INSTALATION GUIDE

Featuring: bspwm, zsh and neovim

light version - for printing

1 - Set the keyboard layout

List all available layouts:

```
# ls /usr/share/kbd/keymaps/**/*.map.gz
```

Select a layout:

```
# loadkeys de-latin1
```

2 - Connecting to internet:

Start the services:

```
# systemctl status dhcpcd
# systemctl start dhcpcd
```

Check connection:

```
# ping www.google.com
```

If you're using wi-fi, start wifi-menu after starting the dhcpcd service.

3 - Update the system clock:

```
# timedatectl set-ntp true
```

You can set it to localtime or UTC later.

4 - Partition the disks; I use fdisk:

```
# fdisk /dev/sdX
```

Where 'X' is the disk you want to partition

5 - Format the new partitions:

```
# mkfs.ext4 /dev/sdXn
```

Where 'X' is the disk you want to partition
Where 'n' is the number of the partition on that specific disk;
e.g: /dev/sda1 /dev/sdd1 /dev/sdc5

6 - If you created a swap partition, initialize it now:

```
# mkswap /dev/sdaX
# swapon /dev/sdaX
```

7 - Mount the filesystem:

```
# mount /dev/sdXn /mnt
```

Create mount points for any other partitions you created in step 5:
For example:

```
# mkdir /mnt/boot
# mount /dev/sdXn /mnt/boot

# mkdir /mnt/home
# mount /dev/sdXn /mnt/home
```

genfstab will later detect mounted filesystems and the swap.

8 - Select the mirrors and sort them so that the closest mirror is also the closes to your location geographically:

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

9 - Install the base packages:

```
# pacstrap /mnt base base-devel linux linux-firmware
```

NOTE: Before running pacstrap, you must have mounted the partitions in step 7, otherwise, it'll not work.
Specially the /boot partition, if you created one.

10 - Generate the fstab:

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

Then check it to see if it's alright.

11 - chroot into the new system:

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

12 - Set the timezone:

```
# ln -sf /usr/share/zoneinfo/Region/City /etc/localtime
```

Run hwclock(8) to generate /etc/adjtime:

```
# hwclock --systohc
```

---------- NOTE ----------

The hardware clock can be queried and set with the timedatectl command.
You can see the current hardware clock time standard of the Arch system using:

```
# timedatectl | grep local
```

To revert to the hardware clock being in UTC, type:

```
# timedatectl set-local-rtc 0
```

These generate /etc/adjtime automatically and update the RTC accordingly; no further configuration is required.

---------- END NOTE ----------

13 - Generate the locales:

Create the /etc/locale.conf file, and set the $LANG variable accordingly:

Add to the file:

```
LANG=en_US.UTF-8
```

Uncomment en_US.UTF-8 UTF-8 and other needed locales in /etc/locale.gen, and generate them with:

```
# locale-gen
```

INSTALATION GUIDE

Featuring: bspwm, zsh and neovim

light version - for printing

14 - Make the keyboard layout persistent

in /etc/vconsole.conf:
Add to the file:
KEYMAP=<your layout>

15 - Network configuration:

Create the hostname file (/etc/hostname):

Add to the file:

<your hostname>

Add matching entries to /etc/hosts:

```
127.0.0.1       localhost
::1                 localhost
127.0.1.1       <your
hostname.localdomain    <your hostname>
```

If the system has a permanent IP address, it should be used instead of 127.0.1.1.
Complete the network configuration for the newly installed environment,
that includes installing your preferred network management software.

16 - Set the root password:

```
# passwd
```

17 - Create the users and set their passwords:

```
# useradd -m -g users -G wheel -s /bin/bash
<your username>
# passwd <your username>
```

18 - Download and install grub, os-prober and ntfs-3g:

NOTE:
os-prober and ntfs-3g are only needed if you have another system you want to dual-boot.
ntfs-3g is used to detect Windows file system, and then grub will add them automatically
at the boot list when you run grub-mkconfig in step 22.
os-prober is used both to detect Windows and another Linux systems.

```
# pacman -S grub os-prober ntfs-3g
# grub-install /dev/sda
```

19 - Configure mkinitcpio:

```
# mkinitcpio -p linux
```

Where 'linux' above should be replaced with whatever kernel you're using.

20 - Install dhcpcd (or dhclient), networkmanager and nmcli (or whatever you're using) and enable the service:

```
# pacman -S dhcpcd networkmanager nmcli

# systemctl enable dhcpcd
# systemctl enable networkmanager
```

21 - Configure grub:

```
# grub-mkconfig -o /boot/grub/grub.cfg
```

This will generate grub.cfg in /boot/grub/grub.cfg and add to it the entries to boot into other systems. And add them to the list.

22 - Exit chroot:

```
# exit
```

23 - Umount all partitions and reboot the system:

```
# umount -a
# reboot
```

24 - Log-in as root with the password you set before, in step 16.
The username is 'root'.

THIS IS STEP DOESN'T MAKE ANY DIFFERENCE AT ALL, AS YOU HAVE ALREADY SET THE LAYOUT IN VCONSOLE.CONF
ABOVE SO YOU CAN SKIP IT

25 - Set the keyboard layout:

```
# loadkeys br-abnt2
# mkdir
/etc/X11/xorg.conf.d
# nano
/etc/X11/xorg.conf.d/10-evdev.conf
```

The file should look like this:

```
Section
InputClass

Identifier evdev "keyboard-hall”

MatchIsKeyboard "on”

MatchDevicePath “/dev/input/event"
                                                Driver
"evdev
                                                Option
XkbLayout "br”
                                                Option
XkbVariant "abnt2"
                                        EndSection
```

25 - Enable sudo in /etc/sudoers:

Search for:
"root ALL=(ALL) ALL" (without the quotes)

And insert:
<your username> ALL=(ALL) ALL
below that line.

It should look like this:

```
[...]
root ALL=(ALL) ALL
user1234 ALL=(ALL) ALL
[...]
```

INSTALATION GUIDE

Featuring: bspwm, zsh and neovim

light version - for printing

26 - Test and enable internet connection:

```
# ping -c3 www.google.com
# systemctl status dhcpcd
# systemctl start dhcpcd
```

Should work if you're on cable.

Check wi-fi, if NetworkManager`s service is not running, then start it:

```
# systemctl status NetworkManager
# systemctl start NetworkManager
```

Identify your network:

```
# nmcli dev wifi
```

Connect:

```
# nmcli device wifi connect <SSID> password <PSW>
```

27 - Create basic folder structure:

```
# mkdir /home/zsucrilhos/_my-disks      - Optional, i use to symlink my other HDDs mounted on /mnt
# mkdir /home/zsucrilhos/_my-scripts    - Optional, i use for my scripts
# mkdir /home/zsucrilhos/_my-sources    - Optional, i use for organization

# mkdir /home/zsucrilhos/Desktop
# mkdir /home/zsucrilhos/Documents
# mkdir /home/zsucrilhos/Downloads
# mkdir /home/zsucrilhos/Music
# mkdir /home/zsucrilhos/Pictures
# mkdir /home/zsucrilhos/Screenshots
# mkdir /home/zsucrilhos/Videos

# mkdir /home/zsucrilhos/RemovableMedia - Optional, i use to mount Removable Media such as pen-drives
# mkdir /home/zsucrilhos/WindowsDesktop - Optional, i use to mount Windows Desktop
```

28 - Install the packes you`ll use:

```
- Window Manager / Desktop Environment
- Text editor
- Music Player
- etc
```

29 - Exit root and log-in as your user:

```
# exit
```

----------NOTE----------
From here on now it is mostly personal preference.
This is what i like to use.
----------END NOTE----------

30 - Install yay:

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

WARNING: DO NOT run makepkg as root as it might damage your system!

31 - Get the my dotfiles.

```
$ cd /home/Downloads/_cloned-repos
$ git clone https://github.com/zsucrilhos/dotfiles
$ rsync -a dotfiles /home/zsucrilhos/
```

32 - Set the default shell, in my case, ZSH:

For root user (run as root):

```
# chsh -s $(which zsh)
```

For normal user:

```
$ chsh -s $(which zsh)
```

Install oh-my-zsh:

```
$ sh -c "$(wget https://raw.githubusercontent.com/robbyrussell/oh-my-zsh/master/tools/install.sh -O -)"
```

And the plugins:

k:

```
$ git clone https://github.com/supercrabtree/k $ZSH_CUSTOM/plugins/k
```

alias-tips:

```
$ cd ${ZSH_CUSTOM1:-$ZSH/custom}/plugins
$ git clone https://github.com/djui/alias-tips.git
```

zsh-autosuggestions:

```
$ git clone https://github.com/zsh-users/zsh-autosuggestions ${ZSH_CUSTOM:-~/.oh-my-zsh/custom}/plugins/zsh-autosuggestions
```

zsh-autocompletions:

```
$ git clone https://github.com/zsh-users/zsh-completions ${ZSH_CUSTOM:=~/.oh-my-zsh/custom}/plugins/zsh-completions
```

33 - Install bspwm (Javyre's fork):

```
$ git clone https://aur.archlinux.org/bspwm-git.git
```

Change the source varible in the PKGBUILD to use Javyre's fork

Compile and install the package with:

```
$ makepkg -si
```

34 - Install vim-plug:

```
$ curl -fLo \
~/.local/share/nvim/site/autoload/plug.vim \
--create-dirs \
https://raw.githubusercontent.com/junegunn/vim-plug/master/plug.vim
```

35 - Install polybar:

```
$ yay -S polybar-git
```

36 - Creating symlinks/shortcuts to other storage:

```
$ ln -s /run/media/zsucrilhos RemovableMedia
$ ln -s /home/zsucrilhos/_my-disks/249g/Users/<username>/Desktop WindowsDesktop
```

INSTALATION GUIDE

Featuring: bspwm, zsh and neovim

light version - for printing

37 - Tailor fstab to mount my others disks.

38 - Restore backed-up files.

39 - Reboot.

Done, it should work now.