

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

## 1. UEFI & KEYBOARD & LOCALE

IMPORTANT: ENABLE SECURE BOOT WITH NO KEYS ENROLLED

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-122b

if yes: setfont ter-122b

if not: pacman -Sy terminus-font

then : setfont ter-122b

## 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 ext4 1025MiB 4097MiB

parted /dev/nvme0n1 mkpart primary btrfs 4097MiB 100%

mkfs.fat -F32 /dev/nvme0n1p1

mkfs.ext4 -L BOOT /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/@pkg

btrfs subvolume create /mnt/@snapshots

umount /mnt

mount -o subvol=@ /dev/nvme0n1p3 /mnt

```
mkdir -p /mnt/{home,var/log,var/cache/pacman/pkg,var/snapshots}
```

```
mount -o subvol=@home /dev/nvme0n1p3 /mnt/home  
mount -o subvol=@log /dev/nvme0n1p3 /mnt/var/log  
mount -o subvol=@pkg /dev/nvme0n1p3 /mnt/var/cache/pacman/pkg  
mount -o subvol=@snapshots /dev/nvme0n1p3 /mnt/var/snapshots
```

```
mkdir -p /mnt/boot
```

```
mount /dev/nvme0n1p2 /mnt/boot
```

```
mkdir -p /mnt/boot/efi
```

```
mount /dev/nvme0n1p1 /mnt/boot/efi
```

## 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 iwd 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-zen.conf
```

```
title Arch Linux Zen
```

```
linux /vmlinuz-linux-zen
```

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

```
initrd /initramfs-linux-zen.img
```

```
options root=PARTUUID=<UUID catched> zswap.enabled=0 rootfstype=btrfs
```

```
rootflags=subvol=@ rw apparmor=1 security=apparmor loglevel=3
```

IMPORTANT: Dracut does not create or manage initramfs fallback

nano /boot/loader/loader.conf

```
timeout 3
console-mode max
editor no

sbctl verify
sbctl create-keys
sbctl verify
sbctl sign -s ... (sign each entrie listed above)
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 -p /etc/pacman.d/hooks (necessary)

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 & 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 --force --verbose

exit ... exiting from chroot environment

```
umount -R /mnt
swapoff -a
```

```
reboot + F2 + Secure Boot + Custom Mode + Import Keys [ only 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" --ask (or -a)
```

```
pacman -Sy terminus-font  
setfont ter-122b
```

## 9. INSTALLING GNOME AND ESSENTIAL PACKAGES

```
[ Gnome Desktop - 36 ] pacman -S gdm adwaita-icon-theme gnome gnome-  
backgrounds gnome-bluetooth gnome-boxes gnome-browser-connector gnome-  
calculator 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 xdg-desktop-portal-gnome xdg-user-dirs-gtk  
xkeyboard-config mutter
```

```
[ Applications - 41 ] pacman -S collision decibels eog extension-manager  
eyedropper file-roller firefox firefox-i18n-pt-br firewalld foliate font-  
manager fragments gedit gparted gthumb gvfs gvfs-afc gvfs-dnssd gvfs-goa  
gvfs-gphoto2 gvfs-ftp gvfs-nfs gvfs-smb gvfs-wsdd libreoffice-still  
libreoffice-still-pt-br loupe mission-center mpv nautilus network-manager-  
applet papers pavucontrol qalculate-gtk seahorse shotwell showtime simple-  
scan snapshot system-config-printer timeshift
```

```
[ CLI-I - 52 ] pacman -S 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  
foomatic-db foomatic-db-engine foomatic-db-ppds fwupd fzf gedit-plugins git  
glances grc grilo-plugins gst-plugins-base gst-plugins-base-libs gutenprint  
hspell htop hunspell hwinfo imagemagick inxi iproute2 less libssh libssh2  
libvncserver libvoikko libwireplumber localsearch lsd lssof
```

```
[ CLI-II - 51 ] pacman -S man-db man-pages mesa-utils meson ninja nmap nss-  
mdns ntfs-3g nuspell p7zip pacman-contrib parted pipewire pipewire-pulse  
powertop ptyxis pwgen qt5-wayland qt6-wayland reflector ripgrep rpcbind rygel  
smartmontools speech-dispatcher speedtest-cli tecla tinysparql 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 yelp yelp-  
tools yelp-xsl zed
```

```
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. setup gnome-control-center
2. setup gnome-tweaker
3. setup extension-manager
4. setup fstab: sudo nano /etc/fstab
5. setup timeshift
6. setup zram-generator
7. setup zsh: chsh -s /bin/zsh \$USER ... then logout / login
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
```

\* \* \*