1、arch\arm\cpu\armv7目录下面为arm公司设计的核心设备驱动文件。board目录下面为各个公司的外设驱动文件（如：三星，飞思卡尔，TI，ST）。

2、arch\arm\cpu\armv7\start.S为u-boot的开始文件，从\_start开始运行。

3、开始为异常处理，异常向量讲解见视频第三集

4、编译u-boot：

make s5p\_goni\_config生成配置文件

make –j4

生成了5个u-boot开头的文件

u-boot：链接器将所有的.o文件都拼接起来，里面有很多符号信息及调试信息，因此这种结果文件很大，不能拷到板卡运行，供调试用

u-boot.bin：使用strip工具将u-boot中的调试信息等剥离掉成为bin

u-boot.lds

u-boot.map

u-boot.srec

5、objdump工具

arm-none-eabi-objdump -S u-boot

-S显示源代码和反汇编代码



objdump可以将.o链接成的elf文件转变为.bin二进制文件，供裸板运行，elf文件在操作系统下可直接运行

6、跳转指令: b（相当于goto）, bl跳转后会跳回原位置， bx跳转完后返回原有的CPU模式

7、ENTRY宏展开

ENTRY(save\_boot\_params) /\*inlcude/linux/linkage\*/

bx lr @ back to my caller

ENDPROC(save\_boot\_params)

.weak save\_boot\_params

ENTRY(save\_boot\_params)展开：

.global save\_boot\_params;

.align 4

save\_boot\_params

ENDPROC(save\_boot\_params)展开：

.type save\_boot\_params STT\_FUNC;

.size save\_boot\_params, .-save\_boot\_params //当前语句地址减去函数地址

第10集：实验使用u-boot点亮led

hexdump –s startbyte –n bytes –C file以16进制和ASCII形式显示。

dd命令:

dd iflag=dsync oflag=dsync if=input\_file of=/dev/sdx seek=1

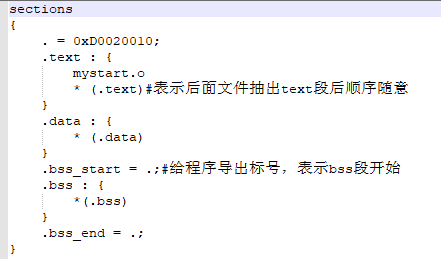
seek=1表示SD卡的第一个block，每个扇区512字节

第13集：myboot点亮LED2

arm-none-eabi-as mystart.S –o mystart.o汇编器将汇编文件生成目标文件，等价于arm-none-eabi-gcc -c mystart.S –o mystart.o其中-c表示只汇编不链接

.o文件至少包含3个段，arm-none-eabi-size file.o可以查看各段的大小

|  |  |
| --- | --- |
| .text | 代码段 |
| .data | 程序初始化数据{1，2，3} |
| .bss | 系统自动初始化数据{0，} |



链接：arm-none-eabi-ld –Tmyboot.lds –o myboot.elf mystart.o mylowlevel\_init.o

剥离符号，得到纯净的二进制可执行文件：

arm-none-eabi-objcopy -O binary myboot.elf myboot.bin