从庖丁解牛说 uboot 如何编译

很多人拿到 uboot，编译不知如何下手！

其实，这个世界上的万事万物，都有一个“纹理”。我读中学的时候劈柴，如果顺着木头的纹理劈下去很轻易的就劈开了，但如果反其道而行之不但劈不开而且斧头还会弹回来伤人！呵呵，城里出生的孩子是没这个体会，即使现在农村的孩子因为家里都烧液化气也没有这个机会体验了。

庖丁解牛之所以游刃有余，是因为他掌握了牛的纹理，顺着这些纹理就应该很容易。

那么我们的 uboot 的纹理在哪里呢？

很多初学者，拿到这种代码从来没有去看过它的 readme 或者 document！这两个文本文件是非常重要的东西，可惜呀！很多人不去看 readme 而去请教别人，google，baidu，跑图书馆。其实，有些东西当你问到别人的时候，聪明的人也是去看 readme 然后给你解答的。

下面我们就去找 uboot 的纹理！

本文 u-boot 版本 U-Boot 1.1.4

我们按正常人的思维（智商 90）来分析。

首先，是要编译，那么编译就要执行命令 make，而 make 实际上就是执行 makefile 文件。第一次 make 肯定是不能成功的。听从观音菩萨的教诲“从哪里来就到哪里去”！make 出问题，我们就去 makefile 里找原因。Makefile 里有这样一段话：

```
TOPDIR ：= $（shell if [ “$$PWD” != “” ]; then echo $$PWD; else pwd; fi）

export TOPDIR

ifeq （include/config.mk，$（wildcard include/config.mk））

# load ARCH， BOARD， and CPU configuration

include include/config.mk

export ARCH CPU BOARD VENDOR SOC

# load other configuration
```

```
include $(TOPDIR)/config.mk

ifndef CROSS_COMPILE

ifeq ($(HOSTARCH),ppc)

CROSS_COMPILE =

else

ifeq ($(ARCH),ppc)

CROSS_COMPILE = powerpc-linux-

endif

ifeq ($(ARCH),arm)

CROSS_COMPILE = /usr/local/arm/2.95.3/bin/arm-linux-

endif

ifeq ($(ARCH),i386)

ifeq ($(HOSTARCH),i386)

CROSS_COMPILE =

else

CROSS_COMPILE = i386-linux-

endif

endif

ifeq ($(ARCH),mips)

CROSS_COMPILE = mips_4KC-

endif

ifeq ($(ARCH),nios)

CROSS_COMPILE = nios-elf-

endif
```

```
ifeq （$（ARCH），nios2）

CROSS_COMPILE = nios2-elf-

endif

ifeq （$（ARCH），m68k）

CROSS_COMPILE = m68k-elf-

endif

ifeq （$（ARCH），microblaze）

CROSS_COMPILE = mb-

endif

endif

endif

export CROSS_COMPILE
```

这段脚本就是设置交叉编译路径 CROSS_COMPILE，在设置这个路径前要判断我们所用的平台是什么，即 ARCH 是什么。我们这里以 ARM 为例。那么我们的 ARCH 从哪里来呢？

上面有这样一段话：

```
export ARCH CPU BOARD VENDOR SOC
```

export 表示从外部引进的变量。那么 ARCH 是从哪里引进来的呢？

上面还有一句话：

```
ifeq （include/config.mk，$（wildcard include/config.mk））

# load ARCH， BOARD， and CPU configuration

include include/config.mk
```

其实已经说的很直观了，是从 include/config.mk 这个文件中装载 ARCH 这些变量的。

那么，我们再来看 include/config.mk 这个文件。

从官方下载的 uboot 是没有这个文件的。

很多人到这里就傻眼了，不过更多的人还跟不到这里！

没有这个文件怎么办！

还有一个很重要的文件没有看 readme。

在 readme 中有这样一段话

Selection of Processor Architecture and Board Type：

For all supported boards there are ready-to-use default

configurations available; just type “make _config”。

Example： For a TQM823L module type：

cd u-boot

make TQM823L_config

For the Cogent platform， you need to specify the cpu type as well;

e.g. “make cogent_mpc8xx_config”。 And also configure the cogent

directory according to the instructions in cogent/README.

也许你不知道 TQM823L_config 是什么，但如果你脑子灵活你应该看到了 configurations available; just type “make _config”。这句话。如果你不能够看到这句话，说明你不怎么适合搞嵌入式。因为嵌入式道路上的难度远不止于此，不过这也算不上难度，这是一个悟性问题。

但如果你看了 uboot 的整个目录结构和文件结构也许会知道 TQM823L 是什么。说实话，你在编译 uboot 之前你应该看一遍 uboot 的代码，至少目录结构应该清楚，如果你什么都没有看，就来编译，也说明你是一个不善于学习的人，也可以说，你不怎么适合搞嵌入式，因为嵌入式知识面比较广，需要你“博览群书”。

TQM823L 是我们 uboot 目录 board 下的一个文件夹名称,board 下面是 uboot 所有所支持的 BSP。

如果我们是 smdk2410，我们就应该是

Make smdk2410 _config

这样我们就可以在 include 下得到 config.mk 这个配置文件，内容如下：

ARCH = arm

CPU = arm920t

BOARD = smdk2410

SOC = s3c24x0

这样，在 makefile 中，就可以导出

export ARCH CPU BOARD VENDOR SOC 给 makefile 所用。

本文仅做抛砖引玉的作用，接下来的事情就靠读者自己去解决了。

祝你好运！(作者：下家山)