
Full Arch Install with Btrfs + GNOME

1. Prep USB / Boot to live image

Download and flash USB with arch linux iso
Update BIOS to boot from USB first
Intercept boot process to start from USB

2. Connect to Wifi using iwctl

```
iwctl
device list (find wireless controller, probably wlan0)
station wlan0 scan
station wlan0 get-networks
station wlan0 connect <your SSID>
Enter password at prompt
station wlan0 show
exit
```

3. Update NTP and Enable (this part got me for a long time)

Update NTP server in /etc/systemd/timesyncd.conf
Uncomment NTP and set =time.google.com
Write / quit txt editor
Restart time service - sudo systemctl restart systemd-timesyncd.service
Verify you're receiving NTP updates - timedatectl timesync-status

4. Run archinstall guided install

Run archlinux and follow guided install - archinstall

Select region - United States

Disk partition -

This is personal but I did best effort and chose btrfs for better snapshot support

Setup root password and new user - Authentication menu

Enable bluetooth and audio - Applications menu

Additional Packages - (again personal, these are what I wanted with arch install; protip, don't scroll - use / to search for packages)

- sudo

- firefox

- nano

Select Timezone - Timezone menu

Whatever other personal changes you want

Install

After install chmod to setup gpu drivers and desktop environment

10. GPU drivers (AMD)

pacman -S mesa vulkan-radeon libva-mesa-driver mesa-vaapi

11. Install GNOME

pacman -S gnome

systemctl enable gdm.service

13. Exit & reboot

exit

Shutdown -r now

Post install / first boot checklist -

Install **yay** (Yet Another Yaourt)

1. Install build tools

You need the base development group for compiling AUR packages:

```
sudo pacman -S --needed base-devel git
```

2. Grab **yay** from the AUR

Clone and build it as your normal user (not root):

```
cd ~  
git clone https://aur.archlinux.org/yay.git  
cd yay  
makepkg -si
```

This builds the package, resolves deps, and installs **yay**.

3. Basic usage

Update everything (official repos + AUR):

- `yay -Syu`

Install a package from the AUR:

- `yay -S spotify`

Remove orphaned packages:

- `yay -Yc`

Search with extra info:

- `yay -Ss <keyword>`

Arch First-Boot Optimization Checklist

1. Enable parallel downloads and multilib in pacman

Edit `/etc/pacman.conf`:

`ParallelDownloads = 10`

Uncomment `[multilib]` and the repo URL below it

2. Use zram for swap (instead of slow disk swap)

`sudo pacman -S zram-generator`

Create `/etc/systemd/zram-generator.conf`:

```
[zram0]
zram-size = ram / 2
compression-algorithm = zstd
```

Enable:

`sudo systemctl daemon-reexec`

3. Adjust swappiness & cache pressure

Create `/etc/sysctl.d/99-sysctl.conf`:

```
vm.swappiness=10
vm.vfs_cache_pressure=50
```

- Lower swappiness keeps stuff in RAM instead of swapping too eagerly.
 - Reducing cache pressure keeps file metadata cached longer → faster apps.
-

4. CPU frequency scaling (AMD Zen)

Install tools:

```
sudo pacman -S cpupower
```

Enable performance governor:

```
sudo systemctl enable --now cpupower.service
```

Edit `/etc/default/cpupower`:

```
governor='performance'
```

5. Enable systemd-oomd (better memory handling)

```
sudo systemctl enable --now systemd-oomd
```

This kills runaway apps before your desktop freezes.

6. Trim SSDs automatically

```
sudo systemctl enable --now fstrim.timer
```

Keeps your SSDs clean and fast.

7. Gaming performance tweaks (optional)

yay -S gamemode goverlay mangohud

- **gamemode** → boosts CPU/GPU performance for games
- **mangohud** → overlay for FPS, temps, frame times
- **goverlay** → GUI for configuring them

Enable gamemode daemon:

```
systemctl --user enable --now gamemoded
```

10. Startup services sanity check

See what's enabled:

```
systemctl list-unit-files --state=enabled
```

Example - disable what you don't need (e.g., **bluetooth.service** if you don't use it).
