



加盟怎么理



关闭广告



佳驹

微博

加好友

发纸条

写留言

加关注

博客地图 world map

博客等级: 13

博客积分: 702 新

博客访问: 37,442

关注人气: 20

获赠金笔: 0支

赠出金笔: 0支

荣誉徽章:



Click to enable Adobe Flash Player

广告

相关博文

更多>>

佳驹的博客

<http://blog.sina.com.cn/wuchao5460> [订阅] [手机订阅][首页](#) | [博文目录](#) | [图片](#) | [关于我](#)

正文

字体大小: 大 中 小

理解Makefile中的KERNELRELEASE (2013-08-05 18:09:25)

标签: kernelrelease makefile 内核模块 校园 分类: 嵌入式开发

Linux内核是一种单体内核，但是通过动态加载模块的方式，使它的开发非常灵活方便。那么，它是如何编译内核的呢？我们可以通过分析它的Makefile入手。以下是一个简单的hello内核模块的Makefile。

#####Makefile1#####

ifneq (\$(KERNELRELEASE),)

obj-m:=hello.o

else

KERNELDIR:=/lib/modules/\$(shell uname -r)/build

PWD:=\$(shell pwd)

default: \$(MAKE) -C \$(KERNELDIR) M=\$(PWD) modules

clean: rm -rf *.o *.mod.c *.mod.o *.ko

endif

#####

#####Makefile2#####

ifneq (\$(KERNELRELEASE),)

obj-m :=hello.o

else

KDIR:= /lib/modules/2.6.29.4-167.fc11.i686.PAE/build

all:

make -C \$(KDIR) M=\$(PWD) modules

clean:

rm -f *.ko *.o *.mod.o *.mod.c .symvers

endif

#####

#####Makefile3#####

ifneq (\$(KERNELRELEASE),)

obj-m := hello.o

else

PWD := \$(shell pwd)

KVER := \$(shell uname -r)

KDIR := /lib/modules/\$(KVER)/build

all:

\$(MAKE) -C \$(KDIR) M=\$(PWD) modules

clean:

rm -rf *.cmd *.o *.mod.c *.ko .tmp_versions

endif

#####

当我们写完一个hello模块，只要使用以上的makefile。然后make一下就行。假设我们把hello模块的源代码放在/home/study/prog/mod/hello/下。当我们在这个目录运行make时，make是怎么执行的呢？ LDD3第二章第四节“编

博



第一批05

·为什么发

·游戏让

·小说：乌



淘 广告

推荐博文



谁说英式教育懒散？我在。



毕业证要配学历证明：奇。



我被职场妈妈养大 现在也。



大学生应聘礼仪成出租女。



高颜值健身情侣高难度秀。



让吃甜食毫无罪恶感的甜。

查看更多>>

谁看过这篇博文

	逆鳞_右痕	11月9日
	临云	10月30日
	陈安	9月16日
	用户34416...	9月5日
	我会一点...	9月3日
	kevin	9月1日
	Nodoka	8月4日
	479851636	8月1日
	heyouf	7月31日
	用户57585...	7月27日
	Aaron刘威	7月18日
	就是有才	7月18日

译和装载”中只是简略地说到该Makefile被执行了两次，但是具体过程是如何的呢？

首先，由于make 后面没有目标，所以make会在Makefile中的第一个不是以.开头的目标作为默认的目标执行。于是default成为make的目标。make会执行 \$(MAKE) -C \$(KERNELDIR) M=\$(PWD) modules

shell是make内部的函数,假设当前内核版本是2.6.13-study,所以\$(shell uname -r)的结果是 2.6.13-study 这里，实际运行的是

make -C /lib/modules/2.6.13-study/build M=/home/study/prog/mod/hello/ modules， /lib/modules/2.6.13-study/build是一个指向内核源代码/usr/src/linux的符号链接。可见，make执行了两次。第一次执行时是读hello模块的源代码所在目录/home/study/prog/mod/hello/下的Makefile。

第二次执行时是执行/usr/src/linux/下的Makefile时。

但是还是有不少令人困惑的问题： 1.这个KERNELRELEASE也很令人困惑，它是什么呢？

在/home/study/prog/mod/hello/Makefile中是没有定义这个变量的，所以起作用的是else...endif这一段。不过，如果把hello模块移动到内核源代码中。例如放到usr/src/linux/driver/中， KERNELRELEASE就有定义了。

在usr/src/linux/Makefile中有 162

KERNELRELEASE=\$(VERSION).\$(PATCHLEVEL).\$(SUBLEVEL)\$(EXTRAVERSION)\$(LOCALVERSION) 这时候，hello模块也不再是单独用make编译，而是在内核中用make modules进行 编译。

用这种方式，该Makefile在单独编译和作为内核一部分编译时都能正常工作。

2.这个obj-m := hello.o什么时候会执行到呢？ 在执行：

make -C /lib/modules/2.6.13-study/build M=/home/study/prog/mod/hello/ modules时， make 去usr/src/linux/Makefile中寻找目标modules: 862 .PHONY: modules 863 modules: \$(vmlinux-dirs) \$(if \$(KBUILD_BUILTIN),vmlinux) 864 @echo ' Building modules, stage 2.'; 865 \$(Q)\$(MAKE) -rR -f \$(srctree)/scripts/Makefile.modpost

可以看出，分两个stage: 1.编译出hello.o文件。 2.生成hello.mod.o hello.ko 在这过程中，会调用 make -f scripts/Makefile.build obj=/home/study/prog/mod/hello 而在 scripts/Makefile.build会包含很多文件： 011 -include .config 012 013 include \$(if \$(wildcard \$(obj)/Kbuild), \$(obj)/Kbuild, \$(obj)/Makefile) 其中就有/home/study/prog/mod/hello/Makefile 这时 KERNELRELEASE已经存在。所以执行的是： obj-m:=hello.o

关于make modules的更详细的过程可以在scripts/Makefile.modpost文件的注释中找到。如果想查看make的整个执行过程，可以运行make -n。

```
#include <linux/init.h>
```

```
#include <linux/module.h>
```

```
MODULE_LICENSE("GPL");
```

```
MODULE_AUTHOR("feixiaoxing");
```

```
MODULE_DESCRIPTION("This is just a hello module!\n");
```

```
static int __init hello_init(void)
```

```
{
    printk(KERN_EMERG "hello, init\n");
    return 0;
}
```

```
static void __exit hello_exit(void)
```

```
{
    printk(KERN_EMERG "hello, exit\n");
}
```

```
module_init(hello_init);
```

```
module_exit(hello_exit);
```

加载模块就是输入insmod hello.ko，卸载模块就是rmmod hello。在这过程中，我们都可以看到相应的打印内容。当然，朋友们可以一直往里面加代码，一步步调试，一步步学习，只要坚持和总结，都是可以学习好linux的驱动代码的。

参考资料：

[1]. ##Makefile1## 《KERNELRELEASE》 <http://yjiuzai.blog.163.com/blog/static/1037567272010101885922998/>

[2]. ##Makefile3## 《linux的四个维度》 <http://blog.csdn.net/feixiaoxing/article/details/7487404>

7


喜欢

0

赠金笔

分享：    

阅读(3956) | 评论 (0) | 收藏(1) | 禁止转载 | 喜欢▼ | 打印 | 举报

已投稿到： 排行榜

前一篇：[X86平台内核配置、编译与安装](#)

后一篇：[IPC实验代码<一>](#)

评论

重要提示：警惕虚假中奖信息

[发评论]

做第一个评论者吧！ 抢沙发>>

发评论

更多>>





登录名： 密码： [找回密码](#) [注册](#) ☒ 记住登录状态

发评论

以上网友发言只代表其个人观点，不代表新浪网的观点或立场。

< 前一篇

[X86平台内核配置、编译与安装](#)

后一篇 >

[IPC实验代码<一>](#)