

Raspberry Pi als LTSP Thinclient verwenden

Berry Terminal

Nativer LTSP-Client

...

Vorbereitung und Konfiguration

<http://cascadia.debian.net/trenza/Documentation/raspberrypi-ltsp-howto/>

Benötigte Pakete:

```
apt-get install ltsp-server gemu-user-static binfmt-support ldm-server
```

Außer gemu-user-static und binfmt-support, die für die Cross-Plattform-Unterstützung nötig sind, sollte alles auf einem laufenden LTSP-Server bereits installiert sein.

Jetzt brauchen wir noch den gnupg-Key der Raspian Debian Binaries und müssen diesen in einen Keyring exportieren, der von der LTSP-Build-Umgebung genutzt werden kann:

Raspian-Gpg-Key suchen und importieren:

```
gpg --search-keys 90FDDD2E  
# jetzt die Nummer des gefundenen Keys zum Importieren angeben
```

In einen extra Keyring exportieren:

```
gpg --export 90FDDD2E >> /etc/ltsp/raspbian.public.key.gpg
```

Wir benötigen noch eine eigene ltsp-Konfiguration mit Angaben für den speziellen Raspian Kernel und den Keyring:

[/etc/ltsp/ltsp-raspbian.conf](#)

```
DEBOOTSTRAP_KEYRING=/etc/ltsp/raspbian.public.key.gpg  
DIST=wheezy  
# For alternate raspbian mirrors,  
# see: <http://www.raspbian.org/RaspbianMirrors>  
MIRROR=http://archive.raspbian.org/raspbian  
SECURITY_MIRROR=none  
KERNEL_PACKAGES=linux-image-3.2.0-4-rpi
```

Client Image bauen und ggf. konfigurieren

Jetzt kann das Client Image wie gewohnt für LTSP gebaut werden:

```
ltsp-build-client --arch armhf --config /etc/ltsp/ltsp-raspbian.conf
```

Für den Fall, dass die ltsp-chroots per nfs exportiert werden (bei Debian Wheezy der default) und nicht squashfs-Images per nbd, liegt die `lts.conf` Konfiguration unter `/opt/ltsp/ARCH/etc/lts.conf`, wobei ARCH in diesem Fall mit armhf ersetzt werden muss.

Das bedeutet, dass für jede Konfiguration eine eigene lts.conf existiert. Evtl. lässt sich das durch symlinks aber auch zentralisieren.

Kernel und Firmware auf die SD-Karte kopieren

Download raspberry pi foundation's raspbian image: <http://www.raspberrypi.org/downloads>. This howto used 2013-02-09-wheezy-raspbian

Copy firmware files to an SD card (/dev/sdc1, replace with the device name for your SD card):

```
mount /dev/sdc1 /mnt mkdir -p /media/rpi mount -o loop,offset=$1) /srv/src/2013-02-09-wheezy-raspbian.img /media/rpi cp -r /media/rpi/* /mnt/ umount /media/rpi
```

Copy kernel and initrd to your SD card:

```
cp -vb /opt/ltsp/armhf/boot/vmlinuz-3.2.0-4-rpi /mnt cp -vb /opt/ltsp/armhf/boot/initrd.img-3.2.0-4-rpi /mnt
```

edit cmdline.txt:

add `init=/sbin/init-ltsp boot=nfs nfsroot=192.168.67.1:/opt/ltsp/armhf` delete the `root=` and `rootfstype=` entries:

edit it with sed:

```
sed -i -e 's,root=.*rootfstype=ext4,boot=nfs init=/sbin/init-ltsp\nnfsroot=192.168.67.1:/opt/ltsp/armhf,g' /mnt/cmdline.txt
```

edit config.txt to use the kernel and initramfs:

```
echo 'kernel vmlinuz-3.2.0-4-rpi' » /mnt/config.txt echo 'ramfsfile initrd.img-3.2.0-4-rpi' »\nmnt/config.txt echo 'ramfsaddr 0x00800000' » /mnt/config.txt
```

Copyright 2013 Vagrant Cascadian vagrant@debian.org This work is licensed under a: Creative Commons Attribution-ShareAlike 3.0 Unported License

<http://creativecommons.org/licenses/by-sa/3.0/deed.en_US>

¹⁾

512*8192

From:

<https://wiki.datenkollektiv.net/> - **datenkollektiv.net**

Permanent link:

<https://wiki.datenkollektiv.net/public/ltsp/raspberry?rev=1427100756>

Last update: **2015/03/23 09:52**

