

ARCH LINUX "FROM SCRATCH" + GNOME + INTEL CHIPSET + UEFI + SECURE BOOT

## 1. UEFI & KEYBOARD & LOCALE

```
ls /sys/firmware/efi/efivars ... if there's an error ... it's not UEFI  
efibootmgr ... should list "Linux Boot Manager"
```

```
loadkeys br-abnt2
```

```
nano /etc/locale.gen ... unmark LANG=pt_BR.UTF-8  
locale-gen  
nano /etc/locale.conf ... set LANG=pt_BR.UTF-8
```

## 2. NETWORK & CONNECTIVITY BY WI-FI

```
ip link  
iwctl  
device list  
station wlan0 scan  
station wlan0 get-networks  
station wlan0 connect  
exit  
ping 1.1.1.3
```

```
timedatectl
```

```
ls /usr/share/kbd/consolefonts/ | grep ter-124b  
if yes: setfont ter-124b  
if not: pacman -Sy terminus-font  
if ok: setfont ter-124b
```

## 3. PARTITIONING & FORMATTING NVME

```
parted /dev/nvme0n1 mklabel gpt  
parted /dev/nvme0n1 mkpart ESP fat32 1MiB 1025MiB  
parted /dev/nvme0n1 set 1 esp on  
parted /dev/nvme0n1 mkpart primary linux-swap 1025MiB 9217MiB  
parted /dev/nvme0n1 mkpart primary btrfs 9217MiB 100%
```

```
mkfs.fat -F32 /dev/nvme0n1p1  
mkswap /dev/nvme0n1p2  
swapon /dev/nvme0n1p2  
mkfs.btrfs -L ROOT /dev/nvme0n1p3
```

## 4. MOUNTING BTRFS SUBVOLUMES

```
mount /dev/nvme0n1p3 /mnt  
btrfs subvolume create /mnt/@  
btrfs subvolume create /mnt/@home  
btrfs subvolume create /mnt/@log  
btrfs subvolume create /mnt/@cache  
btrfs subvolume create /mnt/@snapshots  
umount /mnt  
  
mount -o subvol=@ /dev/nvme0n1p3 /mnt  
mkdir -p /mnt/{home,var/log,var/cache,var/snapshots}  
mount -o subvol=@home /dev/nvme0n1p3 /mnt/home  
mount -o subvol=@log /dev/nvme0n1p3 /mnt/var/log  
mount -o subvol=@cache /dev/nvme0n1p3 /mnt/var/cache
```

```
mount -o subvol=@snapshots /dev/nvme0n1p3 /mnt/var/snapshots
mkdir -p /mnt/boot
mount /dev/nvme0n1p1 /mnt/boot
```

## 5. BASE SYSTEM INSTALLATION FOR INTEL CHIPSET

```
reflector --latest 20 --sort rate --save /etc/pacman.d/mirrorlist
```

```
pacstrap -K /mnt base linux-firmware linux-zen linux-zen-headers dkms dracut
intel-ucode vulkan-intel vulkan-mesa-device-select vulkan-tools apparmor
btrfs-progs efibootmgr iwctl nano networkmanager sbctl sudo
```

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

## 6. CHROOT CONFIGURATION

```
arch-chroot /mnt
```

```
ln -sf /usr/share/zoneinfo/America/Araguaina /etc/localtime
hwclock --systohc
```

```
nano /etc/locale.gen ... check if LANG=pt_BR.UTF-8 is enabled
locale-gen
```

```
nano /etc/locale.conf ... set LANG=pt_BR.UTF-8
```

```
nano /etc/vconsole.conf ... set KEYMAP=br-abnt2
```

```
nano /etc/hostname ... set book (choose a personal name)
```

```
passwd ... setting "root" admin password
```

```
useradd -m -g users -G wheel -s /bin/bash arch (choose a personal name)
passwd arch ... setting arch sudo password
```

```
EDITOR=nano visudo
```

```
%wheel ALL=(ALL:ALL) ALL ... uncomment for enabling sudo for arch user
```

## 7. BOOTLOADER (SYSTEMD-BOOT)

```
bootctl install
```

```
blkid /dev/nvme0n1p3 ... catch the UUID in PARTUUID field
```

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

```
title Arch Linux Zen
linux /vmlinuz-linux-zen
initrd /intel-ucode.img
initrd /initramfs-linux-zen.img
options root=PARTUUID=<UUID catched> rw
rootflags=subvol=@,compress=zstd,ssd,discard=async,space_cache=v2 apparmor=1
security=apparmor quiet loglevel=3
```

```
nano /boot/loader/entries/arch-fallback.conf
```

```
title Arch Linux Fallback
linux /vmlinuz-linux-zen
initrd /intel-ucode.img
initrd /initramfs-linux-zen-fallback.img
options root=PARTUUID=<UUID catched> rw
rootflags=subvol=@,compress=zstd,ssd,discard=async,space_cache=v2 apparmor=1
security=apparmor quiet loglevel=3
```

```
nano /boot/loader/loader.conf
```

```

default arch.conf
timeout 2
console-mode max
editor no

sbctl verify
sbctl create-keys
sbctl verify
sbctl sign -s (above verified list)
sbctl enroll-keys
sbctl list-enrolled-keys
sbctl status

nano /etc/sysctl.d/99-sysctl.conf

kernel.kptr_restrict = 2
kernel.dmesg_restrict = 1
kernel.randomize_va_space = 2
fs.protected_hardlinks = 1
fs.protected_symlinks = 1

sysctl --system

mkdir /etc/pacman.d/hooks
nano /etc/pacman.d/hooks/99-secureboot.hook

[Trigger]
Operation = Upgrade
Type = Package
Target = linux-zen
Target = systemd

[Action]
Description = Signing linux-zen kernel, initramfs and systemd-boot with sbctl
When = PostTransaction
Exec = /bin/sh -c 'sbctl sign -s /boot/vmlinuz-linux-zen 2>/dev/null || true \
&& sbctl sign -s /boot/initramfs-*.img 2>/dev/null || true && sbctl sign \
-s /boot/EFI/BOOT/BOOTX64.EFI 2>/dev/null || true && sbctl sign -s \
/boot/EFI/systemd/systemd-bootx64.efi 2>/dev/null || true'

ls /sys/firmware/efi/efivars ... check if UEFI remain active
efibootmgr ... should list "Linux Boot Manager"

systemctl enable NetworkManager.service ... enable network connection

dracut -f -H -v

exit ... exiting from chroot environment

umount -R /mnt
swapoff -a

restart + F2 + Enable Secure Boot + Custom Mode + Import keys (if you did not
has used sbctl enroll-keys)

```

## 8. POST-INSTALLATION NETWORK CONFIGURATION

```
Login: root
```

```
systemctl start NetworkManager.service
```

```
nmcli general status  
nmcli device status  
nmcli device wifi list  
nmcli device wifi connect "SSID"
```

```
pacman -Sy terminus-font  
setfont ter-124b
```

## 9. INSTALLING GNOME AND ESSENTIAL PACKAGES

```
pacman -S adwaita-icon-theme collision decibels eog extension-manager  
eyedropper file-roller firefox firefox-i18n-pt-br foliate font-manager  
fragments gdm gedit gedit-plugins gnome gnome-backgrounds gnome-bluetooth  
gnome-browser-connector gnome-calendar gnome-characters gnome-clocks gnome-  
color-manager gnome-control-center gnome-disk-utility gnome-firmware gnome-  
info-collect gnome-keyring gnome-logs gnome-menus gnome-music gnome-online-  
accounts gnome-power-manager gnome-session gnome-settings-daemon gnome-shell-  
extensions gnome-terminal gnome-text-editor gnome-themes-extra gnome-tweaks  
gnome-usage gnome-user-docs gnome-weather gparted grilo-plugins gthumb gvfs  
gvfs-afc gvfs-dnssd gvfs-goa gvfs-gphoto2 gvfs-mtp gvfs-nfs gvfs-smb gvfs-  
wsdd libreoffice-still libreoffice-still-pt-br loupe mpv mutter nautilus  
network-manager-applet papers pavucontrol qalculate-gtk rygel seahorse  
shotwell showtime simple-scan snapshot system-config-printer timeshift xdg-  
desktop-portal-gnome xdg-user-dirs-gtk xkeyboard-config yelp yelp-tools yelp-  
xsl zed
```

```
pacman -S alsaview alsaview-libs alsaview-qt alsaview-x alsaview-z  
alsa-utils aspell aspell-en aspell-pt at-spi2-core avahi bashtop  
bat bind-tools bluez bluez-utils colord cronie cups cups-browsed cups-filters  
cups-pdf curl ethtool fail2ban fastfetch fd ffmpeg firewalld foomatic-db  
foomatic-db-engine foomatic-db-ppds fwupd fzf git glances grc gst-plugins-  
base gst-plugins-base-libs gutenprint hspell htop hunspell hwinfo imagemagick  
inxi iproute2 less libssh libssh2 libvncserver libvoikko libwireplumber  
localsearch lsd lsof man-db man-pages mesa-utils meson micro mission-center  
ninja nmap nss-mdns ntfs-3g nuspell p7zip pacman-contrib parted pipewire  
pipewire-pulse powertop ptyxis pwgen qt5-wayland qt6-wayland reflector  
ripgrep rpcbind smartmontools speech-dispatcher speedtest-cli tecla  
tinystrsql tlp tree unzip upower v4l-utils v4l2loopback-utils wget  
wireless_tools wireplumber yt-dlp zip zram-generator zsh zsh-autocomplete  
zsh-autosuggestions zsh-completions zsh-history-substring-search zsh-lovers  
zsh-syntax-highlighting
```

```
systemctl enable gdm.service  
systemctl status gdm.service
```

```
systemctl enable apparmor.service  
systemctl enable avahi-daemon.service  
systemctl enable bluetooth.service  
systemctl enable cronie.service  
systemctl enable cups.service  
systemctl enable cups-browsed.service  
systemctl enable firewalld.service  
systemctl enable tlp.service
```

```
reboot
```

Login: "user" & "password" ... if fail --> make login by CLI

CLI command: sudo systemctl start gdm.service

sbctl status

```
Secure Boot: enabled
Setup Mode: disabled
Vendor keys: disabled
User keys: enrolled
```

bootctl status

```
Secure Boot: enabled (user keys)
```

## 10. CONFIGURATIONS AND OPTIMIZATIONS

1. configure gnome-control-center
2. configure gnome-tweaker
3. configure extension-manager
4. configure fstab: sudo nano /etc/fstab
5. configure timeshift
6. configure zram-generator
7. configure zsh: chsh -s /bin/zsh arch
8. install fonts :

```
sudo pacman -S gnu-free-fonts ttf-anonymous-pro ttf-atkinson-hyperlegible
ttf-bitstream-vera ttf-caladea ttf-carlito ttf-cascadia-code ttf-crimson-pro
ttf-crimson-pro-variable ttf-croscore ttf-dejavu ttf-doulos-sil ttf-droid
ttf-eurof ttf-fantasque-sans-mono ttf-fira-code ttf-fira-mono ttf-fira-sans
ttf-hack ttf-ibm-plex ttf-inconsolata ttf-input ttf-jetbrains-mono ttf-
junicode ttf-junicode-variable ttf-khmer ttf-lato ttf-liberation ttf-
libertinus ttf-linux-libertine ttf-linux-libertine-g ttf-material-icons ttf-
material-symbols-variable ttf-meslo-nerd ttf-mona-sans ttf-monospace-frozen
ttf-monospace-variable ttf-monofur ttf-monoid ttf-montserrat ttf-nunito ttf-
opensans ttf-overpass ttf-roboto ttf-roboto-mono
```

\* \* \*