

Лабораторная работа №0. Основы интерфейса командной строки ОС GNU/Linux

Введение в GNU/Linux

ОС GNU/Linux, часто называемая в обиходе просто Linux — это современная многопользовательская операционная система с сетевой оконной графической системой X Window System. Все компоненты системы, включая исходные тексты, распространяются с лицензией на свободное копирование и установку для неограниченного числа пользователей.

Когда создавалась система UNIX, компьютеры были большими (мейнфреймами), и пользователи работали на них через множество удаленных терминалов. У мейнфреймов также имелся особый терминал, который предназначался для системного администратора и назывался консолью.

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

Кроме консоли, Linux позволяет подключать к компьютеру и удаленные терминалы и, более того, обеспечивает возможность работы с несколькими виртуальными терминалами с одной консоли. Убедиться в этом можно при работе с Linux на локальной машине: 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>`. Если в процессе работы вы забыли, в каком терминале находитесь в данный момент, воспользуйтесь командой `tty`, которая выводит имя терминала в следующем формате: `/dev/tty2`.

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, и 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 командой `exit`.

Введение в командную строку

Командная строка — это самый прямой способ для отправки команд машине. Если вы будете использовать командную строку GNU/Linux, то вы вскоре обнаружите, что она значительно мощнее и обладает более широкими возможностями, чем любой другой интерпретатор команд. Эта мощь доступна благодаря тому, что вы имеете доступ к тысячам утилит в консольном режиме (в противоположность графическому режиму), которые не имеют графических аналогов, с их многочисленными опциями и возможными комбинациями, достичь которых в виде кнопок или меню было бы труднее.

Если ваш дистрибутив Linux загружается в графическом режиме – зайдите в главное меню и найдите в нем несколько эмуляторов консоли, называемых Терминалы. Выберите эмулятор на свой вкус, например Konsole или xterm. В зависимости от вашего пользовательского интерфейса на панели задач также может существовать значок, четко его определяющий (см. рисунок).



Когда вы запускаете эмулятор терминала, на самом деле вы используете командный интерпретатор shell. Вы обнаружите перед собой приглашение:

```
[queen@localhost queen]$
```

Здесь подразумевается, что ваше имя пользователя — queen, а ваша машина называется localhost. Пространство после приглашения предназначено для ввода ваших команд.

Когда вы запускаете shell в первый раз, вы обычно попадаете в свой домашний подкаталог внутри каталога /home/. Чтобы вывести на экран имя каталога, в котором вы в данный момент находитесь, наберите команду pwd (которая означает вывести рабочий каталог, от англ. Print Working Directory):

```
$ pwd /home/queen
```

⁹ cd: Смена каталога

Команда cd такая же, как и в DOS, но с дополнительными возможностями. Она выполняет как раз то, что заявлено в ее аббревиатуре — сменяет рабочий каталог. Вы можете использовать «.» и «..», которые означают текущий и родительский каталоги соответственно. Запуск cd без параметров вернет вас назад в ваш домашний каталог. Запуск cd - вернет вас назад в последний посещенный вами каталог. И, наконец, вы можете указать домашний каталог пользователя peter, набрав cd ~/peter (~ сама по себе означает ваш домашний каталог). Обратите внимание, что как обычный пользователь, вы, как правило, не можете попасть в домашние каталоги других пользователей, если вы не вошли в систему как администратор (пользователь root).

⁹ Некоторые переменные окружения и команда echo

Все процессы имеют свои переменные окружения, а shell позволяет вам увидеть их непосредственно при помощи команды echo. Некоторые интересные переменные:

- HOME: эта переменная окружения содержит строку, в которой отображен путь к вашему домашнему каталогу.

- PATH: содержит список всех каталогов, в которых shell должен будет искать исполняемые файлы, когда вы набираете команду. Обратите внимание, что в отличие от DOS, shell по умолчанию не будет искать команды в текущем каталоге!
- USERNAME: эта переменная содержит ваше имя логина.
- UID: эта переменная содержит ваш ID (числовой идентификатор) пользователя.
- PS1: определяет, как будет выглядеть ваше приглашение, и зачастую является комбинацией специальных последовательностей. Для получения дополнительной информации вы можете прочитать `bash(1)` (страницу руководства), набрав в терминале `man bash`.

```
$ echo Hello
Hello
$ echo $HOME
/home/queen
$ echo $USERNAME
queen
$ echo Hello $USERNAME
Hello queen
$ cd /usr
$ pwd
/usr
$ cd $HOME
$ pwd
/home/queen
```

Как видите, shell подставляет значение переменной перед выполнением команды. В противном случае наш пример `cd $HOME` не заработал бы. По сути командный процессор сначала заменил `$HOME` на ее значение (`/home/queen`), так что строка стала `cd /home/queen`, чего мы и добивались. То же самое происходит и с примером `echo $USERNAME`.

↳ cat: Вывод на экран содержимого одного или более файлов

Нечего добавлять, эта команда делает только это: она выводит содержимое одного или более файлов на стандартный вывод, обычно на экран:

```
$ cat /etc/hosts
#
# /etc/hosts: static lookup table for host names
#

<ip-address>    <hostname.domain.org>    <hostname>
127.0.0.1        localhost.localdomain  localhost

# End of file

$ cd /etc
$ cat modules.conf shells
#
# /etc/modprobe.conf (for v2.6 kernels)
#
```

```
options i915 modeset=1
#
# /etc/shells
#

/bin/sh
/bin/dash
/bin/bash
/bin/false

# End of file
```

▸ less: Пейджер

Пейджер (от англ. page, страница) — это программа, которая позволяет пользователю просматривать большие файлы по частям: страница за страницей (точнее, экран за экраном).

Название пейджера `less` произошло от игры слов, связанной с первым пейджером из когда-либо использовавшихся в UNIX с именем `more` (`more` — больше; `less` — меньше).

Пейджер `more` ведет себя так же, как одноименная команда DOS. Однако пейджер `less` более интуитивен и удобен, и пользуются преимущественно им (хотя `more` обычно также присутствует в Linux-системах).

Вам следует использовать `less` для просмотра больших файлов, которые не умещаются на одном экране. Например:

```
less /etc/termcap
```

Для перемещения по этому файлу используйте клавиши вверх и вниз. Для выхода нажмите `Q`. Возможности `less` значительно шире: нажмите `H` для вызова справки с различными доступными опциями.

▸ ls: Вывод списка файлов

Команда `ls` (LiSt) эквивалентна команде `dir` в DOS, но сделать она может больше, гораздо больше. В действительности это связано с тем, что файлы также могут делать гораздо больше. Синтаксис команды `ls`:

```
ls [опции] [файл|каталог] [файл|каталог...]
```

Если в командной строке не указан файл или каталог, `ls` выведет список файлов в текущем каталоге. Опций довольно много, поэтому мы опишем только некоторые из них:

- `-a`: вывод списка всех файлов, включая скрытые файлы (напомним, что в UNIX скрытые файлы — это те, чьи имена начинаются с точки). Опция `-A` выводит список «почти» всех файлов, т.е. всех файлов, которые были бы показаны опцией `-a` за исключением `«.»` и `«..»`
- `-R`: рекурсивный вывод списка, т.е. все файлы и подкаталоги в каталогах, указанных в командной строке.
- `-h`: вывод для каждого файла его размера в удобном для чтения формате. Это означает, что вы увидите размеры файлов с использованием суффиксов типа `«К»`, `«М»` и `«Г»`, например, `«234 К»` и `«132 М»`. Пожалуйста, обратите внимание, что размеры

вычисляются по основанию 2, а не по основанию 10. Это означает, что «1 К» на самом деле равен 1024 байтам, а не 1000 байт.

- -l: вывод дополнительной информации о файлах: их права доступа, владельцы и группы, размеры файлов и время последнего доступа.
- -i: вывод перед каждым файлом номера inode (уникальный номер файла в файловой системе).
- -d: обработка каталогов, указанных в командной строке так, как если бы они были обычными файлами, вместо вывода списка их файлов.

Вот несколько примеров:

- «@ls -R@» рекурсивно выводит список содержимого текущего каталога.
- «@ls -is images/ ..@» выводит список с номером inode и размером в килобайтах для каждого из файлов в каталоге images/, а также в родительском каталоге по отношению к текущему.
- «@ls -l images/*.png@» выводит список всех файлов в каталоге images/, чьи имена заканчиваются на .png, включая файл .png, если такой существует.

В большинстве Linux-систем в дополнение к команде ls присутствует также команда dir, более привычная пользователям DOS.

⁹ **mkdir, touch: Создание пустых каталогов и файлов**

Команда mkdir (MaKe DIRectory – создать каталог) используется для создания каталогов. Ее синтаксис довольно прост:

```
mkdir [опции] <каталог> [каталог ...]
```

Только на одну опцию имеет смысл обратить внимание: опция -p. Она выполняет два действия:

1. создаёт родительские каталоги, если они не существовали до этого. Без этой опции mkdir просто вызовет сбой, жалуюсь на отсутствие заявленных каталогов;
2. молча завершит работу, если каталог, который вы хотели создать, уже существует. Для сравнения, если вы не указали опцию -p, mkdir возвратит сообщение об ошибке, жалуюсь на то, что каталог уже существует.

Вот несколько примеров:

- mkdir foo: создает каталог foo в текущем каталоге;
- mkdir -p images/misc docs: создает каталог misc в каталоге images. Сначала создается последний каталог, если такой не существует (-p); также создается каталог с именем docs в текущем каталоге.

Изначально команда touch предназначалась не для создания файлов, а для обновления времени последнего доступа к файлу и времени его изменения. Однако touch создаст перечисленные пустые файлы, если они не существуют. Ее синтаксис:

```
touch [опции] файл [файл...]
```

Таким образом, выполнение команды:

```
touch file1 images/file2
```

создаст в текущем каталоге пустой файл с именем file1 и пустой файл file2 в каталоге images, если ранее эти файлы не существовали.

› **rm: Удаление файлов или каталогов**

Команда **rm** (ReMove — удалить) является аналогом команд DOS **del** и **deltree**, и содержит дополнительные опции. Её синтаксис выглядит следующим образом:

```
rm [опции] <файл|каталог> [файл|каталог...]
```

Опции включают:

- **-r** или **-R**: рекурсивное удаление. Эта опция является обязательной для удаления каталога, пустого или нет. Однако для удаления пустых каталогов вы также можете воспользоваться командой **rmdir**.
- **-i**: запрос подтверждения перед каждым удалением. Часто по умолчанию **rm** по соображениям безопасности — это алиас команды **rm -i** (похожие алиасы существуют для **cp** и **mv**). Однако ваше мнение относительно полезности этих алиасов может измениться. Если вы хотите удалить их, вы можете создать пустой файл `~/.alias`, который предотвратит использование общесистемных алиасов. Или же вы можете отредактировать свой файл `~/.bashrc` для отключения некоторых общесистемных алиасов путем добавления этой строки: `unalias rm cp mv`
- **-f**, как противоположность **-i**, принудительно удалит файлы или каталоги, даже если у пользователя нет доступа для записи в файлы.

Несколько примеров:

- `rm -i images/*.jpg file1`: удаляет все файлы с именами, заканчивающимися на `.jpg`, в каталоге `images` и удаляет `file1` в текущем каталоге, запрашивая подтверждение на удаление каждого файла. Отвечайте `y` для подтверждения удаления и `n` для отмены.
- `rm -Rf images/misc/ file*`: удаляет без запроса подтверждения весь каталог `misc/` в каталоге `images/` вместе со всеми файлами в текущем каталоге, чьи имена начинаются с `file`.

Внимание! При использовании **rm** файлы удаляются безвозвратно. Способа для их восстановления не существует! (Ну, вообще-то для этого всё-таки есть несколько способов, но это нетривиальная задача и обычно используется предупреждение удаления файлов). Не раздумывайте при использовании опции **-i**, чтобы убедиться в том, что вы не удалите по ошибке что-нибудь важное.

› **mv: Перемещение или удаление файлов**

Синтаксис команды **mv** (MoVe — переместить) следующий:

```
mv [опции] <файл|каталог> [файл|каталог ...] <назначение>
```

Обратите внимание, что когда вы перемещаете несколько файлов, назначением должен быть каталог. Чтобы переименовать файл, просто переместите его в файл с новым именем.

Некоторые опции:

- -f: принудительное выполнение операции. Если перезаписывается существующий файл, предупреждения не выводятся.
- -i: противоположное действие. У пользователя спрашивается подтверждение перед перезаписью существующего файла.
- -v: подробный режим, сообщает обо всех изменениях и действиях.

Несколько примеров:

- `mv -i /tmp/pics/*.png .:` перемещает все файлы из каталога `/tmp/pics/`, чьи имена заканчиваются на `.png`, в текущий каталог (`.`), но запрашивает подтверждение перед перезаписью в нем любых файлов.
- `mv foo bar:` переименовывает файл `foo` в `bar`. Если каталог `bar` уже существовал, результатом выполнения этой команды будет перемещение файла `foo` или всего каталога (самого каталога плюс всех файлов и каталогов в нем, рекурсивно) в каталог `bar`.
- `mv -vf file* images/ trash/:` перемещает без запроса подтверждения все файлы из текущего каталога с именами, начинающимися с `file`, вместе со всем каталогом `images/` в каталог `trash/`, и показывает порядок выполнения каждой операции.

⁹ **ср: Копирование файлов и каталогов**

Команда `ср` (CoPy — копировать) является аналогом команд DOS `сору` и `хсору`, и содержит дополнительные опции. Её синтаксис выглядит следующим образом:

`ср [опции] <файл|каталог> [файл|каталог ...] <назначение>`

Вот наиболее общие опции `ср`:

- -R: рекурсивное копирование; обязательна для копирования каталога, даже если он пуст.
- -i: запрос подтверждения перед перезаписью любых файлов, которые могут быть перезаписаны.
- -f: противоположность -i, заменяет любые существующие файлы без запроса подтверждения.
- -v: подробный режим, сообщает обо всех действиях, выполняемых `ср`.

Несколько примеров:

- `ср -i /timages/* images/:` копирует все файлы из каталога `/timages/` в каталог `images/`, находящийся в текущем каталоге. Запрашивается подтверждение, если должен быть перезаписан файл.
- `ср -vR docs/ /shared/mp3s/* mystuff/:` копирует весь каталог `docs`, плюс все файлы из каталога `/shared/mp3s` в каталог `mystuff`.
- `ср foo bar:` делает копию файла `foo` в файл с именем `bar` в текущем каталоге.

⁹ **Полезные комбинации клавиш**

Существует большое количество сокращенных клавиатурных команд. Их основное преимущество состоит в том, что они экономят для вас время при наборе на клавиатуре. В этом разделе подразумевается, что вы пользуетесь стандартным `shell`’ом, по умолчанию

поставляемым вместе с большинством дистрибутивов Linux — `bash`, но эти клавиши также могут работать и в других командных процессорах.

Во-первых, клавиши со стрелками. `bash` хранит историю предыдущих команд, которые вы можете увидеть при помощи клавиш вверх и вниз. Вы можете прокрутить назад столько строк, сколько определено в переменной окружения `HISTSIZE`. Кроме того, от сеанса к сеансу история остается неизменной, поэтому вы сохраните все команды, введенные вами в предыдущем сеансе работы. Клавиши со стрелками влево и вправо перемещают курсор влево и вправо в текущей строке, позволяя вам редактировать свои команды. Но при редактировании вы можете не только просто перемещаться на один символ за раз: `Ctrl-A` и `Ctrl-E`, например, переместят вас в начало и конец текущей строки. Клавиши `Backspace` и `Del` работают так, как и должны. `Backspace` и `Ctrl-H` эквивалентны. `Del` и `Ctrl-D` также могут быть использованы поочередно. `Ctrl-K` удалит все от текущей позиции курсора до конца строки, а `Ctrl-W` удалит слово перед курсором (так же как и `Alt-Backspace`).

Нажатие `Ctrl-D` в пустой строке позволит вам закрыть текущий сеанс, что намного проще, чем необходимость вводить `exit`. `Ctrl-C` прервет выполняющуюся в данный момент команду, если только вы не в процессе редактирования своей командной строки, в этом случае редактирование будет отменено и вы вернетесь назад к исходному приглашению. `Ctrl-L` очищает экран. `Ctrl-Z` временно останавливает выполнение задачи, т.е. она приостанавливается; затем можно перевести ее в фоновый режим, введя команду `bg`. Например:

```
$ less MyDocument.txt
```

С этого момента вы больше не сможете воспользоваться своим командным процессором, т.к. приоритетной задачей стал процесс `less`. Чтобы сделать процесс фоновым и восстановить работу своей консоли, просто нажмите `Ctrl-Z` и введите `bg`. Команда `bg` происходит от слова `background` — фон. Фоновый процесс продолжает выполняться параллельно интерпретатору команд и другим программам, на занимая при этом экран консоли. Процесс, помещенный в фоновый режим, можно вернуть обратно в режим взаимодействия с пользователем командой `fg` (от слова `foreground` — передний план).

И в заключение, имеются комбинации `Ctrl-S` и `Ctrl-Q`, которые используются для приостановки и возобновления вывода на экран. Они нечасто используются, но вы можете по ошибке ввести `Ctrl-S` (в конце концов, `S` и `D` на клавиатуре находятся близко друг от друга). Поэтому, если вы попадете в ситуацию, когда вы что-то набираете на клавиатуре, но в Терминале никакие символы не появляются, попробуйте нажать `Ctrl-Q`. Обратите внимание, что все символы, введенные вами между нежелательной комбинацией `Ctrl-S` и `Ctrl-Q` будут одновременно выведены на экран.

⁹ Выполните самостоятельно

1. Войдите в систему, введя свои логин и пароль. Выведите информацию о текущем каталоге и выполните действия с любой командой на выбор. Выйдите из системы, набрав `exit` (или нажав `Ctrl-D`).
2. Повторите вход.
3. Последовательно изучите основные консольные команды, описанные в п.2.1-2.9. Создайте в своем домашнем каталоге папку `Lab_1` и в папке `Lab_1` файл

addition.txt, с помощью стандартного текстового редактора (например, редактора `mcedit`) и внесите свои имя и фамилию.

4. Обратите внимание на файловую систему ОС Linux. Где хранятся папки каждого из пользователей? Как обратиться к другому носителю информации, например, CD-диску или дискете? (найдите каталог `/mnt`).
5. Выясните содержимое любой папки выбор и просмотрите содержимое любого выбранного файла соответствующей командой. Например, для просмотра содержимого файла используется команда `cat`.