В этой статье мы разберем как выполнять основные операции, которые мы привыкли делать в Gparted, такие как разметка диска, удаление и создание раздела, в терминале, а поможет в этом команда parted Linux. Сначала рассмотрим саму команду parted.

Содержание статьи

* [Синтаксис parted](https://losst.pro/razmetka-diskov-v-linux#sintaksis-parted)
* [Опции и команды parted](https://losst.pro/razmetka-diskov-v-linux#opcii-i-komandy-parted)
* [Разметка диска в Linux](https://losst.pro/razmetka-diskov-v-linux#razmetka-diska-v-linux)
* [Выводы](https://losst.pro/razmetka-diskov-v-linux#vyvody)

**Синтаксис parted**

Синтаксис parted выглядит следующим образом:

**$ parted [опции] устройство [команда]**

В квадратных скобках указанны необязательные аргументы.

* **опции** указывают как будет работать программа, например, интерактивный режим работы или скриптовый, или нужно просто вывести справку.
* **устройство** - жесткий диск с которым будет работать программа.
* **команда** - действие которое нужно выполнить над жестким диском.

**Опции и команды parted**

У parted всего несколько опций и огромное количество команд.

Вот опции:

* **-h** - вывести справку о программе
* **-v** - вывести версию программы
* **-s** - скриптовый режим, программа работает без запросов к пользователю
* **-i** - интерактивный режим, пользователь вводит команды в интерактивном режиме. По умолчанию программа работает в этом режиме.

Команд не очень много, но даже помнить их все вовсе не обязательно. Рассмотрим только те, которые нам пригодятся в сегодняшних примерах:

* **help команда** - помощь по выбранной команде;
* **mkpart тип раздела файловая система начало конец** - создание раздела linux с файловой системой начиная с позиции начало заканчивая конец, два последних параметра задаются в мегабайтах по умолчанию;
* **mktable тип** - создать таблицу разделов;
* **print** - отобразить таблицу разделов;
* **quit** - выйти;
* **resizepart раздел конец** - изменить размер раздела;
* **rm раздел** - удалить раздел;
* **select раздел** - установить раздел как текущий;
* **set раздел флаг состояние** - установить флаг для раздела. Состояние может быть on (включен) или off(выключен).

Раньше поддерживались также опции работы с файловой системой mkfs, check, mkpartfs но начиная с версии 2.4 они были удаленны. Но программа не стала хуже.

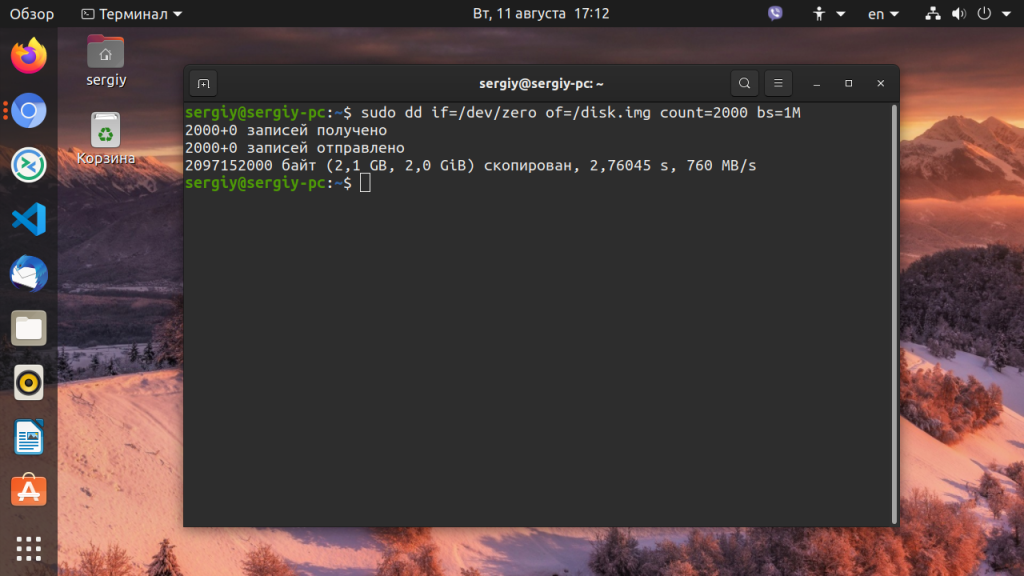
Я понимаю что сейчас разметка дисков в linux для вас непонятна и расплывчата, но с примерами станет намного яснее. Больше не будем о теории, перейдем к примерам.

**Разметка диска в Linux**

Во всем, что вы изучаете нужна практика и работа с жесткими дисками linux не исключение. Но практиковаться на реальном диске, рискуя повредить файловую систему не совсем правильно, а создавать виртуальную машину для таких целей слишком непрактично. Но есть решение. В Linux все считается файлами, и жесткий диск в том числе представлен в виде файла. Из этого выплывает интересная возможность. Мы можем использовать обычный файл вместо жесткого диска для своих экспериментов.

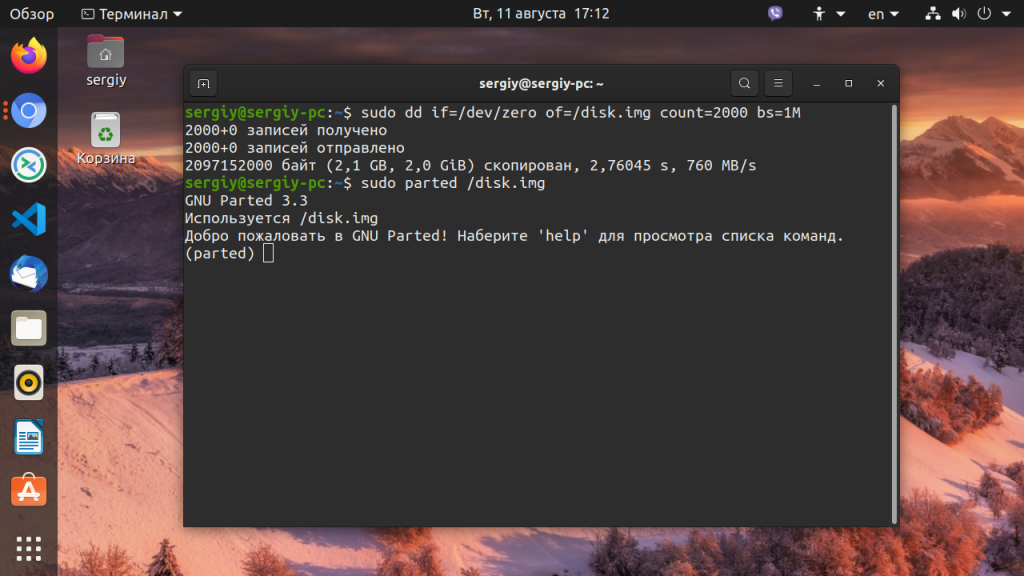
Создадим себе тестовый полигон командой:

sudo dd if=/dev/zero of=/disk.img count=2000 bs=1M

[](https://losst.pro/wp-content/uploads/2016/04/Snimok-ekrana-ot-2020-08-11-17-12-38.png)

Таким образом, мы получим файл размером 2000 мегабайт, заполненный нулями, с которым можно спокойно работать. Запустим parted, передав ему вместо устройства наш файл:

sudo parted /disk.img

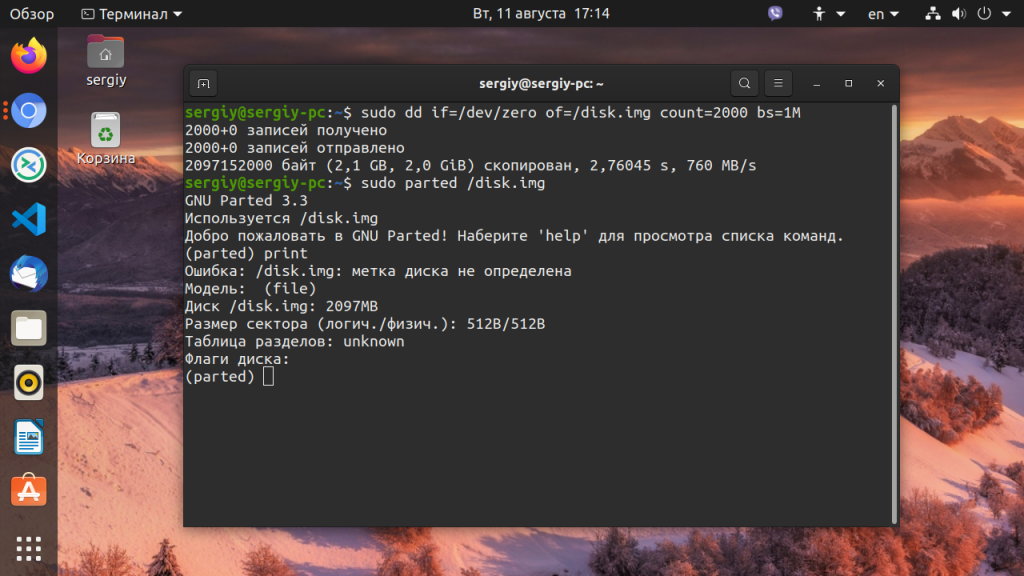
[](https://losst.pro/wp-content/uploads/2016/04/Snimok-ekrana-ot-2020-08-11-17-12-50.png)

Когда надо будет работать с реальным диском, просто передайте путь к файлу его устройства утилите:

sudo parted /dev/sda

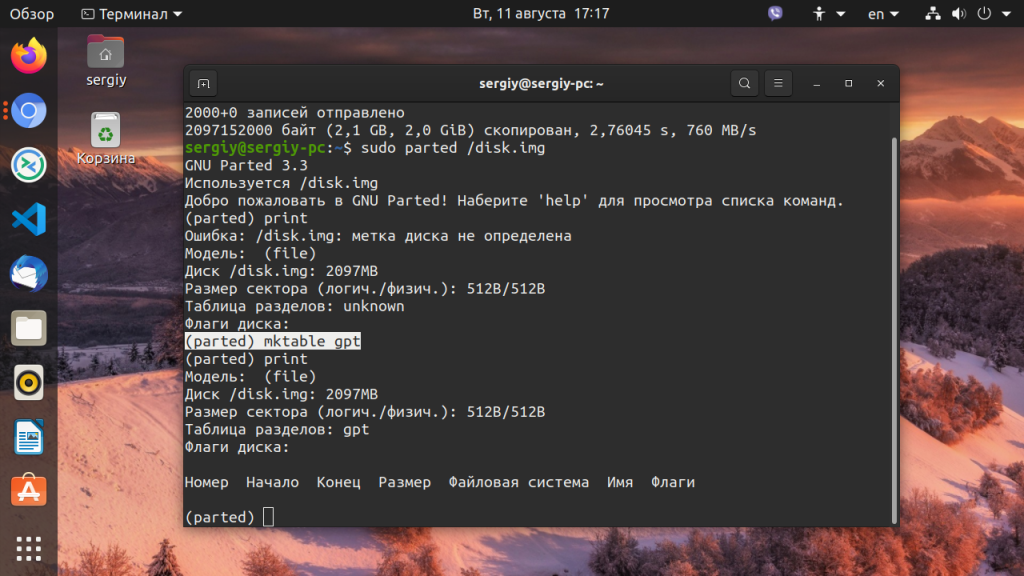
Программа запустится в интерактивном режиме и вы сможете выполнять нужные команды. Сначала попытаемся посмотреть список разделов на устройстве:

(parted) print

[](https://losst.pro/wp-content/uploads/2016/04/Snimok-ekrana-ot-2020-08-11-17-14-53.png)

Он пуст потому что даже нет таблицы разделов. Пока её нет, разметка дисков в Linux не может быть выполнена, нам предстоит ее создать. Для этого воспользуйтесь командой mktable:

(parted) mktable gpt

[](https://losst.pro/wp-content/uploads/2016/04/Snimok-ekrana-ot-2020-08-11-17-17-39.png)

Будем мы создали таблицу разделов типа gpt, но можно выбрать одну из этих: aix, amiga, bsd, dvh, gpt\_sync\_mbr, gpt, mac, msdos, pc98, sun, loop

Теперь давайте создадим новый раздел с файловой системой ext2 размером 100 мегабайт. Для этого есть команда mkpart, как вы помните ей нужно передать три параметра: тип раздела, файловую систему и координаты. Тип раздела может быть:  primary, logical, extended (первичный, логический и расширенный).

В gpt можно создавать нужное количество первичных разделов и не задумываться об их типе. Это все было создано для MBR, в этой таблице есть ограничение на количество первичных разделов - только четыре. Поэтому приходилось создавать три первичных и один расширенный, в который вкладывались уже сколько нужно логических.

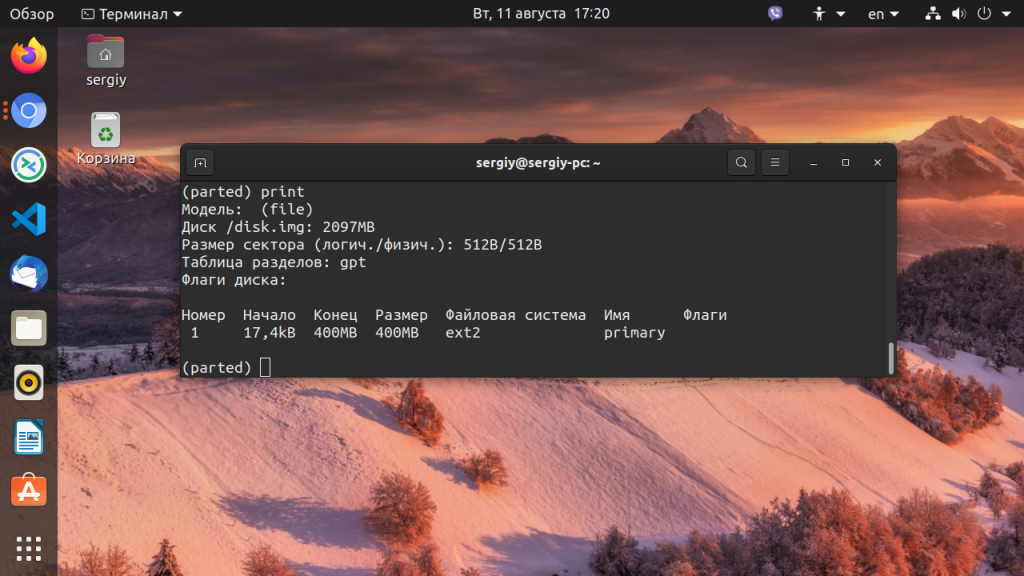
Ну так вот, создаем первичный:

(parted) mkpart primary ext2 0 400М

Поскольку это первый раздел, начинаем мы с нуля, а заканчиваем нужным нам размером - 400 мегабайт.

Смотрим список разделов еще раз:

(parted) print

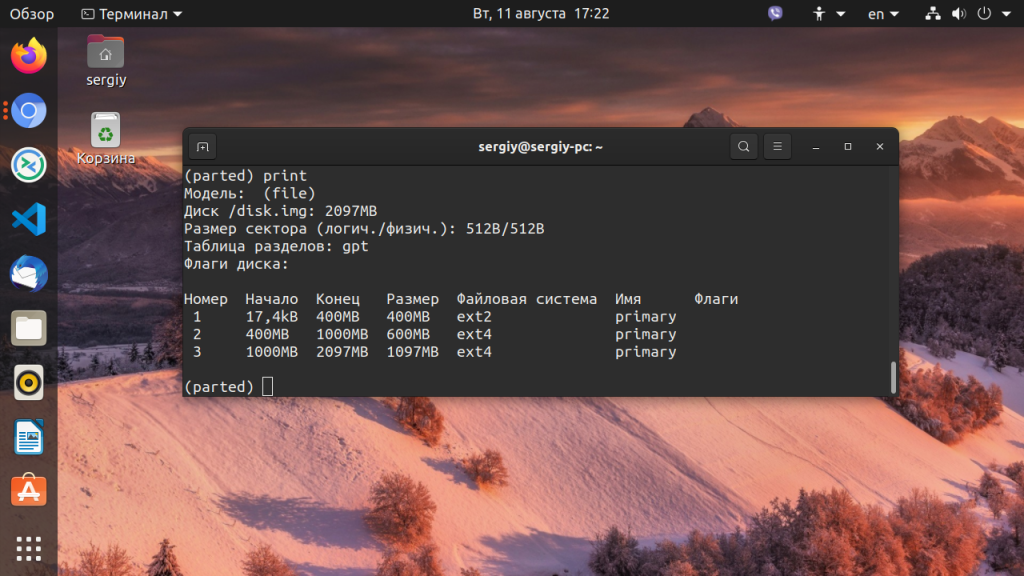
[](https://losst.pro/wp-content/uploads/2016/04/Snimok-ekrana-ot-2020-08-11-17-20-09.png)

Создадим еще несколько разделов, например для корня системы и для домашней папки:

(parted) mkpart primary ext4 400 1000M

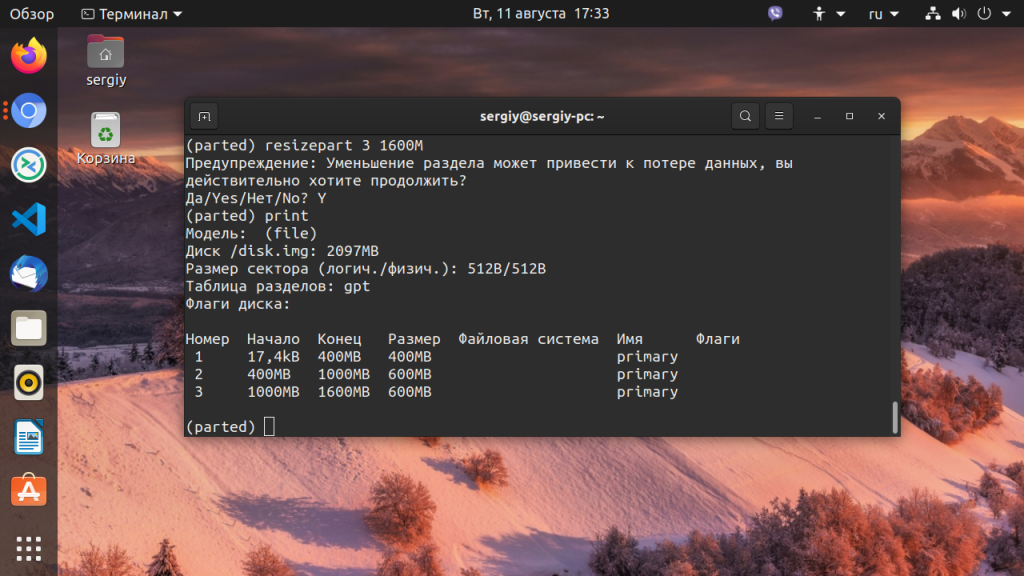
(parted) mkpart primary ext4 1000M -0M

Параметр **-0** означает вести отсчет от конца раздела, таким образом будет занято все доступное пространство под третий раздел. Смотрим что получилось:

[](https://losst.pro/wp-content/uploads/2016/04/Snimok-ekrana-ot-2020-08-11-17-22-19.png)

Работа с диском в Linux осуществляется по их номерам. Давайте уменьшим последний раздел и создадим после него ещё один:

(parted) resizepart 3 1600M

[](https://losst.pro/wp-content/uploads/2016/04/Snimok-ekrana-ot-2020-08-11-17-33-01.png)

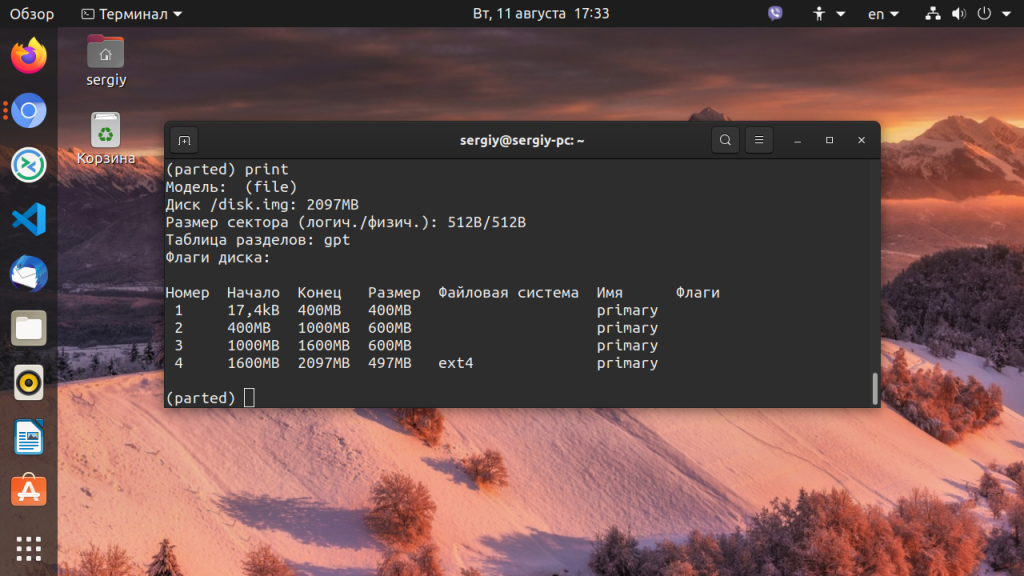
Также можно указать нужный размер в процентах:

(parted) resizepart 3 60%

А теперь создание раздела parted:

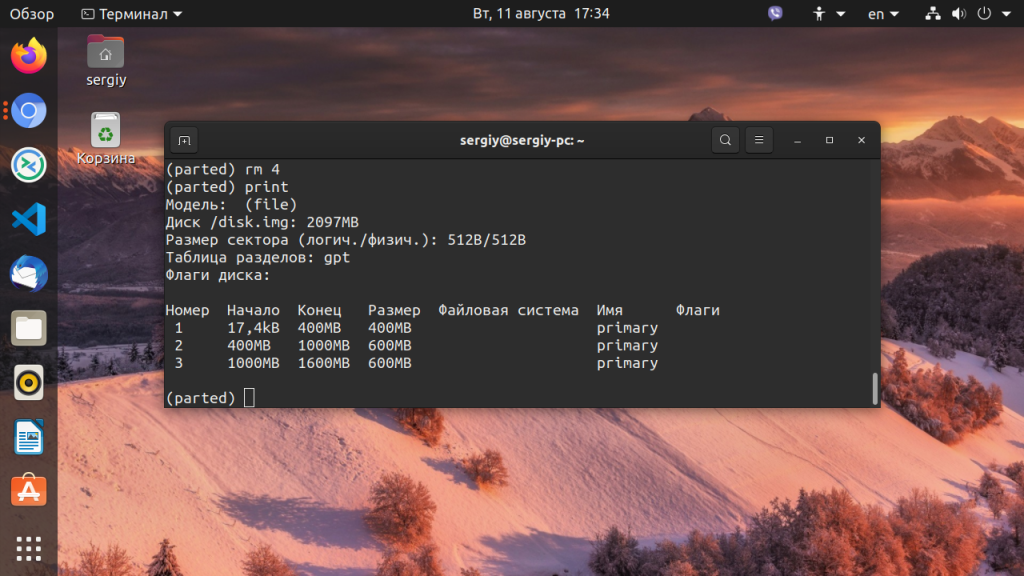
mkpart primary ext4 1600M -0M

Смотрим:

[](https://losst.pro/wp-content/uploads/2016/04/Snimok-ekrana-ot-2020-08-11-17-33-19.png)

Удалим только что созданный четвертый раздел:

rm 4

[](https://losst.pro/wp-content/uploads/2016/04/Snimok-ekrana-ot-2020-08-11-17-34-47.png)

Зададим метки для существующих разделов с помощью команды name:

(parted) name 1 boot

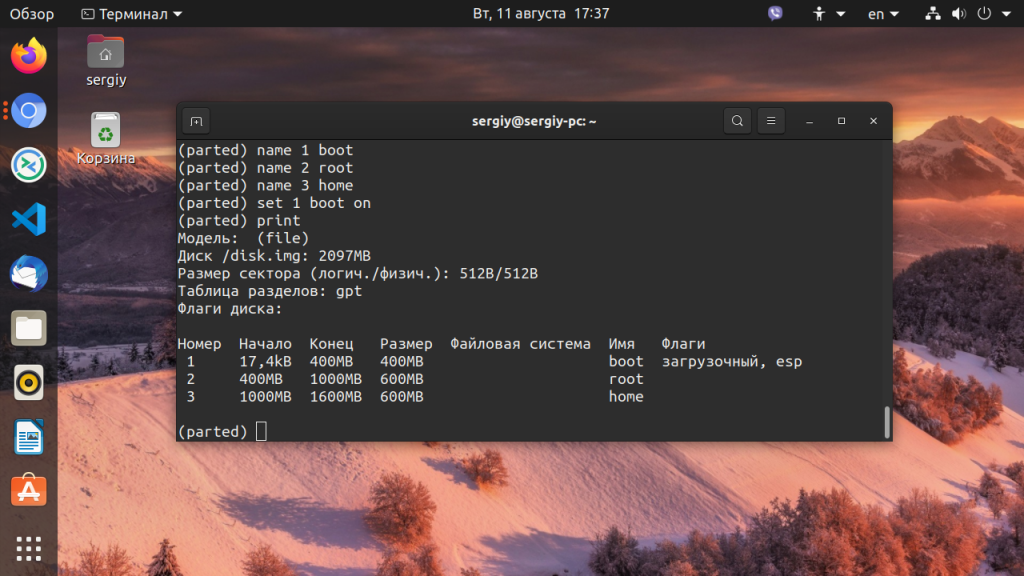
(parted) name 2 root

(parted) name 3 home

Сделаем первый раздел загрузочным командой set:

set 1 boot on

Можно устанавливать следующие флаги: **boot, root, swap, hidden, raid, lvm, lba, hp-service, palo, prep, msftres, bios\_grub, atvrecv, diag, legacy\_boot, type, irst**.

[](https://losst.pro/wp-content/uploads/2016/04/Snimok-ekrana-ot-2020-08-11-17-37-14.png)

Вот в принципе и все. Разбивка диска на разделы Linux завершена. Работу с файловыми системами мы в этой статье опустим, так как для этого используются уже другие команды, которые мы рассмотрим в одной из следующих статей.

**Выводы**

Теперь вы знаете как пользоваться parted Linux и этой информации достаточно чтобы не потеряться в случае, если окажетесь без доступа к графическому интерфейсу и вам понадобится разбивка жесткого диска в консоли Linux.