此文章已于 11:07:32 2016/11/29 发布到 MADHEX

U-boot源码整体框架

# U-boot源码整体框架

源码解压以后，我们可以看到以下的文件和文件夹：

|  |  |  |
| --- | --- | --- |
| **cpu** | | 与处理器相关的文件。每个子目录中都包括cpu.c和interrupt.c、start.S、u-boot.lds。  **cpu.c**初始化CPU、设置指令Cache和数据Cache等  **interrupt.c**设置系统的各种中断和异常  **start.S**是U-boot启动时执行的第一个文件，它主要做最早其的系统初始化，代码重定向和设置系统堆栈，为进入U-boot第二阶段的C程序奠定基础  **u-boot.lds**链接脚本文件，对于代码的最后组装非常重要 |
| **board** | | 已经支持的所有开发板相关文件，其中包含SDRAM初始化代码、Flash底层驱动、板级初始化文件。  其中的**config.mk**文件定义了TEXT\_BASE，也就是代码在内存的其实地址，非常重要。 |
| **common** | | 与处理器体系结构无关的通用代码，U-boot的命令解析代码/common/command.c、所有命令的上层代码**cmd\_\*.c**、U-boot环境变量处理代码env\_\*.c、等都位于该目录下 |
| **drivers** | | 包含几乎所有外围芯片的驱动，网卡、USB、串口、LCD、Nand Flash等等 |
| disk  **fs**  **net** | | 支持的CPU无关的重要子系统：  磁盘驱动的分区处理代码  文件系统：FAT、JFFS2、EXT2等  网络协议：NFS、TFTP、RARP、DHCP等等 |
| **include** | | 头文件，包括各CPU的寄存器定义，文件系统、网络等等  configs子目录下的文件是与目标板相关的配置头文件 |
| doc | | U-Boot的说明文档，在修改配置文件的时候可能用得上 |
| **lib\_arm** | | 处理器体系相关的初始化文件  比较重要的是其中的board.c文件，几乎是U-boot的所有架构第二阶段代码入口函数和相关初始化函数存放的地方。 |
| lib\_avr32  lib\_blackfin  lib\_generic  lib\_i386  lib\_m68k  lib\_microblaze | lib\_mips lib\_nios  lib\_nios2  lib\_ppc  lib\_sh  lib\_sparc |
| api  examples | | 外部扩展应用程序的API和范例 |
| nand\_spl  onenand\_ipl  post | | 一些特殊构架需要的启动代码和上电自检程序代码 |
| libfdt | | 支持平坦设备树(flattened device trees)的库文件 |
| tools | | 编译S-Record或U-Boot映像等相关工具，制作bootm引导的内核映像文件工具mkimage源码就在此 |
| **Makefile**  MAKEALL  config.mk  rules.mk  mkconfig | | 控制整个编译过程的主Makefile文件和规则文件 |
| CHANGELOG  CHANGELOG-before-U-Boot-1.1.5  COPYING  CREDITS  MAINTAINERS  README | | 一些介绍性的文档、版权说明 |

标为红色的是移植时比较重要的文件或文件夹。

# U-boot代码的大致执行流程（以S3C24x0为例）

从链接脚本文件u-boot.lds中可以找到代码的起始：

|  |
| --- |
| OUTPUT\_FORMAT("elf32-littlearm", "elf32-littlearm", "elf32-littlearm")  OUTPUT\_ARCH(arm)  ENTRY(\_start)  SECTIONS  {         . = 0x00000000;         . = ALIGN(4);         .text :         {                cpu/arm920t/start.o      (.text)                \*(.text)         }  …... |

从中知道程序的入口点是\_start，定位于cpu/arm920t /start.S（即u-boot启动的第一阶段）。

# U-boot版本相关

1、版本号变化：

2008年8月及以前

按版本号命名：u-boot-1.3.4.tar.bz2(2008年8月更新)

2008年8月以后均按日期命名。

目前最新版本：u-boot-2011.06.tar.bz2（2011年6月更新）

2、目录结构变化：

u-boot目录结构主要经历过2次变化，u-boot版本第一次从u-boot-1.3.2开始发生变化，主要增加了api的内容；变化最大的是第二次，从2010.6版本开始。

u-boot-2010.03及以前版本

├──api 存放uboot提供的接口函数

├──board 根据不同开发板定制的代码，代码也不少

├──common 通用的代码，涵盖各个方面，已命令行处理为主

├──cpu 与体系结构相关的代码，uboot的重头戏

├──disk 磁盘分区相关代码

├──doc 文档，一堆README开头的文件

├──drivers 驱动，很丰富，每种类型的设备驱动占用一个子目录

├──examples 示例程序

├──fs 文件系统，支持嵌入式开发板常见的文件系统

├──include 头文件，已通用的头文件为主

├──lib\_【arch】 与体系结构相关的通用库文件

├──nand\_spl NAND存储器相关代码

├──net 网络相关代码，小型的协议栈

├──onenand\_ipl

├──post 加电自检程序

└──tools 辅助程序，用于编译和检查uboot目标文件

从u-boot-2010.06版本开始把体系结构相关的内容合并，原先的cpu与lib\_arch内容全部纳入arch中，并且其中增加inlcude文件夹；分离出通用库文件lib。

u-boot-2010.06及以后版本

├──api 存放uboot提供的接口函数

├──arch 与体系结构相关的代码，uboot的重头戏

├──board 根据不同开发板定制的代码，代码也不少

├──common 通用的代码，涵盖各个方面，已命令行处理为主

├──disk 磁盘分区相关代码

├──doc 文档，一堆README开头的文件

├──drivers 驱动，很丰富，每种类型的设备驱动占用一个子目录

├──examples 示例程序

├──fs 文件系统，支持嵌入式开发板常见的文件系统

├──include 头文件，已通用的头文件为主

├──lib 通用库文件

├──nand\_spl NAND存储器相关代码

├──net 网络相关代码，小型的协议栈

├──onenand\_ipl

├──post 加电自检程序

└──tools 辅助程序，用于编译和检查uboot目标文件

3、移植工作涉及的目录情况

从uboot代码根目录，可以看出其已经非常庞大，功能也很丰富。

移植工作最主要的是看对应的处理器和开发板代码，2010.06版本以后处理器相关的代码集中在arch、board目录。(以前版本主要在cpu和board目录)

先看一下arch目录：

arch

├──arm

├──avr32

├──blackfin

├──i386

├──m68k

├──microblaze

├──mips

├──nios2

├──powerpc

├──sh

└──sparc

arch目录内容比以前的版本干净，每个子目录代表一个处理器类型，子目录名称就是处理器的类型名称。

我们移植的是mips的处理器，所以参考一下arch/mips目录：

arch/mips

├──cpu

├──include

└──lib

arch/mips目录下有三个目录，其他的处理器目录下也是这个结构：

cpu子目录对应一种处理器的不同产品型号或者系列;

include子目录是处理器用到的头文件；

lib目录对应用到处理器公用的代码；

下面看看cpu下的内容，arch/mips/cpu目录下的内容：

arch/mips/cpu

├──asc\_serial.c

├──asc\_serial.h

├──au1x00\_eth.c

├──au1x00\_serial.c

├──au1x00\_usb\_ohci.c

├──au1x00\_usb\_ohci.h

├──cache.S

├──config.mk

├──cpu.c

├──incaip\_clock.c

├──incaip\_wdt.S

├──interrupts.c

├──Makefile

└──start.S 整个uboot代码入口点

目前最新版本(2011.6版本开始)中cpu目录中建立mips32目录，把incaip和au1x00也分类放在不同的目录中。

u-boot.lds是ld程序也就是连接器的脚本文件，这个文件描述了如何连接目标文件，ld程序会根据这个文件的指示按照需求把不同的目标文件连接在一起生成供烧写到开发板的程序。

该文件放在board对应的目录中。

4、移植u-boot的版本选择情况

由于u-boot的各版本没有重大变化，各版本移植起来基本相同，也正因为如此，大多数版本均有人移植过，主要是arm体系结构的。

如cortex A8使用 u-boot-1.3.4；cortex M3 上u-boot-1.1.6、u-boot-1.2.0等均有人移植过。

考虑到我们目前的编译器较新，编译旧版本u-boot时会出现错误，警告也很多；新版本的u-boot目录结构也较清晰，因此选用较新版本的u-boot。

最新版本（2011.06）Makefile中没有mips的部分，不知道为什么。（2011.03版本中同样也是）

u-boot-2010.12的Makefile没有问题，编译incaip通过，没有任何警告和错误，因此最终选择u-boot-2010.12作为我们的移植版本。