

## **Day-3**

### **18 June 2025**

On the previous day, we were done with configuring the system. Creating the users, hostname and language configuration, etc. Today, we will move forward to choosing init system, bootloader and more things ahead.

#### ***Choose init system***

What's an init system? It is the very first process which starts after the Kernel is loaded. It brings up the kernel from booting to operating stage. It's PID (Process ID) is 1 as it's the first process to start and process IDs are assigned sequentially. All the processes and programs are derived from this init system only. If this process crashes, then the whole process crashes and sends the kernel into panic.

Think of it like first login screen after boot, then you go to desktop after login, and then use Firefox and enter the url for YouTube. It looks like this: login screen --> desktop --> Firefox --> YouTube...

And why would the kernel panic? An OS is nothing but the running and management of different processes. And if there's no process running or kernel is unable to locate it, then kernel panic will happen. And as we discussed earlier, all the processes are derived from PID 1. And if that stops, everything stops, and kernel doesn't know to restart the PID 1 and what to do next without any process.

For this time, we're using *systemd* as our init system. Because it's monolithic and nature and widely used. It comes by default when you use the command *pacstrap* to install base system. Choosing different init system without *systemd* required artix to install or reinstalling the base system. So, for this time, we will move forward with *systemd* and use different init systems when creating our own OS. Let's continue building arch with *systemd* for this time.

#### ***Install Bootloader***

A bootloader loads our kernel into memory. It starts just after the BIOS menu. Without it, our system cannot boot without ISO environment and will make us lose all the important things we did, including removal of the base system. It reads and loads the kernel to boot. There are many bootloaders to choose, we will now discuss which one is better and what are the differences.

1. **Systemd-boot:** It is best for UEFI systems. An UEFI system is nothing but a modern replacement to old school BIOS menu. This bootloader is fast, minimal, good for single OS not dual booting and clean arch installs.
2. **GRUB:** This bootloader is compatible with both BIOS and UEFI. But this is a bit slower compared to other bootloaders. And it's much more fancy, customizable and bloated. It supports the booting of multiple OSes,
3. **rEFInd:** It is more eye-candied than the previous bootloader. It has icons, theme and more control over aesthetics.

But we don't need beauty for bootloaders, so we will use *systemd-boot* in this case.

Run the command – *bootctl install* to install the bootloader

#### ***Create bootloader configuration***

*nano /boot/loader/loader.conf*

and paste the lines mentioned below in the file.

```
default arch  
timeout 3  
console-mode max  
editor no
```

### **Add your arch linux boot entry**

First, we need *PARTUUID* of our partition. Run the command and note the *PARTUUID* ---> `blkid /dev/vda2`  
Now, we will edit the `arch.conf` file.

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

And paste the following lines mentioned below.

```
title Arch Linux  
linux /vmlinuz-linux  
initrd /initramfs-linux.img  
options root=PARTUUID=xxxxx rw
```

Adjust the *PARTUUID* segment according to your output.