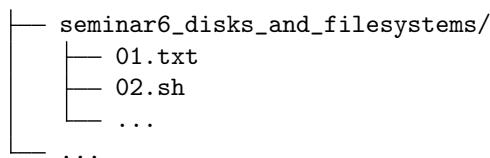


# Семинар #6: Диски и файловые системы. Практика.

## Как сдавать задачи

Для сдачи ДЗ вам нужно создать репозиторий на GitLab (если он ещё не создан) под названием `devtools-homework`. Структура репозитория должна иметь вид:



Для каждой задачи, если не сказано иное, нужно создать 1 файл решения с расширением `.sh`. Если в задаче есть подзадачи, то их нужно оформлять внутри каждого из файлов в следующем формате:

```
# Subtask a  
lsblk  
# Subtask b  
...
```

Для каждой подзадачи нужно прописать все команды, которые исполняются в ходе выполнения этой подзадачи. Если в задаче встречается вопрос, то на этот вопрос нужно ответить в комментариях (начинаются с `#`) скрипта.

## Предварительные замечания

- **Важно! Резервное клонирование ВМ.**

При выполнении данного задания придётся работать от имени суперпользователя (`root`), используя команды `su` и `sudo`. Выполняя такие команды, можно случайно сломать систему. Поэтому перед выполнением данного задания на всякий случай клонируйте вашу виртуальную машину. В VirtualBox для этого нужно нажать правой кнопкой мыши на виртуальной машине и выбрать "Клонировать".

**Это задание нужно делать только в виртуальной машине!**

- **Создание диска для виртуальной машины.**

Это задание нужно выполнять в виртуальной машине. Для выполнения этого задания вам понадобиться создать новый виртуальный диск. В Virtual Box это можно сделать следующим образом:

- Выключите виртуальную машину, если она включена.
- Выберите вашу виртуальную машину и нажмите настроить.
- Выберите вкладку "Носители".
- Выберите "Контроллер SATA" и нажмите "Создать жёсткий диск".
- В верхней панели выберите "Создать".
- Установите размер жёсткого диска в 3.15 гигабайта (это примерно равно 3000 мебибайтам).
- Добавьте новый диск к вашей виртуальной машине (он должен отображаться на вкладке носители).
- Запустите виртуальную машину и выполните `lsblk`, чтобы посмотреть, что создался новый диск размером 3 ГиБ. Скорей всего он будет иметь имя `sdb`.

- **Проверяйте имя диска**

Во всех задачах задания (после задачи 2) необходимо будет работать с новым, специально созданным диском, а не с диском на котором установлена операционная система. Команды, которые вы будете использовать в этом задании будут изменять или стирать важные структуры на диске. Применение этих команд к диску, на котором установлена операционная система, немедленно приведет к поломке системы.

Скорей всего диск, на котором установлена ОС у вас будет называться `sda`, а новый пустой диск будет называться `sdb`, но это необязательно. Более того, имена дисков (хотя это и маловероятно) могут поменяться после перезапуска системы. Поэтому всегда проверяйте к какому диску вы применяете ту или иную команду, используя команду `lsblk`.

## **Задача 1. Единицы измерения информации**

Произведите конвертацию одних величин в другие с точностью до трёх знаков после запятой.

- 1МБ в байты
- 1МиБ в байты
- 1ГиБ в гигабайты
- 1ГБ в мебибайты

Для решения этой задачи создайте файл `01.txt` с ответами.

## **Задача 2. Просмотр информации о дисках и разделах**

Некоторые из этих команд требуют прав суперпользователя, используйте `sudo`, чтобы запустить их.

### **a. Просмотр информации о дисках и разделах**

Просмотрите ваши текущие диски и разделы, используя команду `lsblk`.

### **b. Просмотр информации о UUID и файловых системах**

Просмотрите уникальные идентификаторы дисков и разделов, а также используемые на этих разделах файловые системы. Используйте команду `sudo blkid`.

### **c. Просмотр всех смонтированных файловых систем**

Просмотрите все смонтированные на данный момент файловые системы, используя команду `findmnt`. Команда `findmnt` показывает не только физические разделы и диски, но и виртуальные (`proc`, `sysfs`) и другие.

### **d. Просмотр таблицы разделов**

Просмотрите подробную информацию о разделах `parted -l`. Какая таблица разделов используется на вашем основном диске?

### **e. Просмотрите системный файл `/etc/fstab`**

Файл `/etc/fstab` хранит в себе файловых системах, которые должны быть автоматически смонтированы при загрузке системы. Для просмотра этого файла используйте команду:

```
$ cat /etc/fstab
```

### **f. Просмотрите директорию, содержащую файлы устройств**

Просмотрите директорию `/dev`, используя подробный вывод `ls -la`. Обратите внимание на файлы блочных устройств, например `sda` (файл устройства – диска), `sda1` (файл устройства – раздела диска) и другие подобные файлы.

### **g. Просмотрите подробную информацию о диске/разделе**

Используйте команду `file -s` на файлах устройств, чтобы посмотреть подробную информацию о соответствующем диске и разделе.

```
$ sudo file -s /dev/sda
$ sudo file -s /dev/sda1
```

### **h. Просмотр использованного места**

Просмотрите использованное место на различных разделах, используя команду `df -h`.

### **i. Просмотр общего размера директорий**

Используйте команду `du`, чтобы просмотреть сколько места на диске занимают директории `/home` и `/usr` вместе со всеми внутренними файлами. Сравните полученные числа с тем, что выводит `ls -l`.

### **Задача 3. Разметка диска, создание файловой системы и монтирование**

#### **a. Новый диск**

Найдите как в системе называется новый диск, созданный в части "Предварительные замечания". Далее во всём задании будет предполагаться, что он называется `sdb` и его файл устройства находится в `/dev/sdb`.

#### **b. Новый раздел**

Используйте программу `parted`, чтобы создать новый раздел на диске `sdb`. Раздел должен занимать весь диск. Раздел будет иметь имя `sdb1`, а файл устройства этого раздела будет `/dev/sdb1`. Посмотрите, что новый раздел создался, используя `lsblk`.

#### **c. Создаём файловую систему**

Создайте файловую систему `ext4` на разделе `sdb1`, используя команду `mkfs`. Используйте `lsblk -f`, чтобы посмотреть, что в разделе `sdb1` используется файловая система `ext4`.

#### **d. Монтируем файловую систему**

Создайте новую папку `/mnt/myfs`. Измените владельца этой папки на вашего пользователя, чтобы можно было работать с папкой без использования `sudo`. Используйте команду `mount`, чтобы примонтировать файловую систему к директории `/mnt/myfs`. Используйте команду `findmnt`, чтобы убедиться, что файловая система была примонтирована.

#### **e. Используем файловую систему**

Перейдите в директорию `/mnt/myfs` и создайте там два файла: `a.txt`, который будет содержать строку "Alpaca" и `b.txt`, который будет содержать строку "Bison". Создайте большой пустой файл под названием `large`, размером 100 МиБ, используя команду:

```
dd if=/dev/zero of=./large bs=1M count=100
```

Выполните команду `df -h`, чтобы убедиться что количество занятого места на разделе увеличилось.

#### **f. Перемонтирование в другом месте**

Создайте новую директорию `/home/shared`. Дайте этой папке полные права (`rwxrwxrwx`), чтобы можно было работать с ней без использования `sudo`. Размонтируйте файловую систему с раздела `sdb1` из папки `/mnt/myfs` и примонтируйте её к новой директории `/home/shared`. Зайдите в эту директорию и убедитесь, что все файлы сохранились.

#### **g. Перезагрузка**

Перезагрузите виртуальную машину и проверьте файлы в директории `/home/shared`. Заново примонтируйте файловую систему в папку `/home/shared` и убедитесь, что все файлы сохранились.

#### **h. Запись в `/etc/fstab`**

Файловая система в `/home/shared` была примонтирована временно. После перезагрузки системы её придётся монтировать снова. Чтобы указать, что эту систему нужно монтировать автоматически при загрузке системы, нужно добавить новую запись в `/etc/fstab`. Этот файл нужно редактировать очень осторожно, так как ошибка в этом файле может привести к тому, что система не запустится.

- Укажите файловую систему по UUID раздела, на который она установлена. UUID можно найти, используя команду `sudo blkid`.
- Для поля `options` укажите значение `defaults`.
- Для полей `dump` и `pass` укажите значения 0.

#### **i. Проверьте, что запись в `/etc/fstab` корректна**

Размонтируйте вашу файловую систему и выполните `mount -a`. Эта команда проверит запись в `/etc/fstab` на корректность и, если запись корректна, она смонтирует её. Команда не будет работать, если файловая система уже смонтирована.

```
$ sudo umount /home/shared  
$ sudo mount -a
```

Перезагрузите виртуальную машину и убедитесь, что файловая система в `/home/shared` была автоматически примонтирована.

## Задача 4. Несколько разделов (Таблица разделов MBR)

Для создания таблицы разделов и удаления/создания разделов в этой задаче используйте программу `parted`. Размонтируйте файловую систему на `sdb` и удалите раздел `sdb1`. Удалите запись, соответствующую файловой системе удалённого раздела из `/etc/fstab`.

Создайте на диске таблицу разделов MBR. Создайте на диске 4 раздела: 3 основных (primary) и 1 расширенный (extended). Внутри расширенного раздела создайте 2 логических (logical) раздела. Затем создайте в этих разделах файловые системы, используя `mkfs`, в соответствии со следующей таблицей:

раздел	размер	файловая система
<code>sdb1</code>	1000 MiB	xsf
<code>sdb2</code>	1000 MiB	btrfs
<code>sdb3</code>	100 MiB	ext4
<code>sdb4</code>	—	расширенный раздел
<code>sdb5</code>	100 MiB	ext4
<code>sdb6</code>	800 MiB	fat32

- Для btrfs может понадобиться установить пакет `btrfs-progs`, а для xfs – `xfsprogs`.
- Первый и/или последний раздел может получиться чуть меньше или чуть больше.
- Проверьте, что созданный диск использует таблицу разделов MBR. Для этого выполните `sudo parted -l`. В выводе должна присутствовать строка:

```
Disklabel type: msdos
```

- После создания разделов может потребоваться вызвать команду `partprobe`, чтобы система их увидела.
- Проверьте, что все разделы были созданы вместе с соответствующими файловыми системами, используя команду `lsblk -f`.

## Задача 5. Несколько разделов (Таблица разделов GPT)

Для создания таблицы разделов и удаления/создания разделов в этой задаче используйте программу `parted`. Создайте на диске `sdb` таблицу разделов GPT и 5 разделов. Затем создайте в этих разделах файловые системы, используя `mkfs`, в соответствии со следующей таблицей:

раздел	размер	файловая система
<code>sdb1</code>	1000 MiB	xsf
<code>sdb2</code>	1000 MiB	btrfs
<code>sdb3</code>	100 MiB	ext4
<code>sdb4</code>	100 MiB	ext4
<code>sdb5</code>	800 MiB	fat32

- Проверьте, что созданный диск использует таблицу разделов GPT. Для этого выполните `sudo parted -l`. В выводе должна присутствовать строка:

```
Disklabel type: gpt
```

- Проверьте, что все разделы были созданы вместе с соответствующими файловыми системами, используя команду `lsblk -f`.
- Примонтируйте эти разделы к директориям `/mnt/01`, `/mnt/02`, ... `/mnt/05`.

## Задача 6. Программа dd

### (a) Файл из нулевых байт

Создайте файл из нулевых байт размером в 1 кибибайт, используя программу `dd` и псевдоустройство `/dev/zero`. Используйте программу `xxd` или `hexdump -C`, чтобы просмотреть все байты созданного файла.

### (b) Файл из случайных байт

Создайте файл из случайных байт размером в 500 байт, используя программу `dd` и псевдоустройство `/dev/random`. Используйте `xxd`, чтобы просмотреть все байты созданного файла.

(c) **Обнуление части файла**

В созданном файле из предыдущей подзадачи обнулите байты с 100-го до 200-го используя `dd` и `/dev/zero`. Просмотрите байты изменённого файла. Используйте опцию `conv=notrunc`, иначе `dd` будет обрезать файл после последнего записанного байта.

(d) **Патч**

В файле из предыдущей задачи задайте байты с индексами от 50 до 55 значениями A1 B2 C3 D4 E5 F6. Остальные байты файла измениться не должны, размер файла должен оставаться прежним. Используйте `printf` и `dd`. Просмотрите байты изменённого файла.

(e) **Текстовый файл**

Создайте файл `a.txt`, содержащий фразу `Sapere Aude`.

- Используйте `dd`, чтобы вывести содержимое этого файла на экран (`stdout`).
- Используйте `dd`, чтобы вывести первые 6 символов файла на экран.
- Используйте `dd`, чтобы вывести символы начиная с 8-го и до конца файла на экран.
- Используйте `dd`, чтобы скопировать файл `a.txt` в файл `b.txt`.
- Используйте `dd`, чтобы скопировать файл `a.txt` в файл `c.txt`, переведя все строчные буквы в прописные.

(f) **Байты диска**

Используйте `dd` и `xxd`, чтобы просмотреть первые 1000 байт диска `/dev/sdb`.

(g) **Копирование файловой системы** *Будьте осторожны при указании раздела в этой подзадаче! Если вы укажите системный раздел, то система сломается.*

В файловой системе на разделе `sdb3` создайте 100 файлов с именами `file000`, `file001` ... `file100`. Размонтируйте файловые системы на разделах `sdb3` и `sdb4`. Скопируйте содержимое файловой системы на разделе `sdb3` в раздел `sdb4`, используя `dd`. Используйте значение `bs` как минимум 1 МиБ для ускорения копирования. Используйте опцию `status=progress` для отслеживания хода выполнения. Примонтируйте файловые системы обратно и проверьте содержимое раздела `sdb4`.

## Задача 7. Блоки хранения данных

Дайте определения следующим понятиям, указав для каждого типичные размеры.

- Сектор (sector) – в контексте жёстких дисков (HDD).
- Страница (page) – в контексте твердотельных накопителей (SSD).
- Блок (block) – в контексте твердотельных накопителей (SSD).
- Блок (block) – в контексте файловых систем (например, ext4).
- Кластер (cluster) – в контексте файловых систем FAT и NTFS.
- Страница (page) – в контексте оперативной и виртуальной памяти.

Для решения этой задачи создайте файл `07.txt` и напишите в этом файле все определения.

## Задача 8. Задача на ссылки

- Создайте новый файл `file.txt`, содержащий строку "I am file".
- Используйте `ls -li` чтобы посмотреть на значение номера `inode` файла. Также обратите внимание на число 1, следующее после прав. Это количество жёстких ссылок на этот файл.
- Используйте программу `stat`, чтобы посмотреть метаинформацию о заданном файле. Обратите внимание на поля `Inode` и `Links`.
- Создайте жёсткую ссылку (*hard link*) на `file.txt` по имени `hl.txt`.
- Создайте мягкую ссылку (*soft link*) на `file.txt` по имени `sl.txt`. Мягкая ссылка также называется как символьическая ссылка или симлинк.

- (f) Используйте `ls -li`, чтобы посмотреть значения номеров inode для файла и двух ссылок. Также обратите на значение количества жёстких ссылок. Чему равны эти значения?
- (g) Используйте `stat` что посмотреть номера inode и количество жёстких ссылок для начального файла и двух ссылок.
- (h) Допишите в файл строку "From hard link", используя жёсткую ссылку `hl.txt`.

```
$ echo "From hard link" >> hl.txt
```

Что содержит файл после этой операции?

- (i) Допишите в файл строку "From soft link", используя мягкую ссылку `sl.txt`. Что содержит файл после этой операции?
- (j) Измените права жёсткой ссылки на 111.

```
$ chmod 111 hl.txt
```

Как изменились права начального файла и ссылок после этого?

- (k) Измените права мягкой ссылки на 222.

```
$ chmod 222 sl.txt
```

Как изменились права начального файла и ссылок после этого?

- (l) Установите права файла, мягкой и жёсткой ссылки на 777.
- (m) Удалите изначальный файл `file.txt`.
- (n) Можно ли теперь прочитать файл по жёсткой ссылке?
- (o) Можно ли теперь прочитать файл по мягкой ссылке?
- (p) Создайте новый файл с тем же именем `file.txt`, содержащий "Second file".
- (q) Что будет, если напечатать файл через жёсткую ссылку `hl.txt`?
- (r) Что будет, если напечатать файл через мягкую ссылку `sl.txt`?

## Задача 9. Поиск ссылок

- (a) Используйте команду `find` чтобы найти в системе все обычные файлы, на которых существует хотя бы 2 жёстких ссылки.
- (b) Выберете один из файлов из предыдущей подзадачи и найдите все жёсткие ссылки для этого файла.
- (c) Примените команду `diff` для двух жёстких ссылок на один файл, чтобы убедиться, что эти файлы побайтово совпадают.
- (d) Используйте команды `xxd` и `head`, чтобы посмотреть первые байты этих файлов и также убедиться, что они совпадают.
- (e) Используйте команду `find`, чтобы найти все мягкие ссылки в системе, вызовите команду `ls -l` для каждой из этих ссылок, чтобы понять куда они указывают.
- (f) Просмотрите директорию `/usr/bin`, используя `ls`, чтобы посмотреть какие файлы в этой директории являются мягкими ссылками.
- (g) Создайте символьическую ссылку `/usr/bin/dog` на файл `/usr/bin/cat`. Директория `/usr/bin` является системной директорией, в которой происходит поиск исполняемых файлов. Если туда поместить исполняемый файл или ссылку на него, то его можно будет вызывать без прописывания пути. Используйте новую команду `dog` для просмотра файлов.

## Задача 10. Свойства файловых систем

### (a) Ссылки в разных файловых системах

- Зайдите в директорию `/mnt/01`, в которой содержатся файлы ФС xfs.
- Создайте файл `a.txt` с содержимым "File from xfs".
- В этой же директории создайте жёсткую и мягкую ссылку на этот файл. Работают ли эти ссылки?
- Перейдите в директорию `/mnt/02`, в которой содержатся файлы файловой системы btrfs.
- Создайте в директории `/mnt/02` жёсткую и мягкую ссылку на `/mnt/01/a.txt`. Работают ли ссылки?

### (b) FAT

- Зайдите в директорию `/mnt/05`, в которой содержатся файлы ФС FAT32 и создайте файл `b.txt`.
- Можно ли создать в этой директории жёсткую и мягкую ссылку на файл `b.txt`?
- Просмотрите права файла `b.txt`. Попробуйте изменить права файлов, используя `sudo chmod`. Получится ли у вас это сделать и, если нет, то почему?
- Просмотрите владельца и группу владельца файлов. Попробуйте их изменить, используя команды `sudo chown` и `sudo chgrp`. Получится ли у вас это сделать и, если нет, то почему?

### (c) Переполнение таблицы inode-ов в файловой системе ext4

- Зайдите в директорию `/mnt/03`, в которой содержатся файлы ФС ext4 размером 100 МиБ.
- Просмотрите и запомните количество свободного места в этой файловой системе, используя `df -h`.
- Просмотрите и запомните количество свободных inode в этой файловой системе, используя `df -i`.
- Попытайтесь создать как можно больше пустых файлов на этой файловой системе, например 50000 файлов с именами `file00000`, `file00001` и т. д. Создавайте пустые файлы в этом разделе пока команда не выдаст ошибку "No space left on device". Используйте команду `touch` вместе с brace expansion.
- Просмотрите количество свободного места в этой файловой системе, используя `df -h`.
- Просмотрите количество свободных inode в этой файловой системе, используя `df -i`.
- Сравните значение свободного места и количества свободных inode с тем, что было до создания файлов.
- Используйте `du -sh`, чтобы найти размер папки `/mnt/03` и убедиться, что он меньше 100 МиБ.

## Задача 11. Какая информация хранится в inode?

В файловой системе ext4 inode каждого файла хранится в специальной таблице inode-ов. Обычно inode занимает 256 байт и содержит множество полей, также называемых метаданными. Напишите, какие поля содержатся в inode для разных типов файлов.

- Поля inode общие для всех типов файлов.
- Поля inode, характерные для обычных файлов. inode для обычных файлов содержит указатели на блоки данных. Сколько указателей на блоки данных содержит такой inode? Что хранится в этих блоках данных?
- Поля inode, характерные для директорий. inode для файлов-директорий содержит указатели на блоки данных. Что и в каком формате хранится в этих блоках?
- Поля inode, характерные для мягких ссылок. inode для таких файлов может содержать указатели на блоки данных. В каких случаях мягкая ссылка может использовать блоки данных и что хранится в этих блоках?
- Поля inode, характерные для файлов блочного и символьных устройств.

Для решения этой задачи создайте файл `11.txt` с ответами на вопросы.

## Задача 12. Расширение раздела и файловой системы

После выполнения предыдущих задач, раздел `sdb3` должен быть полностью заполнен пустыми файлами. Создать новый файл без удаления существующих на этом разделе не получится. Используйте `parted`, чтобы удалить раздел `sdb4`, следующий после раздела `sdb3`. Расширьте раздел `sdb3` и соответствующую файловую систему. Теперь в эту файловую систему, помимо блоков данных, добавились inode-таблицы. Создайте новый файл в этой файловой системе. Посмотрите на процент свободных inode в расширенной файловой системе.