ARCH LINUX - XMONAD & XMOBAR

BASIC ARCH OS

=> Make bootable pendrive

```
# dd if=<filename>.iso of=/dev/<dev> bs=4M status='progress'
```

=> Show network interface & status

```
# iw dev
# iw dev <interface> link
```

=> Switch on the interface

```
# ip link set <interface> up
```

=> Scan & list all the wireless networks

```
# iw dev <interface> scan | less
```

=> Start iwd service

```
# systemctl start iwd.service
```

=> Go into the iwd interface

```
# iwctl
```

=> List all the access points & ssid & connect with one ssid

```
[iwd]# station <interface> get-networks
[iwd]# station <interface> connect <ssid>
```

=> Enter password and exit

```
[iwd]# exit
```

=> Ping to check connection

```
# ping archlinux.org
```

=> Edit mirrorlist for fast package server & update it

```
# vim /etc/pacman.d/mirrorlist
# pacman -Syyy
```

=> List drives & make partitions

fdisk -l

fdisk /dev/<device>

=> For gpt : g & dos : o

command : g

=> Efi partition

command : n

partition no. : 1

first sector (default 2048) : ↵ (default)

last sector : +300M

=> Swap partition

command : n

partition no. : 2

first sector : ↵ (default)

last sector : +2G

=> Linux filesystem

command : n

partition no. : 3

first sector : ↵ (default)

last sector : ↵ (default)

=> Change partition types

command : t

partition no. : 1

partition type : 1 (efi)

command : t

partition no. : 2

partition type : 19 (linux swap)

command : p (print the partitions)

command : w (write and exit)

```
=> Format the partitions
        #  mkfs.fat  -F32  /dev/<dev1>
        #  mkswap  /dev/<dev2>
        #  mkfs.btrfs  /dev/<dev3>
=> Switch on the swap
        #  swapon  /dev/<dev2>

=> Mount at live image & install kernel & utilities
        #  mount  /dev/<dev3>
        #  pacstrap  /mnt  base  linux-lts  linux-firmware  intel-ucode
=> Generate file system table & login as root
        #  genfstab  -U  /mnt  >>  /mnt/etc/fstab
        #  arch-chroot  /mnt
=> List & set time zone
        #  ls  /usr/share/zoneinfo
        #  ln  -sf  /usr/share/zoneinfo/Asia/Kolkata  /etc/localtime
=> Set hardware clock
        #  hwclock  --systohc
=> Install vim, edit locale.gen & generate locale
        #  pacman  -S  vim
        #  vim  /etc/locale.gen
              ** uncomment #en_IN UTF-8 and save
        #  locale-gen
=> Set hostname & hosts
        #  vim /etc/hostname
              ** <hostname>
        #  vim /etc/hosts
              ** 127.0.0.1          localhost
                 ::1                      localhost
                 127.0.1.1                <hostname>.localdomain
        <hostname>
=> Create and set password for root and new user
        #  passwd
              **  root password
        #  useradd  -m  <user>
```

```
# passwd pluto
=> Make new user the member of groups (wheel to give sudo privileges)
        # usermod -aG wheel,audio,video,optical,storage <user>
=> Install sudo and add <username> sudo privileges <user>
        # visudo
            ** uncomment # % wheel ALL = (ALL) ALL and save

=> Install grub and basic utilities
        # pacman -S grub efibootmgr dosfstools os-prober mtools
        networkmanager network-manager-applet dialog wpa-supplicant
        bluez bluez-utils xdg-utils xdg-user-dirs alsa-utils pulseaudio
        pulseaudio-bluetooth base-devel linux-headers

=> Make efi directory and mount /dev/<dev1> here
        # mkdir /boot/efi
        # mount /dev/<dev1> /boot/efi

=> Install grub in efi & generate its config. file
        # grub-install --target=x86_64-efi --bootloader-id=GRUB
        --efi-directory=boot/efi --no-nvram --removable
        # grub-mkconfig -o /boot/grub/grub.cfg
=> Enable networkmanager
        # systemctl enable NetworkManager

=> Install some important packages
        # pacman -S git htop neofetch tree mlocate
=> Exit and unmount /mnt
        # exit
        # umount -l /mnt
        # reboot

=> Connect with the network interface
        $ sudo systemctl start systemd-networkd
        $ sudo systemctl start NetworkManager
=> Show the network interface
        $ ip link
```

=> Switch on the interface for wi-fi

```
$ ip link set <interface> up
```

=> Connect with the interface using gui

```
$ nmtui
```

=> Set up firewall using ufw

```
$ sudo pacman -S ufw
$ sudo ufw enable
$ sudo ufw status verbose
$ sudo systemctl enable ufw.service
** reboot and check the status again
```

=> List and remove the orphan packages

```
$ pacman -Qdt
$ sudo pacman -Rns $(pacman -Qtdq)
```

=> Check the errors

```
$ sudo systemctl --failed
$ sudo journalctl -p 3 -xb
```

=> Disable grub delay

=> Add the following to /etc/default/grub

```
$ sudo vim /etc/default/grub
** GRUB_FORCE_HIDDEN_MENU="true"
```

=> Then put file 31_hold_shift to /etc/grub.d/

```
$ vim 31_hold_shift
$ sudo mv 31_hold_shift /etc/grub.d/
```

=> Get the 31_hold_shift file from https://github.com/sudo-pluto/dotfiles

=> Make it executable & regenerate grub conf.

```
$ sudo chmod a+x /etc/grub.d/31_hold_shift
$ sudo grub-mkconfig -o /boot/grub/grub.cfg
```

XMONAD & XMOBAR

=> Install xmonad, xmobar, lightdm, firefox, terminator,etc

```
$ sudo pacman -S xorg lightdm lightdm-gtk-greeter xmonad
 xmonad-contrib xmobar dmenu nitrogen chromium
terminator
```

=> Enable lightdm

```
$ sudo systemctl enable lightdm
```

=> Create .xprofile file in ~ directory

```
$ vim .xprofile
   ** #  wallpaper
       nitrogen  --restore  &
       #  compositor
       picom  -f  &
```

=> Create .xmonad/ dir in home dir and xmonad.hs file in it

```
$ mkdir  .xmonad
$ cd  .xmonad
$ vim  xmonad.hs
    ** import XMonad
        main       =  xmonad  def
                      {  terminal           =  “terminator”
                       ,  modMask         =  mod4Mask
                      }
```

=> Reboot the system

```
$ reboot
```

=> Clone xmonad & xmobar settings from github

https://github.com/sudo-pluto/dotfiles