Linux Artix

AISK11

March 2022

Contents

1. System Info

Hardware	Value
CPU architecture	amd64
CPU vendor	intel
Motherboard	UEFI
Hard Drive	NVMe
WiFi	iwlwifi
GPU	Intel + Nvidia

Table 1.1: System hardware overview.

Software	Value
RAID	-
LVM	-
Encryption	LUKS
Root filesystem	btrfs
SWAP	4GB (file)
Bootloader	rEFInd
Kernel	linux (artix)
Initramfs	dracut
Init	dinit
Shell	zsh
Virtualization	?

Table 1.2: System software overview.

Software	Value
Display server	Xorg
Window manager	bspwm
Desktop environment	-
Terminal emulator	rxvt-unicode

Table 1.3: Display software overview.

2. Download and Boot

2.1 Download Artix ISO

1. **Download Artix ISO**
   - **Artix download page:** https://artixlinux.org/download.php
   - **Select install file:** Official ISO images » base » artix-base-dinit-<YYYYMMDD>-x86_64.iso
2. **Verify download: @TODO**

2.2 USB Preparation

1. **Unmount USB FS!**
2. **Flash image to USB:**

```
root# dd if=<./artix-base-runit-<YYYYMMDD>-x86_64.iso>.iso> of=</dev/sdX>
[bs=4M] [conv=fsync] [status=progress] && sync
```

2.3 Boot USB

2.3.1 Disable Secure Boot

1. **During POST press key to access BIOS/UEFI:** https://techofide.com/blogs/boot-menu-option-keys-for-all-computers-and-laptops-updated-list-2021-techofide/
2. **Disable Secure Boot.**
3. **Poweroff/Restart.**

2.3.2 Boot

1. **Plug in flashed USB.**
2. **During POST press key to access Boot Menu:** https://techofide.com/blogs/boot-menu-option-keys-for-all-computers-and-laptops-updated-list-2021-techofide/
3. **Select USB entry (UEFI).**

3. Pre-Installation

3.1 Login

- **Login to live environment:**

```
> artix
> artix
```

- **Access root:**

```
user$ sudo passwd root
> <NEW_ROOT_PASSWORD>
> <NEW_ROOT_PASSWORD (VERIFY)>
user$ su -
```

3.2 Disable pcspkr Module

1. **See if pcspkr is active:**

```
root# lsmod | grep -i pcspkr
```

2. **Disable pcspkr module:**

```
root# modprobe -r pcspkr
```

3.3 Verify UEFI

- **UEFI is used if following directory exists:**
  /sys/firmware/efi/

- **Verify via kernel:**

```
root# dmesg | grep -i efi
```

3.4 Connect To The Internet

1. **Connect to the internet:**
   See section: 8.

3.5 Time Sync

1. **Synchronize time:**
   See section: 6.2.

3.6 Install Additional Software

1. **Synchronize database:**

   ```
   root# [yes |] pacman -Syy
   ```

2. **Install parted:**

   ```
   root# [yes |] pacman -S [--needed] parted
   ```

3.7 Check Disk Sectors

3.7.1 Theory

- **Sector:** smallest unit size on disk. 512 or 4096 bytes (Advanced Format). 4096 Advanced Format disks have usually 512-byte conversion firmware.
- **Block:** allocation size the FS uses. Cannot be smaller than size of the sector. Can be group of sectors (4096b: 8 x 512b sectors).
  - **512b** = good for lot of small files. More blocks = more metadata.
  - **4096b** = good for larger files, less metadata. Waste if there are mostly small files.

3.7.2 Disk Information

- **Find disks (block devices):**

  ```
  user$ lsblk [-ap | -apf]
  root# fdisk -l [/dev/sdX]
  root# gdisk -l </dev/sdX>
  root# blkid
  ```

- **Get raw disk info:**
  - **Get disk physical sector size:**

    ```
    root# blockdev [-v] --getpbsz </dev/sdX[Y]>
    ```

  - **Get disk logical sector size (usually 512):**

    ```
    root# blockdev [-v] --getss </dev/sdX[Y]>
    ```

  - **Disk size in bytes:**

    ```
    root# blockdev [-v] --getsize64 </dev/sdX[Y]>
    ```

  - **Check if disk is readonly (1 = ro, 0 = rw):**

```
root# blockdev [-v] --getro </dev/sdX[Y]>
```

- **See partitions:**

```
root# parted -s </dev/sdX> (p)rint [free]
```

3.7.3 Check Disk For Bad Sectors

1. **Unmount FS!**
2. **Check disk for bad blocks:**

```
root# badblocks [-b 4096] [-w [-t 0xaa]] [-v] [-s]
</dev/sdX[Y]> | tee -a <OUTPUT_FILE>
```

4. Installation

4.1 Disk Partitioning

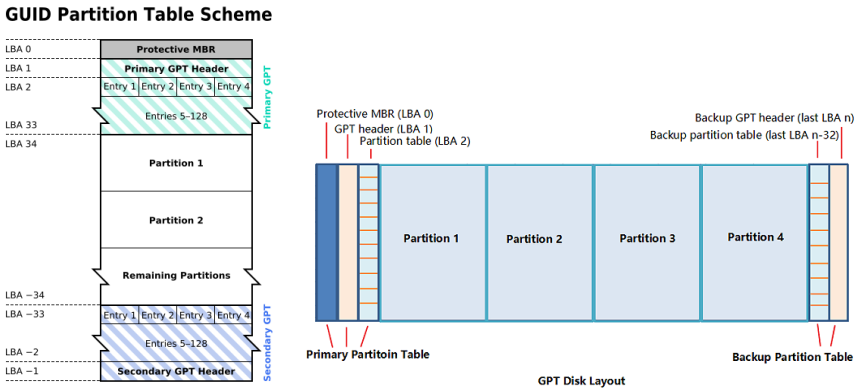


Figure 4.1: GPT Partition Table.

4.1.1 Partition Table

1. **Unmount FS from disk on which linux will be installed!**
2. **Create GPT partition table:**

```
root# parted -s </dev/sdX> mktable gpt
```

4.1.2 Basic Partitions

1. **Enter cfdisk:**

```
root# cfdisk </dev/sdX>
```

2. **Create EFI partition (recommended 550 MiB):**

```
cfdisk> n
cfdisk> 550MiB
cfdisk> t
cfdisk> EFI System
```

3. **Create encrypted partition:**

```
cfdisk> n
cfdisk>  (Enter)
```

4. **Write changes:**

```
cfdisk> W
cfdisk> yes
```

5. **Quit cfdisk:**

```
cfdisk> Q
```

6. **Name partitions:**

```
root# parted -s </dev/sdX> name 1 ESP
root# parted -s </dev/sdX> name 2 LUKS
```

7. **Format partitions:**

   (a) **ESP:**

   ```
   root# mkfs.fat -F 32 </dev/sdX1>
   root# fatlabel </dev/sdX1> <ESP>
   ```

   (b) **LUKS:**

   ```
   root# cryptsetup luksFormat --label <LUKS> </dev/sdX2>
   > YES
   > <NEW_LUKS_PASSWORD>
   > <NEW_LUKS_PASSWORD (VERIFY)>
   ```

4.1.3 (Advanced LUKS Stuff)

- **Open LUKS:**

```
root# cryptsetup open --type luks </dev/sdX2> <luks>
> <PASSWORD>
```

- **Close LUKS:**

```
root# cryptsetup close <luks>
```

- **LUKS header:**

  1. **See LUKS header:**

     ```
     root# cryptsetup luksDump </dev/sdX2>
     ```

  2. **Make LUKS header backup:**

     ```
     root# cryptsetup luksHeaderBackup </dev/sdX2>
     --header-backup-file <FILE>
     ```

  3. **Destroy LUKS header:**

```
root# cryptsetup luksErase </dev/sdX2>
> YES
```

4. **Restore LUKS header:**

```
root# cryptsetup luksHeaderRestore </dev/sdX2>
--header-backup-file <FILE>
> YES
```

- **Passwords**
  - **Change password:**

```
root# cryptsetup luksChangeKey </dev/sdX2>
> <OLD_PASSWORD>
> <NEW_PASSWORD>
> <NEW_PASSWORD (VERIFY)>
```

4.1.4 Encrypted Partition

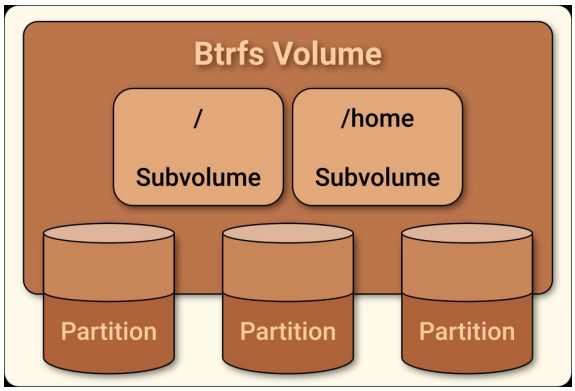


Figure 4.2: Btrfs structure.

1. **Open encrypted partition:**

```
root# cryptsetup open --type luks </dev/sdX2> <luks-root>
> <PASSWORD>
```

2. **Format root partition:**

```
root# mkfs.btrfs </dev/mapper/luks-root>
root# btrfs filesystem label </dev/mapper/luks-root> <LUKS-ROOT>
```

4.2 Base Artix Installation

1. **Create mount directory:**

```
root# mkdir </mnt/artix/>
```

2. **Mount Artix:**

```
root# mount </dev/mapper/luks-root> </mnt/artix/>
```

3. **Install packages to new directory:**

```
root# basestrap </mnt/artix/> base
```

4.3 Chroot

1. **Mount all filesystems:**

```
root# mount -t proc /proc/ </mnt/artix/proc/>
root# mount --rbind /sys/ </mnt/artix/sys/>
root# mount --make-rslave </mnt/artix/sys/>
root# mount --rbind /dev/ </mnt/artix/dev/>
root# mount --make-rslave </mnt/artix/dev/>
root# mount --bind /run/ </mnt/artix/run/>
root# mount --make-slave </mnt/artix/run/>
```

2. **Chroot:**

```
root# chroot </mnt/artix/> /bin/bash
root# source /etc/profile
root# export PS1="(chroot) ${PS1}"
```

3. **Mount boot partition:**

```
(chroot) root# mount </dev/sdX1> /boot/
```

4. **Add DNS server:**

```
(chroot) root# echo "nameserver 1.1.1.1" > /etc/resolv.conf
```

4.4 Pacman Configuration

1. **Add additional Arch packages:**

```
(chroot) root# [yes |] pacman -S [--needed] artix-archlinux-support
```

2. **Pacman configuration:**
File (/etc/pacman.conf):
https://github.com/AISK11/Artix/blob/main/configs/artix/pacman.conf

3. **Synchronize new repos [and update]:**

```
(chroot) root# [yes |] pacman -Syy[u]
```

4.5 Additional Packages

- **Text editor:**

```
(chroot) root# [yes |] pacman -S [--needed] vim
```

- **Git:**

```
(chroot) root# [yes |] pacman -S [--needed] git
```

- **Partitioning**

```
(chroot) root# [yes |] pacman -S [--needed] parted
```

- **Filesystems:**

```
(chroot) root# [yes |] pacman -S [--needed] dosfstools cryptsetup
btrfs-progs
```

- **Bootloader:**

```
(chroot) root# [yes |] pacman -S [--needed] refind
```

- **Initramfs:**

```
(chroot) root# [yes |] pacman -Rns mkinitcpio
(chroot) root# [yes |] pacman -S [--needed] dracut
```

- **Microcode:**

```
(chroot) root# [yes |] pacman -S [--needed] intel-ucode
```

- **Kernel and drivers:**

```
(chroot) root# [yes |] pacman -S [--needed] linux linux-firmware
```

- **Init:**

```
(chroot) root# [yes |] pacman -S [--needed] dinit elogind elogind-dinit
```

- **Networking:**

```
(chroot) root# [yes |] pacman -S [--needed] ethtool wpa_supplicant
[iw] dhcpcd
```

4.6 SWAP File

1. **Allocate space on COW filesystem:**

```
(chroot) root# truncate -s 0 </swap>
(chroot) root# chattr +C </swap>
(chroot) root# fallocate -l 2G </swap>
(chroot) root# chmod 0600 </swap>
```

2. **Make SWAP:**

```
(chroot) root# mkswap </swap>
(chroot) root# swapon </swap>
```

4.7 Fstab and Crypttab

4.7.1 Fstab

1. **Find UUIDs for block devices:**

```
(chroot) root# blkid
```

2. **Configure fstab:**
File (/etc/fstab):

```
## <partition>  <mount>   <fs>  <options>         <dump> <pass>
LABEL=ESP       /boot/    vfat  umask=0077        0 1
LABEL=LUKS-ROOT /         btrfs defaults,noatime 0 0
/swap           none      swap  defaults          0 0
```

4.7.2 Crypttab

1. **Add root filesystem partition to crypttab:**
File (/etc/crypttab):

```
## </dev/mapper/name> <device>              <password>  <options>
luks-root             UUID=<UUID_/dev/sdX2> none        luks
```

4.8 Local Settings

1. **Configure local settings:**
See section: 6.

4.9 Boot

4.9.1 ESP File Structure

1. **Create directories for entries:**

```
(chroot) root# mkdir -p </boot/EFI/BOOT/> </boot/EFI/artix/>
```

2. **Remove auto-generated initramfs:**

```
(chroot) root# rm /boot/init*
```

3. **Move microcode to correct directory:**

```
(chroot) root# mv /boot/intel-ucode.img </boot/EFI/artix/>
```

4. **Move kernel to linux directory:**

```
(chroot) root# mv /boot/vmlinuz-linux </boot/EFI/artix/>
```

4.9.2 Initramfs

1. **Add numlock module to dracut modules directory:**
   https://github.com/AISK11/Artix/tree/main/configs/dracut/50numlock
   (Original author: https://github.com/FivEawE/dracut-numlock)

```
(chroot) root# git clone https://www.github.com/aisk11/artix
(chroot) root# cp -r <./artix/configs/dracut/50numlock/>
/usr/lib/dracut/modules.d/
```

2. **Find kernel version:**
   - **Kernel version in boot:**
     ```
     (chroot) root# file </boot/EFI/artix/vmlinuz-linux>
     ```
   - **Available kernels:**
     /lib/modules/

3. **Generate initramfs:**

```
(chroot) root# dracut -f </boot/EFI/artix/initramfs-linux.img>
--kver <5.16.16-artix1-1> -H
```

4.9.3 Bootloader

1. **Copy rEFInd to ESP:**

```
(chroot) root# cp /usr/share/refind/refind_x64.efi </boot/EFI/BOOT/>
```

2. **Copy rEFInd to fallback:**

```
(chroot) root# cp </boot/EFI/BOOT/refind_x64.efi> </boot/EFI/BOOT/BOOTX64.EFI>
```

3. **Copy default rEFInd icons and fonts:**

```
(chroot) root# cp -r /usr/share/refind/icons/ /usr/share/refind/fonts/
</boot/EFI/BOOT/>
```

4. **Copy rEFInd config file:**
   - **Github:**
     https://github.com/AISK11/Artix/blob/main/configs/refind/refind.conf
   - **Default rEFInd file:**
     ```
     (chroot) root# cp /usr/share/refind/refind.conf-sample
     </boot/EFI/BOOT/refind.conf>
     ```

5. **Add manual entry to refind config file:**
   File (**/boot/EFI/BOOT/refind.conf**):

```
scanfor manual,internal,external,optical

menuentry "Artix" {
    #volume  "ESP"
    icon    /EFI/BOOT/themes/refind-theme/icons/128-48/os_artix.png
    loader  /EFI/artix/vmlinuz-linux
    options "initrd=/EFI/artix/intel-ucode.img \
    initrd=/EFI/artix/initramfs-linux.img rw root=/dev/mapper/luks-root quiet"
    submenuentry "Debug" {
        options "initrd=/EFI/artix/intel-ucode.img \
        initrd=/EFI/artix/initramfs-linux.img rw root=/dev/mapper/luks-root \
        rd.debug"
    }
    #disabled
}
```

6. **Create efibootmgr entry for rEFInd:**

```
(chroot) root# efibootmgr -c -d </dev/sdX> -p 1
-l </EFI/BOOT/refind_x64.efi> -L <"rEFInd"> -v
```

7. **Apply rEFInd theme:**

   (a) **Create directory for themes:**

   ```
   (chroot) root# mkdir </boot/EFI/BOOT/themes/>
   ```

   (b) **Copy theme to themes directory:**
   https://github.com/AISK11/Artix/tree/main/configs/refind/themes/refind-theme

   ```
   (chroot) root# git clone https://www.github.com/aisk11/artix
   (chroot) root# cp -r <./artix/configs/refind/themes/refind-theme/>
   </boot/EFI/BOOT/themes/>
   ```

4.10 Root User

1. **Set password for root user:**

```
(chroot) root# passwd root
> <NEW_ROOT_PASSWORD>
> <NEW_ROOT_PASSWORD (VERIFY)>
```

4.11 Finishing

1. **Reboot:**

```
(chroot) root# reboot
```

5. Post-Installation

5.1 Blacklist pcspkr

1. **See if pcspkr is in use:**

```
root# lsmod | grep -i pcspkr
```

2. **Add pcspkr to blacklisted file:**
File (/etc/modprobe.d/blacklist.conf):

```
blacklist pcspkr
```

3. **Reboot PC:**

```
root# reboot
```

5.2 Create doas group

1. **Install doas:**

```
root# [yes |] pacman -S [--needed] opendoas
root# [yes] | pacman -Rns sudo
```

2. **Create symlink for apps that needs sudo:**

```
root# ln -sf /usr/bin/doas /usr/bin/sudo
```

3. **Create group for privilege escalation:**

```
root# groupadd <doas>
```

4. **Add rights to the group:**
File (/etc/doas.conf):
https://github.com/AISK11/Artix/blob/main/configs/doas/doas.conf

```
## <permit|deny> [nopass|persist] <USER>[:GROUP] [as <USER2>]
[cmd <COMMAND> [args <ARGUMENTS>]
permit nopass :doas
```

5. **Change read permissions for doas config file:**

```
root# chmod 0600 /etd/doas.conf
```

5.3 Create non-root user

1. **Create user with home directory:**

```
root# useradd -m <USER>
root# passwd <USER>
> <NEW_USER_PASSWORD>
> <NEW_USER_PASSWORD (VERIFY)>
```

2. **Add user to the doas group:**

```
root# usermod -aG <doas> <USER>
```

5.4 Additional Basic Tools

- **Manuals:**

```
root# [yes |] pacman -S [--needed] man-db man-pages tldr
user$ tldr -u
```

- **Hardware monitoring:**

```
root# [yes |] pacman -S [--needed] usbutils inxi
```

- **System monitoring:**

```
root# [yes |] pacman -S [--needed] htop neofetch ncdu tree
```

- **Networking:**

```
root# [yes |] pacman -S [--needed] mtr
```

- **File management:**

```
root# [yes |] pacman -S [--needed] zip unzip
```

- **Multimedia:**
  - **Music:**

```
root# [yes |] pacman -S [--needed] mpv youtube-dl ffmpeg
```

6. Local Settings

6.1 Hostname

Valid hostname characters: **a-z**, **0-9** and hyphens (**-**), but not on start.

1. **Set hostname:**

```
root# echo "<HOSTNAME>" > /etc/hostname
```

6.2 Date & Time

6.2.1 Timezone

1. **List timezones:**
   /usr/share/zoneinfo/
2. **Set timezone:**

```
root# ln -sf </usr/share/zoneinfo/Europe/Copenhagen> /etc/localtime
```

6.2.2 System Clock

1. **Show current datetime:**
   - **Default:**
     ```
     user$ date
     ```
   - **ISO 8601 (YYYY-MM-DDThh:mm:ss(+|-)xx:xx):**
     ```
     user$ date '+%Y-%m-%dT%H:%M:%S%:z'
     ```
   - **Hardware clock (RTC):**
     ```
     user$ hwclock
     ```
2. **Set time:**

```
root# date <MMDDhhmmYYYY>
```

3. **Update hwclock (RTC):**

```
root# hwclock --systohc
```

6.3 Locales

1. **Show current locales:**

   ```
   user$ locale
   ```

2. **Supported locales:**
   /usr/share/i18n/locales/

3. **Choose locales:**
   File (/etc/locale.gen):

   ```
   ...
   en_DK.UTF-8 UTF-8
   en_DK ISO-8859-1
   en_GB.UTF-8 UTF-8
   en_GB ISO-8859-1
   ...
   en_US.UTF-8 UTF-8
   en_US ISO-885-1
   ...
   ```

4. **Generate locales:**

   ```
   root# locale-gen
   ```

5. **Show all available locales:**

   ```
   user$ locale -a
   ```

6. **Set system locale:**
   File (/etc/locale.conf):
   https://github.com/AISK11/Artix/blob/main/configs/artix/locale.conf

7. **Apply changes:**

   ```
   root# reboot
   ```

6.4 CLI Keyboard

Dracut initramfs will include keymap and font on initramfs on next re-generation!

- **Keymap:**

  1. **List all available keyboards:**
     /usr/share/kbd/keymaps/[i386/]
  2. **Set CLI keyboard language:**
     File (/etc/vconsole.conf):

     ```
     KEYMAP=us
     ```

  3. **Apply changes:**

```
root# reboot
```

- **Font:**
  1. **Show current CLI font:**
     ```
     user$ showconsolefont
     ```
  2. **Show available fonts:**
     /usr/share/kbd/consolefonts/
  3. **Set font:**
     - **Temporary:**
       ```
       user$ setfont <lat2-16>
       ```
     - **Permanently:**
       File (/etc/vconsole.font):
       ```
       FONT=eurlatgr
       ```
  4. **Apply changes:**
     ```
     root# reboot
     ```

7. Dinit

7.1 List services

- **List all services:**
  /etc/dinit.d/
- **List all loaded services:**

```
root# dinitctl list
```

- **Status of a service:**

```
root# dinitctl status <SERVICE>
```

7.2 Start/Stop

- **Start service:**

```
root# dinitctl start <SERVICE>
```

- **Stop service:**

```
root# dinitctl stop <SERVICE>
```

7.3 Enable/Disable

- **List enabled services:**
  /etc/dinit.d/boot.d/
- **Enable service on startup:**

```
root# dinitctl enable <SERVICE>
```

- **Disable service on startup:**

```
root# dinitctl disable <SERVICE>
```

7.4 Logs

- **See logs for services:**
  /var/log/dinit/

8. Network

8.1 Rfkill

1. **List rfkill rules:**

   ```
   root# rfkill list
   ```

2. **Block all RF devices:**

   ```
   root# rfkill block all
   ```

8.2 Ethernet

8.2.1 Monitoring

- **Show link info:**

  ```
  root# ethtool <eth0>
  ```

8.2.2 Connect

1. **Put interface logically UP:**

   ```
   root# ip l set <eth0> up
   ```

2. **Connect to network:**

   - **Static IP:**

     ```
     root# ip a add <10.0.0.11>/<24> dev <eth0>
     root# ip r add default via <10.0.0.1> [dev <eth0>]
     ```

   - **DHCP:**

     ```
     root# dhcpcd <eth0>
     ```

8.3 Wi-Fi

8.3.1 Configuration

1. **Configure wpa_supplicant:**
   File (/etc/wpa_supplicant/wpa_supplicant.conf):
   https://github.com/AISK11/Artix/blob/main/configs/wpa_supplicant/wpa_supplicant.conf

```
## Basic settings (needed to be able to launch wpa_cli).
ctrl_interface=/run/wpa_supplicant
update_config=1

## Country for wireless channels (2 letter ISO country code, e.g. DK = Denmark).
country=DK

## UNPROTECTED:
network={
  ssid="<ESSID>"
  scan_ssid=1                   ## Find hidden network
  key_mgmt=NONE
  priority=3                    ## Connection priority
}

## WPA-PSK:
network={
  ssid="<ESSID>"
  scan_ssid=1                   ## Find hidden network
  key_mgmt=WPA-PSK
  psk="<PASSWORD>"
  #psk=<32byte-HEX-NUMBER>
  priority=1                    ## Connection priority
}

## WPA-EAP:
network={
  ssid="<ESSID>"
  scan_ssid=1                   ## Find hidden network
  key_mgmt=WPA-EAP
  #eap=PEAP
  identity="<EMAIL@ADDRESS>"
  password="<PASSWORD>"
  #psk=<32byte-HEX-NUMBER>
  #ca_cert="/etc/cert/ca.pem"
  #phase1="peaplabel=0"
  phase2="auth=MSCHAPV2"
  priority=2                    ## Connection priority
}
```

2. **Change config file permissions:**

```
root# chmod 0600 </etc/wpa_supplicant/wpa_supplicant.conf>
```

8.3.2 Connect

1. **Unblock Wi-Fi in rfkill:**

   ```
   root# rfkill unblock wlan
   ```

2. **Put interface logically UP:**

   ```
   root# ip l set <wlan0> up
   ```

3. **Find ESSID:**

   (a) **Run wpa_supplicant process:**

   ```
   root# wpa_supplicant -B -D wext -i <wlan0>
   -c </etc/wpa_supplicant/wpa_supplicant.conf>
   ```

   (b) **See available interfaces and select one:**

   ```
   root# wpa_cli interface
   root# wpa_cli interface <wlan0>
   ```

   (c) **Show current wireless link status:**

   ```
   root# wpa_cli status
   user$ iw <wlan0> link
   ```

   (d) **See current wlan status and scan for networks:**

   ```
   root# wpa_cli scan
   root# wpa_cli scan_results
   ```

   (e) **Add ESSID to </etc/wpa_supplicant/wpa_supplicant.conf>.**

   (f) **Remove current wpa_supplicant process to avoid problems:**

   ```
   user$ ps -ef | grep '<wlan0>' | grep 'wpa_supplicant'
   root# kill -9 <PID>
   root# rm -f </run/wpa_supplicant/<wlan0>>
   ```

4. **Connect to configured ESSID:**

   ```
   root# wpa_supplicant -B -D wext -i <wlan0>
   -c </etc/wpa_supplicant/wpa_supplicant.conf>
   ```

5. **Connect to network:**

   - **Static IP:**

     ```
     root# ip a add <10.0.0.11>/<24> dev <wlan0>
     root# ip r add default via <10.0.0.1> [dev <wlan0>]
     ```

   - **DHCP:**

     ```
     root# dhcpcd <wlan0>
     ```

8.4 DHCP

8.4.1 Configuration

1. **Configuration file:**
   File (/etc/dhcpcd.conf):
   https://github.com/AISK11/Artix/blob/main/configs/dhcpcd/dhcpcd.conf

8.5 DNS

DNS resolvers list: https://www.privacytools.io/#dns

1. **Configure DNS server list:**
   File (/etc/resolv.conf):
   https://github.com/AISK11/Artix/blob/main/configs/artix/resolv.conf

8.6 Bluetooth @TODO

9. Pacman

9.1 Files

- **Configuration:**
  - /etc/pacman.conf
- **Mirrors:**
  - /etc/pacman.d/mirrorlist
  - /etc/pacman.d/mirrorlist-arch

9.2 Update System

1. **Synchronize DB and upgrade pkgs:**

```
root# [yes |] pacman -Syyu
```

2. **Remove all files from cache:**

```
root# [yes |] pacman -Scc
```

9.3 Install Packages

1. **Search for a package:**

```
user$ pacman -Ss[q] <REGEX>
```

2. **Show info about package:**

```
user$ pacman -Sii <PACKAGE>
```

3. **Install package:**

```
root# [yes |] pacman -S [--needed] <PACKAGE>
```

9.4 List Packages

- **List all installed packages:**

```
user$ pacman -Q[q]
```

- **List explicitly installed packages:**

```
user$ pacman -Qe[q]
```

- **List installed dependency packages:**

```
user$ pacman -Qd[q]
```

- **List unrequired [optional] dependencies:**

```
user$ pacman -Qdt[t][q]
```

- **Search within locally installed packages:**

```
user$ pacman -Qs[q] <REGEX>
```

- **Show info about installed package:**

```
user$ pacman -Qii <PACKAGE>
```

9.5 Package Files

- **Check for missing files:**

```
user$ pacman -Qk[k] [PACKAGE]
```

- **Show which package owns the file:**

```
user$ pacman -Qo <FILE>
```

- **List all files of an installed package:**

```
user$ pacman -Ql <PACKAGE>
```

- **List all files of a remote package:**

```
user$ pacman -Fyy
user$ pacman -Fl <PACKAGE>
```

9.6 Remove Packages

- **Uninstall package with dependencies and config files):**

```
root# [yes |] pacman -Rns <PACKAGE>
```

10. AUR

10.1 Setting Up

1. **Install dependencies:**

```
root# [yes |] pacman -S [--needed] base-devel
```

2. **Change compilation variables:**
File (/etc/makepkg.conf):

```
...
MAKEFLAGS="-j$(nproc)"
...
```

3. **Create directory for AUR packages:**

```
root# mkdir </aur/>
```

4. **Transfer directory ownership to doas group:**

```
root# chown -R :doas </aur/>
```

5. **Make directory writable for doas group:**

```
root# chmod -R 0775 </aur/>
```

10.2 Install Package

1. **Find package in AUR repository:**
https://aur.archlinux.org/

2. **Clone AUR package to AUR directory:**

```
user$ git clone <https://aur.archlinux.org/bvi.git> </aur/bvi/>
```

3. **Go to cloned directory:**

```
user$ cd </aur/bvi/>
```

4. **Check AUR package content:**

```
user$ less ./PKGBUILD
```

5. **Compile package:**

```
user$ makepkg -si
```

6. **If there was GPG key fault:**
   (a) **Import key:**
   ```
   user$ gpg --recv-keys <KEY>
   ```
   (b) **Compile again:**
   ```
   user$ makepkg -si
   ```

10.3 DEB Packages

1. **Install:**
   ```
   root# [yes |] pacman -S dpkg
   ```
2. **Download .deb package.**
3. **Install package with dpkg:**

11. Vim & Zsh

11.1 Vim

1. **Create symlink for visudo (ignores EDITOR and VISUAL):**

```
root# ln -sf </usr/bin/vim> /usr/bin/vi
```

2. **Set as default editor:**
   File (**/etc/profile**):

```
...
VISUAL="/bin/vim"
EDITOR="/bin/vim"
```

3. **Configuration:**
   File (**~/.vimrc**):
   https://github.com/AISK11/Artix/blob/main/dotfiles/.vimrc

11.2 Bvi

1. **Install:**
   - https://aur.archlinux.org/packages/bvi/
   - **See section: 10.**
2. **Configuration:**
   File (**~/.bvirc**):
   https://github.com/AISK11/Artix/blob/main/dotfiles/.bvirc

11.3 Zsh

1. **Install packages:**

```
root# [yes |] pacman -S [--needed] zsh zsh-syntax-highlighting
zsh-autosuggestions [zsh-completions]
```

2. **Set as default shell:**
   - **User by user:**

```
user$ chsh -l
user$ chsh -s </bin/zsh>
> <USER_PASSWORD>
```

- **User by root:**

```
root# usermod -s </bin/zsh> <USER>
```

3. **Configuration:**
   File (~/**.zshrc**):
   https://github.com/AISK11/Artix/blob/main/dotfiles/.zshrc

12. Audio

12.1 Install

1. **Install:**

```
root# [yes |] pacman -S [--needed] alsa-utils pulseaudio
```

12.2 Audio

- **Get audio:**

```
user$ amixer get <Master|PCM|Headphone|Speaker>
```

- **Mute and unmute:**

```
user$ amixer set <Master|PCM|Headphone|Speaker> <mute|unmute|toggle>
```

- **Set volume:**
  - **Specific volume:**
    ```
    user$ amixer set <Master|PCM|Headphone|Speaker> <0-100>%
    ```
  - **Increase/decrease volume:**
    ```
    user$ amixer set <Master|PCM|Headphone|Speaker> <0-100>%<+|->
    ```

12.3 Microphone

- **Get mic:**

```
user$ amixer get Capture
```

- **Mute and unmute:**

```
user$ amixer set Capture <cap|nocap|toggle>
```

- **Set volume:**
  - **Specific volume:**
    ```
    user$ amixer set Capture <0-100>%
    ```
  - **Increase/decrease volume:**
    ```
    user$ amixer set Capture <0-100>%<+|->
    ```

13. Xorg

13.1 GPU Drivers

1. **Detect GPUs:**

   ```
   root# lspci -v | grep -i -e 'VGA' -e '3D'
   ```

2. **Install drivers:**

   - **Intel (open source):**

     ```
     root# [yes |] pacman -S [--needed] xf86-video-intel mesa lib32-mesa
     vulkan-intel lib32-vulkan-intel [intel-gpu-tools]
     ```

   - **Nvidia (proprietary):**

     ```
     root# [yes |] pacman -S [--needed] nvidia nvidia-utils lib32-nvidia-utils
     [nvtop] [nvidia-settings]
     ```

3. **Monitor drivers:**

   - **Intel:**

     ```
     root# intel_gpu_top [-s <MILISECONDS>]
     ```

   - **Nvidia:**

     ```
     user$ nvtop [-d <TENTHS_OF_A_SECOND>]
     user$ nvidia-settings
     ```

13.2 Touchpad Drivers

1. **Uninstall synaptics driver:**
   Explanation: causing middle button to be almost whole area bug.

   ```
   root# pacman -Rns xf86-input-synaptics
   ```

13.3 Xorg

1. **Install:**

   ```
   root# [yes |] pacman -S [--needed] xorg xorg-drivers xorg-xinit
   ```

2. **Start X on tty1:**
   File (~/.zshrc||~/.bashrc):

```
...
if [[ -z ${DISPLAY} ]] && [[ $(tty) = '/dev/tty1' ]]; then
    source /etc/profile
    startx
fi
...
```

13.4 Brightness

1. **Install:**

```
root# [yes |] pacman -S [--needed] light
```

2. **Get current brightness:**

```
user$ light -G
```

3. **Set minimum brightness:**

   (a) **See current minimum brightness:**

   ```
   root# light -P
   ```

   (b) **Set minimum brightness**

   ```
   root# light -N 1
   ```

4. **Set brightness:**

   - **Specific value:**

     ```
     root# light -S <MIN-100>
     ```

   - **Increase/decrease value:**
     - **Increase:**

       ```
       root# light -A <0-100>
       ```

     - **Decrease:**

       ```
       root# light -U <0-100>
       ```

14. Window Manager @WIP

14.1 Bspwm

1. **Install:**

```
root# [yes |] pacman -S [--needed] bspwm
```

14.2 I3

1. **Install:**

```
root# [yes |] pacman -S [--needed] bspwm
```

2. **Run i3 after Xorg:**
   File (~/.xinitrc):
   https://github.com/AISK11/Artix/blob/main/dotfiles/.xinitrc

3. **I3 configuration:**
   File (~/.config/i3/config):

14.3 I3 Tools

```
root# [yes |] pacman -S [--needed] feh light rxvt-unicode
```

15. WM Tools @WIP

15.1 Fonts

- **See all installed fonts:**

  ```
  user$ fc-list
  ```

- **Install fonts from repository:**
  - **Unicode coverage and emojis:**

    ```
    root# [yes |] pacman -S [--needed] noto-fonts
    ```

  - **Chinese, Japanese and Korean:**

    ```
    root# [yes |] pacman -S [--needed] noto-fonts-cjk
    ```

  - **Other:**
    - **Monospaced:**

      ```
      root# [yes |] pacman -S [--needed] ttf-fira-mono ttf-dejavu
      ```

- **Install custom font:**
  1. **Create global and local directory for fonts:**

     ```
     user$ mkdir <~/.local/share/fonts/>
     root# mkdir </usr/local/share/fonts/>
     ```

  2. **Download font (TTF or OTF).**
  3. **Move font to the directory:**
     - **Global:**

       ```
       root# mv <FONT> </usr/local/share/fonts/>
       ```

     - **Local:**

       ```
       user$ mv <FONT> <~/.local/share/fonts/>
       ```

- **Install fontawesome (desktop free version):**
  1. **Download fontawesome:**
     https://fontawesome.com/download
  2. **Install fontawesome (globally):**

     ```
     user$ unzip <~/Downloads/fontawesome-free-6.1.1-desktop.zip>
     root# cp -r <~/Downloads/fontawesome-free-6.1.1-desktop/otfs/>
     /usr/local/share/fonts/fontawesome/
     user$ rm -rf ~/Downloads/fontawesome*
     ```

15.2 Terminal Emulator

1. **Install terminal emulator:**

```
root# [yes |] pacman -S [--needed] rxvt-unicode
```

2. **Configure terminal emulator:**
File (~/.Xresources):
https://github.com/AISK11/Artix/blob/main/dotfiles/.Xresources

3. **Apply changes for current X session:**

```
user# xrdb ~/.Xresources
```

15.3 Bar

1. **Install:**

```
root# [yes |] pacman -S [--needed] polybar
```

2. **Config:**
File (~/.config/polybar/config.ini):
https://github.com/AISK11/Artix/blob/main/dotfiles/.config/polybar/config.ini

15.4 Compositor

15.5 Lxappearance

16. GUI Programs

16.1 Atom

1. **Download dependencies:**

```
root# [yes |] pacman -S [--needed] atom
```

16.2 Steam

1. **Install:**

```
root# [yes |] pacman -S [--needed] lib32-fontconfig ttf-liberation
wqy-zenhei steam
```

2. **Enable Proton support:**
Steam » Settings » Steam Play » Enable Steam Play for all other titles » Restart Steam

3. **Disable shader pre-caching:**
Steam » Settings » Shader Pre-Caching » Disable Shader Pre-Caching

4. **Run game with Nvidia instead of intel:**

```
__NV_PRIME_RENDER_OFFLOAD=1 __GLX_VENDOR_LIBRARY_NAME=nvidia %command%
```

17. Virtualization @WIP

17.1 Theory

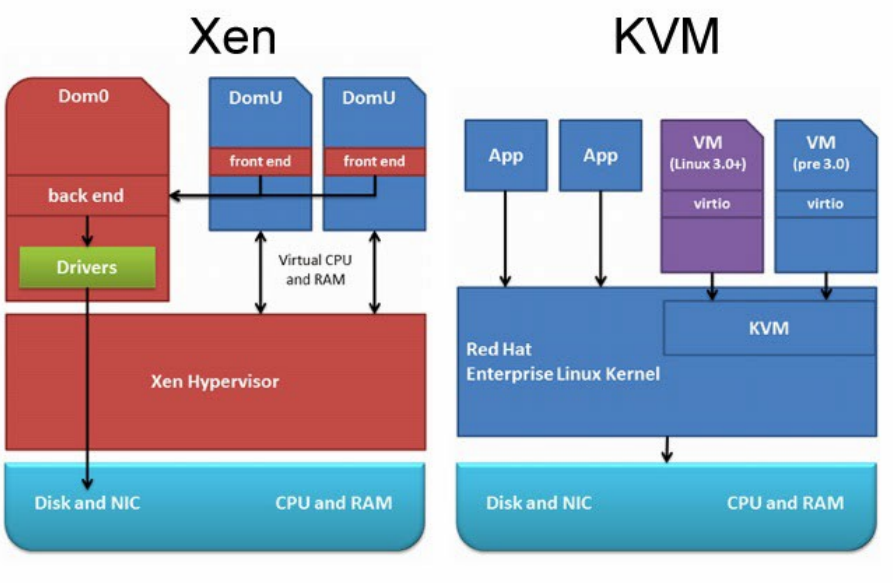


Figure 17.1: Comparison between Xen, KVM and QEMU hypervisors.

- **Hardware VM (HVM):** full virtualization, VMs are not aware, that they are sharing processing time with other clients on the same hardware.
- **Paravirtual (PV):** lighter virtualization, near native speed compared to HVM.

17.2 Virtualization Support

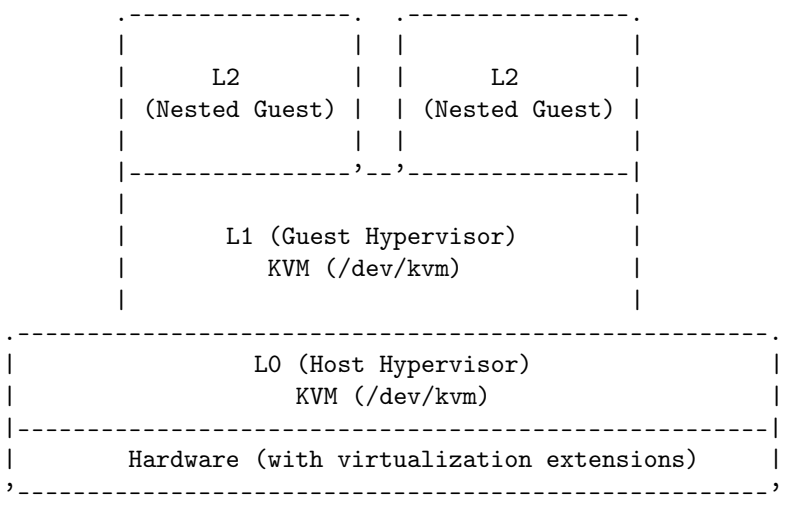
1. **See if virtualization is supported:**

   ```
   user$ grep -E 'vmx|svm|0xc0f' /proc/cpuinfo
   ```

2. **If virtualization is not supported, check if it is not disabled in BIOS/UEFI.**

17.3 Virtualbox (T2 Hypervisor)

17.4 KVM/QEMU (T1/T2 Hypervisor)



17.4.1 Installation

1. **Enable nested KVM:**
   (a) **Check if KVM module is loaded:**
   ```
   user$ lsmod | grep -i kvm_intel
   ```
   (b) **Configure kvm module to allow nesting:**
   File (/etc/modprobe.d/virtualization.conf):
   ```
   options kvm-intel nested=y
   ```
   (c) **Reload module:**
   ```
   root# modprobe -r kvm_intel
   root# modprobe kvm_intel
   ```
   (d) **Verify if nested is enabled (Y):**
   ```
   user$ cat /sys/module/kvm_intel/parameters/nested
   ```
2. **Install:**
   ```
   root# [yes |] pacman -S [--needed] qemu libvirt libvirt-dinit edk2-ovmf
   iptables-nft dnsmasq virt-manager virt-viewer
   ```
3. **Do not start libvirtd on startup and now (if running):**
   ```
   root# dinitctl disable libvirtd
   root# dinitctl stop libvirtd
   ```
4. **Create directory for iso files:**
   ```
   root# mkdir /iso/
   ```

17.4.2 Start virtualizing:

1. **Start libvirt and virtlog daemons:**

```
root# dinitctl start libvirtd
root# virtlogd -d
```

2. **Start (default) network:**

```
root# virsh net-list --all
root# virsh net-start <default>
```

3. **Start libvirt GUI client:**

```
root# virt-manager
```

17.4.3 Libvirt files:

- **Images (qcow2):** /var/lib/libvirt/images/

17.5 Xen (T1 Hypervisor) @TODO

1. **Install:**

```
root# [yes |] pacman -S [--needed]
```

18. Programming & Development @TODO

18.1 Git

18.1.1 Configuration

1. **Change default branch from 'master' to 'main':**

18.2 Atom

19. Additional Software

19.1 Fosscord (Discord Client)

19.1.1 Client @WIP

1. **Install:**

```
root# [yes |] pacman -S [--needed] yarn
user$ git clone https://github.com/fosscord/fosscord-client
user$ cd ./fosscord-client/
user$ yarn install
user$ yarn run
> linux
```

19.1.2 Server @TODO

20. Anonymity & Security

`motd /etc/issue`

`usbguard`

`macchanger`

`nftables`

21. Xen???

- https://wiki.archlinux.org/title/Xen
- https://aur.archlinux.org/xen.git

`/etc/AUR/xen/pkg/xen/boot/xen.gz`
`/etc/AUR/xen/pkg/xen/boot/xen.efi`
`/etc/AUR/xen/xen.conf`

22. ToDo

```
zsh parameter colors
bspwm
polybar
lockscreen
rofi
compton
power saving

virtualization (KVM or Xen)

GUI programs for: wifi

Programming: Atom, git, Unity

GUI: lxappearance, keepassxc, Firefox, Fosscord/Webcord

security + anonymity

VIRTUALIZATION:
See archcraft apps and rice
See debian getty program
See QubesOS Xen solution
test optimization Arch, Debian, Artix


Rice:
.zshrc = PS1
rxvt-unicode = color scheme + Font
bspwm + polybar + compton = desktop 9 Font
lxappearance = gtk theme
icons
(rEFInd + CLI font)
```

23. Quick Reference

23.1 Filesystem

- **Btrfs:**
  - https://btrfs.readthedocs.io/en/latest/Introduction.html
  - https://wiki.archlinux.org/title/Btrfs
  - https://wiki.gentoo.org/wiki/Btrfs

23.2 Bootloaders

- **rEFInd:**
  - https://www.rodsbooks.com/refind/
- **GRUB:**
  - https://gnu.huihoo.org/grub-0.90/html_chapter/grub_12.html
  - https://web.mit.edu/rhel-doc/3/rhel-rg-en-3/s1-grub-configfile.html

23.3 Linux

- **Kernel:**
  - https://www.kernel.org/doc/html/
  - https://www.kernel.org/doc/html/v4.14/admin-guide/kernel-parameters.html

23.4 Microcode

- **Microcode:**
  - https://gms.tf/check-cpu-microcode-version-on-linux.html
  - https://wiki.archlinux.org/title/Microcode
  - https://wiki.gentoo.org/wiki/Microcode
  - https://wiki.debian.org/Microcode

23.5 Init Systems

- **Dinit:**
  - https://davmac.org/projects/dinit/
  - https://github.com/davmac314/dinit/
  - https://wiki.artixlinux.org/Main/Dinit
- **Elogind:**
  - https://wiki.artixlinux.org/Main/Elogind
  - https://wiki.gentoo.org/wiki/Elogind

23.6 X11 and WM

- **Xorg:**
  - **https://wiki.archlinux.org/title/Xorg**
- **i3:**
  - **https://wiki.archlinux.org/title/I3**

23.7 Useful

- **Zsh documentation:** https://zsh.sourceforge.io/Doc/
- **Zsh highlighting** https://github.com/zsh-users/zsh-syntax-highlighting/blob/master/docs/highlighters.md
- **Linux app list:** https://wiki.archlinux.org/title/List_of_applications
- **I3 config options:** https://i3wm.org/docs/userguide.html
- **Polybar:** https://github.com/polybar/polybar/wiki/Configuration
- **Polybar themes:** https://github.com/kiddae/polybar-themes

23.8 Virtualization

- **KVM:**
  - **Nested KVM guests:** https://www.kernel.org/doc/html/v5.16-rc1/virt/kvm/running-nested-guests.html
  - **Installation:** https://wiki.archlinux.org/title/Libvirt
  - **Default network manual config:** https://jamielinux.com/docs/libvirt-networking-handbook/nat-based-network.html
- **Xen:**
  - **Installation:** https://wiki.archlinux.org/title/Xen