INTRODUCTION

After some tests on opnBSD/FreeBSD/TrueOS (the 2 last for zfs usage) I need to have a working laptop with Rolling Release method, simple to tweak and maintain. I will use archlinux-2016.12.01-dual.iso, install with btrfs and switch to openrc instead of systemd Process is painful... an easier way would be to use archbang Linux...

Hardware

I enhanced the almighty Thinkpad with a Crucial M4 mSATA Drive of 256GB I intend to put the system on this SSD and home, swap and /var on the spinning 500GB disk :) like so:

Partition table

Disk	Partition	Name	Label	Size	Format	Comment
SSD	1	sdb1	ARCHROOT	256+GB	BTRF	/ for Archlinux
HD	1	sda1	SWAP	32GB	Linux Swap	swap for Linux
HD	2	sda2	VAR	50GB	BTRFS	/var for Linux
HD	3	sda3	HOME	389GB	NTRFS	/home for Linux

Simple 🙂

Base install

References

- Installation guide
- Thinkpad x230

ssh root login

ssh root login is already allowed grab the ip

```
ip addr show
```

Start sshd

```
systemctl start sshd
```

change root passwd

```
passwd
```

Now you can connect remotely from another host set the partition using cfdisk like displayed in the table above

Formatting

Let's format all these partitions in btrfs/swap format with LABELS!!.

```
mkfs.btrfs -L ARCHROOT /dev/sdb1
mkfs.btrfs -L ARCHVAR /dev/sda2
mkfs.btrfs -L ARCHHOME /dev/sda3
mkswap -L SWAP /dev/sda1
```

Mount partitions

```
mkdir /mnt/{home,var}
mount /dev/sdb1 /mnt
mount /dev/sda2 /mnt/var
mount /dev/sda3 /mnt/home
swapon /dev/sda1
```

Check all is fine:

```
mount
```

Thank you systemd for this horrible output....

Base system

Then install base packages plus base-devel packages

```
pacstrap /mnt base base-devel
```

Fstab

For SSD tweaks and to make it shine: SSD

- generate fstab

```
genfstab -L -p /mnt >> /mnt/etc/fstab
```

Yes I use Labels, UUID sucks

- check fstab for options and label in place (for me it did a good job)

Bootloader

I use Syslinux since it does its job well and didn't fall into useless overkill configuration files and tools! Syslinux

```
pacstrap /mnt syslinux
```

Configuration

Environment

Let's go to our new system!

```
arch-chroot /mnt
```

- /etc/hostname

```
 echo 30L3 > /etc/hostname
```

- /etc/locale.gen

I'll use en_US.utf-8/iso8859 so uncomment:

```
en_US.UTF-8 UTF-8
en_US ISO-8859-1
```

The generate locales

```
 locale-gen
```

- /etc/locale.conf Locale

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

# Keep the default sort order (e.g. files starting with a '.'
# should appear at the start of a directory listing.)
LC_COLLATE="C"
```

- /etc/vconsole.conf

Default keyboard in console (US variant international ... with a different mapping than on X11! Well done)

```
echo "KEYMAP=us-acentos" > /etc/vconsole.conf
```

- /etc/localtime

```
ln -s /usr/share/zoneinfo/Europe/Paris /etc/localtime
```

- /etc/mkinitcpio.conf

Double check the content, then just regenerate it in case ...

```
mkinitcpio -p linux
```

got scary:

```
==> ERROR: file not found: `fsck.btrfs'
==> WARNING: No fsck helpers found. fsck will not be run on boot.
```

- /boot/syslinux/syslinux.cfg

I left it by default (for now 😉) WARNING: No fsck helpers found. fsck will not be run on boot.
❗ Check that Linux start from /dev/sdb1 not sdaX ... Then the magic:

```
 /usr/sbin/syslinux-install_update -i -a -m
```

output:

```
Syslinux BIOS install successful
Attribute Legacy Bios Bootable Set - /dev/sdb1
Installed MBR (/usr/lib/syslinux/bios/gptmbr.bin) to /dev/sdb
```

if errors, install gptfdisk package or what's required :)

- root password

```
passwd
```

Reboot

Umount stuff cleanly

First use Ctrl+D to escape from the chroot
then:

```
umount /mnt/var /mnt/home /mnt
```

Now is the time to light a candle and type

```
reboot
```

SSHd

Access remotely:

```
 pacman -Sy openssh
```

Enable it at boot

User

```
useradd -g users -m -s /bin/bash warnaud
passwd warnaud
```

use **visudo** to set appropriate rights for the user

Install some X

pacman -S xorg-server xorg-apps xorg-fonts xorg-fonts-100dpi xorg-fonts-75dpi xorg-twm xorg-xclock xorg-xinit xorg-xdm xterm xf86-video-intel xorg-xmessage xorg-xcalc xorg-xfontsel xorg-utils vim We have intel video card so we need to add some modules at boot for KMS

```
vi /etc/mkinitcpio.conf
```

```
...
MODULES="i915"
...
```

Acceleration method

```
vi /etc/X11/xorg.conf.d/20-intel.conf
```

```
Section "Device"
   Identifier  "Intel Graphics"
   Driver      "intel"
   Option      "AccelMethod"  "sna"
EndSection
```

```
 vi /etc/X11/xorg.conf.d/10-evdev.conf
```

```
Section "InputClass"
        Identifier "evdev keyboard catchall"
        MatchIsKeyboard "on"
        MatchDevicePath "/dev/input/event*"
        Driver "evdev"
        Option "XkbLayout" "us"
```

```
        Option "XkbVariant" "intl"
EndSection
```

XDM

Reference: https://wiki.archlinux.org/index.php/XDM

```
pacman -S xorg-xdm
```

Enable systemd service

```
systemctl enable xdm.service
```

i915

Power consumption for Intel Graphics:

```
vi /etc/modprobe.d/modprobe.conf
```

```
options i915 i915_enable_rc6=1 i915_enable_fbc=1 lvds_downclock=1
```

Rebuild the kernel:

```
mkinitcpio -p linux && mkinitcpio -p linux-ck
```

Thinkpad

Wacom

Reference: https://wiki.archlinux.org/index.php/Wacom Install the driver:

```
pacman -S xf86-input-wacom
```

Fix Me! la suite ...

Disable Touchpad

Deactivate the useless touchpad:

```
 vi /etc/X11/xorg.conf.d/10-evdev.conf
```

```
Section "InputClass"
        Identifier "evdev touchpad catchall"
        MatchIsTouchpad "off"
```

```
        MatchDevicePath "/dev/input/event*"
        Driver "evdev"
EndSection
```

Fix Me! Check alternate solution:https://wiki.archlinux.org/index.php/Synaptics_Touchpad

TrackNav enhancement

This hack will enable scrolling using the middle button + the TrackPoint™ ® ©

```
vi /etc/X11/xorg.conf.d/20-thinkpad.conf
```

```
Section "InputClass"
    Identifier  "Trackpoint Wheel Emulation"
    MatchProduct        "TPPS/2 IBM TrackPoint|DualPoint Stick|Synaptics
Inc. Composite TouchPad / TrackPoint|ThinkPad USB Keyboard with
TrackPoint|USB Trackpoint pointing device"
    MatchDevicePath     "/dev/input/event*"
    Option          "EmulateWheel"        "true"
    Option          "EmulateWheelButton"  "2"
    Option          "Emulate3Buttons"      "false"
    Option          "XAxisMapping"      "6 7"
    Option          "YAxisMapping"        "4 5"
EndSection
```

Other software

Archlinuxfr repository

We need yaourt to install a bunch of aur packages for openrc

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

```
...
[archlinuxfr]
SigLevel = Never
Server = http://repo.archlinux.fr/$arch
```

\o/ No signature

Sudo

```
pacman -S sudo
```

```
visudo
```

```
...
root ALL=(ALL) ALL
someuser ALL=(ALL) ALL
```

Just add the 'someuser' line and change by the user you need

Compilation options

❗ just for fun not really mandatory Reference: https://wiki.archlinux.org/index.php/Makepkg.conf

```
vi /etc/makepkg.conf
```

Modify CFLAGS/CXXFLAGS:

```
# -march=native also sets the correct -mtune=
CFLAGS="-march=native -O2 -pipe -fstack-protector --param=ssp-buffer-size=4
-D_FORTIFY_SOURCE=2"
CXXFLAGS="${CFLAGS}"
```

Modify MAKEFLAGS (4 = output of nproc)

```
MAKEFLAGS="-j4"
```

❗❗❗❗❗❗ WORK IN PROGRESS ❗❗❗❗❗❗❗❗

Switch to openrc

bye bye systemd!

```
yaourt -S openrc-sysvinit
```

This will install sysvinit also. This will conflict with systemd crap, remove it:

```
Remove systemd-sysvcompat? [y/N] y

 yaourt -S openrc
```

use openrc-git to be super bleeding edge :) </code> Next, install some arch services + network service tools

```
 yaourt -S openrc-arch-services-git net-tools
```

openrc & bootloader

I use syslinux, the trick is to add init=/usr/bin/init-openrc to the kernel parameters of your bootloader

configuration file for me on **/boot/syslinux/syslinux.cfg**:

```
LABEL arch-ck
        MENU LABEL Arch Linux CK
        LINUX ../vmlinuz-linux-ck-ivybridge
        APPEND root=/dev/sdb1 ro resume=/dev/sda1 vga=792
init=/usr/bin/init-openrc
        INITRD ../initramfs-linux-ck-ivybridge.img quiet vga=current
ipv6.disable=1 elevator=bfq pcie_aspm=force acpi_backlight=vendor
```

After a reboot, systemd is gone 🤣

recommended services

```
rc-update add udev sysinit
rc-update add dcron default
rc-update add dbus default
rc-update add alsa default
rc-update add syslog-ng default
```

Then it is safe to reboot ;)

Nightmare after reboot, first xdm do not work/start... second network card naming is the same as systemd...

References

- https://wiki.archlinux.org/index.php/OpenRC
- https://blog.notfoss.com/posts/openrc-on-arch-linux/
- https://github.com/throwawaygh/arch-openrc-guide
- Overview my previous install

rc-update add dcron default

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

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

Last update: **2021/12/29 20:01**