

Практическое занятие 1. Начальные сведения о работе с OCLinux.

Работа с файлами в UNIX/Linux

Цель занятия: знакомство с командной строкой UNIX, командами по работе с файловой системой и командами обработки текстовых файлов.

Введение в bash

Shell

Если вы используете линукс, то знаете что после логина вас приветствует приглашение командного интерпретатора. Например такое:

```
\$
```

Если после логина загружается графическая оболочка, то чтобы добраться до командного интерпретатора нужно запустить эмулятор терминала (gnome-terminal, xfce4-terminal, konsole, xterm, rxvt...).

Приглашение командного интерпретатора на вашем компьютере может отличаться от того что показано в примере. Оно может содержать имя пользователя, имя компьютера и название текущей рабочей директории. Но несмотря на все эти различия, программа которая печатает это приглашение называется "**shell**" (оболочка), и скорее всего в роли вашей командной оболочки выступает программа которая называется **bash**.

У вас запущен bash?

Проверить запущен ли bash можно следующей командой:

```
\$ echo \$SHELL /bin/bash
```

Если в результате выполнения этой команды вы получили ошибку или её вывод отличается от того что в примере, то возможно в вашей системе в качестве командной оболочки используется не bash. Несмотря на это, большая часть материала будет актуальна, но все же рекомендуем вам переключиться на bash. Сделать это можно (если bash установлен в системе) командой:

```
\$ bash
```

Что такое bash

Bash (акроним от "**B**ourne-again **S**hell") это стандартный интерпретатор команд на большинстве линукс систем. В его обязанности входит обработка и исполнение команд с помощью которых пользователь управляет компьютером. После того как вы завершили работу, можно завершить процесс командного интерпретатора. После нажатия клавиш **Ctrl-D**, команд **exit** или **logout** процесс командного интерпретатора будет завершен и на экране снова появится приглашение ввести имя пользователя и пароль.

Работа с командной строкой Linux.

Описание команд разбито на несколько разделов — по функциональному назначению команд.

Команды по работе с файловой системой

pwd

Команда **pwd** (print working directory — печать рабочего каталога) позволяет узнать имя текущего каталога:

```
[user@localhost ~]$ pwd
/home/user
```

Т.е. в настоящее время Вы находитесь в корневом каталоге.

Если вы в процессе путешествия по файловой системе вы «потерялись» и не знаете, где находитесь, просто наберите эту команду, которая выведет на экран полное имя текущего каталога, начиная от корневого.

Если вместо имени текущего каталога на экран будет выдано диагностическое сообщение типа: Cannot open.. («Не может открыть..») и Read error in.. («Ошибка считывания в..»), то значит произошло нарушение файловой системы, возможно у Вас не хватает прав на чтение этого каталога.

cd

Команда **cd** используется для смены текущего каталога, т.е. для перехода в другой каталог и является встроенной командой оболочки (аналог соответствующей команды для DOS-систем). В качестве аргумента этой команды укажите имя каталога, в который Вы хотите перейти. Например:

Например, **cd /etc** — перейти в каталог /etc, т.е. сделать его текущим.

Если Вы хотите перейти в подкаталог, то «/» не указывается. Допустим Вы находитесь в каталоге /usr и вам надо перейти в каталог local, который является подкаталогом этого каталога. В этом случае вы можете выполнить следующие действия:

```
[user@localhost usr]$ pwd
/usr
[user@localhost usr]$ cd local
[user@localhost local]$ pwd
/usr/local
```

Таким образом каталог /usr/local стал текущим.

Если ввести команду **cd** без аргументов, то вы вернетесь в свой домашний (начальный) каталог. При этом используется системная переменная окружения HOME.

```
[user@localhost usr]$ cd
[user@localhost ~]$ pwd
/home/user
```

Приведем еще несколько примеров использования этой команды:

- **cd ..** — вернуться в родительский каталог (подняться на один каталог вверх по дереву);
- **cd ../..** — подняться на два каталога вверх по дереву;
- **cd /** — перейти в корневой каталог;

- **cd ../local** — вернуться в родительский каталог и перейти в его подкаталог local.

ls

Команды **ls** выдает различную информацию о файлах и каталогах (аналог команды **DIR** для DOS-систем). Как и большинство команд, **ls** имеет возможные аргументы и параметры (флаги), которые позволяют изменить её действие.

Перейдите с помощью команды **cd /** в корневой каталог и посмотрите его содержимое. Если опции не указаны команда выводит отсортированные по алфавиту имена файлов и каталогов.

```
[user@localhost ~]$ cd /  
[user@localhost /]$ ls  
bin dev home lib mnt proc sbin sys usr  
boot etc image media opt root srv tmp var
```

К сожалению, пока не очень понятно, что является каталогом, а что файлом. Помочь в этом может следующая команда, которая выдает информацию в расширенном формате, который содержит большой объем информации о каждом файле (права доступа, размер, дата последней модификации и т.д.):

```
[user@localhost /]$ ls -l  
total 6  
drwxr-xr-x 2 root root 40 Feb 10 17:57 bin  
drwx----- 2 root root 172 Feb 10 2007 boot  
drwxr-xr-x 8 root root 3200 Feb 10 17:58 dev  
drwxr-xr-x 50 root root 480 Feb 10 18:00 etc  
drwxr-xr-x 3 root root 60 Feb 10 18:00 home  
dr-xr-xr-x 5 root root 2048 Feb 10 2007 image  
drwxr-xr-x 11 root root 40 Feb 10 17:57 lib  
drwxr-xr-x 2 root root 40 Feb 10 2007 media  
drwxr-xr-x 5 root root 100 Feb 10 17:57 mnt  
drwxr-xr-x 2 root root 3 Feb 10 2007 opt  
dr-xr-xr-x 53 root proc 0 Feb 10 17:56 proc  
drwxr-xr-x 3 root root 40 Feb 10 17:57 root  
drwxr-xr-x 2 root root 40 Feb 10 17:57 sbin  
drwxr-xr-x 2 root root 3 Feb 10 2007 srv  
drwxr-xr-x 11 root root 0 Feb 10 17:56 sys  
drwxrwxrwt 7 root root 140 Feb 10 18:00 tmp  
drwxr-xr-x 13 root root 152 Feb 10 2007 usr  
drwxr-xr-x 19 root root 40 Feb 10 17:57 var
```

Столбцы слева направо: тип файла и права доступа, число жестких ссылок, имя владельца, группа владельца, размер в байтах, время модификации, имя.

Если в качестве операнда указать имя файла то **ls** выдаст информацию только об этом файле:

```
[user@localhost /]$ ls -l bin/ls  
-rwxr-xr-x 1 root root 73704 Feb 10 2007 bin/ls
```

Если качестве операнда указать имя каталога, то будет выдано содержимое этого каталога, т.е. имена файлов в этом каталоге:

```
[user@localhost /]$ ls -l bin
total 4366
-rwxr-xr-x 1 root root 2852 Feb 1 23:01 arch
-rwxr-xr-x 1 root root 10732 Feb 18 2005 aumix-minimal
lrwxrwxrwx 1 root root 4 Feb 10 2007 awk -> gawk
-rwxr-xr-x 1 root root 13004 Feb 10 2007 basename
-rwxr-xr-x 1 root root 458000 Feb 14 2006 bash
...
```

Если не указаны какие-либо операнды, будет выдано содержимое текущего каталога. В случае, если указано несколько операндов, сначала выводится информация о файлах, а потом о каталогах. Вывод происходит в алфавитном порядке.

Можно использовать следующие опции:

-C

Вывод содержимого каталога в несколько столбцов. Принимается по умолчанию при выводе на терминал.

-F

Добавить «слэш» (/) к имени каталога, звездочку (*) к имени исполнимого файла, знак (@) к каждой символической ссылке.

-R

Рекурсивный обход встретившихся подкаталогов. Может быть полезна для создания списка всех файлов системы.

-s

Вывод сортируется по времени последней модификации файла.

-d

Вывод только имени каталога (но не содержимого).

```
[user@localhost /]$ ls -ld bin
drwxr-xr-x 2 root root 40 Feb 10 17:57 bin
```

-f

Вывод не сортируется, т.е. порядок вывода соответствует порядку файлов в каталоге.

```
[user@localhost /]$ ls -f
. bin dev home lib mnt proc sbin sys usr
.. boot etc image media opt root srv tmp var
```

-l

Для каждого файла выводится номер индексного дескриптора (уникальный номер файла).

```
[user@localhost /]$ ls -l
27 bin 2 etc 11 lib 484 opt 31 sbin 1063 tmp
1216 boot 32 home 867 media 1 proc 485 srv 1215 usr
1467 dev 1856 image 829 mnt 33 root 1 sys 10 var
```

-r

Изменение порядка сортировки на обратный (по алфавиту или по времени последнего использования).

-t

Сортировать по времени последней модификации файла (последний модифицированный выводится первым).

-1

Вывод одного имени на каждой строке. Принимается по умолчанию при выводе не на терминал.

mkdir

В процессе освоения системы Вам необходимо будет создать свою собственную структуру каталогов. После установки системы, каталогов в ней довольно немного и в них пока еще трудно затеряться начинающему пользователю. В процессе расширения файловой системы вам придется постоянно создавать новые и уничтожать старые каталоги, поэтому рассмотрим, как это все можно сделать.

Создать каталог довольно легко. Для этого существует команда **mkdir** (аналог её есть и в DOS системах). В качестве аргумента необходимо указать имя создаваемого каталога:

mkdir [-p] имя_каталога

```
[user@localhost /]$ cd
[user@localhost ~]$ mkdir test
[user@localhost ~]$ ls
Documents test tmp
```

Стандартные элементы точка (.) для самого каталога и две точки (..) для его родительского каталога создаются автоматически.

Команда **mkdir** может быть использована и для создания нижележащих подкаталогов непосредственно из текущего каталога с указанием полного пути к ним. В этом случае все указанные в пути каталоги должны существовать и быть доступны. Если же Вам необходимо моздать иерархию каталогов, удобно пользоваться опцией **-p**:

```
[user@localhost ~]$ mkdir -p test/xxx/yyy/zzz
[user@localhost ~]$ ls -R test
test:
xxx

test/xxx:
yyy

test/xxx/yyy:
zzz

test/xxx/yyy/zzz:
```

Без использования опции **-p** при попытке создать уже существующий каталог будет выдано соответствующее предупреждение.

rmdir

После создания разнообразнейших каталогов, через некоторое время у вас возникнет естественное желание удалить некоторые из них. В этом вам может помочь команда **rmdir**. Ее формат также прост, как и формат предыдущей команды **mkdir**:
rmdir имя_каталога

Команда **rmdir** удаляет каталог, имя которого указано в качестве параметра. Удаляемый каталог должен быть пустым, то есть перед его уничтожением вы должны удалить все находящиеся в нем файлы. Практически это приходится делать очень редко, так как с двумя этими проблемами успешно справляется команда **rm**, которая будет рассмотрена позже.

```
[user@localhost ~]$ rmdir test
rmdir: test: Directory not empty
```

Указанные имена каталогов обрабатываются по порядку. При одновременном удалении каталога и его подкаталога, подкаталог должен быть удален раньше.

touch

Команда **touch** только и делает, что меняет время последнего доступа к файлу. Замечательной особенностью этой команды является то, что с её помощью можно создавать новые пустые файлы — если в качестве аргумента передаётся имя несуществующего файла:

```
[user@localhost test]$ ls
xxx
[user@localhost test]$ touch file.test
[user@localhost test]$ ls -l
total 0
-rw-r--r-- 1 user user 0 Feb 10 18:26 file.test
drwxr-xr-x 3 user user 80 Feb 10 18:23 xxx
```

rm

В процессе работы с системой довольно часто возникает необходимость удаления файлов. Для этого существует команда **rm**, которая позволяет удалять как файлы, так и каталоги. Пользоваться ей нужно с большой осторожностью, так как UNIX-системы не имеют привычки, в отличие от Windows, переспрашивать пользователя перед удалением файла, а делают это быстро и навсегда. Поэтому, связи с тем, что в UNIX-системах нет привычных программ восстановления удаленных файлов, крепко подумайте перед тем, как что-то удалять.

Команда **rm** имеет следующий формат:

rm [-f | -i] [-dRr] имя_файла ...

С помощью данной команды вы можете удалить файлы, имена которых указаны в качестве параметров. Если файл защищен от записи и стандартным устройством ввода данных является терминал, пользователю будет выдан запрос на подтверждение удаления файла.

При попытке удаления каталога с помощью этой команды будет выдано сообщение об ошибке.

Вы можете использовать следующие опции:

-d

Если удаляемый файл окажется каталогом, то также его удалить.

-f

Удалить защищенный от записи файл без подтверждения. Если данный файл не существует, то не выводить сообщение об его отсутствии. Опция **-f** отменяет предыдущую опцию **-i**.

-i

Запрашивать подтверждение на удаление любого файла, независимо от того, является ли терминал стандартным устройством ввода или нет. Опция **-i** отменяет предыдущую опцию **-f**.

-r

Удалить все файлы и подкаталоги данного каталога, а после этого удалить и сам каталог. Опция **-R** подразумевает опцию **-d**. Если также указана опция **-i**, то пользователю выдается запрос на подтверждение удаления файлов и каталогов.

cp

Программа копирует содержимое файла в файл с другим именем либо в другой каталог с сохранением существующего имени файла, также применяется для копирования каталогов с их содержимым. Программа имеет следующий формат:

cp [-prg] имя_файла ... имя_файла_назначения

Вы можете использовать следующие опции:

-i

Пользователю выдается запрос на подтверждение при перезаписи существующих файлов и каталогов.

-p

Сохранять существующий режим доступа к файлу.

-r

Копирование каталога с его подкаталогами.

Копирование файла в текущий каталог с новым именем:

```
[user@localhost ~]$ cp /etc/resolv.conf resolv
```

Копирование файла в текущий каталог с сохранением имени:

```
[user@localhost ~]$ cp /var/log/apache/access_log .
```

Копирование каталога:

```
[user@localhost ~]$ cp -r test test.old
```

mv

Переименование файла или перемещение одного либо нескольких файлов (или каталогов) в другой каталог. Формат команды:

mv [-i | -f] имя_файла ... имя_файла_назначения

При переносе в рамках одной файловой системы команда только меняет путевое имя файла, поэтому переименование и перенос идентичны по реализации.

Вы можете использовать следующие опции:

-f

Запрет запроса подтверждений при перезаписи существующих файлов.

-i

Пользователю выдается запрос на подтверждение при перезаписи существующих файлов и каталогов.

Пример использования команды для переноса файла в каталог:

```
[user@localhost ~]$ mv text/user.html text/htmls/
```

ln

Эта команда создаёт ссылки на файлы как жёсткие, так и символические. Она имеет следующий формат (аналогичный команде **mv** по порядку аргументов):

ln [-s] имя_файла имя_файла_ссылки

Вы можете использовать следующие опции:

-f

Запрет запроса подтверждений при перезаписи существующих файлов (ссылок).

-s

Создание символической ссылки.

Примеры использования команды:

```
[user@localhost ~]$ ln text/alex/linux.html working/linux-todo.html
```

Создание символической ссылки на каталог:

```
[user@localhost ~]$ ln -s images/my/photos photos
```

du

Команда выводит размер пространства на диске, занятого каталогом (и всеми его подкаталогами), в блоках (по умолчанию, 1 блок составляет 512 байт). По умолчанию выводится информация о текущем каталоге.

Программа имеет следующие параметры:

-a

Выводить информацию не только о каталогах, но и о файлах.

-h

Вывод размера каталога в «человеческом» формате: килобайтах, мегабайтах и т.п..

-s

Вывод только общего итога, без отображения промежуточной информации.

Пример выполнения программы:

```
[user@localhost ~]$ du -sh test
925K  test
```

tar

Команда **tar** предназначена для архивации групп файлов и каталогов. Эта команда позволяет объединять группу файлов и каталогов со всеми атрибутами в единый файл, который имеет расширение «.tar». Полученный файл затем с помощью той же команды **tar**. Команда **tar** имеет множество опций, мы же остановимся на следующем формате использования команды:

tar [-c | -x] [-z | -j] -f имя_tar_файла имена_файлов_и_каталогов

Опция **-c** соответствует созданию архива, **-x** — разворачивание архива.

Созданный архив можно также сжать с помощью архиваторов **gzip** или **bzip2**. Для этого применяются соответственно опции **-z** и **-j**. Здесь хорошо видна разница между понятием

«архивирование», которое выполняет команда **tar**, и «сжатием», которые производят специализированные программы.

Имя архива, который создаётся или разворачивается передаётся после параметра **-f**. Рассмотрим примеры работы с этой программой:

Создание архива **etc.tar.bz2** каталога **/etc** с использованием сжатия **bzip2**:

```
[user@localhost ~]$ tar -cjf etc.tar.bz2 /etc
[user@localhost ~]$ ls
etc.tar.bz2
```

Разворачивание архива **etc.tar.bz2** в текущий каталог с использованием разсжатия **bzip2**:

```
[user@localhost ~]$ tar -xjf etc.tar.bz2
[user@localhost ~]$ ls
etc etc.tar.bz2
```

find

При работе с файлами часто бывает необходимо найти что-нибудь. В UNIX для этого существует команда **find**. В общем виде эта команда имеет следующий формат:

find [где_искать] [условие_поиска] [действие]

Первый аргумент команды **find** — это *имя каталога*, в которой производится поиск. По умолчанию, это текущий каталог.

Далее указываются *условия поиска*, которые могут принимать следующие значения:

-name

Имя файла. В нём можно использовать шаблоны, но при этом необходимо заключить имя в двойные кавычки.

-type

Тип файла: **f** — обычный файл, **d** — каталог, **l** — символическая ссылка и т.п..

-user

Имя владельца файла или его уникальный идентификатор (UID).

-group

Имя группы владельца файла или её уникальный идентификатор (GID).

-perm

Права доступа файла.

-size

Размер файла. Обычно за числом следует буква — в чём измеряется размер (в блоках, байтах, килобайтах и т.п.).

-atime

Время доступа к файлу.

-ctime

Время последнего изменения владельца файла.

-mtime

Время последнего изменения содержимого файла.

-newer

Поиск всех файлов новее указанного.

Условия поиска могут быть объединены с помощью модификаторов: **-a** — И, **-o** — ИЛИ, **!** — НЕ, **\(...\)** — группа условий.

К найденным файлам могут применяться следующие *действия*:

-print

Вывести на экран имя найденного файла вместе с путём. Это действие выполняется по умолчанию.

-delete

Удалить найденные файлы.

-exec команда {} \;

Исполнение указанной команды для каждого найденного файла с передачей имени файла в качестве аргумента ({}).

-ok команда {} \;

Аналогична действию *-exec*, только для каждого файла запрашивается подтверждение перед выполнением команды.

Рассмотрим примеры использования команды:

Вывод всех файлов в текущем каталоге и подкаталогах:

```
[user@localhost ~]$ find
.
./xxx
./xxx/yyy
./xxx/yyy/zzz
./file.test
```

Вывод всех файлов в каталоге /etc, начинающихся на «re»:

```
[user@localhost ~]$ find /etc -name "re*"
find: /etc/tcb: Permission denied
find: /etc/default: Permission denied
/etc/chroot.d/resolv.all
/etc/chroot.d/resolv.conf
/etc/chroot.d/resolv.lib
/etc/redhat-release
/etc/remounttab
/etc/resolv.conf
...
```

Вывод всех подкаталогов в каталоге /etc, начинающихся на «re»:

```
[user@localhost test]$ find /etc -name "rp*" -type d
/etc/rpm
```

Вывод всех файлов в каталоге /etc, изменённых за последние сутки:

```
[user@localhost test]$ find /etc -mtime -1
/etc
/etc/issue
/etc/issue.net
/etc/mtab
```

Вывод всех файлов в каталоге /tmp, не принадлежащих пользователю user:

```
[user@localhost test]$ find /tmp \! -user user
/tmp
```

```
/tmp/.private  
/tmp/.font-unix  
/tmp/.X11-unix
```

Удаление в текущем каталоге всех устаревших файлов (оканчивающихся на «~»):

```
[user@localhost test]$ find -name "*~" -delete
```

Вывод на экран имён всех текстовых файлов в домашнем каталоге:

```
[user@localhost test]$ find ~ -name "*.txt" -exec echo {} \  
/home/user/a.txt  
/home/user/unix_commands.txt
```

Копирование всех текстовых файлов на дискету:

```
[user@localhost test]$ find ~ -name "*.txt" -exec cp {} /mnt/floppy/ \  
/;
```

Сценарий: Знакомство с каталогами UNIX

В этом сценарии будет освещено знакомство с командами UNIX, изучение файловой системы и базовых каталогов, создание рабочей среды в домашнем каталоге пользователя для выполнения всех последующих команд.

Начальные условия: Командная строка после входа в систему.

Получить имя текущего каталога с помощью команды **pwd**:

```
[user@localhost ~]$ pwd  
/home/user/
```

Перейти в корневой каталог с помощью команды **cd /**:

```
[user@localhost ~]$ cd /  
[user@localhost /]$ pwd  
/
```

Отметить, как изменилась строка приглашения.

Посмотреть содержимое корневого каталога с помощью команды **ls**:

```
[user@localhost /]$ ls  
bin dev home lib mnt proc sbin sys usr  
boot etc image media opt root srv tmp var
```

Сравнить с использованием «расширенного» вывода команды **ls -F**:

```
[user@localhost /]$ ls -F  
bin/ dev/ home/ lib/ mnt/ proc/ sbin/ sys/ usr/  
boot/ etc/ image/ media/ opt/ root/ srv/ tmp/ var/
```

Каталоги отмечаются синим цветом и знаком «/» после имени.

Посмотреть содержимое домашнего каталога с помощью команды **ls ~**:

```
[user@localhost /]$ ls ~  
Documents tmp
```

Домашний каталог содержит набор стандартных каталогов.
Вернуться в домашний каталог с помощью **cd** без параметров:

```
[user@localhost /]$ cd  
[user@localhost ~]$
```

Создать каталог test с помощью команды **mkdir test**:

```
[user@localhost ~]$ mkdir test
```

Посмотреть обновлённое содержимое домашнего каталога **ls**:

```
[user@localhost ~]$ ls  
Documents test tmp
```

Создать подкаталог subtest в каталоге test командой **mkdir test/subtest**:

```
[user@localhost ~]$ mkdir test/subtest
```

Посмотреть содержимое домашнего каталога и его подкаталогов с помощью ключа рекурсивного просмотра **-R** в команде **ls -R**:

```
[user@localhost ~]$ ls -R  
..  
Documents test tmp  
  
./Documents:  
  
./test:  
subtest  
  
./test/subtest:  
  
./tmp:
```

Сценарий: Изучение типов файлов в UNIX

В этом сценарии рассматриваются типы файлов в UNIX: простые файлы, каталоги, ссылки. Изучаются команды создания и копирования файлов. Показывается разница между жесткими и символическими ссылками, между копированием и переносом файла.
Начальные условия: Командная строка. Каталог test после предыдущего сценария.
Создать пустой файл с помощью команды **touch first.txt**:

```
[user@localhost test]$ touch first.txt  
[user@localhost test]$ ls
```

```
first.txt subtest
```

Добавить строку текста в конец файла с помощью команды **echo "Hello, world" >> first.txt** и перенаправления вывода:

```
[user@localhost test]$ echo "Hello, world" >> first.txt
```

Посмотреть содержимое файла с помощью команды **cat first.txt**:

```
[user@localhost test]$ cat first.txt
Hello, world
[user@localhost test]$
```

Посмотреть расширенную информацию о каталоге, используя ключ **-l** команды **ls**. **ls -l test**:

```
[user@localhost test]$ ls -l
total 4
-rw-r--r-- 1 user user 13 Feb 14 20:12 first.txt
drwxr-xr-x 2 user user 60 Feb 14 19:42 subtest
```

Добавить псевдоним командной оболочки, чтобы сократить размер команды, с помощью команды **alias ls='ls -F -l'**:

```
[user@localhost test]$ alias ls='ls -F -l'
[user@localhost test]$ ls test
total 4
-rw-r--r-- 1 user user 13 Feb 14 20:12 first.txt
drwxr-xr-x 2 user user 60 Feb 14 19:42 subtest
```

Посмотреть глубже на каталог **test**, используя ключи **-a** и **-i**. **ls -a -i**

```
[user@localhost test]$ ls -a -i
total 4
1014 drwxr-xr-x 3 user user 100 Feb 14 20:07 ./
941 drwx----- 8 user user 340 Feb 14 19:28 ../
1081 -rw-r--r-- 1 user user 13 Feb 14 20:12 first.txt
1015 drwxr-xr-x 2 user user 60 Feb 14 19:42 subtest/
```

Первый столбец — номера индексных узлов файловой системы. Третий столбец — число жёстких ссылок файла.

Скопировать файл с помощью команды **cp first.txt copy1.txt**:

```
[user@localhost test]$ cp first.txt copy1.txt
[user@localhost test]$ ls -a -i
total 8
1014 drwxr-xr-x 3 user user 120 Feb 14 20:33 ./
941 drwx----- 8 user user 340 Feb 14 19:28 ../
1082 -rw-r--r-- 1 user user 13 Feb 14 20:33 copy1.txt
1081 -rw-r--r-- 1 user user 13 Feb 14 20:12 first.txt
1015 drwxr-xr-x 2 user user 60 Feb 14 19:42 subtest/
```

Новый файл имеет свой собственный индексный узел.

Переименовать файл с помощью команды **mv first.txt orig.txt. mv first.txt orig.txt:**

```
[user@localhost test]$ mv first.txt orig.txt
[user@localhost test]$ ls -a -i
total 8
1014 drwxr-xr-x 3 user user 120 Feb 14 20:37 ./
941 drwx----- 8 user user 340 Feb 14 19:28 ../
1082 -rw-r--r-- 1 user user  13 Feb 14 20:33 copy1.txt
1081 -rw-r--r-- 1 user user  13 Feb 14 20:12 orig.txt
1015 drwxr-xr-x 2 user user  60 Feb 14 19:42 subtest/
```

Отметим, что изменилось только имя файла, все остальные атрибуты остались прежними.

Создать жёсткую ссылку командой **ln orig.txt copy2.txt:**

```
[user@localhost test]$ ln orig.txt copy2.txt
[user@localhost test]$ ls -a -i
total 12
1014 drwxr-xr-x 3 user user 140 Feb 14 20:41 ./
941 drwx----- 8 user user 340 Feb 14 19:28 ../
1082 -rw-r--r-- 1 user user  13 Feb 14 20:33 copy1.txt
1081 -rw-r--r-- 2 user user  13 Feb 14 20:12 copy2.txt
1081 -rw-r--r-- 2 user user  13 Feb 14 20:12 orig.txt
1015 drwxr-xr-x 2 user user  60 Feb 14 19:42 subtest/
```

Добавилась ещё одна ссылка на тот же файл, число ссылок увеличилось на 1.

Создать символическую ссылку командой **ln -s orig.txt orig.lnk:**

```
[user@localhost test]$ ln -s orig.txt orig.lnk
[user@localhost test]$ ls -a -i
total 12
1014 drwxr-xr-x 3 user user 160 Feb 14 20:45 ./
941 drwx----- 8 user user 340 Feb 14 19:28 ../
1082 -rw-r--r-- 1 user user  13 Feb 14 20:33 copy1.txt
1081 -rw-r--r-- 2 user user  13 Feb 14 20:12 copy2.txt
1083 lrwxrwxrwx 1 user user   8 Feb 14 20:45 orig.lnk -> orig.txt
1081 -rw-r--r-- 2 user user  13 Feb 14 20:12 orig.txt
1015 drwxr-xr-x 2 user user  60 Feb 14 19:42 subtest/
```

Новый файл имеет новый индексный узел и размер, равный имени файла orig.txt.

Сравним содержимое файлов при обращении к ним по имени:

```
[user@localhost test]$ cat orig.txt
Hello, world.
[user@localhost test]$ cat copy2.txt
Hello, world.
[user@localhost test]$ cat orig.lnk
Hello, world.
```

К одним и тем же данным можно обратиться через жесткую или символическую ссылку.

Сценарий: Поиск системных журналов

В этом сценарии изучается команда поиска файлов и каталогов.

Начальные условия: Командная строка, каталог test после прошлого сценария.

Вывести список всех файлов и каталогов в текущей директории, включая содержимое подкаталогов с помощью команды **find**:

```
[user@localhost test]$ find
.
./orig.lnk
./copy2.txt
./orig.txt
./copy1.txt
./subtest
```

Найти все файлы и дериктории в текущем каталоге и ее подкаталогах, которые начиняются на «о» с помощью команды **find -name "o*"**:

```
[user@localhost test]$ find -name "o*"
./orig.lnk
./orig.txt
```

Найти все файлы и каталоги в каталоге /etc и его подкаталогах, которые начиняются на «о» с помощью команды **find /etc -name "o*"**:

```
[user@localhost test]$ find /etc -name "o*"
find: /etc/tcb: Permission denied
find: /etc/default: Permission denied
find: /etc/buildreqs: Permission denied
...
```

Список найденных файлов может быть слишком большим и для его просмотра можно воспользоваться «прокруткой» терминала с помощью клавиш **Shift-PgUp** и **Shift-PgDn**. Вы увидели множество сообщений об ошибках доступа («Permission denied»), которые можно подавить с помощью перенаправления ошибок следующим образом: **find /etc -name "o*" 2>/dev/null**

```
[user@localhost test]$ find /etc -name "o*" 2>/dev/null
/etc/modprobe.d/options
/etc/modutils.d/oss
/etc/net/ifaces/default/fw/options
/etc/net/ifaces/default/options
...
/etc/pam.d/other
/etc/rc.d/init.d/outformat
```

Найти все каталоги в /etc, которые начинаются на «о» с помощью команды **find /etc -name "o*" -a -type d 2>/dev/null**:

```
[user@localhost test]$ find /etc -name "o*" -a -type d 2>/dev/null
/etc/net/options.d
```

```
/etc/openssh  
/etc/openssl  
/etc/opt
```

Найти все обычные файлы в каталоге /var и его подкаталогах, заканчивающиеся на «log»:

```
[user@localhost test]$ find /var -name "*log" -a -type f 2>/dev/null  
/var/log/Xorg.0.log  
/var/log/faillog  
/var/log/lastlog
```

Создать каталог logs с помощью команды **mkdir logs**:

```
[user@localhost test]$ mkdir logs
```

Скопировать найденные файлы в локальный каталог с помощью параметра -exec команды find. Для этого выполнить: **find /var -name "*log" -a -type f -exec cp {} test/logs/ 2>/dev/null**:

```
[user@localhost test]$ find /var -name "*log" -a -type f -exec cp {} test/logs/ 2>/dev/null  
[user@localhost test]$ ls test/logs  
итого 789  
-rw-r----- 1 user users 601033 Окт 16 18:37 emerge.log  
-rw-r--r-- 1 user users 292292 Окт 16 18:37 lastlog  
-rw-r--r-- 1 user users 37383 Окт 16 18:37 Xorg.0.log
```

Были скопированы все файлы, права на чтение которых у нас есть.

Сценарий: Архивирование и деархивирование файлов и каталогов

В этом сценарии изучается команда архивирования файлов и каталогов. Командная строка, каталог test после прошлого сценария.

Начальные условия:

Перейти в домашний каталог с помощью команды **cd**:

```
[user@localhost test]$ cd  
[user@localhost ~]$
```

Создать архив с именем test.tar.gz с применением сжатия, содержащий каталог test с помощью команды **tar -czf test.tar.gz test**:

```
[user@localhost ~]$ tar -czf test.tar.gz test  
[user@localhost ~]$ ls  
drwxr-xr-x 4 user users 208 Окт 16 18:36 test/  
-rw-r--r-- 1 user users 79173 Окт 16 18:49 test.tar.gz
```

Создать новый каталог для содержимого архива командой **mkdir test2**:

```
user@desktop ~ $ mkdir test2
```

Перейти в новый каталог с помощью команды **cd test2**:


```
user@desktop ~ $ cd test2
```

Развернуть содержимое архива в текущий каталог с помощью команды **tar -xzf ../test.tar.gz**:

```
user@desktop test2 $ tar -xzf ../test.tar.gz
```

Убедиться, что содержимое распакованного архива совпадает с оригинальным каталогом. Ввести команду **ls -l -F**:

```
user@desktop test2 $ ls -l -F
итого 0
drwxr-xr-x  4 user users 208 Окт 16 18:36 test/
user@desktop test2 $ ls -l -F test
итого 12
-rw-r--r--  1 user users  13 Окт 15 20:54 copy1.txt
-rw-r--r--  2 user users  13 Окт 15 20:48 copy2.txt
drwxr-xr-x  2 user users 136 Окт 16 18:37 logs/
lrwxrwxrwx  1 user users   8 Окт 16 18:54 orig.lnk -> orig.txt
-rw-r--r--  2 user users  13 Окт 15 20:48 orig.txt
drwxr-xr-x  2 user users  48 Окт 13 21:33 subtest/
```

Сценарий: Создание новых текстовых файлов

В этом сценарии изучаются команды для создания и модифицирования текстовых файлов. Командная строка.

Начальные условия:

Ввести команду **echo "One line"**:

```
user@desktop test $ echo "One line"
One line
```

Данная команда принимает строку в виде аргумента и выводит её на стандартный вывод. Ввести ту же команду, но перенаправить вывод в файл **second.txt** с помощью «>». Ввести команду **echo "One line" > second.txt**:

```
user@desktop test $ echo "One line" > second.txt
```

Добавить в конец файла **second.txt** строку «A line» с помощью другого перенаправления **echo "A line" >> second.txt**:

```
user@desktop test $ echo "A line" >> second.txt
```

Вывести содержимое файла с помощью команды **cat second.txt**:

```
user@desktop test $ cat second.txt
One line
A line
```

С помощью команды `cat` можно создавать многострочные файлы —, если переопределить вывод программы в файл и вводить текст до нажатия **Ctrl-D** (конец ввода). Ввести команду **`cat >multiline.txt`** и набрать текст

```
user@desktop test $ cat >multiline.txt
Simple text:
blah-blah-blah
1 2 3 4 5 6 7 8 9 0

bye!

(Ctrl+D)
user@desktop test $
```

Убедиться, что содержимое файла совпадает с введённым текстом, включая все переводы строки. Для этого ввести команду **`cat multiline.txt`**

```
user@desktop test $ cat multiline.txt
Simple text:
blah-blah-blah
1 2 3 4 5 6 7 8 9 0

bye!
```

1. Основным назначением команды `cat` является объединение файлов, имена которых передаются как аргументы командной строки. Объединить файлы с помощью команды **`cat orig.txt second.txt multiline.txt > big.txt`**:

```
user@desktop test $ cat orig.txt second.txt multiline.txt > big.txt
```

2. Убедиться, что новый файл содержит строки из перечисленных файлов с помощью команды **`cat big.txt`**:

```
user@desktop test $ cat big.txt
Hello, world
One line
A line
Simple text:
blah-blah-blah
1 2 3 4 5 6 7 8 9 0

bye!
```

Контрольные вопросы.

1. Создайте каталог `test1` в домашнем каталоге. Сравните время создания системных каталогов `/bin`, `/tmp` с каталогом `test1`.
2. Скопируйте файл `/bin/l` в локальный каталог. Посмотрите атрибуты этого файла. Попробуйте запустить его.

3. Создайте в локальном каталоге символическую ссылку `tmpInk` на каталог `/tmp`. Скопируйте несколько файлов в каталог `tmpInk`.
4. Сравните файлы `/dev/tty1` и `/dev/hda1`. Какой тип они имеют? Чем они отличаются?
5. Найдите все файлы в системе, которые были модифицированы не более суток назад.
6. С помощью одной команды найдите все файлы с расширением `.html` в каталоге `/usr` и скопируйте их в локальный каталог `htmls`.
7. Создайте архив каталога `/etc` и узнайте его размер. Попробуйте создать сжатый архив того же каталога. Сравните степень сжатия `gzip` и `bzip2` на этом примере.