

Установка BlackArch, часть вторая: Установка Arch

Updated on 20.01.2016 By [Alexey](#) [Leave a comment](#)

[Инструкция составлена из трёх источников:](#)

- <http://learnaholic.me/2013/11/10/archlinux-virtualbox-install-notes/>
- <http://www.tecmint.com/arch-linux-installation-guide/>
- <http://jorisvandijk.com/2014/installing-arch-on-a-uefi-gpt-system/>

Здесь будет показана установка с использованием интерфейса EFI вместо GRUB. Это связано с тем, что в некоторых ситуациях возникает критическая ошибка при загрузке Arch (BlackArch) в VirtualBox при использовании GRUB. Поэтому если вы устанавливаете в VirtualBox, то убедитесь, что у вас установлена галочка для опции «Включить EFI (только специальные ОС)».

Загрузка с диска с EFI занимает некоторое время (до полутора минут), при этом на экране ничего не отображается - не беспокойтесь, просто дождитесь завершения.

Важно: это уже упоминалось, но подчеркну ещё раз - без Интернета ничего не установится, у вас обязательно должно быть подключение.

Разметка диска

Скачайте LIVE-образ Arch (только с официального сайта!): <https://www.archlinux.org/download/>

Загрузитесь с него, выберите вариант, соответствующий битности вашего процессора:



Arch Linux
Boot Arch Linux (x86_64)
Boot Arch Linux (i686)
Boot existing OS
Run Memtest86+ (RAM test)
Hardware Information (HDT)
Reboot
Power Off

Press [Tab] to edit options

Boot the Arch Linux (x86_64) live medium.
It allows you to install Arch Linux or perform system maintenance.

[Логин и пароль вводить не нужно, поскольку будет произведён автоматический вход:](#)

```
Arch Linux 4.3.3-2-ARCH (tty1)
archiso login: root (automatic login)
root@archiso ~ # _
```

[Проверим, какие диски имеются на компьютере:](#)

1 fdisk -l

```
root@archiso ~ # fdisk -l
Disk /dev/sda: 100 GiB, 107374182400 bytes, 209715200 sectors
Units: sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes

Disk /dev/loop0: 300.3 MiB, 314888192 bytes, 615016 sectors
Units: sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
root@archiso ~ # _
```

В моём случае один диск (другой - это образ CD) под названием **/dev/sda**

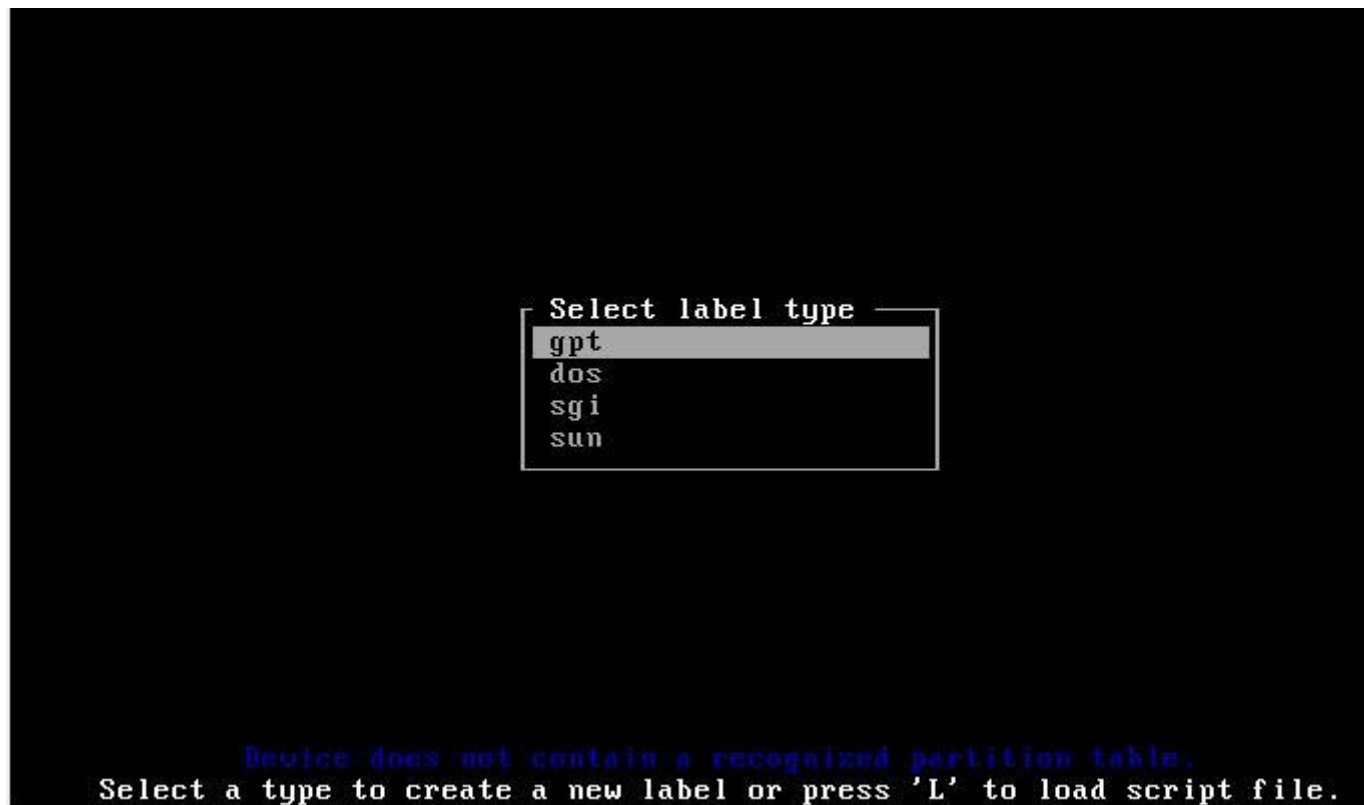
Нам нужно разбить этот диск на разделы. Могут быть варианты (сколько разделов и прочее, делать ли файл подкачки, выделять ли отдельный раздел для загрузчика), я предлагаю сделать следующие области:

- Загрузочная область (/dev/sda1) размером 200 мегабайт, отформатированная в fat32.
- Раздел подкачки (/dev/sda2) размером 4 гигабайта, Swap On.
- Корневой раздел (/dev/sda3) размером на весь оставшийся диск отформатированный в ext4.

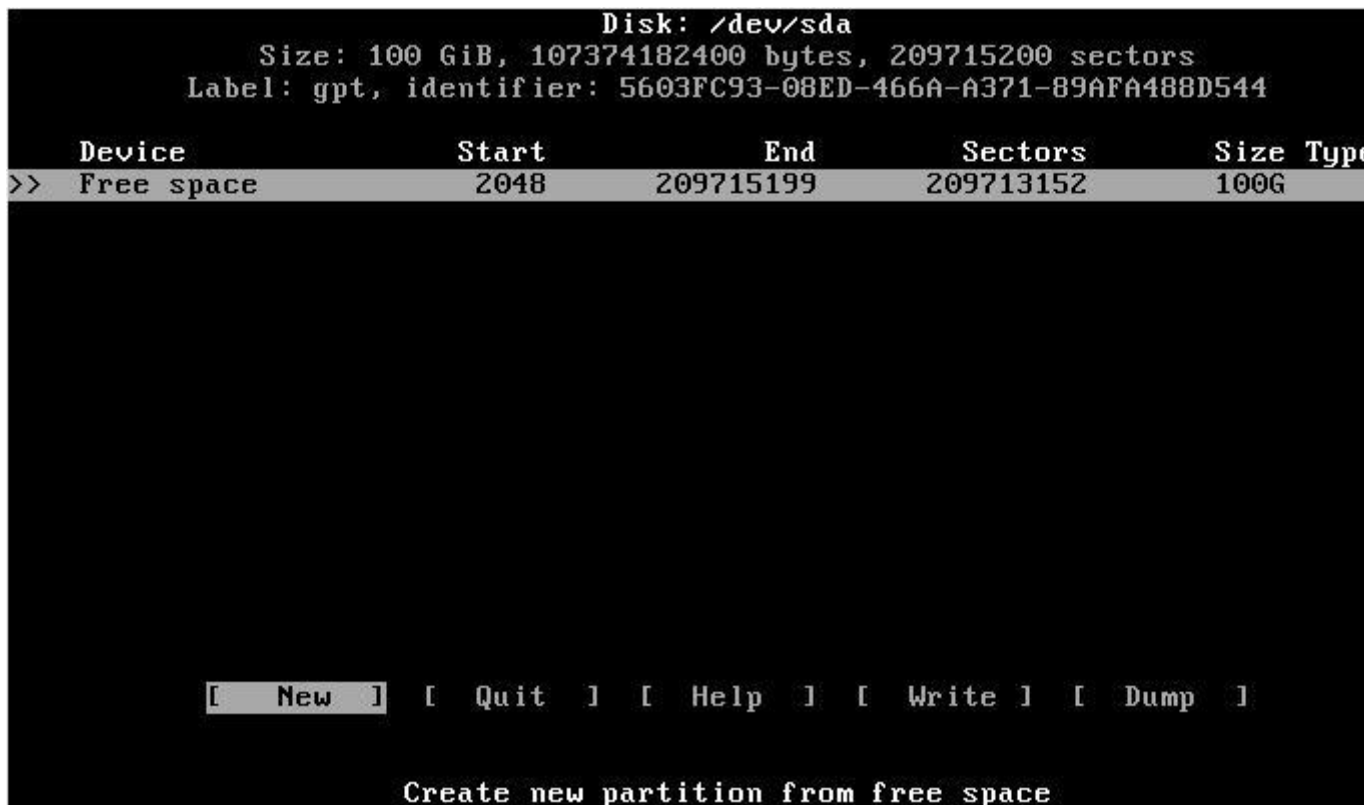
Это можно сделать разными программами, я для себя выбираю **cgdisk**

1 cgdisk /dev/sda


Выбираем **gpt**:



Теперь нажимаем **New**:



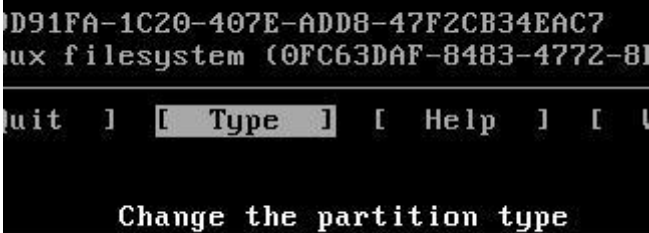
Устанавливаем размер:



```
Partition size: 200M_
```

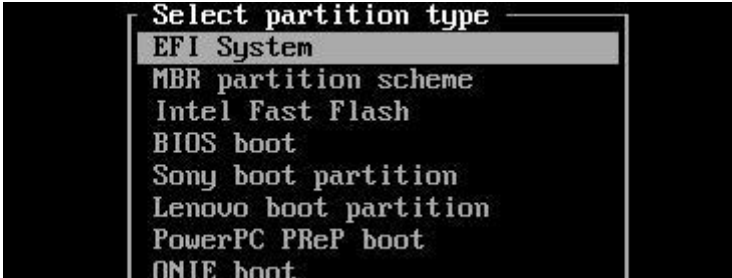
May be followed by M for MiB.

Указываем тип:



```
0D91FA-1C20-407E-ADD8-47F2CB34EAC7  
Linux filesystem (0FC63DAF-8483-4772-81  
quit ] [ Type ] [ Help ] [ U  
  
Change the partition type
```

Нам нужно выбрать **EFI System**:



```
Select partition type  
EFI System  
MBR partition scheme  
Intel Fast Flash  
BIOS boot  
Sony boot partition  
Lenovo boot partition  
PowerPC PReP boot  
ONIE boot
```

Переходим к оставшейся свободной области (стрелочка вниз), опять нажимаем **New**, устанавливаем размер **4 гигабайта**, в качестве типа выбираем **Linux swap** (раздел подкачки):



```
HP-UX data  
HP-UX service  
Linux swap  
Linux filesystem  
Linux server data  
↓  
0657FD6D-A4AB-43C4-84E5-0933C84B4F4F
```

Опять переходим к оставшемуся свободному месту, выбираем весь незанятый размер, в качестве типа ставим **Linux root (x86-64)** - или другой, который соответствует архитектуре вашего процессора:

```
Select partition type ———↑
Linux root (x86)
Linux root (ARM)
Linux root (x86-64)
Linux root (ARM-64)
Linux reserved
Linux home
Linux RAID
```

Теперь выбираем **Write**, чтобы записать сделанные изменения:

```
9-A08E0F4CD732
F68BCE3-E8CD-4DB1-96E7-FBCAF984B709
[ Help ] [ Write ] [ Dump ]
k (this might destroy data)
```

Пишем **yes**:

```
Partition UUID: D32FEE3E-0728-459F-B479-A08E0F4CD732
Partition type: Linux root (x86-64) (4F68BCE3-E8CD-4DB1-96E7-FBCAF984B709)
Are you sure you want to write the partition table to disk? yes_
Type "yes" or "no", or press ESC to leave this dialog.
```

В результате должно получиться примерно так:

```
Disk: /dev/sda
Size: 100 GiB, 107374182400 bytes, 209715200 sectors
Label: gpt, identifier: E26C809A-E089-4B7C-8256-5ED
Device      Start      End        Sectors
/dev/sda1   2048       411647    409600
/dev/sda2   411648    8800255   8388608
>> /dev/sda3 8800256   209715166 200914911
```

Покидаем программу.

Можно ещё раз проверить вывод команды, чтобы убедиться, что изменения сохранены:

`fdisk -l`

```

root@archiso ~ # fdisk -l
Disk /dev/sda: 100 GiB, 107374182400 bytes, 209715200 sectors
Units: sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disklabel type: gpt
Disk identifier: E26C809A-E089-4B7C-8256-5ED273458474

Device        Start      End      Sectors  Size Type
/dev/sda1     2048      411647   409600   200M EFI System
/dev/sda2     411648    8800255  8388608   4G Linux swap
/dev/sda3     8800256   209715166 200914911 95.8G Linux root (x86-64)

Disk /dev/loop0: 300.3 MiB, 314888192 bytes, 615016 sectors
Units: sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
root@archiso ~ #

```

[Форматируем наши разделы:](#)

- 1 mkfs.fat -F32 /dev/sda1
- 2 mkfs.ext4 /dev/sda3

```

mke2fs 1.42.13 (17-May-2015)
Creating filesystem with 204800 1k blocks and 51200 inodes
Filesystem UUID: 6236ed69-e8ea-4cc7-803d-016bc7c098ee
Superblock backups stored on blocks:
    8193, 24577, 40961, 57345, 73729

Allocating group tables: done
Writing inode tables: done
Creating journal (4096 blocks): done
Writing superblocks and filesystem accounting information: done

root@archiso ~ # mkfs.ext4 /dev/sda3
mke2fs 1.42.13 (17-May-2015)
Creating filesystem with 25114363 4k blocks and 6283264 inodes
Filesystem UUID: a5391d23-27ab-41ad-bd5a-0bd9838827b9
Superblock backups stored on blocks:
    32768, 98304, 163840, 229376, 294912, 819200, 884736, 1605632, 2654144,
    4096000, 7962624, 11239424, 20480000, 23887872

Allocating group tables: done
Writing inode tables: done
Creating journal (32768 blocks): done
Writing superblocks and filesystem accounting information: done

root@archiso ~ # _

```

[Форматируем и инициализируем раздел SWAP:](#)

- 1 mkswap /dev/sda2

2 swapon /dev/sda2

```
root@archiso ~ # mkswap /dev/sda2
Setting up swapspace version 1, size = 4 GiB (4294963200 bytes)
no label, UUID=f911d910-97c1-4b0d-8487-709f3a28c9aa
root@archiso ~ # swapon /dev/sda2
root@archiso ~ # _
```

[Ещё раз убедимся, что всё именно так, как мы и планировали:](#)

1 lsblk

Установка базовой системы Arch

[Теперь нам нужно смонтировать созданные разделы жёсткого диска. Для этого выполните в следующей последовательности команды:](#)

1 mount /dev/sda3 /mnt
2 mkdir /mnt/boot
3 mount /dev/sda1 /mnt/boot

```
root@archiso ~ # mount /dev/sda3 /mnt
root@archiso ~ # mkdir /mnt/boot
root@archiso ~ # mount /dev/sda1 /mnt/boot
root@archiso ~ # _
```

[Начинаем саму установку](#)

1 pacstrap -i /mnt base base-devel

[Когда будет построен список пакетов, то два раза нажимаем ENTER, чтобы одобрить его, когда будет написан размер скачиваемых данных, а также размер после установки, нажимаем ENTER ещё раз для согласия:](#)


```

root@archiso ~ # pacstrap -i /mnt base base-devel
==> Creating install root at /mnt
==> Installing packages to /mnt
:: Synchronizing package databases...
core                121.9 KiB   1257K/s  00:00 [#####]
extra               1765.9 KiB  181K/s   00:10 [#####]
community           3.2 MiB   196K/s   00:17 [#####]
:: There are 50 members in group base:
:: Repository core
   1) bash      2) bzip2   3) coreutils  4) cryptsetup  5) device-mapper  6) dhcpcd
   7) diffutils 8) e2fsprogs  9) file      10) filesystem  11) findutils  12) gawk
  13) gcc-libs 14) gettext 15) glibc    16) grep       17) gzip       18) inetutils
  19) iproute2 20) iputils  21) jfsutils  22) less       23) licenses   24) linux
  25) logrotate 26) lvm2     27) man-db   28) man-pages  29) mdadm      30) nano
  31) netctl    32) pacman   33) pciutils 34) pcmciautils 35) perl
  36) procps-ng 37) psmisc   38) reiserfsprogs 39) s-nail    40) sed
  41) shadow    42) sysfsutils 43) systemd-sysvcompat 44) tar      45) texinfo
  46) usbutils 47) util-linux 48) vi      49) which     50) xfsprogs

Enter a selection (default=all): _

```

```

tzdata-2015g-1  xz-5.2.2-1  zlib-1.2.8-4  autoconf-2.69-2
automake-1.15-1  bash-4.3.042-4  binutils-2.25.1-3
bison-3.0.4-1   bzip2-1.0.6-5   coreutils-8.24-1
cryptsetup-1.7.0-1  device-mapper-2.02.138-1  dhcpcd-6.10.0-1
diffutils-3.3-2  e2fsprogs-1.42.13-1  fakeroot-1.20.2-1
file-5.25-1     filesystem-2015.09-1  findutils-4.4.2-6
flex-2.6.0-1    gawk-4.1.3-1     gcc-5.3.0-3    gcc-libs-5.3.0-3
gettext-0.19.6-2  glibc-2.22-3     grep-2.22-1    groff-1.22.3-5
gzip-1.6-1      inetutils-1.9.4-2  iproute2-4.1.1-1
iputils-20140519.fad11dc-1  jfsutils-1.1.15-4  less-481-2
libtool-2.4.6-4  licenses-20140629-1  linux-4.3.3-2
logrotate-3.9.1-1  lvm2-2.02.138-1  m4-1.4.17-1    make-4.1-1
man-db-2.7.5-1   man-pages-4.04-1   mdadm-3.3.4-1  nano-2.5.1-1
netctl-1.11-1    pacman-4.2.1-4     patch-2.7.5-1  pciutils-3.4.0-1
pcmciautils-018-7  perl-5.22.1-1     pkg-config-0.29-1
procps-ng-3.3.11-2  psmisc-22.21-3    reiserfsprogs-3.6.24-1
s-nail-14.8.6-1  sed-4.2.2-3       shadow-4.2.1-3  sudo-1.8.15-1
sysfsutils-2.1.0-9  systemd-sysvcompat-228-3  tar-1.28-1
texinfo-6.0-1    usbutils-008-1     util-linux-2.27.1-1
vi-1:070224-2    which-2.21-1      xfsprogs-4.3.0-1

Total Download Size:   219.77 MiB
Total Installed Size:  744.55 MiB

:: Proceed with installation? [Y/n] _

```

[Создаём файл fstab:](#)

```
1      genfstab -U -p /mnt >> /mnt/etc/fstab
```

Настройка системы

[Выполним следующую команду:](#)

```
1 arch-chroot /mnt
```

[Устанавливаем Vim:](#)

```
1 pacman -Sy
2 pacman -S vim
```

[Теперь в файле locale.gen нам нужно раскомментировать две строки, для этого](#)

```
1 vim /etc/locale.gen
```

[Ищем там и раскомментируем строки:](#)

```
1 en_US.UTF-8 UTF-8
2 ru_RU.UTF-8 UTF-8
```

[После этого выполняем:](#)

```
1 locale-gen
```



```
#ro_RU ISO-8859-2
#ru_RU.KOI8-R KOI8-R
ru_RU.UTF-8 UTF-8
"/etc/locale.gen" 484L, 9533C written
sh-4.3# locale-gen
Generating locales...
  en_US.UTF-8... done
  ru_RU.UTF-8... done
Generation complete.
sh-4.3# _
```

[Создадим locale.conf и экспортируем локаль:](#)

```
1 echo LANG=en_US.UTF-8 > /etc/locale.conf
2 export LANG=en_US.UTF-8
```



```
sh-4.3# echo LANG=en_US.UTF-8 > /etc/locale.conf
sh-4.3# export LANG=en_US.UTF-8
sh-4.3# _
```

[Установим Time Zone, связав информацию о временной зоне с localtime:](#)

[Чтобы увидеть все временные зоны:](#)

```
1 ls /usr/share/zoneinfo
```

[Чтобы увидеть подкатегории](#)

```
1 ls /usr/share/zoneinfo/Europe
```

[Теперь, когда вы выбрали город, делаем ссылку:](#)

```
1 ln -s /usr/share/zoneinfo/Europe/Moscow /etc/localtime
```

```
sh-4.3# ln -s /usr/share/zoneinfo/Asia/Bangkok /etc/localtime
sh-4.3# hwclock --systohc --utc
sh-4.3# echo HackWare > /etc/hostname
sh-4.3# systemctl enable dhcp.service
Operation failed: No such file or directory
sh-4.3# systemctl enable dhcpd.service
Created symlink from /etc/systemd/system/multi-user.target.wants/dhcpd.serv
to /usr/lib/systemd/system/dhcpd.service.
sh-4.3# _
```

[Устанавливаем аппаратные часы на UTC:](#)

```
1 hwclock --systohc --utc
```

[Устанавливаем имя хоста:](#)

```
1 echo HackWare > /etc/hostname
```

[Настраиваем сеть:](#)

```
1 systemctl enable dhcpd.service
```

[Устанавливаем пароль рута:](#)

```
1 passwd
```

```
sh-4.3# passwd
Enter new UNIX password:
Retype new UNIX password:
passwd: password updated successfully
sh-4.3# _
```

[Создаём нового пользователя с sudo:](#)

```
1 useradd -m -g users -G wheel,video -s /bin/bash <имя пользователя>
```

Установка sudo с pacman:

```
1 pacman -S sudo
```

Раскомментируйте группу wheel из sudoers %wheel ALL=(ALL), чтобы пользователи сразу после создания могли sudo:

```
1 vim /etc/sudoers
```

Установите пароль для созданного пользователя:

```
1 passwd <имя пользователя>
```

Загрузчик

```
1 bootctl install
```

Редактируем содержимое файла:

```
1 vim /boot/loader/loader.conf
```

Удалите то, что там есть и впишите туда:

```
1 default arch
```

Создайте конфигурационный файл для добавления пункта Arch Linux в менеджер gummiboot

```
1 vim /boot/loader/entries/arch.conf
```

Содержимое файла должно быть следующим:

```
1 BlackArch title
2 /vmlinuz-linux linux
3 /initramfs-linux.img initrd
4 root=/dev/sda3 rw options
```

Выйдем из chroot, размонтируем смонтированные разделы и перезагрузимся:

- 1 `exit`
- 2 `umount -R /mnt/boot`
- 3 `umount -R /mnt`
- 4 `reboot`

Можно вынимать установочный диск.