

Lab 4

Kernel Patching and Cross-Compilation for RPi Embedded Linux

Julia Desmazes
Michael Nissen

October 24, 2017

1 Download / Set the good version of Linux sources

To execute the tasks for this report, the first step is to get the Raspberry Pi cross-compilation tools, and specific version of the linux kernel that one wants to patch. This is going to be done using `git clone`, which ofcourse assumes a working installation of `git`.

To get the latest version of the kernel source code and the cross-compilation tools, the following commands will be copied to the terminal:

```
1 >> git clone https://github.com/raspberrypi/tools.git
2 >> git cone -b rpi-4.4.y https://github.com/raspberrypi/linux.git
```

One now has to make sure to get the same configuration as is used by ones Raspberry Pi during the kernel compilation, which can be done by checking out the `git branch` that corresponds to the hash of the Linux kernel running on the Raspberry Pi.

This will be done by by running the shells script seen in Code Snippet 1. This shell script will grep the firmware hash found on the Raspberry pi, and then get the `git hash` from the github link relative to the hardware hash.

```
1 #!/bin/bash
2
3 PLATFORM='uname -s '
4
5 if [ ${PLATFORM} = "Darwin" ]
6 then
7     CMD="gunzip -c"
8 elif [ ${PLATFORM} = "Linux" ]
9 then
10    CMD="zcat"
11 else
12    echo "Sorry, the platform ${PLATFORM} is not supported !!!"
13    exit -1
14 fi
15
16 FWHASH='${CMD} /usr/share/doc/raspberrypi-bootloader/changelog.Debian.gz | grep -m 1 'as
17 of' | awk '{print $NF}''
18 #echo "Firmware Hashcode: fwhash = $FWHASH"
19
20 LINUXHASH='wget -qO- https://raw.githubusercontent.com/raspberrypi/firmware/$FWHASH/extra/git_hash '
21
22 echo "Linux Hashcode: linuxhash = $LINUXHASH"
```

Code Snippet 1: Shell script to get Linux kernel git hash

When one has figured out the `git hash`, one can change directory to the linux git repository and checkout the respective brach by the following command:

```
1 >> cd linux
2 >> git checkout <git_hash>
3 >> make clean -mrproper
```

At line 3, we are cleaning kernel tree from all unneeded files of the first version which is a recommended practice prior to each kernel compilation!

2 Patch the Kernel

Before starting to patch the kernel, one should make sure to identify the Linux sources that one is trying to install. This can be done using the `head` command, which will look like this:

```
1 >> head Makefile
2
3 # Ouput generated by running head on the Makefile #
4 VERSION = 4
5 PATCHLEVEL = 4
6 SUBLEVEL = 21
7 EXTRAVERSION =
8 NAME = Blurry Fish Butt
```

Now that one know the specific version of the Linux kernel, it is time to download the latest PREEMPT-RT patch that corresponds to ones kernel version - if follows the format: `version.patch-level.sub-level`.

To download the patch from the patch from `kernel.org` using the `wget` command like so: `wget https://cd.kernel.org/`

Now one can actually start patching the kernel sources by executing the following commands in succession:

```
1 host$ gunzip patch-<version.patch-level.sub-level>-rt<last>.patch.gz
2 host$ cat patch-<version.patch-level.sub-level>-rt<last>.patch | patch -p1
```