Archlinux With BTRFS + systemd-boot + LARBS + Blackarch repos

Experience the thrill ! systemd-boot looks less bloated than grub while graphically horrible 🙂
Machine has 2 hard drives:

- sdb 256GB msata
- sda 1T SSD

The concept will be to put all system on sda and and data/VMs on sdb

Dowload ISO

Enable SSH

On the live system,

Start SSHD

```
systemctl start sshd.service
```

Set a password for root

```
passwd
```

Look up the IP address of the live system

```
ip a
```

Remote PC

On a second PC, connect to the live system via SSH:

```
ssh root@<IP-OF-THE-LIVE-SYSTEM>
```

Partition disk

A very simple setup:

- sdb1 fat32 EFI crap (from Windows)550MB
- sdb2 swap 10G
- sdb3 btrfs with tons of subvolumes

Adapt if you have only one drive with full Linux install like:

- sda1 (home) btrfs all drive

```
gdisk /dev/sda
# tout sda en linux
o [enter]
n[enter]
[enter x 4]
w [enter]
```

```
gdisk /dev/sdb
```

Create new partition table

```
Command (? for help): o
```

Create an EFI partition (optional if you already have Windows installed)

```
Command (? for help): n [enter]
Partition number (1-128, default 1):[enter]
First sector (34-500118158, default = 2048) or {+-}size{KMGTP}:[enter]
Last sector (2048-500118158, default = 500118158) or {+-}size{KMGTP}: +550M
Hex code or GUID (L to show codes, Enter = 8300): EF00
```

(choose size 550M and hex code EF00):

SWAP

```
 Command (? for help): n
Partition number (2-128, default 2): [enter]
First sector (34-500118158, default = 1128448) or {+-}size{KMGTP}:[enter]
Last sector (1128448-500118158, default = 500118158) or {+-}size{KMGTP}:
+10G[enter]
Hex code or GUID (L to show codes, Enter = 8300): 8200 [enter]
```

Create a root partition

Select default values

```
Command (? for help): n [enter]
Partition number (3-128, default 3): [enter]
First sector (34-500118158, default = 22099968) or {+-}size{KMGTP}: [enter]
Last sector (22099968-500118158, default = 500118158) or {+-}size{KMGTP}:
[enter]
Hex code or GUID (L to show codes, Enter = 8300): [enter]
```

Write the new partitions to disk

```
Command (? for help): w
```

Format partitions

```
mkfs.vfat -F32 -n EFI /dev/sdb1 ## if not already existing
mkswap -L SWAP /dev/sdb2 # Adapt if necessary
mkfs.btrfs -L ROOT /dev/sdb3 # Adapt if necessary
mkfs.btrfs -L HOME /dev/sda1
```

Subvolumes creation

```
# Adapt if necessary
mount /dev/sdb3 /mnt
btrfs sub create /mnt/@
btrfs sub create /mnt/@pkg
btrfs sub create /mnt/@snapshots
btrfs sub create /mnt/@btrfs
umount /mnt

mount /dev/sda1 /mnt
btrfs sub create /mnt/@home
umount /mnt
```

Mount the subvolumes

```
mount -o noatime,nodiratime,compress=zstd,space_cache,ssd,subvol=@
/dev/disk/by-label/ROOT /mnt
mkdir -p /mnt/{boot,home,var/cache/pacman/pkg,.snapshots,btrfs}
#mount -o noatime,nodiratime,compress=zstd,space_cache,ssd,subvol=@home
/dev/disk/by-label/ROOT /mnt/home
```

```
mount -o noatime,nodiratime,compress=zstd,space_cache,ssd,subvol=@home
/dev/disk/by-label/HOME /mnt/home
mount -o noatime,nodiratime,compress=zstd,space_cache,ssd,subvol=@pkg
/dev/disk/by-label/ROOT /mnt/var/cache/pacman/pkg
mount -o noatime,nodiratime,compress=zstd,space_cache,ssd,subvol=@snapshots
/dev/disk/by-label/ROOT /mnt/.snapshots
mount -o noatime,nodiratime,compress=zstd,space_cache,ssd,subvolid=5
/dev/disk/by-label/ROOT /mnt/btrfs
```

```
mount /dev/sdb1 /mnt/boot # Adapt if necessary
swapon /dev/sdb2 # Adapt if necessary
```

Base System

Install Arch Linux with (adjust this list to your needs):

```
pacstrap /mnt linux base base-devel btrfs-progs intel-ucode vim linux-
firmware dhcpcd networkmanager
```

Optional, for über security ⚠ If Multibooting... Microsoft creates a 100MB EFI partition so you cannot have 20 linux kernel...) Second warning ⚠ some modules need recompiling (acpi_call/tpacpi-bat):

```
pacstrap /mnt linux-hardened base base-devel btrfs-progs intel-ucode vim
linux-firmware dhcpcd networkmanager
```

Generate /etc/fstab

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

System Configuration

chroot into the new system

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

Set host name

```
echo <YOUR-HOSTNAME> > /etc/hostname
```

Set locale

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

Uncomment the following appropriates locales in /etc/locale.gen

```
en_US.UTF-8
```

Generate locale

```
locale-gen
```

Set keyboard layout and font

```
echo KEYMAP=us-acentos > /etc/vconsole.conf
echo FONT=lat9w-16 >> /etc/vconsole.conf
```

Set time zone

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

/etc/hosts

```
#<ip-address>   <hostname.domain.org>    <hostname>
127.0.0.1    <YOUR-HOSTNAME>.localdomain    <YOUR-HOSTNAME>
```

Initramfs

Configure the creation of initramfs by editing /etc/mkinitcpio.conf.
Change the line HOOKS=... to:

```
HOOKS=(base keyboard udev autodetect modconf block keymap btrfs filesystems)
```

Recreate initramfs

```
mkinitcpio -p linux
```

Boot Manager

Install systemd-boot

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

Arch entry

Create file **/boot/loader/entries/arch.conf** and fill it with:

```
title Arch Linux
linux /vmlinuz-linux
initrd /intel-ucode.img
initrd /initramfs-linux.img
options root="LABEL=ROOT" rootflags=subvol=@ rw
```

Edit file **/boot/loader/loader.conf** and fill it with:

```
default  arch.conf
timeout  4
console-mode max
#editor   no
```

Final Steps

Exit chroot, unmount partitions and reboot:

```
passwd
exit
umount -R /mnt
reboot
```

Troubleshoot

Network

If you forgot dhcpcd... like I did first time:

```
ip addr add 192.168.1.116/24 dev enp0s25
```

```
ip link set enp0s25 up
ip route add 192.168.1.0/24 via 192.168.1.1 dev ens0p25
echo "nameserver 192.168.1.1">> /etc/resolv.conf
```

Post install

Wifi

```
pacman -S linux-firmware
```

To know the module:

```
lspci -k
```

Set the interface up

```
 ip link set wlp3s0 up
```

Enp0s25

Isn't that a beautiful name? so romantic and of course more simple than eth0.... Thank you Lenard

```
systemctl enable --now NetworkManager
```

If you don't want NetworkManager

```
systemctl enable dhcpcd@enp0s25
```

Tools

Start building and installing nice stuff

```
pacman -Sy yay git zsh htop openssh man-pages
```

Configure pacman

```
vi /etc/pacman.conf
```

See: [[https://man.archlinux.org/man/pacman.conf.5| man 5 pacman.conf]

Color

Remove the “#” in front of the line

```
#Color
```

LARBS

```
curl -LO larbs.xyz/larbs.sh
sh larbs.sh
```

FAILOVER method:

```
 git clone https://github.com/LukeSmithxyz/LARBS.git
cd LARBS
sh larbs.sh
```

Battery

tp-battery-mode

- https://wiki.archlinux.org/index.php/Tp-battery-mode

```
yay -S tp-battery-mode
```

Config

```
vi /etc/tp-battery-mode.conf
```

```
START_THRESHOLD=85
STOP_THRESHOLD=100
```

Enable

```
systemctl enable tp-battery-mode
systemctl start tp-battery-mode
```

Battery Modes

- https://wiki.archlinux.org/index.php/Laptop#Power_management
- https://wiki.archlinux.org/index.php/Laptop_Mode_Tools
- https://wiki.archlinux.org/index.php/Power_management#Laptop_Mode

```
pacman -S acpi acpid tpacpi-bat
```

```
systemctl enable --now acpid
```

Then add

```
vm.laptop_mode = 5
```

in **/etc/sysctl.d/laptop.conf**

Laptop Mode Tools

From aur

```
 yay laptop-mode-tools
systemctl enable --now laptop-mode
```

CPU frequenct Squaling

```
pacman -Sy i7z thermald cpupower
systemctl enable --now thermald
systemctl enable --now cpupower
acpi -i -b
```

```
vi /etc/default/cpupower
```

Uncomment #ondemand

Backlight

Fn+F8/F9 works when adding

```
acpi_backlight=vendor
```

to **/boot/loader/entries/arch.conf**

Firewall

- https://wiki.archlinux.org/index.php/Firewalld

```
pacman -S firewalld
systemctl enable --now firewalld
```

SUSPEND to disk

This requires a SWAP partition/file

- https://wiki.archlinux.org/index.php/Mkinitcpio (resume HOOK)
- https://wiki.archlinux.org/index.php/Hibernation

```
resume="PARTLABEL=SWAP"
```

Fix Me! DOESN'T WORK

```
resume=UUID=8e3dfb9c-a8df-4312-9a7f-bf82120fb0ab
```

in **/boot/loader/entries/arch.conf**

```
HOOKS=(base udev autodetect modconf block keymap btrfs resume filesystems)
```

in **/etc/mkinitcpio.conf** then </code bash>mkinitcpio -P</code>

TROUBLESHOOTING

```
 journalctl -p 3 -xb
```

BLUETOOTH

- https://wiki.archlinux.org/index.php/Bluetooth

```
 pacman -S bluez bluez-utils
modprobe btusb
systemctl enable --now bluetooth
yay bcm20702a
bluetoothctl
```

NTP

- https://wiki.archlinux.org/index.php/Chrony

```
systemctl disable systemd-timesyncd
pacman -S chrony
```

- /etc/chrony.conf

Uncomment / change iburst to offline

```
server 0.arch.pool.ntp.org offline
```

```
server 1.arch.pool.ntp.org offline
server 3.arch.pool.ntp.org offline
```

- Run

```
systemctl enable --now chronyd
chronyc
chronyc> online
200 OK
chronyc> exit
```

Blackarch

```
curl -O https://blackarch.org/strap.sh
chmod +x strap.sh
sudo ./strap.sh
```

print

- https://wiki.archlinux.org/index.php/CUPS

```
pacman -S cups cups-pdf
systemctl enable --now cups
```

TODO

AUDIO

- https://wiki.archlinux.org/index.php/Advanced_Linux_Sound_Architecture#Unmuting_the_channels

```
pacman -S alsa-utils alsa-plugins
alsamixer
speaker-test -c 2
```

TIMESHIFT

- https://github.com/teejee2008/timeshift

Suspend to disk/ram

check if suspend works on hardened linux + if reboot works after no more batteries → 5% → hibernate

rule etc,,,

Mutt-Wizzard

Cybersecurity tools

Hardened linux

Sound

Optional utilities

ccze libreoffice discord irssi newsflash kitty zsh-* starship

Starship prompt

- https://starship.rs/

Battery

cpufreq ?

MISSING: backlight on battery

US International layout

if using larbs:

```
vi ~/.local/bin/remaps
```

add

```
setxkbmap -layout us -variant intl
```

Référence

- https://nerdstuff.org/posts/2020/2020-004_arch_linux_luks_btrfs_systemd-boot/
- https://wiki.archlinux.org/index.php/Pacman#Configuration

- https://lukesmith.xyz | https://larbs.xyz (https://github.com/lukesmithxyz/larbs) | https://github.com/lukesmithxyz/voidrice (dotfiles)

From:
https://wiki.fortier-family.com/ - **Warnaud's Wiki**

Permanent link:
https://wiki.fortier-family.com/os/archlinux/30l3-2021

Last update: **2023/01/06 14:50**