Arch Linux Installation

Schritt für Schritt

Anleitung

Präsentator:
Jürgen Mattheis

Korrekturen bitte an:
https://github.com/matthejue/PicoC-Compiler_Uebungsblatt/issues.

12. Oktober 2022

Universität Freiburg, Lehrstuhl für Betriebssysteme

Gliederung

Before Installation

Before Installation - 1 I

- in der UEFI firmware fast-boot auf [Disabled] setzen.
- „Schnellstart“ in Windows deaktivieren, da die EFI Systempartition beschädigt werden kannn.

1. Windows-Taste + X drücken / Systemsteuerung starten.
2. Hier nun System und Sicherheit / Energieoptionen starten.
3. Links nun „Auswählen, was beim Drücken des Netzschalters geschehen soll“ anklicken.
4. Im neuen Fenster nun oben auf: Einige Einstellungen sind momentan nicht verfügbar anklicken.

Before Installation - 1 II

5. Nun wird unten bei „Einstellungen für das Herunterfahren“ der Haken bei Schnellstart aktivieren (Empfohlen) anklickbar. Nun kann man den Haken entweder entfernen oder setzen.

- use Belena Etcher (‘sudo balena-etcher-electron‘) to put the ‘.iso‘ on a usb-device, then go into UEFI firmware settings (start with e.g. f2 during system start, ‘systemctl reboot –firmware-setup‘ or select it in GRUB with the option ‘UEFI firmware settings‘) and change the bootorder to have the usb-device having a higher boot priority then the esp partition with it’s bootloader that is usually loaded.
- maybe disable secure boot

Base Installation

Keyboard Layout (for the Installation)

- `> ls /usr/share/kbd/keymaps/**/*.map.gz | less`

- `> loadkeys de-latin1-nodeadkeys`

Wifi connection I

- `> ls /usr/share/kbd/keymaps/**/*.map.gz | less`
- `> ping -c 1 google.com`
- `> ip link or ip a (addr show)`

Wifi connection II

Sidenote

- sometimes one has to manually start the dhcp client `dhcpcd`.
- netctl (and therefore wifi-menu) got removed from the Arch Linux ISO starting July 2020. To get connected while installing Arch, use `iwctl`. If it's blocked, either use that physical switch, or use `rfkill unblock wifi`. Then, type in `iwctl`. When you're in `iwctl`, use `device list` to see the name that the wifi router is using. For commands after this, replace device with the name of the device as found using the `device list` command. If you want to get the name of the network you want to use, use `station device scan` and then `station device get-networks`. After that, type in `station device connect SSID*`, with the *SSID being the name of the internet you want to use. If there is a password on the wifi, type that in when it asks for the wifi. After that, press Ctrl+C to get back to the terminal/root@archiso.

Optional

- reorder mirrolist: `nvim etc/pacman.d/mirrorlist` (first entry will be taken first, if offline the second etc.)
- before that: `pacman -S neovim` (already root)
- check for uefi mode: `ls /sys/firmware/efi/efivars` check if exists

Time

- `timedatectl set-ntp true`
- verify with `timedatectl status` (or `date`), should be utc

Partitioning I

- `lsblk -f`, `fdisk -l | more` or `df -h`, `mount` (for currently mounted filesystem)
- `cfdisk /dev/sda`
- `gpt`, enter
- move with arrow keys to `New`, enter, type `512M`, enter (efi-partition)
- select `Type`, `EFI-System`
- move down to next Free space and next `2G` (swap-partition)

Partitioning II

- select `Type`, `Linux swap`
- next `20G` (root-Partition)
- for floating number: `17.5G`
- it's type is automatically `Linux filesytem`
- next e.g. `80G` (home-Partition)
- it's type is automatically `Linux filesytem`
- move to `Write` and answer `yes` (enter doesn't write anything)

Partitioning III

- move to `Quit`
- (`fdisk` is the old way)
- `mkfs.fat -F32 /dev/sda1`
- `mkswap /dev/sda2`
- `swapon /dev/sda2`
- `mkfs.ext4 /dev/sda3` and `mkfs.ext4 dev/sda4`

Install to Partition I

- `mount /dev/sda3 /mnt`
- `mkdir /mnt/home`
- `mount /dev/sda4 /mnt/home`
- check mounting points with `lsblk`

Install to Partition II

- `pacstrap /mnt base linux linux-firmware`

Sidenote

- `linux-firmware` is required in order that the wifi adapter will be automatically recognized after installation is completed
- install linux kernels:
  `pacman -S linux linux-headers linux-lts linux-lts-headers` (long term support kernel to have the possibility to choose if the other one stops working
- all installed previously has to be installed again, because it was only in the installation media

- `genfstab -U /mnt >> /mnt/etc/fstab` (generate filesystem table file)

Install to Partition III

- `cat /mnt/etc/fstab` (take a look at it)
- other options: `-p` and `-h` (help)
- `arch-chroot /mnt` (change into root directory of new installation, now root user in new linux system)
- `/bin/bash`

Sidenote

- create ramdisk for the linux kernel: `mkinitcpio -p linux`, `mkinitcpio -p linux-lts` (or `-P`)

Timezone I

- `ln -sf /usr/share/zoneinfo/Europe/Berlin /etc/localtime` (use `tab` to see possible options)
- `hwclock --systohc --utc`: write the current software UTC time to the hardware clock
- If you specify neither `--utc` nor `--localtime` then the one last given with a set function (`--set`, `--systohc`, or `--adjust`), as recorded in `/etc/adjtime`, will be used. If the adjtime file doesn't exist, the default is UTC
- the `date` time should correspond to current localtime

Timezone II

- `sudo hwclock --show` (does already add up the winter time (+1) and the summer time (+2))
- `vim /etc/locale.gen`, uncomment locale `en_US.UTF-8 UTF-8`
- determines the language, monetary values, time and date formats etc. of the system
- `pacman -S neovim`
- `locale-gen` to generate the choosen locale
- `echo LANG=en_US.UTF-8 > /etc/locale.conf`

Timezone I

timedatectl

- `timedatectl set-ntp true`: Controls whether network time synchronization is active and enabled (if available). If the argument is true, this enables and starts the first existing network synchronization service
- old way: `sudo ntpd -qg` to manually synchronize your clock with the network, ignoring large deviations between local UTC and network UTC
- `timedatectl set-timezone Europe/Berlin`: Set the system time zone to the specified value
- this will create an `/etc/localtime` symlink that points to a zoneinfo file under `/usr/share/zoneinfo/`

Timezone II

timedatectl

- `timedatectl list-timezones`: list available time zones
- `timedatectl set-time [TIME]`: set the system clock to the specified time. This will also update the RTC time accordingly. The time may be specified in the format "2012-10-30 18:17:16".
- `timedatectl`: check the current system clock time (presented both in local time and UTC) as well as the RTC (hardware clock)
- there are two time standards: localtime and Coordinated Universal Time (UTC). The localtime standard is dependent on the current time zone, while UTC is the global time standard and is independent of time zone values

Timezone III

timedatectl

- the standard used by the hardware clock (CMOS clock, the BIOS time) is set by the operating system. By default, Windows uses localtime
- an OS that uses the UTC standard will generally consider the hardware clock as UTC and make an adjustment to it to set the OS time at boot according to the time zone

User and Root I

- `echo ArchPC > /etc/hostname`, type in username
- `passwd` for root user
- `useradd -m -g users -G wheel areo` or
  `useradd -m -G users,wheel areo`
- or `sudo useradd -m (-g username) -G additional_groups -s login_shell username` or
  `useradd -n areo` and
  `usermod -aG wheel.audio.video.optical.storage areo`
- other options: `-s /bin/bash`

User and Root II

- `passwd areo`
- `pacman -S sudo`
- find out if it's installed with `which sudo`
- else `pacman -S which sudo` or just directly `pacman -S base-devel`
- `EDITOR=nvim visudo` to edit sudoers file in nvim and uncomment `%whell ALL=(ALL) ALL`

User and Root III

Sidenote

- user, group and password management tools on Arch Linux come from the shadow package, which is a dependency of the base package

Keyboard-Layout I

- `echo KEYMAP=de-latin1-nodeadkeys > /etc/vconsole.conf`
  - for the tty, but no in X

Network I

- `nvim /etc/hosts`:

```
# blablabla
# blablabla

127.0.0.1 localhost
::1 localhost
127.0.1.1 ArchPC.localdomain ArchPC
```

- `pacman -S networkmanager`
- `systemctl enable NetworkManager` (create symlink)
- `nm-applet`: symbol in systray to configure and have easy access to NetworkManager (`sudo pacman -S network-manager-applet`)

Network II

- put `nm-applet &` into `~/.xinitrc`
- there's a autostart desktop entry automaticaly created under `/etc/xdg/autostart/nm-applet.desktop`
- i3 already autostarts it in it's `~/.configs/i3/config`:
  `exec --no-startup-id nm-applet`

Sidenote

- there is also `yay -S networkmanager-dmenu-git`.

Network III

Sidenote

- other packages:
  `pacman -S wpa_supplicant wireless_tools netctl`, if there's no wired connection one can use `wifi-menu` from the `netctl` package.

Grub I

- `pacman -S grub efbootmgr dosfstools os-prober mtools`
- `mkdir /boot/EFI`
- `mount /dev/sda1 /boot/EFI`
- `grub-install --target=x86_64-efi --bootloader-id=OSName`
  - `x86_64-efi` is for x86˙64 systems

Grub II

- other options: `--efi-directory=/boot/EFI --removable` or `--bootloader-id=GRUB` (bootloader identifier, here named GRUB. A directory of that name will be created in `esp/EFI/` to store the EFI binary and this is the name that will appear in the UEFI boot menu to identify the GRUB boot entry)
- by default the generation scripts automatically add menu entries for all installed Arch Linux kernels to the generated configuration. After installing or removing a kernel, you just need to re-run the above grub-mkconfig command

Grub III

Sidenote

- `mkdir /boot/grub/locale` and then `cp /usr/share/locale/en\@quot/LC_MESSAGES/grub.mo /boot/grub/locale/en.mo` is propably not rly needed
- `--recheck` propably not rly needed

- Dualbooot with Windows:
  - use the EFI-Partition from Windows: `mount /dev/sda1 /boot/EFI`

Grub IV

- if two EFI-Partitions exist (one from Windows: `/dev/sda1` and one for Arch: `dev/sda5`): `mount /dev/sda1/ /mnt` (EFI-Partition of Windows has to be mounted, so that the os-prober can find it) or `mdkdir /mnt2` and `mount /dev/sda1/ /mnt2`

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

Sidenote

- or don't use grub and just choose with e.g. `f12` a bootloader from the bootmenu (maybe has to be enabled in the uefi-firmware settings)

Swapile (Optional)

```
[root@archiso /]# fallocate -l 2G /swapfile
[root@archiso /]# chmod 600 /swapfile
[root@archiso /]# mkswap /swapfile
Setting up swapspace version 1, size = 2 GiB (2147479552 bytes)
no label, UUID=3fdd0cc7-f04a-4c70-aee8-66271aaac211
```

```
[root@archiso /]# echo '/swapfile none swap sw 0 0' | tee -a /etc/fstab
/swapfile none swap sw 0 0
[root@archiso /]# cat /etc/fstab
# /dev/nvme0n1p2
UUID=dd243497-8dd6-4d5b-bc44-28292763e059       /               ext4            rw,relatime     0 1

# /dev/nvme0n1p3
UUID=9145035f-3acf-487a-ac2b-69dd55d61e67       /home           ext4            rw,relatime     0 2

/swapfile none swap sw 0 0
```

Finish

- `exit`, to exit out of chroot
- `umount -R /mnt`
- `umount -l /mnt` (to force unmount) or `umount -a`
- `poweroff` (not `reboot` to remove the iso from storage in virtualbox)
- in the UFEI firmware settings choose the right bootloader for the esp on which the bootloader was installed (maybe secure boot has to be enabled for this) and give the esp with the bootloader the highest boot priority

After Base Installation

General

- if one forgot one step in the base installation with `su`, one can get root again.
- `sudo pacman -S base-devel, xorg-xkill, man-db texinfo openssh e2fsprogs, dialog`: `base-devel` is for building aur packages and `sudo` and `which` are in there, enable `openssh` with `systemsctl enable sshd`, `dialog` is a cli-textbox some programs use.
- if sth. goes wrong with the DE one can change tty with `ctrl + alt + fX` and make e.g. `killall i3`.

Desktop-Environment / WM I

- `sudo pacman -S xorg-server xorg-xinit`
- i3:
  - `sudo pacman -S i3-gaps i3status alacritty dmenu`
  - install fonts (i3 doesn't pull fonts), e.g. `sudo pacman -S noto-fonts`
- xfce:
  - `sudo pacman -S xfce4`
- `cp /etc/X11/xinit/xinitrc /home/areo/.xinitrc`

Desktop-Environment / WM II

- `nvim ~/.xinitrc`: write `exex i3` or `exec xfce4-session` in there
- `startx` to start
- `xrandr` to show all available screen resolutions and then e.g. `xrandr -s 1920x780`

Start DE directly after login or set up a display manager (login screen)

- `~/.zshrc` or `~/.bash_profile`:

```
if [[ "$(tty)" = "/dev/tty1" ]]; then
  pgrep  startx
fi
```

- displaymanager:
  - `sudo pacman -S lightdm lightdm-gtk-greeter`
  - `sudo systemctl enable lightdm.service`: systemd command to tell systemd to start lightdm when one does log in
  - useful to be able to choose between differnt desktop environments

Compiling yay (make arch package)

- `git clone https://aur.archlinux.org/yay-git.git`
- `cd yay-git` and then `makepkg -si`
  - `base-devel` needed for it

Arch in Virtualbox (in case)

- `pacman -S virtualbox-guest-utils xf86-video-vmware`

Wifi

- NetworkManager manages everything ones it is activated (ethernet an wifi)
- `wifi-menu` doesn't work once the NetworkManager is activated or if there's already a ethernet connection
- `nmcli device wifi list`
- `nmcli device wifi connect 'FRITZ!Box Gastzugang Herbert' password PASSWORD`

CPU/GPU

- `pacman -S amd-ucode` or `pacman -S intel-ucode`
- `pacman -S xf86-video-intel`
- `pacman -S mesa` (if intel or amd for graphics) or
  `pacman -S nvidea nvidea-utils` (nvideo for graphics) and
  `pacman -S nvidea-lts` (if one installed the lts-kernel)

Right Keyboard Layout in Xorg I

- for xorg the keyboard layout isn't related to the keyboard layout in the tty with it's file: `/etc/vconsole.conf` but has to be configured in e.g. `/etc/X11/xorg.conf.d/00-keyboard.conf` (one of many keyboard layouts for xorg)
  - xorg.conf is parsed by the X server at start-up. To apply changes, restart X
- get overview:

```
localectl list\itemx11-keymap-models
localectl list\itemx11-keymap-layouts
localectl list\itemx11-keymap-variants [layout] (e.g. de)
localectl list\itemx11-keymap-options
```

Right Keyboard Layout in Xorg II

- set one for the current session: `sudo setxkbmap de nodeadkeys` or `sudo setxkbmap -layout de -variant nodeadkeys` (long variant)
  - `setxkbmap [-model xkb_model] [-layout xkb_layout] [-variant xkb_variant] [-option xkb_options]`
  - or persistent in `~/.xinitrc`
- make persistent in `/etc/X11/xorg.conf.d`:
  - `localectl set-x11-keymap de "" nodeadkeys ""`: autogenerates the keyboard layout file
  - `localectl [--no-convert] set-x11-keymap layout [model [variant [options]]]`

Right Keyboard Layout in Xorg III

- if `--no-convert` option is passed, the specified keymap is also converted to the closest matching console keymap and applied to the console configuration in `vconsole.conf`
- to set a model, variant or options, all preceding fields need to be specified, but the preceding fields can be skipped by passing an empty string with

Desktop Background

- `feh --bg-scale "/home/areo/Pictures/Wallpaper/linux wallpaper/urban-1597922375998-8560.jpg"`
  - best into `~/.xinitrc`

Sound

- `sudo pacman -S pulseaudio`
- `/usr/bin/start-pulseaudio-x11`
  - best into `~/.xinitrc`
- `pavucontrol` is a gui to have an overview

Compositor

- `picom &`
  - best into `~/.xinitrc`

Screen-Brightness

- `sys/class/backlight`

Screenshot

- `scrot` (→ configuration in `~/.config/i3/config` file)

SysRq-Key einsetzen

- reboot: `alt` + `print` + each of `reisub`.
- shut down: `alt` + `print` + each of `reisuo`.
- Bedeutung der Keys kann hier nachgelesen werden: https://en.wikipedia.org/wiki/Magic_SysRq_key.

Sidenote

- Nach dem Auslösen von e sollte man den Prozessen ein paar Sekunden Zeit lassen, der Aufforderung, sich sauber zu beenden, nachzukommen.
- SysRq may be released before pressing the command key, as long as Alt remains held down.
- this keys are for the querty keyboard.

SysRq-Key aktivieren

- direkt aktivieren, aber nicht persistent `echo "1" | sudo tee /proc/sys/kernel/sysrq`.
- persistent aktivieren `> echo kernel.sysrq=1 | sudo tee /etc/sysctl.d/99-sysctl.conf`.

Sidenote

- geht auch: `> sudo bash -c "echo kernel.sysrq=1 > /etc/sysctl.d/99-sysctl.conf"`.
- ~~`> sudo echo "kernel.sysrq=1" > /etc/sysctl.d/99-sysctl.conf`~~.
  - The reason it doesn't work is that ones gives root privileges to echo, which it doesn't need to print to stdout. It's bash doing the writing to file and that's running under your user.
- `tee -a` or `>>` for appending.

SysRq-Key checken

- `cat /proc/sys/kernel/sysrq`
  - 0 - disable sysrq completely.
  - 1 - enable all functions of sysrq.
  - 2 - enable control of console logging level.
  - 4 - enable control of keyboard (SAK, unraw).
  - 8 - enable debugging dumps of processes etc.
  - 16 - enable sync command.
  - 32 - enable remount read-only.
  - 64 - enable signaling of processes (term, kill, oom-kill).
  - 128 - allow reboot/poweroff.
  - 256 - allow nicing of all RT tasks.

Sidenote

- 438 is obtained from the sum of $2 + 4 + 16 + 32 + 128 + 256$, so all the corresponding functions are enabled.

Restore boobtable USB-Stick to normal

Explanation

- if one writes a iso-image onto a flash drive there're e.g 2 partitions encoded in the iso image and a lot of free space.
- if one writes a iso to the flash drive, it will get a boot flag (that can be seen with `sudo fdisk -l`. If one only formats it, it won't work correctly (can't remove partitions etc.).
- one has to wipe filesystem completely from the flash drive, to restore it to it's original state.

Restore boobtable USB-Stick to normal I

Format / repartition a storage device

- `sudo fdisk -l`
- `lsblk -o NAME,FSTYPE,SIZE,MOUNTPOINT,LABEL`
- `sudo wipefs --all /dev/sdc`
  - whole drive, not `sdc1`!
- `sudo cfdisk /dev/sdc`
  - GPT wählen

Restore boobtable USB-Stick to normal II

Format / repartition a storage device

- DOS ist eine andere Bezeichnung für MBR
- `sudo mkfs.ext4 /dev/sdc1`
  - `-n 'My_USB'` to give it a name
- `sudo chmod 755 . -R`
- `sudo chown areo:users . -R`

Restore boobtable USB-Stick to normal III

Format / repartition a storage device

Sidenote

- need to `sudo umount /dev/sdX` flash drive before `wipefs` / `mkfs.vfat` etc.

Literatur

Bücher

Artikel

Vorlesungen

Online

Sonstiges