

## 文件的内容:

1. BIN文件是 raw binary 文件, 这种文件只包含机器码。
2. ELF文件除了机器码外, 还包含其它额外的信息, 如段的加载地址, 运行地址, 重定位表, 符号表等。

所以ELF文件的体积比对应的BIN文件要大。

## 文件的执行:

1. 执行raw binary很简单, 只需要将程序加载到其起始地址, 就可以执行;

```
FILE *fp = fopen("vmlinux.bin", "rb");  
fread(VMLINUX_START, 1, VMLINUX_SIZE, fp);  
((void (*)(void))VMLINUX_START)();
```

2. 执行ELF程序则需要一个ELF Loader。

现在知道了吧, uboot和Linux kernel启动的时候是没有ELF Loader的  
所以烧在flash上的文件只能是raw binary格式的, 即镜像文件image

## 文件的转换:

1. 通过gcc编译出来的是elf文件
2. 通过objcopy可以把elf文件转换为bin文件

```
1. CC=ppc-gcc  
2. LD=ppc-ld  
3. OBJCOPY=ppc-objcopy  
4.  
5. $(CC) -g $(CFLAG) -c boot.S  
6. #先将boot.S文件生成boot.o  
7.  
8. $(LD) -g -Bstatic -T$(LDFILE) \  
9. -Ttext 0x12345600 boot.o \  
10. --start-group -Map boot.map -o boot.elf  
11. #再将boot.o生成boot.elf  
12.  
13. $(OBJCOPY) -O binary -R .note -R .comment -S boot.elf boot.bin  
14. #接着将 boot.elf 转换为 boot.bin  
15. #使用 -O binary (或--out-target=binary) 输出为原始的二进制文件  
16. #使用 -R .note (或--remove-section) 输出文件中不要.note这个section, 缩小了文件尺寸  
17. #使用 -S (或--strip-all) 输出文件中不要重定位信息和符号信息, 缩小了文件尺寸
```

编译完uboot后生成:

u-boot	ELF文件可用来调试
u-boot.bin	BIN文件用来烧在Flash上

编译linux生成:

vmlinux	ELF文件可用来调试
vmlinux.bin	BIN文件, 没直接用过

zImage/vmlinuz/bzimage

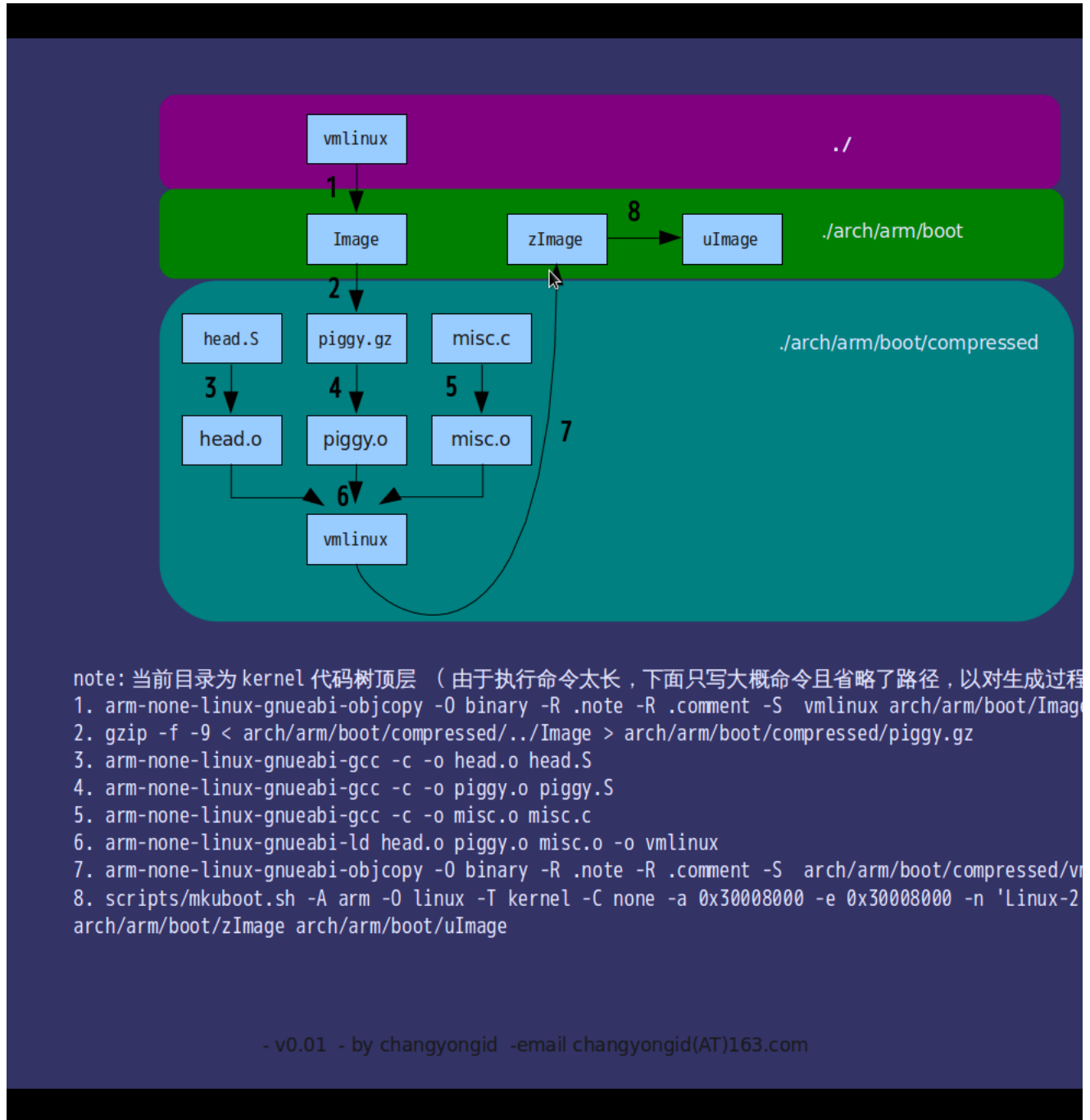
将vmlinux.bin压缩, 并加上一段解压代码得到的, 不可和bootloader共存?

uImage

uboot专用的内核镜像，在zImage前加了一个64字节的头，描述内核版本、加载地址生成时间，文件大小等等。其0x40后的内容和zImage一样  
它是由uboot的工具mkImage生成的。

uImage相对于zImage的优点在于：uImage可以和uboot共存。

网上的一个图片很强大，讲解了arm平台的uImage的生成：



ppc平台uImage有点不同，make uImage后查看源代码目录的verbose.log

1. ....省略
- 2.
3. make -f scripts/Makefile.build obj=arch/ppc/boot/images arch/ppc/boot/images/uImage
4. ppc\_85xx-objcopy -O binary vmlinux arch/ppc/boot/images/vmlinux.bin (第一步)

```
5. gzip -f -9 < arch/ppc/boot/images/vmlinux.bin > arch/ppc/boot/images/vmlinux.gz.$ && mv arch/ppc/boot/images/vmlinux.gz.$  
arch/ppc/boot/images/vmlinux.gz (第二步)  
6. rm -f arch/ppc/boot/images/ulmage  
7. ./bin/sh /root/project/eldk4.2/ppc_85xx/usr/src/8541-reducece-ppc-linux-2.6.24/scripts/mkuboot.sh -A ppc -O linux -T kernel -C gzip -a  
00000000 -e 00000000 -n 'Linux-2.6.24.2' -d arch/ppc/boot/images/vmlinux.gz arch/ppc/boot/images/ulmage (第三步)
```

## 文件的调试:

1. 我们调试一般都是使用elf文件，比如：

1. nm elf文件 #得到符号表
2. objdump -D elf文件 #反汇编，且汇编代码与源码混排

2. bin文件比较杯具，里面全是机器码，所以只能反汇编

1. objdump -b binary -m powerpc uboot.bin

全是汇编代码.....+\_+