

Debian linux kernel / armmp フレーバ

東京エリア Debian 勉強会 第102回 2013年7月度

岩松 信洋


2013年7月20日

arm multi platform とその仕組み

- arm multi platform(ARMMP)。
- 一つの linux kernel バイナリで複数の ARM Soc、ターゲットボードをサポートする仕組み。
- カーネル起動時に渡される Device Tree (DT) の blob によって有効化され動作するように初期化される。

ARMMP カーネルの起動方法

起動方法は3つほどある。



zImage に dtb (DT blob) を付加する。そしてその zImage を起動する

カーネルの CONFIG_ARM_APPENDED_DTB が必要。

```
$ cat zImage foo.dtb > zImage+dtb
```

```
$ tftpboot 2000000 zImage+dtb  
$ go 2000000
```

先で作成したzImage を uImage に変換して起動する

```
$ cat zImage foo.dtb > zImage+dtb  
$ cut -f 3- -d ' ' < arch/${ARCH}/boot/.uImage.cmd | \  
sed -e 's/zImage/zImage+dtb/g' \  
-e 's/uImage/uImage+dtb/g'
```

```
$ tftpbboot 2000000 uImage+dtb  
$ bootm
```

先で作成した zImage を ulmage に変換して起動する

- ただしカーネルがロードされるアドレスなどは SoC やボード毎に異なるので、AMMMMP の場合は mkimage 実行時にアドレスを指定する必要がある。
- -a でロードアドレス、-e でエントリポイントアドレスを指定。

```
$ cat zImage foo.dtb > zImage+dtb
$ mkimage -A arm -O linux -T kernel -C none \
-a 0x2000000 -e 0x2000040 -n 'Linux-marvell' \
-d zImage+dtb uImage+dtb
```

ulmage と dtb blob 別に読み込んで起動する。 (ブートローダ依存)

- ARMMP は一つのバイナリで複数の ARM SoC、ボードをサポートすることが目的なので、上記の方法ではサポートできない。
- ulmage と dtb blob を分けて起動させるのが理想。
- U-Boot の場合は カーネルを起動するコマンド bootm を使う。

```
$ make zImage  
$ mkimage -A arm -O linux -T kernel -C none \  
-a 0x2000000 -e 0x2000040 -n 'Linux-marvell' \  
-d arch/arm/boot/zImage uImage
```

ulmage と dtb blob 別に読み込んで起動する。 (ブートローダ依存)

```
$ tftpboot 2000000 uImage  
$ tftpboot 3000000 dtb  
$ bootm 2000000 - 3000000
```

- 第1引数は ulmage がロードされているアドレス
- 第2引数は ulnitrd(initrd イメージの ulmage) がロードされているアドレス
- 第3引数には dtb がロードされているアドレス
- -は 指定なし

カーネルモジュールの対応

- カーネルモジュールも DT によって設定できる。
- 必要なカーネルドライバを組み込みにしてカーネルを起動させるのもよいがカーネルが肥大化する。
- SoC のコア部分は最低限の機能は組み込みを設定し、各デバイス用ドライバはモジュールにして `initrd` などからロードするようにするのがよい。

Debian でのサポート

- Debian で ARMMP をサポートするため、armhf に armmp フレーバを追加。
- Plat'Home さんの Openblocks AX3 を Debian でサポート。
- どうせ後で必要になるし。

Debian でのカーネル提供と利用方法

- Debian はカーネルイメージを `ulmimage` で提供していない。 `zlmimage(vmlinuz)` のみ。
- カーネルがロードされるアドレスが固定値になってしまうので、複数のデバイスをサポートできない。

- 設定されたデータをもとにカーネルや initrd を U-Boot などが扱える形式に変換、フラッシュメモリに書き込む機能等を提供する

```
Machine: Marvell Armada 370/XP (Device Tree)
Kernel-Flavors: armmp
DTB-Id: armada-xp-openblocks-ax3-4.dtb
DTB-Append: yes
U-Boot-Kernel-Address: 0x2000000
U-Boot-Initrd-Address: 0x0
Boot-Device: /dev/sda1
Boot-Kernel-Path: /boot/uImage
Boot-Initrd-Path: /boot/uInitrd
Required-Packages: u-boot-tools
Bootloader-Sets-root: no
```

- 実行すると設定されているデータと起動しているカーネル情報(/proc/cpuinfo、または/proc/device-tree/model)を元にカーネルと inittd を変換し、インストールする。

flash-kenrel

```
$ cat /proc/cpuinfo | tail -3
Hardware       : Marvell Armada 370/XP (Device Tree)
Revision      : 0000
Serial        : 0000000000000000
```

```
$ flash-kernel
flash-kernel: installing version 3.10-1-armmp
Generating kernel u-boot image... done.
Installing new uImage.
Generating initramfs u-boot image... done.
Installing new uInitrd.
Installing new dtb.
```

```
$ ls /boot
System.map-3.10-1-armmp  initrd.img-3.10-1-armmp
uInitrd                 config-3.10-1-armmp
uImage                  vmlinuz-3.10-1-armmp
```

flash-kernel によって作成されたイメージを使った起動方法はボード毎に異なります。OpenBlocks AX3 の場合は以下の方法で起動できるはずですが。

```
$ ide reset
$ ext2load ide 0 2000000 /boot/uImage
$ ext2load ide 0 3000000 /boot/uInitrd
$ bootm 2000000 3000000
```

終わりに

- ARMMP の仕組みと、Debian の対応方法について説明した。



終わりに

- ARMMP の仕組みと、Debian の対応方法について説明した。
- Debian で ARMMP を使うには armmp フレーバを使う。

終わりに

- ARMMP の仕組みと、Debian の対応方法について説明した。
- Debian で ARMMP を使うには armmp フレーバを使う。
- Debian で提供されているカーネルを ulmage に変換するには flash-kernel を使う。

終わりに

- ARMMP の仕組みと、Debian の対応方法について説明した。
- Debian で ARMMP を使うには armmp フレーバを使う。
- Debian で提供されているカーネルを ulmage に変換するには flash-kernel を使う。
- flash-kernel のデータはぷらっとホームさんと相談して入れようと思っているのでまだコミットされていない。

終わりに

- ARMMP の仕組みと、Debian の対応方法について説明した。
- Debian で ARMMP を使うには armmp フレーバを使う。
- Debian で提供されているカーネルを ulmage に変換するには flash-kernel を使う。
- flash-kernel のデータはぷらっとホームさんと相談して入れようと思っているのでまだコミットされてない。
- また、エントリーポイントを設定する項目と DT の model を指定する項目がないのでこれも対応する。

終わりに

- ARMMP の仕組みと、Debian の対応方法について説明した。
- Debian で ARMMP を使うには armmp フレーバを使う。
- Debian で提供されているカーネルを ulmage に変換するには flash-kernel を使う。
- flash-kernel のデータはぷらっとホームさんと相談して入れようと思っているのでまだコミットされてない。
- また、エントリーポイントを設定する項目と DT の model を指定する項目がないのでこれも対応する。
- 残作業としてはカーネル 3.10 では mvebu が動作しない。パッチを当てる必要がある

終わりに

- ARMMP の仕組みと、Debian の対応方法について説明した。
- Debian で ARMMP を使うには armmp フレーバを使う。
- Debian で提供されているカーネルを ulmage に変換するには flash-kernel を使う。
- flash-kernel のデータはぷらっとホームさんと相談して入れようと思っているのでまだコミットされてない。
- また、エントリーポイントを設定する項目と DT の model を指定する項目がないのでこれも対応する。
- 残作業としてはカーネル 3.10 では mvebu が動作しない。パッチを当てる必要がある
- ということでまだまだやることはたくさんあります。