

Отчет к лабораторной работе №5

Common information

discipline: Операционные системы

author: Бабина Юлия Олеговна

group: НПМбд-01-21

Цель работы

Ознакомление с файловой системой Linux, её структурой, именами и содержанием каталогов. Приобретение практических навыков по применению команд для работы с файлами и каталогами, по управлению процессами (и работами), по проверке использования диска и обслуживанию файловой системы.

Ход работы

Выполним все примеры, приведённые в первой части описания лабораторной работы (рис.1).

```
[babinayuliaolegovna@yobabina ~]$ touch abc1
[babinayuliaolegovna@yobabina ~]$ touch april may
[babinayuliaolegovna@yobabina ~]$ cp abc1 april
[babinayuliaolegovna@yobabina ~]$ cp abc1 may
[babinayuliaolegovna@yobabina ~]$ mkdir monthly
[babinayuliaolegovna@yobabina ~]$ cp april may monthly
[babinayuliaolegovna@yobabina ~]$ touch june
[babinayuliaolegovna@yobabina ~]$ cp monthly/may monthly/june
[babinayuliaolegovna@yobabina ~]$ ls monthly
april june may
[babinayuliaolegovna@yobabina ~]$ mkdir monthly.00
[babinayuliaolegovna@yobabina ~]$ cp -r monthly monthly.00
[babinayuliaolegovna@yobabina ~]$ cp -r monthly.00 /tmp
[babinayuliaolegovna@yobabina ~]$ mv april july
mv: не удалось выполнить stat для 'april': нет такого файла или каталога
[babinayuliaolegovna@yobabina ~]$ mv april july
[babinayuliaolegovna@yobabina ~]$ mv july monthly.00
[babinayuliaolegovna@yobabina ~]$ ls monthly.00
july
[babinayuliaolegovna@yobabina ~]$ ls monthly
april june may
[babinayuliaolegovna@yobabina ~]$ mv monthly.00 monthly.01
[babinayuliaolegovna@yobabina ~]$ mkdir reports
[babinayuliaolegovna@yobabina ~]$ mv monthly.00 reports
```

рис.1

Скопируем файл /usr/include/sys/io.h в домашний каталог и назовем его equipment. (рис.2.1).

```
babinayuliaolegovna@yobabina sys]$ cp /usr/include/sys/io.h ~
babinayuliaolegovna@yobabina sys]$ cd ~
babinayuliaolegovna@yobabina ~]$ mv io.h equipment
```

рис.2.1

В домашнем каталоге создадим директорию ~/ski.places.(рис.2.2)

```
[babinayuliaolegovna@yobabina ~]$ mkdir ~/ski.places
```

рис.2.2

Переместим файл equipment в каталог ~/ski.places.(рис.2.3).

```
[babinayuliaolegovna@yobabina ~]$ mv equipment ~/ski.places
```

рис.2.3

Переименуем файл ~/ski.places/equipment в ~/ski.places/equiplist (рис.2.4).

```
[babinayuliaolegovna@yobabina ~]$ mv ~/ski.places/equipment ~/ski.places/equiplist
```

рис.2.4

Создим в домашнем каталоге файл abc1 и скопируем его в каталог ~/ski.places, назовем его equiplist2.(рис.2.5)

```
[babinayuliaolegovna@yobabina ~]$ cd ~
[babinayuliaolegovna@yobabina ~]$ mv abc1 ~/ski.places
[babinayuliaolegovna@yobabina ~]$ cd ski.places
[babinayuliaolegovna@yobabina ski.places]$ mv abc1 equiplist2
```

рис.2.5

Создим каталог с именем equipment в каталоге ~/ski.places.Переместим файлы ~/ski.places/equiplist и equiplist2 в каталог ~/ski.places/equipment.(рис.2.6)

```
[babinayuliaolegovna@yobabina ski.places]$ mkdir equipment
[babinayuliaolegovna@yobabina ski.places]$ mv equiplist equipment
[babinayuliaolegovna@yobabina ski.places]$ mv equiplist2 equipment
```

рис.2.6

Создадим и переместим каталог ~/newdir в каталог ~/ski.places и назовем его plans.(рис.2.7)

```
[babinayuliaolegovna@yobabina ~]$ cd ~/newdir
[babinayuliaolegovna@yobabina newdir]$ mv ski.places plans
[babinayuliaolegovna@yobabina newdir]$
```

рис.2.7

Определив опции команды chmod, необходимые для того, чтобы присвоить перечисленным ниже файлам выделенные права доступа, считая, что в начале таких прав нет (рис.3).

```
[babinayuliaolegovna@yobabina ~]$ chmod 744 australia
[babinayuliaolegovna@yobabina ~]$ mkdir play
[babinayuliaolegovna@yobabina ~]$ touch my_os
[babinayuliaolegovna@yobabina ~]$ touch feathers
[babinayuliaolegovna@yobabina ~]$ chmod 711 play
[babinayuliaolegovna@yobabina ~]$ chmod 544 my_os
[babinayuliaolegovna@yobabina ~]$ chmod 664 feathers
```

рис.3

Посмотрим содержимое файла /etc/passwd (рис.4.1).

```
[babinayuliaolegovna@yobabina ~]$ cd /etc
[babinayuliaolegovna@yobabina ~]$ mkdir etc
[babinayuliaolegovna@yobabina ~]$ cd etc
[babinayuliaolegovna@yobabina etc]$ touch password
[babinayuliaolegovna@yobabina etc]$ ls password
password
[babinayuliaolegovna@yobabina etc]$ cat password
abcdefghijklmnopqrstuvwxyz
```

рис.4.1

Скопируем файл ~/feathers в файл ~/file.old. Переместим файл ~/file.old в каталог ~/play(рис.4.2).

```
[babinayuliaolegovna@yobabina ~]$ cd ~
[babinayuliaolegovna@yobabina ~]$ cp feathers file.old
[babinayuliaolegovna@yobabina ~]$ mv file.old play
```

рис.4.2

Скопируем каталог ~/play в каталог ~/fun. Переместим каталог ~/fun в каталог ~/play и назовем его games (рис.4.3).

```
[babinayuliaolegovna@yobabina ~]$ cp -r play fun
[babinayuliaolegovna@yobabina ~]$ mv fun play
[babinayuliaolegovna@yobabina ~]$ mv play games
```

рис.4.3

Лишим владельца файла ~/feathers права на чтение. Далее вернем (рис.4.4).

```
[babinayuliaolegovna@yobabina ~]$ chmod u-r feathers
[babinayuliaolegovna@yobabina ~]$ cat feathers
cat: feathers: Отказано в доступе
[babinayuliaolegovna@yobabina ~]$ chmod u+r feathers
```

рис.4.4

Лишим владельца каталога ~/games права на выполнение. Далее вернем (рис.4.5).

```
[babinayuliaolegovna@yobabina ~]$ chmod u-x games
[babinayuliaolegovna@yobabina ~]$ cd games
bash: cd: games: Отказано в доступе
[babinayuliaolegovna@yobabina ~]$ chmod u+r games
```

рис.4.5

Прочитаем man по командам mount (рис.5.1), fsck (рис.5.2), mkfs (рис.5.3), kill (рис.5.4). Команда mount выводит информацию о файловой системе. Команда fsck позволяет проверить целостность системы файлов. Команда kill может прекратить процесс, который в настоящее время выполняется.

```
[babinayuliaolegovna@yobabina ~]$ man mount
[babinayuliaolegovna@yobabina ~]$ man fsck
[babinayuliaolegovna@yobabina ~]$ man mkfs
[babinayuliaolegovna@yobabina ~]$ man kill
```

рис.5

```
babinayuliaolegovna@yobabina:~$ man mount
MOUNT(8)                                System Administration                                MOUNT(8)

NAME
    mount - mount a filesystem

SYNOPSIS
    mount [-h|-V]

    mount [-l] [-t fstype]

    mount -a [-ffnrsvw] [-t fstype] [-O optlist]

    mount [-fnrsvw] [-o options] device|mountpoint

    mount [-fnrsvw] [-t fstype] [-o options] device mountpoint

    mount --bind|--rbind|--move olddir newdir

    mount
    --make {shared|slave|private|unbindable|rshared|rslave|rprivate|runbindable}
    mountpoint
```

рис.5.1

```
babinayuliaolegovna@yobabina:~ — man fsck
FSCK(8) System Administration FSCK(8)
NAME
    fsck - check and repair a Linux filesystem
SYNOPSIS
    fsck [-lsAVRTMNP] [-r [fd]] [-C [fd]] [-t fstype] [filesystem...] [--]
    [fs-specific-options]
DESCRIPTION
    fsck is used to check and optionally repair one or more Linux
    filesystems. filesystem can be a device name (e.g., /dev/hdc1,
    /dev/sdb2), a mount point (e.g., /, /usr, /home), or an filesystem
    label or UUID specifier (e.g.,
    UUID=8868abf6-88c5-4a83-98b8-bfc24057f7bd or LABEL=root). Normally, the
    fsck program will try to handle filesystems on different physical disk
    drives in parallel to reduce the total amount of time needed to check
    all of them.

    If no filesystems are specified on the command line, and the -A option
    is not specified, fsck will default to checking filesystems in
    /etc/fstab serially. This is equivalent to the -As options.

Manual page fsck(8) line 1 (press h for help or q to quit)
```

рис.5.2

```
babinayuliaolegovna@yobabina:~ — man mkfs
MKFS(8) System Administration MKFS(8)
NAME
    mkfs - build a Linux filesystem
SYNOPSIS
    mkfs [options] [-t type] [fs-options] device [size]
DESCRIPTION
    This mkfs frontend is deprecated in favour of filesystem specific
    mkfs.<type> utils.

    mkfs is used to build a Linux filesystem on a device, usually a hard
    disk partition. The device argument is either the device name (e.g.,
    /dev/hda1, /dev/sdb2), or a regular file that shall contain the
    filesystem. The size argument is the number of blocks to be used for
    the filesystem.

    The exit status returned by mkfs is 0 on success and 1 on failure.

    In actuality, mkfs is simply a front-end for the various filesystem
    builders (mkfs.fstype) available under Linux. The filesystem-specific
    builder is searched for via your PATH environment setting only. Please

Manual page mkfs(8) line 1 (press h for help or q to quit)
```

рис.5.3

```
babinayuliaolegovna@yobabina:~ — man kill
KILL(1) User Commands KILL(1)
NAME
    kill - terminate a process
SYNOPSIS
    kill [-signal|-s signal|-p] [-q value] [-a] [--timeout milliseconds]
    signal [--] pid/name...

    kill -l [number] | -L
DESCRIPTION
    The command kill sends the specified signal to the specified processes
    or process groups.

    If no signal is specified, the TERM signal is sent. The default action
    for this signal is to terminate the process. This signal should be used
    in preference to the KILL signal (number 9), since a process may
    install a handler for the TERM signal in order to perform clean-up
    steps before terminating in an orderly fashion. If a process does not
    terminate after a TERM signal has been sent, then the KILL signal may
    be used; be aware that the latter signal cannot be caught, and so does
    not give the target process the opportunity to perform any clean-up

Manual page kill(1) line 1 (press h for help or q to quit)
```

рис.5.4

Контрольные вопросы

Вопрос 1

Чтобы узнать, какие файловые системы существуют на жёстком диске моего компьютера, использую команду «df -Th». На моем компьютере есть следующие файловые системы: devtmpfs, tmpfs, ext4, iso9660. devtmpfs позволяет ядру создать экземпляры tmpfs с именем devtmpfs при инициализации ядра, прежде чем регистрируется какое-либо устройство с драйверами. Каждое устройство с майором /

минором будет предоставлять узел устройства в devtmpfs. devtmpfs монтируется на /dev и содержит специальные файлы устройств для всех устройств. tmpfs – временное файловое хранилище во многих Unix-подобных ОС. Предназначена для монтирования файловой системы, но размещается в ОЗУ вместо ПЗУ. Подобная конструкция является RAM диском. Данная файловая система также предназначена для быстрого и ненадёжного хранения временных данных. Хорошо подходит для /tmp и массовой сборки пакетов/образов. Предполагает наличие достаточного объёма виртуальной памяти. Файловая система tmpfs предназначена для того, чтобы использовать часть физической памяти сервера как обычный дисковый раздел, в котором можно сохранять данные (чтение и запись). Поскольку данные размещены в памяти, то чтение или запись происходят во много раз быстрее, чем с обычного HDD диска. ext4 – имеет обратную совместимость с предыдущими версиями ФС. Эта версия была выпущена в 2008 году. Является первой ФС из «семейства» Ext, использующая механизм «extent file system», который позволяет добиться меньшей фрагментации файлов и увеличить общую производительность файловой системы. Кроме того, в Ext4 реализован механизм отложенной записи (delayed allocation – delalloc), который так же уменьшает фрагментацию диска и снижает нагрузку на CPU. С другой стороны, хотя механизм отложенной записи и используется во многих ФС, но в силу сложности своей реализации он повышает вероятность утери данных. Характеристики: - максимальный размер файла: 16 TB; - максимальный размер раздела: 16 TB; - максимальный размер имени файла: 255 символов.

Рекомендации по использованию: - наилучший выбор для SSD; - наилучшая производительность по сравнению с предыдущими Ext-системами; - она так же отлично подходит в качестве файловой системы для серверов баз данных, хотя сама система и моложе Ext3.

ISO 9660 – стандарт, выпущенный Международной организацией по стандартизации, описывающий файловую систему для дисков CD-ROM. Также известен как CDFS (Compact Disc File System). Целью стандарта является обеспечить совместимость носителей под разными операционными системами, такими, как Unix, Mac OS, Windows.

Вопрос 2

Файловая система Linux/UNIX физически представляет собой пространство раздела диска разбитое на блоки фиксированного размера, кратные размеру сектора – 1024, 2048, 4096 или 8120 байт. Размер блока указывается при создании файловой системы. В файловой структуре Linux имеется один корневой раздел – / (он же root, корень). Все разделы жесткого диска (если их несколько) представляют собой структуру подкаталогов, “примонтированных” к определенным каталогам. - / – корень Это главный каталог в системе Linux. По сути, это и есть файловая система Linux. Адреса всех файлов начинаются с корня, а дополнительные разделы, флешки или оптические диски подключаются в папки корневого каталога. Только пользователь root имеет право читать и изменять файлы в этом каталоге. - /BIN – бинарные файлы пользователя

Этот каталог содержит исполняемые файлы. Здесь расположены программы, которые можно использовать в однопользовательском режиме или режиме восстановления. - /sbin – системные исполняемые файлы

Так же как и /bin, содержит двоичные исполняемые файлы, которые доступны на

ранних этапах загрузки, когда не примонтирован каталог /usr. Но здесь находятся программы, которые можно выполнять только с правами суперпользователя. - /ETC – конфигурационные файлы

В этой папке содержатся конфигурационные файлы всех программ, установленных в системе. Кроме конфигурационных файлов, в системе инициализации Init Scripts, здесь находятся скрипты запуска и завершения системных демонов, монтирования файловых систем и автозагрузки программ. - /DEV – файлы устройств

В Linux все, в том числе внешние устройства являются файлами. Таким образом, все подключенные флешки, клавиатуры, микрофоны, камеры – это просто файлы в каталоге /dev/. Выполняется сканирование всех подключенных устройств и создание для них специальных файлов. - /PROC – информация о процессах

По сути, это псевдофайловая система, содержащая подробную информацию о каждом процессе, его Pid, имя исполняемого файла, параметры запуска, доступ к оперативной памяти и так далее. Также здесь можно найти информацию об использовании системных ресурсов. - /VAR – переменные файлы

Название каталога /var говорит само за себя, он должен содержать файлы, которые часто изменяются. Размер этих файлов постоянно увеличивается. Здесь содержатся файлы системных журналов, различные кеши, базы данных и так далее. - /TMP – временные файлы

В этом каталоге содержатся временные файлы, созданные системой, любыми программами или пользователями. Все пользователи имеют право записи в эту директорию. - /USR – программы пользователя

Это самый большой каталог с большим количеством функций. Здесь находятся исполняемые файлы, исходники программ, различные ресурсы приложений, картинки, музыку и документацию. - /HOME – домашняя папка

В этой папке хранятся домашние каталоги всех пользователей. В них они могут хранить свои личные файлы, настройки программ и т.д. - /BOOT – файлы загрузчика

Содержит все файлы, связанные с загрузчиком системы. Это ядро vmlinuz, образ initrd, а также файлы загрузчика, находящие в каталоге /boot/grub. - /LIB – системные библиотеки

Содержит файлы системных библиотек, которые используются исполняемыми файлами в каталогах /bin и /sbin. - /OPT – дополнительные программы

В эту папку устанавливаются проприетарные программы, игры или драйвера. Это программы созданные в виде отдельных исполняемых файлов самими производителями. - /MNT – монтирование

В этот каталог системные администраторы могут монтировать внешние или дополнительные файловые системы. - /MEDIA – съемные носители

В этот каталог система монтирует все подключаемые внешние накопители –USB флешки, оптические диски и другие носители информации. - /SRV – сервер

В этом каталоге содержатся файлы серверов и сервисов. - /RUN - процессы Каталог, содержащий PID файлы процессов, похожий на /var/run, но в отличие от него, он размещен в TMPFS, а поэтому после перезагрузки все файлы теряются.

Вопрос 3

Чтобы содержимое некоторой файловой системы было доступно операционной системе необходимо воспользоваться командой mount.

Вопрос 4

Целостность файловой системы может быть нарушена из-за перебоев в питании, неполадок в оборудовании или из-за некорректного/внезапного выключения компьютера. Чтобы устранить повреждения файловой системы необходимо использовать команду `fsck`.

Вопрос 5

Файловую систему можно создать, используя команду `mkfs`. Ее краткое описание дано в пункте 5 в ходе выполнения заданий лабораторной работы.

Вопрос 6

Для просмотра текстовых файлов существуют следующие команды: - `cat`

Задача команды `cat` очень проста – она читает данные из файла или стандартного ввода и выводит их на экран.

Синтаксис утилиты:

`cat` опции файл1 файл2 ...

Основные опции:

-b – нумеровать только непустые строки

-E – показывать символ \$ в конце каждой строки

-n – нумеровать все строки

-s – удалять пустые повторяющиеся строки -T – отображать табуляции в виде ^I

-h – отобразить справку

-v – версия утилиты - nl

Команда `nl` действует аналогично команде `cat`, но выводит еще и номера строк в столбце слева. - less

Существенно более развитая команда для пролистывания текста.

При чтении данных со стандартного ввода она создает буфер, который позволяет листать текст как вперед, так и назад, а также искать как по направлению к концу, так и по направлению к началу текста.

Синтаксис аналогичный синтаксису команды `cat`.

Некоторые опции:

-g – при поиске подсвечивать только текущее найденное слово (по умолчанию подсвечиваются все вхождения)

-N – показывать номера строк - head

Команда `head` выводит начальные строки (по умолчанию – 10) из одного или нескольких документов. Также она может показывать данные, которые передает на вывод другая утилита.

Синтаксис аналогичный синтаксису команды `cat`.

Основные опции:

-c (-bytes) – позволяет задавать количество текста не в строках, а в байтах

-n (-lines) – показывает заданное количество строк вместо 10, которые выводятся по умолчанию

-q (-quiet, -silent) – выводит только текст, не добавляя к нему название файла

-v (-verbose) – перед текстом выводит название файла -z (-zero-terminated) – символы перехода на новую строку заменяет символами завершения строк - tail

Эта команда позволяет выводить заданное количество строк с конца файла, а также

выводить новые строки в интерактивном режиме.

Синтаксис аналогичный синтаксису команды cat.

Основные опции:

- c – выводить указанное количество байт с конца файла
- f – обновлять информацию по мере появления новых строк в файле
- n – выводить указанное количество строк из конца файла
- pid – используется с опцией -f, позволяет завершить работу утилиты, когда завершится указанный процесс
- q – не выводить имена файлов
- retry – повторять попытки открыть файл, если он недоступен
- v – выводить подробную информацию о файле

Вопрос 7

Утилита cp позволяет полностью копировать файлы и директории.

Синтаксис:

cp опции файл-источник файл-приемник

После выполнения команды файл-источник будет полностью перенесен в файл-приемник. Если в конце указан слэш, файл будет записан в заданную директорию с оригинальным именем.

Основные опции:

- attributes-only – не копировать содержимое файла, а только флаги доступа и владельца
- f, -force – перезаписывать существующие файлы
- i, -interactive – спрашивать, нужно ли перезаписывать существующие файлы
- L – копировать не символические ссылки, а то, на что они указывают -p – не перезаписывать существующие файлы
- P – не следовать символическим ссылкам
- r – копировать папку Linux рекурсивно
- s – не выполнять копирование файлов в Linux, а создавать символические ссылки
- u – скопировать файл, только если он был изменён
- x – не выходить за пределы этой файловой системы
- p – сохранять владельца, временные метки и флаги доступа при копировании
- t – считать файл-приемник директорией и копировать файл-источник в эту директорию

Вопрос 8

Команда mv используется для перемещения одного или нескольких файлов (или директорий) в другую директорию, а также для переименования файлов и директорий.

Синтаксис:

mv -опции старый_файл новый_файл

Основные опции:

- help – выводит на экран официальную документацию об утилите
- version – отображает версию mv
- b – создает копию файлов, которые были перемещены или перезаписаны
- f – при активации не будет спрашивать разрешение у владельца файла, если речь идет о перемещении или переименовании файла -i – наоборот, будет спрашивать разрешение у владельца
- n – отключает перезапись уже существующих объектов

-strip-trailing-slashes — удаляет завершающий символ / у файла при его наличии
-t директория — перемещает все файлы в указанную директорию
-u — осуществляет перемещение только в том случае, если исходный файл новее объекта назначения
-v — отображает сведения о каждом элементе во время обработки команды
Команда rename также предназначена, чтобы переименовать файл.

Синтаксис:

rename опции старое_имя новое_имя файлы

Основные опции:

-v — вывести список обработанных файлов
-n — тестовый режим, на самом деле никакие действия выполнены не будут
-f — принудительно перезаписывать существующие файлы

Вопрос 9

Права доступа – совокупность правил, регламентирующих порядок и условия доступа субъекта к объектам информационной системы (информации, её носителям, процессам и другим ресурсам) установленных правовыми документами или собственником, владельцем информации.

Права доступа к файлу или каталогу можно изменить, воспользовавшись командой chmod. Сделать это может владелец файла (или каталога) или пользователь с правами администратора.

Синтаксис команды:

chmod режим имя_файла

Режим имеет следующие компоненты структуры и способ записи: - = установить право
- - лишить права - +дать право - r чтение - w запись - x выполнение - u (user) владелец файла - g (group) группа, к которой принадлежит владелец файла - o (others) все остальные

Вывод

В ходе данной лабораторной работы я ознакомилась с файловой системой Linux, её структурой, именами и содержанием каталогов. Приобрела практические навыки по применению команд для работы с файлами и каталогами, по управлению процессами (и работами), по проверке использования диска и обслуживанию файловой системы.