

Просмотр таблицы разделов

Чтобы просмотреть какие диски имеются на системе, выполните:

```
# parted -l
```

Или:

```
# parted -s /dev/sda print
```

Где, /dev/sda — это номер вашего диска.

PS: если не установлена данная утилита, можно заюзать:

```
# fdisk -l
```

Нужно нанести разметку на диск (если ее еще нет — если диск чистый).

Создание разметки на диске

Разметка бывает нескольких типов:

- MSDOS
- GPT
- Про остальные не знаю, не было необходимости их использовать.

Чтобы нанести разметку, можно использовать 2 варианта:

- Если знаете номер диска, то можно использовать:

```
# parted /dev/sda
```

Если не знаете еще номер диска, то можно использовать:

```
# parted
```

```
(parted) print
```

```
(parted) select /dev/sda
```

Т.к у меня /dev/sda — это чистый диск, то на нем необходимо создать разметку. Команда «print» выводит разметку дисков.

Создание разметки

Чтобы создать разметку gpt, используем:

Подключаемся к нужному диску:

```
# parted /dev/sda
```

И выполняем:

```
(parted) mklabel gpt
```

Разметка с gpt уже создана, нужно разбить диск.

Чтобы создать разметку msdos, используем:

Подключаемся к нужному диску:

```
# parted /dev/sda
```

И выполняем:

```
(parted) mklabel msdos
```

Разметка с dos уже создана, нужно разбить диск.

Работа с разделами (создание, удаление, изменение)

Прежде чем создать раздел, загрузитесь в режиме восстановления (или размонтируйте все разделы и отключите пространства подкачки, расположенные на этом устройстве).

Запустите parted как показано ниже, где /dev/sda — устройство, на котором вы хотите создать раздел:

```
# parted /dev/sda
```

Просмотрите текущую таблицу разделов, чтобы определить достаточно ли свободного места:

```
(parted) print
```

Видим что нет ничего (ну предположим это).

Чтобы создать раздел, используйте команду вида:

```
(parted) mkpart part-type fs-type start end
```

Где,

- part-type — Данный аргумент принимает значение: primary (первичный), extended (расширенный) или logical (логический).
- fs-type — Данный аргумент принимает значение: ext2, ext3, ext4 и другие чтобы задать тип ОС.
- start- Данный аргумент принимает значение которое задает начало позиции (создаваемого раздела) относительно 0-го байта. Можно задавать в байтах, но лучше всего (так проще) указывать — 100M/1G.
- end -Данный аргумент принимает значение которое задает конец позиции (создаваемого раздела) относительно 0-го байта. Можно задавать в байтах, но лучше всего (так проще) указывать — 100M/1G, 100%.

Примеры для BIOS/MBR

Если вы хотите использовать MBR разметку для систем с BIOS, то выполните команду:

```
(parted) mklabel msdos
```

И так, для того чтобы создать 1 раздел и который будет занимать все пространство (и так же, будет использоваться для загрузки ОС), выполните:

```
(parted) mkpart primary ext4 2M 100%
```

```
(parted) set 1 boot on
```

Или:

```
(parted) mkpartf primary ext4 2% 100%
```

```
(parted) set 1 boot on
```

PS: НИКОГДА НЕ ПРОБУЙТЕ ПОСТАВИТЬ НАЧАЛО СЕКТОРА В 0 (0%) — попросту потраченное время!

Можно создать корневой раздел, предположим на 15Gb но вынести домашнюю директорию (/home) на отдельный раздел, используйте:

```
(parted) mkpart primary ext4 1M 15G
```

```
(parted) set 1 boot on
```

```
(parted) mkpart primary ext4 15G 100%
```

Можно создать /boot раздел, которому выделить 100Mb для загрузки загрузчика, так же — создать swap, который будет 2 Гб. Так же, под /root раздел, я выделил 8Гб а все остальное — будет для /home раздела:

```
(parted) mkpart primary ext4 1M 100M
```

```
(parted) set 1 boot on
```

```
(parted) mkpart primary linux-swarp 100M 2.1G
```

```
(parted) mkpart primary ext4 2.1G 8.1G
```

```
(parted) mkpart primary ext4 8.1G 100%
```

Не очень сложно, правда?

Примеры для UEFI/GPT

Если вы хотите использовать GPT разметку для систем с UEFI, то выполните команду:

```
(parted) mklabel gpt
```

PS: Если вы используете UEFI/GPT для 2-й (допустим, уже установлена Windows ОС), то запомните — ЧТО НЕЛЬЗЯ УДАЛЯТЬ УЖЕ ИМЕЮЩИЙСЯ UEFI РАЗДЕЛ!. Иначе, больше не запустите Windows.

И так, для того чтобы создать 1 раздел и который будет занимать 200Mb, выполните:

```
(parted) mkpart ESP fat32 1M 200M
```

```
(parted) set 1 boot on
```

ИЛИ:

```
(parted) mkpart ESP fat32 2M 200M
```

```
(parted) set 1 boot on
```

А для остального, я выделю еще 1 раздел который заполню всем оставшимся местом:

```
(parted) mkpart primary ext4 200M 100%
```

Допустим, нужно создать было не все место, а только его часть. Например, под сам корень, я выделю 50.5Гб, а для домашней папки — остальное имеющееся:

```
(parted) mkpart ESP fat32 1M 200M
```

```
(parted) set 1 boot on
```

```
(parted) mkpart primary ext4 200M 50.5G
```

```
(parted) mkpart primary ext4 50.5G 100%
```

Иногда, нужно создать swap, а выполнить это можно:

```
(parted) mkpart ESP fat32 1M 200M
```

```
(parted) set 1 boot on
```

```
(parted) mkpart primary ext4 200M 50.5G
```

```
(parted) mkpart primary linux-swap 50.5G 8.5G
```

```
(parted) mkpart primary ext4 58.5G 100%
```

В последнем примере, я создал 3 раздела (не включая раздела с загрузкой).

Удаление разделов

Запустите parted как показано ниже, где /dev/sda — устройство, на котором вы хотите создать раздел:

```
# parted /dev/sda
```

Просмотрите текущую таблицу разделов, чтобы определить достаточно ли свободного места:

```
(parted) print
```

Чтобы удалить определенный раздел, используйте:

```
(parted) rm device_number
```

Где, device_number — номер раздела.

Например:

```
(parted) rm 1
```

Изменение разделов

Запустите parted как показано ниже, где /dev/sda — устройство, на котором вы хотите создать раздел:

```
# parted /dev/sda
```

Чтобы удалить определенный раздел, используйте:

```
(parted) resizepart device_number
```

Где, device_number — номер раздела.

Например:

```
(parted) resizepart 1
```

Восстановление разделов в Unix/Linux

Parted поддерживает утилиту «rescue», которая поможет вам восстановить потерянный раздел между начальной и конечной точками. Если раздел найден в этом диапазоне, он попытается восстановить его. Например:

```
(parted) rescue
```

Изменить флаг раздела Unix/Linux

Используя parted, вы можете изменять поддерживаемые флаги для дисковых разделов.

Поддерживаемые флаги:

- boot
- root
- swap
- hidden
- raid
- lvm
- lba
- legacy_boot
- rst
- esp
- palo

Чтобы изменить флаг, используйте:

```
(parted) set 2 lba on
```

Данная команда установит «lba» флаг на включение. Чтобы выключить, используйте:

```
(parted) set 2 lba off
```

