su**, чтобы приобрести соответствующие права.

Команда **shutdown** имеет следующий синтаксис:

**[root]# shutdown <options> <time> <warning-massage>**

**Замечание**: *Существует некоторая вероятность того, что, запустив команду, вы получите ответ "command not found". Это значит, что оболочка не знает, где находится файл программы. В таком случае вам необходимо ввести команду с указанием полного пути, в данном случае в виде* ***/sbin/shutdown -h****, поскольку для команды* ***shutdown*** *файл программы лежит в каталоге* ***/sbin****.*

Из опций программы **shutdown** наиболее часто используются две:

**-h** — полная остановка системы (компьютер будет выключен);

**-r** — перезагрузить систему.

Параметр time указывает время, когда должна быть выполнена команда (не обязательно выполнять ее немедленно). Время можно указать в форме задержки от текущего момента. Например, если вы хотите, чтобы система остановилась через 5 минут, вводите команду

**[root]# shutdown -r +5**

что будет означать "остановить систему через 5 минут и перезагрузиться после того, как работа будет корректно завершена''. Для вас пока наиболее актуальной формой этой команды будет, скорее всего,

**[root]# shutdown -h 0**

когда вы захотите просто выключить компьютер.

Эквивалентом команды **shutdown -h 0** является команда **halt**.

При нажатии известной **Ctrl+Alt+Del** в Linux выполняются действия, аналогичные команде: **shutdown -r 0**

Так что таким образом тоже можно выключить компьютер, только надо в момент перезагрузки отключить питание.