ZRouter: Remote update of firmware

Hiroki Mori
yamori813@yahoo.co.jp
Editor: Mchael Zhilin

ZRouter is a system that builds FreeBSD for small targets like routers. In addition to building ordinary kernel and commands, you can create images that can be used with u-boot etc. In FreeBSD 12R, SDRAM 16M / Flash 4M is the lowest line spec. ZRouter is mainly targeting modules using SOC of mips.

Flash has built-in target, and it starts from there. This is inside of Flash.

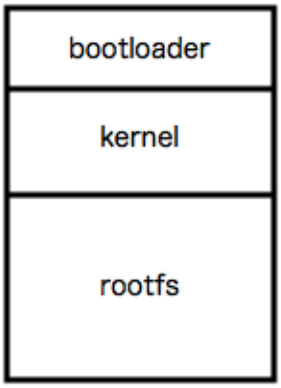

Figure 1: Flash map

Inside embedded modules there is flash memory chip, which contains image with operating system and application programs. This image is read-only and usually compressed, but ability to update image is required.

Image 1: CFI Flash chip

What is the reason why update is necessary? For instance, added new function of the program or fix program defect. It can also correspond to security considerations.

In the early stages of development, it's possible to use UART (serial interface) to force boot loader to update Flash. However, if number of devices increases or serial connection is not available at time, updating of flash image via the network improves operational efficiency.

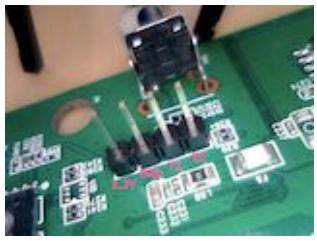

Image 2: UART Connector

Let's consider this method with SOHO (Small Office/Home Office) router based on SoC of MIPS architecture.

ZRouter project provides FreeBSD build tool with routine to update flash image. But there were two problems:

- root partition (rootfs) was mounted and there was possibility of failure while updating flash image (that contains rootfs)
- since libraries were statically linked to the routine, the binary size was enlarged

When updating flash, it is necessary to unmount root partition and resource necessary

for execution should be independent from flash.

The functionality called reroot is present in FreeBSD since 10.3 release. This functionality does not completely reset the system during reboot but it restarts boot process from mounting root partition. This behavior is provided by flag “-r” of “reroot” routine (see reboot(8)).

Using this function, I thought about a method of updating Flash by transferring all resources necessary for execution to memory.

The routine “reroot” identifies the next rootfs from kernel environment variables (also known as kenv(1)). If you put a minimalistic root partition on the memory disk (see mdconfig(8)) and set it as rootfs, you can put the flash resources in a state not used by runtime environment:

```
kenv vfs.root.mountfrom =
    cd9660:md0.uzip
```

If flash chip size is 4 or 8 megabytes, you can copy current compressed flash root partition to the memory disk as is and choose it as next root partition after “reroot” operation.

If flash chip size is 16 megabytes or more, you can copy necessary files to the memory disk and reconstruct rootfs. Because the original compressed rootfs is bigger than UFS filesystem on uncompressed memory disks.

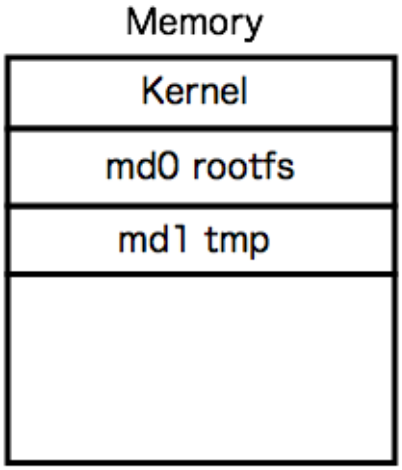

Figure 2: memory layout at flash update

Since the kernel was originally running on the memory and rootfs is also moved to the memory, functions of operating system does not affect any operation with flash chip.

Sometimes rerooting fails when target box has small amount of memory. Thanks to debugging, it has been found out that the function of reroot is using TMPFS, but there was a remaining memory check code in TMPFS and an error occurred due to lack of memory.

As solution, kernel option was introduced to reduce the check size to pass through this process (see review request D13583). If the check size is set to 1 MiB, target boxes with 16 MiB of memory can be rerooted without problem.

After rerooting, we download a new image file from the server via TFTP protocol according to the configuration file, and execute script to write image to Flash via dd & pipe & rc.

This mechanism is provided by ZRouter's profile “reupdate”. It worths to note that It is also possible to develop and execute a command that accepts HTTP download function.

For upgrading, it is necessary to define the area to write the image with geom_map or geom_flash. In Figure 3, the free space that can be written as an upgrade as a partition is taken as a partition.

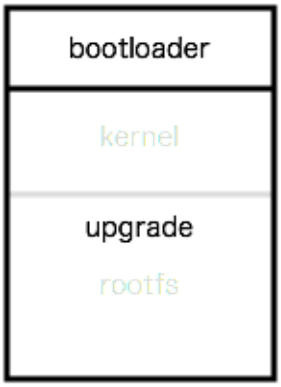

Figure 3: upgrade partition

The flash area is partitioned from geom_map (via hints) and geom_flashmap (via FDT tree). In the case of u-boot image layout, rootfs partition is placed next to kernel partition. Kernel size can be different for firmware versions, so geom_map has a scanning function to identify rootfs partition's starting address. But geom_flashmap has lack of this scanning functions. To add it, patch has been proposed (see review request D13648).

Various targets with SPI flash has been tested. It works with no a problem, but update of CFI flash failed. Patch has bee proposed, but not yet committed (see review request D14279)

ZRouter make image MD5 value at build time. It is thought that by checking the hash value of the image to be updated with the following script, it is possible to prevent security and mistakes.

```
mkfifo /tmp/tftp
MD5=`cat /tmp/tftp | md5&
echo "bin
get Fon_FON2305E_FDT.zimage /tmp/tftp
quit" | tftp 10.10.10.3 69 >/dev/null 2>&1`
echo $MD5
```

Although this mechanism can be used for environments with u-boot boot loader, it is not compatible with the environment with RedBoot boot loader. RedBoot approach is that there is no need to deal with remote control because there is a remote control function in the boot loader itself so that it can be remotely updated.

The different concerns should be considered. At first, fixability and testability: if remote update fails, it is necessary to operate with serial. Then security: if it is a closed network, there is no problem with TFTP usage, but if it is an open network it will be necessary to consider secure protocols different from TFTP. Finally, access to box: if remote update fails, serial operation is required, but if it is stable to a certain extent, remote update seems very useful.

Because it is difficult to read the whole image into memory due to memory constraint, it is difficult to perfect check and it is thought that method registration is necessary.

Consideration about correspondence with NAND memory is also necessary

Finally, this mechanism was developed thanks to the excellent build environment called ZRouter. I would like to thank Oleksandr Rybalko who started ZRouter.
I also thank you for implementing reroot.

Reference

[1] ZRouter.org
https://zrouter.org/

[2] How to put FreeBSD power into small MIPS switch/router
Oleksandr Rybalko
EuroBSDcon2012