

ЛАБОРАТОРНАЯ РАБОТА №2

Тема: Структура файловой системой ОС Linux. Работа с файлами и каталогами. Устройства в Linux. Типы файлов.

Цель: Ознакомиться с файловой системой ОС Linux, ее структурой каталогов, основными командами работы с файловой подсистемой. Овладеть навыками навигации по каталогам операционной системы, а также работы с устройствами.

Содержание работы:

1. Организация файловой системы. Дерево каталогов.
2. Типы файлов и устройств в Linux
3. Ознакомиться с основными командами для навигации по файловой системе (pwd, cd, dir, ls).
4. Ознакомиться с основными командами для работы с файлами (каталогами) (rm, mv, cp, mkdir; cat, less, more).
5. Мягкие (символические) и жесткие ссылки (ln).
6. Организация поиска (find, locate, grep).
7. Стандартный ввод/вывод и их перенаправление.
8. Работа с дисками (mount, umount).
9. Программа Midnight Commander.

Дополнительная информация:

Структура каталогов. Понятие каталога позволяет систематизировать все объекты, размещенные на носителе данных (например, на диске). В большинстве современных файловых систем используется иерархическая модель организации данных: существует один каталог, объединяющий все данные в файловой системе – это "корень" всей файловой системы, **корневой каталог**. Корневой каталог может содержать любые объекты файловой системы, и в частности, подкаталоги (каталоги первого уровня вложенности). Таким образом, все, что записано на диске - файлы, каталоги и специальные файлы - обязательно "принадлежит"

корневому каталогу: либо непосредственно (содержится в нем), либо на некотором уровне вложенности.

Структуру файловой системы можно представить наглядно в виде дерева (см. рис. 1), "корнем" которого является корневой каталог, а в вершинах расположены все остальные каталоги.

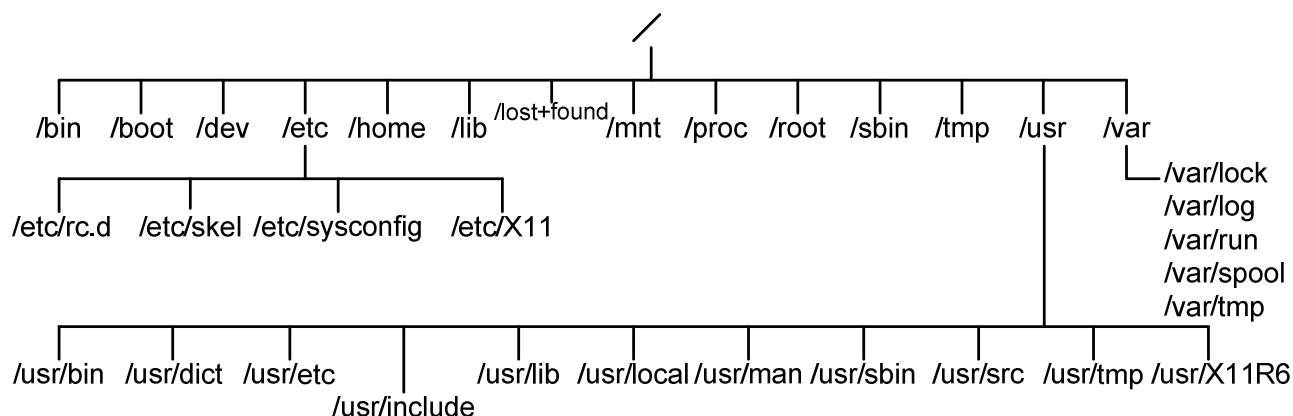


Рисунок 1. – Файловая структура ОС Linux

Назначение каталогов приведено в таблице ниже

Каталог	Подкаталоги	Назначение
/bin		Этот каталог содержит в основном готовые к исполнению программы, большинство из которых необходимы во время старта системы (или в однопользовательском системном режиме, используемом для отладки). Здесь хранится значительное количество общеупотребительных команд Linux
/boot		Содержит основные постоянные файлы для загрузки системы, в частности загружаемое ядро. Файлы из этого каталога нужны только во время загрузки системы
/dev		Каталог специальных файлов или файлов устройств. О них мы поговорим чуть подробнее в одном из следующих подразделов. Можете также заглянуть в man mknod(1)
/etc		Этот каталог и его подкаталоги содержат большинство данных, необходимых для начальной загрузки системы и основные конфигурационные файлы. В /etc находятся, например, файл inittab, определяющий загружаемую конфигурацию, и файл паролей пользователей passwd. Часть конфигурационных файлов может находиться и в /usr/etc. Каталог /etc не должен содержать двоичных файлов (их следует перенести в /bin или /sbin). Ниже приводится назначение основных (но далеко не всех!) подкаталогов каталога /etc
	/etc/rc.d	Этот подкаталог содержит файлы, которые используются в процессе начальной загрузки системы. Подробнее о них и вообще о процессе загрузки будет рассказано в разд. 8.2
	/etc/skel	Когда создается новый пользователь и account для него, то файлы из этого каталога копируются во вновь созданный домашний каталог пользователя
	/etc/sysconfig	Каталог, содержащий некоторые (но не все) конфигурационные файлы системы
	/etc/X11	Каталог для конфигурационных файлов подсистемы X11 (например, XF86Config)
/home		Обычно в этом каталоге находятся домашние каталоги пользователей
/lib		Этот каталог содержит разделяемые библиотеки функций, необходимых компилятору языка C и модули (драйверы устройств). Даже если в системе не установлен компилятор языка C, разделяемые библиотеки необходимы, поскольку они используются многими прикладными программами. Они загружаются в память по мере необходимости выполнения каких-то функций, что позволяет уменьшить объем кода программ — в противном случае один и тот же код многократно

		повторялся бы в различных программах
/lost+found		Этот каталог используется при восстановлении файловой системы командой fsck. Если fsck обнаруживает файл, родительский каталог которого определить невозможно, она помещает такой файл в каталог /lost+found. Поскольку родительский каталог потерян, то таким файлам присваиваются имена, совпадающие с номерами их индексных дескрипторов
/mnt		Это точка монтирования для временно монтируемых файловых систем. Если на компьютере запускается поочередно Linux и MS DOS, то этот каталог обычно используется, чтобы монтировать файловую систему MS DOS. Если вы имеете привычку монтировать несколько дополнительных носителей, например, дискеты, CD-ROM, дополнительный жесткий диск и т. д., то можно создать в нем соответственно дополнительные подкаталоги для каждого носителя
/proc		Это точка монтирования для файловой системы proc, которая обеспечивает информацию о выполняющихся процессах, ядре, оборудовании вычислительной установки и т. д. Это псевдо-файловая система, подробности о которой можно узнать по команде man 5 proc. Специальные файлы из этого каталога используются для получения и передачи данных ядру
/root		Это домашний каталог суперпользователя. Обратите внимание на то, что он расположен не там, где располагаются личные каталоги остальных пользователей (в /home)
/sbin		Подобно каталогу /bin содержит в основном исполняемые файлы — программы и утилиты ОС, используемые в процессе загрузки и запускаемые системным администратором. В стандарте FHS говорится, что в этот каталог надо помещать те исполняемые файлы, которые используются после успешного подключения файловой системы /usr. Минимальное содержимое этого каталога включает программы clock, getty, init, update, mkswap, swapon, swapoff, halt, reboot, shutdown, fdisk, fsck.*, mkfs.*, lilo, arp, ifconfig, route
/tmp		Каталог для временных файлов. В любой момент суперпользователь может удалить файлы из этого каталога без большого ущерба для остальных пользователей. Однако, не стоит удалять файлы из этого каталога, если вам не стало ясно, что конкретный файл или группа файлов мешают продолжению продуктивной работе на машине. Система сама периодически очищает этот каталог, поэтому не следует хранить тут файлы, которые вам могут понадобиться в дальнейшем
/usr		Этот каталог огромен и его структура в основном повторяет структуру корневого каталога. В его подкаталогах находятся все основные приложения. В соответствии со стандартом FHS рекомендуется выделять для этого каталога отдельный раздел диска или вообще располагать его на сетевом диске, общем для всех компьютеров в сети. Такой раздел или диск монтируют только для чтения и располагают в нем общие конфигурационные и исполняемые файлы, документацию, системные утилиты и библиотеки, а также включаемые файлы (файлы типа include)
	/usr/bin	Готовые к исполнению программы — утилиты и приложения, которые часто вызывают обычные пользователи.
	/usr/dict	Этот каталог содержит файлы со словарным запасом для программ проверки корректности написания слов.
	/usr/etc	Здесь содержатся конфигурационные файлы для группы машин. Однако, команды и программы должны смотреть в каталог /etc, в котором должны быть ссылки к файлам в каталоге /usr/etc
	/usr/include	Этот каталог содержит исходный код стандартных библиотек языка C, подставляемый в программы директивой препроцессора include. Поэтому пользователю надо иметь, по крайней мере, право на чтение из этого каталога. Ни в коем случае не следует модифицировать файлы в этом каталоге, потому что они тщательно отлажены разработчиком системы (разве что вы знаете систему лучше разработчика)
	/usr/lib	В данном каталоге содержится объектные библиотеки подпрограмм, динамические библиотеки, некоторые готовые к исполнению программы, которые не вызываются непосредственно. Сложные программные системы могут иметь свои подкаталоги в этом каталоге.
	/usr/local	Обычно здесь помещают программы и подкаталоги, которые являются локальными (уникальными) для данной машины. /usr/local/bin — Обычно здесь помещают готовые к исполнению программы, которые являются локальными (уникальными) для данной машины. /usr/local/doc — Здесь располагается документация ко всем установленным на

		<p>Вашем компьютере пакетам прикладного ПО.</p> <p>/usr/local/etc — Конфигурационные файлы для локально установленных программ.</p> <p>/usr/local/lib — Библиотеки и файлы для локально установленных программ и систем.</p> <p>/usr/local/info — Страницы описаний, которые просматриваются посредством программы info, для локально установленных программ.</p> <p>/usr/local/man — Страницы описаний, которые просматриваются посредством программы man, для локально установленных программ.</p> <p>/usr/local/sbin — Локальные программы системного администратора.</p> <p>/usr/local/src — Исходные тексты программ, установленных на данной машине</p>
	/usr/man	Страницы интерактивного руководства man в исходном формате (не подготовленные для просмотра). /usr/man/<locale>/man[1-9] — Эти каталоги содержат страницы руководств на различных языках (в зависимости от значения locale). Системы, которые используют один язык и один кодовый набор, могут не использовать подстроку <locale>
	/usr/sbin	Этот каталог содержит готовые к исполнению программы для системного администрирования, которые не используются во время загрузки
	/usr/src	Исходные тексты для различных частей Linux. /usr/src/linux — исходные тексты для ядра Linux
	/usr/tmp	Еще одно место для хранения временных файлов. Обычно это символическая ссылка на каталог /var/tmp
	/usr/X11R6	Файлы относящиеся к системе X-Window
/var		Этот каталог содержит файлы, в которых сохраняются различные переменные данные, определяющие конфигурацию некоторых программ при следующем запуске или временно сохраняемую информацию, которая будет использоваться позже в ходе текущего сеанса. Объем данных в этом каталоге может сильно изменяться, поскольку он содержит, например, файлы протоколов (логи), файлы спулинга и блокировки (locking), временные файлы и т. д.
	/var/lock	Здесь содержатся управляющие файлы системы, которые используются для резервирования использования тех или иных ресурсов системы
	/var/log	Различные файлы протоколов (логи)
	/var/run	Переменные файлы времени выполнения различных программ. Они содержат идентификаторы процессов (PIDs) и записывают текущую информацию (utmp). Файлы в этом каталоге обычно очищаются во время загрузки системы
	/var/spool	<p>Файлы различных программ, поставленные в очередь на обслуживание.</p> <p>/var/spool/at — Файлы заданий, запущенных посредством команды at.</p> <p>/var/spool/cron — Файлы системы cron.</p> <p>/var/spool/lpd — Файлы, ожидающие вывода на печать.</p> <p>/var/spool/mail — Пользовательские почтовые ящики.</p>
	/var/tmp	Временные файлы

Основные команда навигации по файловой системе (pwd, cd, dir, ls).

Каждая выполняемая программа "работает" в строго определенном каталоге файловой системы. Такой каталог называется текущим каталогом. В зависимости от текущего каталога поведение программы может меняться: зачастую программа будет по умолчанию работать с файлами, расположенными именно в текущем каталоге – до них она "дотянется" в первую очередь. Текущий каталог есть у любой программы, в том числе и у командной оболочки (shell) пользователя. Поскольку взаимодействие пользователя с системой обязательно опосредовано командной оболочкой, можно говорить о том, что пользователь "находится" в том

каталоге, который в данный момент является текущим каталогом его командной оболочки.

Все команды, отдаваемые пользователем при помощи shell, наследуют текущий каталог shell, т. е. "работают" в том же каталоге. По этой причине пользователю важно знать текущий каталог shell. Для этого служит утилита `pwd`:

```
[globus@fedora ~]$ pwd  
/home/globus
```

Команда `pwd` (`print working directory`) возвращает полный путь текущего каталога командной оболочки – естественно, именно той командной оболочки, при помощи которой была выполнена команда `pwd`. В данном случае globus узнал, что в этот момент (на данной виртуальной консоли) текущим является каталог `"/home/globus"`.

Относительный путь(relative path) – путь к объекту файловой системы, не начинающийся в корневом каталоге. Для каждого процесса Linux определен текущий каталог, с которого система начинает относительный путь при выполнении файловых операций.

Относительный путь строится точно так же, как и полный – перечислением через `"/"` всех названий каталогов, встретившихся при движении к искомому каталогу или файлу. Между полным и относительным путем есть только одно существенное различие: относительный путь начинается **от текущего каталога**, в то время как полный путь всегда начинается **от корневого каталога**. Относительный путь любого файла или каталога в файловой системе может иметь любую конфигурацию: чтобы добраться до искомого файла, можно двигаться как по направлению к корневому каталогу, так и от него. Linux различает полный и относительный пути очень просто: если имя объекта начинается на `"/"` - это полный путь, в любом другом случае – относительный.

Информация о каталоге (`ls` и `dir`). Чтобы иметь возможность ориентироваться в файловой системе, нужно знать, что содержится в каждом каталоге. Запомнить всю структуру файловой системы невозможно и не нужно: в

любой момент можно просмотреть содержимое любого каталога при помощи утилиты `ls` (сокращение от англ. "list" - "список"):

```
[root@fedora globus]# ls
BUILD build.log
```

Применение данной команды без параметров позволяет отобразить содержимое текущего каталога, как видим, у пользователя `globus` в домашнем каталоге находится всего два объекта `BUILD` и `build.log`. Утилита `ls` принимает один параметр: имя каталога, содержимое которого нужно вывести. Имя может быть задано любым доступным способом: в виде полного или относительного пути. Например, чтобы получить список файлов в своем домашнем каталоге, пользователь `globus` мог бы использовать команды "`ls /home/globus`" или находясь в домашнем каталоге "`ls .`" или просто `ls` – результат был бы аналогичным. *Примечание: точка (.) означает текущий каталог, а (..) – родительский каталог.* Ссылки на текущий и на родительский каталог обязательно присутствуют в **каждом каталоге** в Linux. Даже если каталог пуст, т. е. не содержит ни одного файла или подкаталога, команда "`ls -a`" выведет список из двух имен: "." и "..".

Кроме параметра, утилита `ls` "понимает" множество ключей, которые нужны главным образом для того, чтобы выводить дополнительную информацию о файлах в каталоге или выводить список файлов выборочно. Чтобы узнать обо всех возможностях `ls`, нужно, прочесть руководство по этой утилите ("`man ls`"). Основные ключи:

`ls -a` – Выводить все файлы, включая скрытые:

```
[root@fedora globus]# ls -a
.      .bash_logout  .bashrc  build.log  .joe_state  .viminfo
..     .bash_profile .bashrc~ .ccache    .lessht
.bash_history .bash_profile~ BUILD     .globus    .rnd
```

Как мы видим, в каталоге пользователя содержится достаточно много скрытых файлов.

`ls -F` – используется для того, чтобы отличать файл от каталога

```
[root@fedora globus]# ls -F
```

BUILD/ build.log

`ls -l` – расширенный формат вывода:

```
[root@fedora globus]# ls -l
```

итого 12

```
drwxrwxr-x 3 globus globus 4096 Июл 14 15:05 BUILD
```

```
-rw-rw-r-- 1 globus globus 7610 Июл 14 15:05 build.log
```

Как было сказано ранее, ключи можно объединять, например:

```
[root@fedora globus]# ls -laF
```

итого 80

```
drwx----- 5      globus globus  4096   Авг 16 14:29  ./
drwxr-xr-x  4      root  root    4096   Июл 14 15:28  ../
-rw-----  1      globus globus  9669   Авг 14 01:07  .bash_history
-rw-r--r--  1      globus globus   33     Июл  7 22:45  .bash_logout
-rw-r--r--  1      globus globus  279    Июл 14 17:30  .bash_profile
-rw-r--r--  1      globus globus  279    Июл 14 17:30  .bash_profile~
-rw-r--r--  1      globus globus  142    Авг 16 14:29  .bashrc
-rw-r--r--  1      root  root     141    Авг 16 14:29  .bashrc~
drwxrwxr-x  3      globus globus  4096   Июл 14 15:05  BUILD/
-rw-rw-r--  1      globus globus  7610   Июл 14 15:05  build.log
drwxrwxr-x 18      globus globus  4096   Июл  9 12:47  .ccache/
drwxrwxr-x  5      globus globus  4096   Июл 14 17:37  .globus/
-rw-----  1      globus globus  2822   Июл 17 10:52  .joe_state
-rw-----  1      globus globus   35     Авг 13 14:43  .lessht
-rw-----  1      globus globus  1024   Июл 14 16:21  .rnd
-rw-----  1      globus globus  1012   Июл 14 16:23  .viminfo
```

За дополнительной информацией по команде `ls` смотреть справку.

Более детальную информацию рассмотрим дальше. Столбцы: тип файла, права доступа к файлу, количество ссылок на файл, имя владельца, имя группы, размер файла (в байтах), временной штамп и имя файла.

Типы файлов могут принимать следующие значения: - для обычного файла, `d` для каталога, `b` для блочного устройства, `c` для символьного устройства, `l` для символической ссылки, `p` для FIFO и `s` для гнезда (socket).

Пользователь может работать с файлами не только в своем домашнем каталоге, но и в других каталогах. В этом случае будет удобно **сменить текущий каталог**, т. е. "переместиться" в другую точку файловой системы. Для смены текущего каталога командной оболочки используется команда `cd` (от англ.

"change directory" - "сменить каталог"). Команда `cd` принимает один параметр: имя каталога, в который нужно переместиться – сделать текущим. Как обычно, в качестве имени каталога можно использовать полный или относительный путь:

```
[root@fedora globus]# cd /home/globus/
[root@fedora globus]# cd ../globus/
```

Вторая команда перемещает вас обратно в текущий каталог, так как вы подымаетесь на уровень вверх, а затем обратно в текущую папку.

Необходимость вернуться в домашний каталог из произвольной точки файловой системы возникает довольно часто, поэтому командная оболочка поддерживает обозначение домашнего каталога при помощи символа `"~"`. Поэтому чтобы перейти в домашний каталог из любого другого, достаточно выполнить команду `"cd ~"`. При исполнении команды символ `"~"` будет заменен командной оболочкой на полный путь к домашнему каталогу пользователя.

Дополнительная информация в справочной системе.

Команды просмотра файлов `more`, `less`. Название `less` произошло от игры слов, связанной с первым пейджером из когда-либо использовавшихся в UNIX® с именем `more` (`more` - больше; `less` - меньше). Пейджер (`page` - страница) – это программа, которая позволяет пользователю просматривать большие файлы по частям, страница за страницей (точнее, экран за экраном). Причина, по которой мы рассматриваем `less`, а не `more`, заключается в том, что `less` более интуитивна. Вам следует использовать `less` для просмотра больших файлов, которые не уместятся на одном экране. Например:

```
[root@fedora ~]# less .bashrc
# .bashrc
# User specific aliases and functions
alias rm='rm -i'
alias cp='cp -i'
alias mv='mv -i'
alias lm='ls -laF'

# Source global definitions
if [ -f /etc/bashrc ]; then
    . /etc/bashrc
fi
```


(END)

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.

Программа Midnight Commander. Хотя для управления файловой системой и вообще для работы с файлами можно использовать такие команды операционной системы, как pwd, ls, cd, mv, mkdir, rmdir, cp, rm, cat, more и т. д., гораздо удобнее делать большую часть работы по обслуживанию файловой системы с помощью программы Midnight Commander, которая наглядно представляет все выполняемые действия, облегчая тем самым выполнение указанных операций.

Midnight Commander (или просто mc) - это программа, которая позволяет просмотреть структуру каталогов и выполнить основные операции по управлению файловой системой. Другими словами, это файловый менеджер. Если вы имеете опыт работы с Norton Commander (nc) в MS-DOS или с FAR в Windows, то вы легко сможете работать и с mc, поскольку даже основные комбинации "горячих клавиш" у них совпадают.

Задание на лабораторную работу.

- 1) Изучить команду **tree**. Освоить технику применения шаблона, например, вывести все файлы, которые содержат символ **c**, или файлы, которые содержат определенную последовательность символов. Вывести подкаталоги корневого каталога до второго уровня вложенности включительно.
- 2) С помощью какой команды можно определить тип файла (например, текстовый или бинарный)? Привести пример.
- 3) Овладеть навыками навигации по файловой системе при помощи относительных и абсолютных путей. Как можно вернуться в домашний каталог из любого места в файловой системе?
- 4) Ознакомиться с различными ключами команды **ls**. Привести примеры распечатки каталогов с использованием различных ключей. Объяснить выведенную на терминал информацию с помощью ключей **-l** и **-a**.
- 5) Выполнить следующую последовательность операций:

- создать в домашнем каталоге подкаталог;

- в этом подкаталоге создать файл, содержащий информацию о каталогах, находящихся в корневом каталоге (используя операции перенаправления ввода-вывода);
- просмотреть созданный файл;
- скопировать созданный файл в домашний каталог используя относительную и абсолютную адресацию.
- удалить созданный ранее подкаталог с файлом с запросом на удаление;
- удалить файл, скопированный в домашний каталог.

6) Выполнить следующую последовательность операций:

- создать в домашнем каталоге подкаталог **test**;
- скопировать в этот каталог файл **.bash_history** при этом сменив его имя на **labwork2**;
- создать жесткую и мягкую ссылку на файл **labwork2** в подкаталоге **test**;
- как определить мягкую и жесткую ссылку, что означают эти понятия;
- измените данные, открыв символическую ссылку. Какие изменения произойдут и почему
- переименуйте файл жесткой ссылки в файл **hard_lnk_labwork2**;
- переименуйте файл мягкой ссылки в файл **symlnk_labwork2**;
- затем удалите файл **labwork2**. Какие произошли изменения и почему?

- 7) Используя утилиту **locate** найти все файлы, в которых встречается последовательность **squid** и **traceroute**.
- 8) Определить, какие разделы смонтированы в системе, а также типы этих разделов.
- 9) Подсчитать количество строк содержащих заданную последовательность символов в заданном файле.
- 10) Используя команду **find** найти все файлы в каталоге **/etc** содержащие последовательность символов **host**.
- 11) Вывести все объекты каталога **/etc**, содержащие последовательность символов **ss**. Как можно продублировать аналогичную команду используя связку с командой **grep**.

- 12) Организовать постраничную распечатку содержимого каталога */etc*.
Подсказка: необходимо использовать операции перенаправления потоков.
- 13) Какие бывают типы устройств и как определить тип устройства?
Привести примеры.
- 14) Как определить тип файла в системе, какие типы файлов бывают?
- 15) * Вывести первые 5 файлов каталога, к которым был недавно осуществлен доступ в каталоге */etc*