## Работа с файловыми системами и памятью

### Цель работы

Познакомиться с работой с файловой системой из командной строки - научиться производить простые операции - монтирование и размонтирование файловых систем, мониторинг свободного места.

Освоить команды *free, du, dd, df, mount, umount*.

### Задания к выполнению

1. Выведите в командной строки информацию о заполненности оперативной памяти и файла подкачки.
2. Выведите в консоли объем занимаемой домашней папкой, папкой /etc и папкой /tmp
3. Найдите в домашней папке все файлы с расширением txt.
4. Выведите в консоль общее количество файлов в домашней папке и всех подпапках
5. Познакомьтесь с файлом /etc/fstab найдите в нем строки, соответствующие корневой файловой системе и разделу подкачки;
6. Если в данном файле есть и другие файловые системы, то покажите их идентификатор и точку монтирования;
7. Найдите в файле /etc/fstab или в графической утилите управления дисками параметры монтирования, а именно: имя файла устройства, точку монтирования, тип файловой системы.

### Методические указания

#### Команда du

Команда Linux du (Disk Usage, т.е. использование диска) – это стандартная команда Unix/Linux, используется для показа суммарной информации об использовании диска файлами и директориями. Команда du имеет много опций параметров и может выводить результаты в различных форматах. Команда du также отображает размеры файлов и директорий в рекурсивной манере.

Для определения информации о сводном использовании диска деревом директории и каждой поддиректорией, введите команду вида:

du путь/до/директории

Использование с командой du опции -h предоставляет результаты в формате, понятном человеку. Имеется ввиду, что размеры будут представлены в байтах, килобайтах, мегабайтах, гигабайтах и т.д.

Чтобы узнать общий размера директории, т.е. сколько место занимает папка и её подпапки, используется опция -s.

Использование с командой du флага -a отображает использование диска всеми файлами, а не только каталогами.

Узнать размер папки /home/user/data/backup/ :

# du -sh /home/user/data/backup/

Размеры папок, включая файлы внутри папки:

du -sh /var/log/\*

Тоже самое, но с сортировкой по занимаемому месту:

# du -sh /var/log/\* | sort -hr

#### Команда free

Команда free предоставляет информацию об использованной и неиспользованной памяти, а так же о разделе подкачки (swap). В этой статье разберем несколько полезных примеров использования этой команды.

Синтаксис команды free весьма прост:

free [параметры]

Если требуется узнать какое количество памяти и свопа используется, наберите команду free в терминале. Информация отобразится в килобайтах:

# free

Чтобы увидеть всю информацию в байтах, укажите ключ -b:

# free -b

Для вывода в мегабайтах используйте ключ -m:

# free -m

Для гигабайтов -g:

# free -g

Если вы хотите увидеть полную память (включая своп), используйте ключ -t, который выведет информацию типа той, что приведена ниже.

# free -t

Ключ -s используется для указания времени обновления информации. К примеру, для того, чтобы выводить информацию в мегабайтах каждые 5 секунд, запустите команду.

# free -ms 5

Ключ -с указывает количество раз, доступных для обновления информации, инициируемой командой -s. Нижеприведенная команда отобразит всего 2 вывода информации о памяти с 5 секундным интервалом в мегабайтах

# free -m -s 5 -c 2

Теперь разберем, что означают значения в выводе команды free

Строки:

* Mem – физическая память
* Swap – виртуальная память

Колонки:

* total – общий размер ОЗУ
* used – реально использующая в данный момент и зарезервированная системой память
* free – свободная память (total - used)
* shared – разделяемая память
* buffers – буферы в памяти – страницы памяти, зарезервированные системой для выделения их процессам, когда им это потребуется
* cached – файлы, которые недавно были использованы системой/процессами и хранящиеся в памяти на случай, если они снова потребуются.

Больше информации доступно в справочной информации man free.

#### Команда df

Утилита df поставляется по умолчанию во всех дистрибутивах Linux и имеет очень простой синтаксис. Фактически вы можете просто набрать df и уже получить результат, но чтобы сделать вывод более читаемым используются дополнительные опции. Вот основной синтаксис:

$ df опции устройство

Устройство указывать необязательно, но можно указать раздел диска, о котором мы хотим посмотреть информацию. А теперь рассмотрим основные опции утилиты:

* -a, --all - отобразить все файловые системы, в том числе виртуальные, псевдо и недоступные;
* -B - изменить размер одного блока перед выводом данных, например, можно использовать BM, чтобы вывести все данные в мегабайтах;
* -h - выводить размеры в читаемом виде, в мегабайтах или гигабайтах;
* -H - выводить все размеры в гигабайтах;
* -i - выводить информацию об inode;
* -k - выводить размеры в килобайтах;
* --output - использовать специальный формат вывода, если не задано, выводит все поля. Доступны такие варианты: 'source', 'fstype', 'itotal', 'iused', 'iavail', 'ipcent', 'size', 'used', 'avail', 'pcent', 'file' и 'target';
* -P - использовать формат вывода POSIX;
* --total - выводить всю информацию про использованное и доступное место;
* -t, --type - выводить информацию только про указанные файловые системы;
* -x - выводить информацию обо всех, кроме указанных файловых систем;

Для получения информации о свободном пространстве в смонтированных разделах достаточно использовать параметр -h, активирующий режим генерации читаемого человеком вывода:

$ df -h

Для того, чтобы получить информацию о свободном пространстве в разделе с заданной директорией достаточно передать утилите путь к этой директории:

$ df -h /home

Для того, чтобы получить информацию о свободном пространстве в разделах с файловой системой заданного типа следует использовать параметр -t и указать тип этой файловой системы:

$ df -h -T -t ext4

Для получения более подробной информации смотрите страницу man.

$ man df

#### Команда dd

Довольно часто системным администраторам приходится копировать различные двоичные данные. Например, иногда может понадобиться сделать резервную копию жесткого диска, создать пустой файл, заполненный нулями для организации пространства подкачки или другой виртуальной файловой системы.

Фактически, это аналог утилиты копирования файлов cp только для блочных данных. Утилита просто переносит по одному блоку данных указанного размера с одного места в другое. Поскольку в Linux все, в том числе, устройства, считается файлами, вы можете переносить устройства в файлы и наоборот.

Обычные пользователи используют команду dd чаще всего для создания образов дисков DVD или CD. Например, чтобы сохранить образ диска в файл можно использовать такую команду:

sudo dd if=/dev/sr0 of=~/CD.iso bs=2048 conv=noerror

Фильтр noerror позволяет отключить реагирование на ошибки. Дальше, вы можете создать образ жесткого диска или раздела на нем и сохранить этот образ на диск. Только смотрите не сохраните на тот же жесткий диск или раздел, чтобы не вызвать рекурсию:

dd if=/dev/sda of=~/disk.img

Мы можем создать файл размером 512 мегабайт, заполнив его нулями из /dev/zero или случайными цифрами из /dev/random:

sudo dd if=/dev/zero of=file.img bs=1M count=512

Если образ диска слишком большой, можно перенаправить весь вывод на стандартный поток вывода утилиты gzip:

dd if =/dev/sda2 | bzip2 disk.img.bz2

Также можно использовать утилиту dd linux для копирования файлов, хотя это и не является ее прямым предназначением:

dd if=/home/sergiy/test.txt of=/home/sergiy/test1.txt

Как вы знаете, команда dd linux пишет данные на диск непосредственно в двоичном виде, это значит, что записываются нули и единицы. Они переопределяют то, что было раньше размещено на устройстве для записи. Поэтому чтобы стереть диск вы можете просто забить его нулями из /dev/zero.

sudo dd if=/dev/zero of=/dev/sdb

Такое использование dd приводит к тому что весь диск будет полностью стерт.

MBR расположена в первых 512 байтах жесткого диска, и состоит из таблицы разделов, загрузчика и пары доп. байт. Иногда, ее приходится бекапить, восстанавливать и т.д. Бекап выполняется так:

dd if=/dev/sda of=mbr.img bs=512 count=1

Восстановить можно проще:

dd if=mbr.img of=/dev/sda

#### Монтирование файловых систем

Для использования какой-либо файловой системы в операционной системе она должна быть смонтирована (подключена) к ней. Монтирование - это многоэтапный процесс, включающий проверку файловой истемы на ошибки, загрузку драйверов для конкретной файловой системы, подключение новой файловой системы к точке монтирования - какому-либо существующему каталогу. К счастью для пользователя, все это можно произвести всего одной командой - mount.

Некоторые файловые системы монтируются автоматически, при загрузке ОС. Их список вместе с параметрами монтирования указан в специальном конфигурационном файле /etc/fstab Типичный пример - корневая файловая система - раздел жесткого диска, на котором, собственно, и лежит сама ОС всегда автоматически монтируется. Его точка монтирования - корневой каталог. Вы можете изменять этот, чтобы управлять процессом автомонтирования файловых систем.

Чтобы вручную смонтировать файловую систему (например, подключенный к компьютеру сменный носитель, или новый раздел на жестком диске) нужно ввести команду mount с двумя обязательными параметрами - именем файла-устройства с нужным разделом диска и имя каталога, который послужит точкой монтирования. например:

mount /dev/sda5 /media/user/volume

В данном примере мы подключаем пятый раздел на первом жестком диске (устройство sda - serial disk A) к каталогу /media/user/volume. Стандартный каталог /media/ часто используется в некоторых дистрибутивах для хранения точек монтирования. Как мы можем видеть, они также разделены для разных пользователей. По поводу принятых соглашений в отношении точек монтирования обратитесь к документации к вашему дистрибутиву.

Большинство современных дистрибутивов автоматически монтирует разделы на сменных носителях при их подключении, поэтому вам не надо будет вводить команды каждый раз, когда вы вставляете флешку в разъем.

Для отключения файловой системы используется команда размонтирования - umount. Изучите ее синтаксис и основные параметры самостоятельно.

### Контрольные вопросы

1. Как можно из командной строки узнать размер свободного места на жестком диске?
2. В чем разница между файлом устройства и разделом жесткого диска?
3. Как можно настроить автоматическое монтирование разделов при загрузке системы?
4. Что такое файл подкачки? Чем он отличается от раздела подкачки? Для чего они служат?
5. Чем отличается механизм подключения файловых систем в Linux и Windows?

### Дополнительные задания

1. Переустановите операционную систему. При разметке диска выберите ручной режим и создайте отдельные разделы для папок /, /home, /boot и /tmp. Также создайте отдельный раздел для файла подкачки. Загрузите систему и в командной строке посмотрите размер свободного места на каждом разделе.
2. Если вы работаете на физической машине, вставьте в компьютер флешку или другой извлекаемый носитель данных. Убедитесь, что файловая система на нем подмонтировалась автоматически. Отмонтируйте сменный носитель вручную и затем примонтируйте его в другой каталог.
3. Если вы работаете в виртуальной машине, установите дополнения гостевой ОС и настройте общую папку между хостовой и гостевой операционными системами. Убедитесь, что она функционирует в обе стороны.
4. (\*) Если на вашей системе не настроена гибернация, настройте ее. При необходимости, увеличьте размер файла подкачки.