

Arch install for DELL XPS 9570

Simone Ruffini

March 2020

Read everything while keeping an eye to [Arch installation Guide](#) and [GloriusEggroll](#)

1 Pre-Installation

1.1 Create Bootable Media

Create a bootable device with RUFUS with dd method () or by using dd.

- Windows: RUFUS with dd
- *nix:

```
sudo dd bs=4M if=<path/to/input>.iso of=/dev/sd<?> conv=fdatasync status=progress
```

Boot in arch linux. Remember to disable Secure Boot and RAID intel otherwise nvme devices wont be shown.

1.2 Verify the boot mode

If [UEFI](#) mode is enabled on an [UEFI](#) motherboard, Archiso will [boot](#) Arch Linux accordingly via [systemd-boot](#). To verify this, list the efivars directory:

```
ls /sys/firmware/efi/efivars
```

1.3 Connect to the internet

To set up a network connection, go through the following steps:

Ensure your [network](#) interface is listed and enabled, for example with [ip-link\(8\)](#):

```
ip link
```

[Connect to the wireless LAN](#). Try with wifi-menu of [netctl](#) package:

```
wifi-menu
```

If something goes wrong you can modify the network profile with
`vim /etc/netctl/<profile_name>`

The connection may be verified with [ping](#):

```
ping archlinux.org
```

1.4 Update the system clock

Use `timedatectl(1)` to ensure the system clock is accurate:

```
timedatectl set-ntp true
```

1.5 Partition the disks

When recognized by the live system, disks are assigned to a [block device](#) such as `/dev/sda` or `/dev/nvme0n1`. To identify these devices, use `lsblk` or `fdisk`.

```
lsblk
```

Results ending in `rom`, `loop` or `airoot` may be ignored. The following [partitions](#) are **required** for a chosen device:

- One partition for the root directory `/`.
- If [UEFI](#) is enabled, an [EFI system partition](#).

Using [LVM](#) and with a graphical interface to partition disk use `cdisk`.

```
cdisk /dev/<device>
```

Inside `cdisk`:

1. Make a GPT partition table if not already existing
2. New 512M partition for EFI, change partition type with `EFI System`
3. Remaining of disk size (no option) with default type `Linux filesystem`
4. write and quit

1.5.1 LVM partitions

Visit [LVM ita](#). **In this section the storage device will be sda!**

1. Create physical volume and check with `pvdisk`

```
pvcreate /dev/sda2
```

2. Create Volume Group and check with `vgdisplay`

```
vgcreate VolGrp0 /dev/sda2
```

3. Create Logical Volumes and check with `lvdisplay`

```
lvcreate -L 20G VolGrp0 -n root
lvcreate -C y -L 8G VolGrp0 -n swap
lvcreate -l 100%FREE VolGrp0 -n home
```

1.6 Format the partitions

Once the partitions have been created, each must be formatted with an appropriate [file system](#).

```
mkfs.fat -F 32 /dev/<boot_partition>
mkfs.f2fs -l root /dev/VolGrp00/root
mkfs.f2fs -l home /dev/VolGrp0/home
```

Initialize the swap

```
mkswap /dev/VolGrp0/swap
swapon /dev/VolGrp0/swap
```

The partitioning scheme and mount point of the EFI System Partition (ESP) are tied to the type of bootloader used. In this guide [systemd-boot](#) will be used (look at installation section). The mount point of the ESP partition in Systemd-boot needs to contain kernel and initramfs files. So boot is effectively the ESP.

1.7 Mount the file systems

Non-existent directories must be created first

```
mount /dev/VolGrp0/root /mnt
mkdir /mnt/home
mkdir /mnt/boot
mount /dev/VolGrp0/home /mnt/home
mount /dev/<boot_partition> /mnt/boot
```

2 Installation

2.1 Select mirrors

Packages to be installed must be downloaded from [mirror servers](#), which are defined in `/etc/pacman.d/mirrorlist`. The higher a mirror is placed in the list, the more priority it is given when downloading a package. This file will later be copied to the new system by `pacstrap`, so it is worth getting right.

Make a backup of the mirror list:

```
cp /etc/pacman.d/mirrorlist /etc/pacman.d/mirrorlist.backup
```

Install [pacman-contrib](#) package containing the `rankmirrors` script.

```
pacman -Sy
pacman -S pacman-contrib
```

If `pacman` is not working then probably all servers in the mirror list are commented. Uncomment one by modifying `mirrorlist` and after the install recomment it.

To be sure that all servers are available for ranking run the command to uncomment all the lines in `mirrorlist`:

```
sed -i 's/^#Server/Server/' /etc/pacman.d/mirrorlist.backup
```

Now we will run `mirrors`, it can take a while and no output is made in the process so just wait and monitor on another tty with `top`. The script will rank the first 6 best mirrors deleting the other ones.

```
rankmirrors -n 6 /etc/pacman.d/mirrorlist.backup > /etc/pacman.d/mirrorlist
```

If you want add the other servers for precaution.

2.2 Install essential packages

Use the `pacstrap` script to install the `base` package, Linux `kernel` and firmware for common hardware. The `base` package does not include all tools from the live installation, so installing other packages may be necessary for a fully functional base system. In particular, consider installing:

- userspace utilities for the management of `file systems` that will be used on the system,
- utilities for accessing `RAID` or `LVM` partitions,
- specific firmware for other devices not included in `linux-firmware`,
- software necessary for `networking`,
- a `text editor`,
- packages for accessing documentation in `man` and `info` pages: `man-db`, `man-pages` and `texinfo`.

To `install` other packages or package groups, append the names to the `pacstrap` command above (space separated) or use `pacman` while `chrooted into the new system`. For comparison, packages available in the live system can be found in `packages.x86_64`.

```
pacstrap /mnt base base-devel linux linux-firmware vim man-db man-pages lvm2
```

3 Configure the system

3.1 Generate Fstab

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

Check if there is a entry for every partition and the one for swap too with `vim`.

3.2 Chroot

Now we are going to `chroot` into our newly installed system and begin to configure its booting, time, and language:

```
arch-chroot /mnt
```

3.3 Time zone

Set the `time zone`:

```
ln -sf /usr/share/zoneinfo/Europe/Rome /etc/localtime
```

Run `hwclock(8)` to generate `/etc/adjtime`:

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

3.4 Localization

Uncomment `en_US.UTF-8 UTF-8` and other needed `locale` in `/etc/locale.gen`, and generate them with:

```
locale-gen
```

Create the `locale.conf(5)` file, set the `LANG` variable as your language:

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

3.5 Network configuration

3.5.1 Hostname

Create the `hostname` file:

```
echo DellXPS-Simone > /etc/hostname
```

Add matching entries to `hosts(5)`:

```
vim /etc/hosts
-----
127.0.0.1      localhost
::1           localhost
127.0.1.1      DellXPS-Simone.localdomain    DellXPS-Simone
```

3.5.2 Network Manager

Install network tools and manager:

```
pacman -S NetworkManager
```

Find any network devices starting with `enp/wlp` with:

```
ip link
```

Disable `DHCPD` and `netctl` on any network device since `networkmanager` will replace both.

```
systemctl disable dhcpcd@<eth_device>.service
systemctl disable netctl-auto@<wirl_device>.service
```

Enable `networkmanager`

```
systemctl enable NetworkManager.service
```

3.6 Enable trim support

For safe, weekly [TRIM](#) service on [SSDs](#) and all other devices that enable [TRIM](#) support:

```
systemctl enable fstrim.timer
```

3.7 Enabling multilib and Arch AUR

If you are running a 64bit system then you need to enable the [multilib repository](#).
Uncomment in `/etc/pacman.conf`:

```
[multilib]
Include = /etc/pacman.d/mirrorlist
```

Update the sistem with pacman.

3.8 Boot loader

Install [systemd-boot](#) [bootloader](#) Recheck if the EFI variables are mounted

```
mount -t efivarfs efivarfs /sys/firmware/efi/efivars
```

With the ESP mounted to `/boot`, use [bootctl\(1\)](#) to install systemd-boot into the EFI system partition by running:

```
bootctl --path=/boot install
```

This will copy the systemd-boot boot loader to the EFI partition, it will then set systemd-boot as the default EFI application (default boot entry) loaded by the EFI Boot Manager.

Create configuration file to add an entry for Arch Linux:

```
vim /boot/loader/entries/arch.conf
-----
title Arch Linux
linux /vmlinuz-linux
initrd /initramfs-linux.img
```

Next add the root partition to the loader entry, since ours is `/dev/VolGrp0/root`:

```
echo "options root=PARTUUID=$(blkid -s PARTUUID -o value /dev/VolGrp0/root) rw" >>
/boot/loader/entries/arch.conf
```

3.9 Add Intel microcode

Install [microcode](#):

```
pacman -S intel-ucode
```

Update the arch systemd-boot loader file to load microcode:

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

```
-----  
...  
initrd /intel-ucode.img  
initrd /initramfs-linux.img  
...
```

3.10 Root password

Set the root [password](#):

```
passwd
```

3.11 User setup

Add a default [user](#) with:

```
useradd -m -g users -G wheel,storage,power -s /bin/bash simone
```

set a [password](#) for that user:

```
passwd simone
```

3.12 Setting up sudoers

Edit the sudoers file to give this user [sudo](#) privileges. The configuration file for sudo is in `/etc/sudoers`. It should always be edited with the [visudo\(8\)](#) command. `visudo` locks the `sudoers` file, saves edits to a temporary file, and checks that file's grammar before copying it to `/etc/sudoers`.

Uncomment:

```
wheel ALL=(ALL) ALL
```

Make sudoers require typing the root password instead of their own password by adding:

```
Defaults rootpw
```

3.13 LVM checks

Check that `lvm1`, `udev` hooks and `dm_mod` module are enabled in [mkinitcpio](#):

```
vim /minted/mkinitcpio.conf  
-----  
HOOKS="base udev ... lvm2 filesystems"  
...  
MODULES="dm_mod..."
```

if changes to the file are made, `initramfs` has to be [rebuilt](#).
Kernel option `dolvm` could be passed.

3.14 Install additional Packages

```
pacman -S bash-completion
```

add systemd pacman hook to update systemd