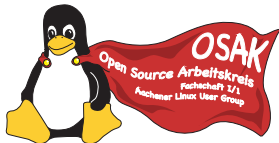


Über das Buildhauen oder Das casper'le Theater

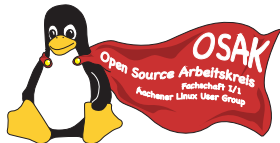
Christopher & Daniel

Open Source Arbeitskreis
der Fachschaft Mathematik/Physik/Informatik an der RWTH Aachen
und der Aachener Linux User Group

26. September 2015



- 1 Der Bootvorgang
- 2 Inhalt eines LIPIImages
- 3 Aufbau und Arbeitsweise von LIPCK
- 4 Erste Schritte



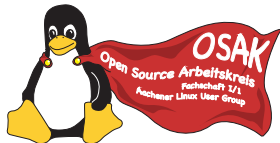
squashfs Readonly Dateisystem

initramfs Minimales Userland – im Arbeitsspeicher

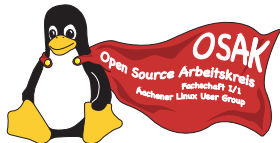
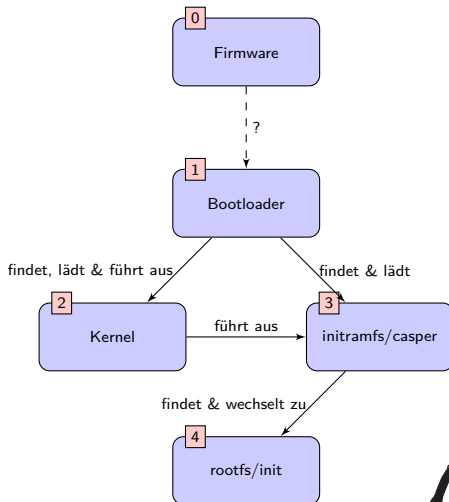
- ▶ als Archiv auf ISO/Image

casper Debian LiveCD Umgebung

- ▶ Kernel
- ▶ initramfs
- ▶ ISO/Image

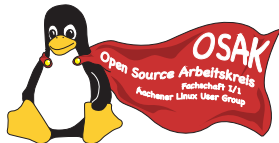
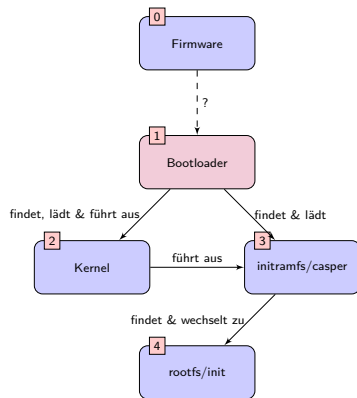


Bootvorgang – Übersicht



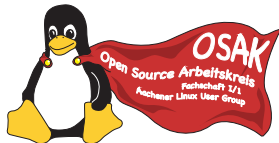
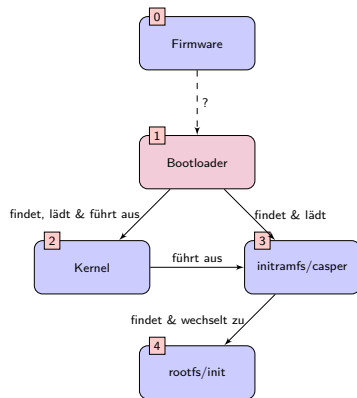
Bootvorgang – Bootloader

- ▶ shim+grub: 64Bit (secure boot) EFI
- ▶ syslinux: BIOS
- ▶ sucht den USB-Stick
- ▶ lädt Kernel und initramfs
- ▶ startet den Kernel



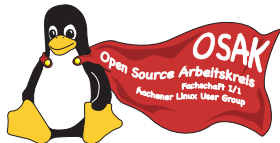
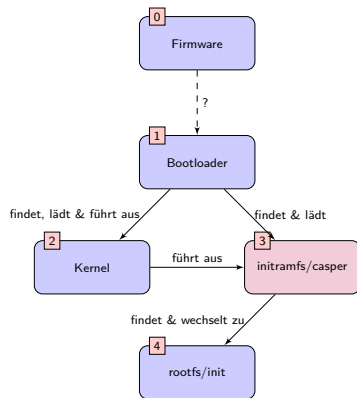
Bootvorgang – Bootloader

- ▶ shim+grub: 64Bit (secure boot) EFI
- ▶ ~~syslinux: BIOS~~ grub: BIOS, EFI 32/64Bit
- ▶ sucht den USB-Stick
- ▶ lädt Kernel und initramfs
- ▶ startet den Kernel
- ▶ Erkennung 32/64Bit
- ▶ selektieren des korrekten Kernels/initramfs
- ▶ GParted und FreeDOS starten



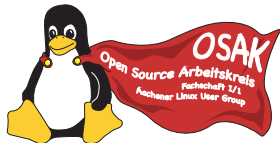
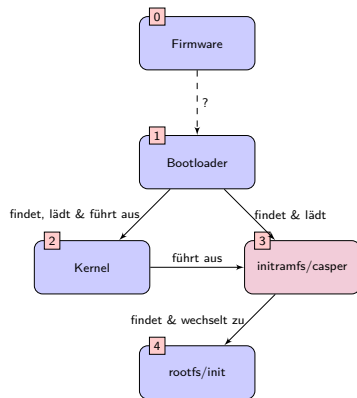
Bootvorgang – initramfs/casper

- ▶ Achtung: nur eingeschränkte busybox
- ▶ sucht den USB-Stick
- ▶ baut das rootfs zusammen
- ▶ legt den Benutzer des Live-Systems an
- ▶ Manipulation des rootfs
- ▶ lädt Kernel Module



Bootvorgang – initramfs/casper

- ▶ Achtung: nur eingeschränkte busybox
- ▶ sucht den USB-Stick
- ▶ baut das rootfs **besser/anders** zusammen
- ▶ legt den Benutzer des Live-Systems an
- ▶ Manipulation des rootfs
- ▶ lädt Kernel Module
- ▶ **nmtelekinese**
- ▶ **lädt Hooks vom USB-Stick nach**
 - ▶ **lipnsa**

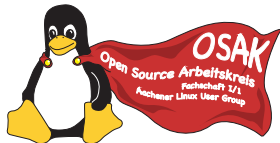


Bootvorgang – rootfs

Aus Sicht des initramfs:

filesystem.squashfs	squashfs	/filesystem.squashfs
tmpfs	tmpfs	/cow
/filesystem.squashfs /cow	overlayfs	/root
/filesystem.squashfs	bindmount	/root/rofs
/cow	bindmount	/root/cow

- „richtiges“ rootfs wird /root



Bootvorgang – rootfs

Aus Sicht des initramfs:

lipcommon.squashfs

lip\$\$\$bit.squashfs

tmpfs

/lipcommon.squashfs /lip\$\$\$bit.squashfs

/overlay /cow

/overlay

/cow

squashfs /lipcommon.squashfs

squashfs /lip\$\$\$bit.squashfs

tmpfs /cow

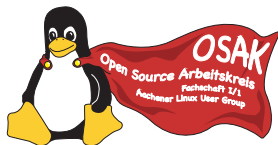
overlayfs /overlay

overlayfs /root

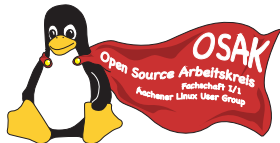
bindmount /root/rofs

bindmount /root/cow

- ▶ „richtiges“ rootfs wird /root
- ▶ *modprobe.d* und *sysctl.d* angepasst
- ▶ diverse Pakete sind vorinstalliert
- ▶ Deduplizierung von 64Bit und 32Bit

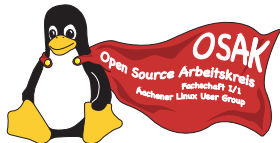


- 1 Der Bootvorgang
- 2 Inhalt eines LIPIImages
- 3 Aufbau und Arbeitsweise von LIPCK
- 4 Erste Schritte



LIPImage – versionierte Dateien

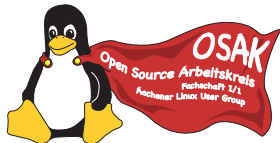
- ▶ aus dem lipstick git
- ▶ lassen sich mit git einfach aktualisieren
- ▶ zum großen Teil handgeschrieben
- ▶ kleine Dateien



LIPImage – versionierte Dateien

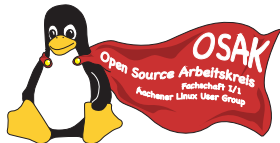
- ▶ aus dem lipstick git
- ▶ lassen sich mit git einfach aktualisieren
- ▶ zum großen Teil handgeschrieben
- ▶ kleine Dateien

- ▶ grub Konfiguration
- ▶ linuxparty.sh inklusive Paketdokumentation
- ▶ Hilfsskripte
- ▶ initrd Hooks (lipnsa)
- ▶ EFI-Shell, FreeDOS



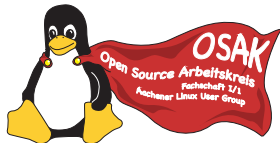
LIPImage – generierte Dateien

- ▶ werden von LIPCK generiert
- ▶ Aktualisierung nur durch Neubau des Image möglich
- ▶ sollten flexibel sein
- ▶ große oder stark abhängige Dateien

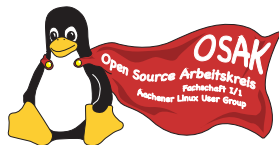


LIPImage – generierte Dateien

- ▶ werden von LIPCK generiert
- ▶ Aktualisierung nur durch Neubau des Image möglich
- ▶ sollten flexibel sein
- ▶ große oder stark abhängige Dateien
- ▶ Bootloader (4x grub + 1x shim)
- ▶ Kernel und initramfs (je 2x)
- ▶ rootfs (common, 32Bit, 64Bit)
- ▶ .disk

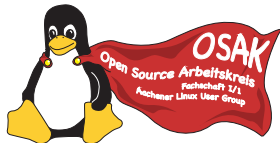


- 1 Der Bootvorgang
- 2 Inhalt eines LIPIImages
- 3 Aufbau und Arbeitsweise von LIPCK
- 4 Erste Schritte



Was ist LIPCK?

LIPCK = **L**inux **I**nstall **P**arty **C**ustomization **K**it

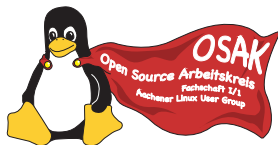


Was ist LIPCK?

LIPCK = **L**inux **I**nstall **P**arty **C**ustomization **K**it

Was ist LIPCK **nicht**?

- ▶ Ein allgemeines Remaster-Tool
- ▶ Software für Laien



Was ist LIPCK?

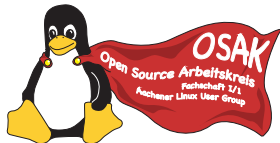
LIPCK = **L**inux **I**nstall **P**arty **C**ustomization **K**it

Was ist LIPCK **nicht**?

- ▶ Ein allgemeines Remaster-Tool
- ▶ Software für Laien

Und was ist LIPCK nun wirklich?

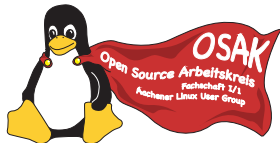
- ▶ Expertensoftware
- ▶ Geschichte
- ▶ Dokumentation
- ▶ Unterstützung



LIPCK ist in GNU make implementiert

Einfache Regel:

```
rootfs_console : $(call archdir,$(ARCH))$(STATE_DIR)/  
    rootfs_extracted | $(APT_CACHE_DIR)  
$(MAKE) ARCH=$(ARCH) rootfs_prepare  
mkdir -p "$$(call archdir,$(ARCH))$(LXC_DIR)"  
...
```



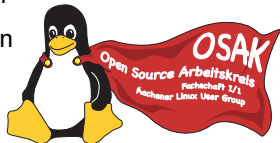
LIPCK – Beispiel 2

```
rootfs_checksums : $(ARCH_DIR)$(CHECKSUMS)

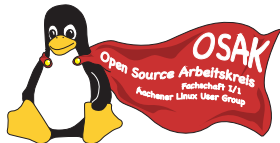
$(call gentargets,$(CHECKSUMS)) : $(call archdir,%)$(STATE_DIR)/
    rootfs_remastered

    cd "$(call archdir,$*)$(ROOTFS)" \
    && find . -type f -print0 | sort -z | xargs -0 md5sum \
    > "$(call archdir,$*)$(CHECKSUMS)"
```

- ▶ \$(ARCH_DIR) verweist auf die primäre Architektur
- ▶ gentargets erzeugt Targets für beide Architekturen
- ▶ % und \$* sind Platzhalter für die Architektur

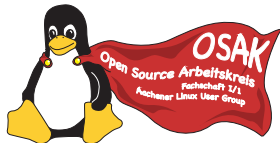


- 1 Der Bootvorgang
- 2 Inhalt eines LIPIImages
- 3 Aufbau und Arbeitsweise von LIPCK
- 4 Erste Schritte



Erste Schritte

- 1 Neuen Branch im *lipstick* git anlegen
- 2 config/Makefile.conf anpassen; dabei nachdenken
- 3 Sind die Paketlisten in contrib aktuell?
- 4 `$ sudo make clean_really_all`
- 5 `$ sudo make image`
 - ▶ Fehler verstehen
 - ▶ LIPCK anpassen
 - ▶ immer **verallgemeinern**, nie neu spezialisieren
 - ▶ nochmal!
- 6 `$ sudo make repo`
- 7 `$ sudo make image_deploy \`
`IMAGE_FILE=/my/final/image`



Erste Schritte – Infrastruktur

Repositories

- ▶ FS-GitLab
- ▶ `git@git.fsmpi.rwth-aachen.de:osak/lipstick.git`
- ▶ `git@git.fsmpi.rwth-aachen.de:osak/lipck.git`

OSAK Build-Server

- ▶ `osak-build.fsmpi.rwth-aachen.de`
- ▶ nur aus dem RWTH-Netz erreichbar
- ▶ `$ sudo -i -u buildhauer`
- ▶ `$ screen -x || screen`

