[字体: 大中小]

linux-2.6.22.6内核启动分析之内核编译体验

[日期: 2018-07-07] 来源: Linux社区 作者: 坦然~

2.6-jz2440.patch上传到服务器/work/hardware目录下

40.patch 命令对内核进行打补丁操作

make: *** No rule to make target 'menuconfig'.

1解压缩、打补丁操作

1.2 输入cd /work/hardware 进入压缩包所在目录,再使用 tar xjf linux-2.6.22.6tar.bz2 命令对linu

1.1 打开Ubuntu,通过FTP将Windows相应文件夹下的linux-2.6.22.6.tar.bz2和补丁文件linux-2.6.2

- x内核源码压缩包进行解压缩 1.3 输入cd /linux-2.6.22.6 进入解压缩后内核的源码目录,执行 patch -p1 < ../linux-2.6.22.6-jz24
- 2 内核的配置(3种方法)

在进行介绍三种内核配置方法之前先说明在我电脑编译环境下,执行meke menuconfig命令遇到的问题。我使用的虚拟机版 本是Ubuntu-16.04.2,GNU Make的版本是4.1(在linux环境下输入make -v可以查看GNU Make版本信息),遇到编译问题

book@www.100ask.org:/work/hardware/linux-2.6.22.6\$ make menuconfig Makefile:416: *** mixed implicit and normal rules: deprecated syntax

book@www.100ask.org:/work/hardware/linux-2.6 网上搜索找到了报错原因,因为所编译linux-2.6.22.6内核版本比较旧,而我使用的GNU Make版本较新,新版本可能对于m ake的规则可能做出了一些修改,所以出现了上面的错误提示,因此需要对内核项层的Makefile文件做出如下修改:

Stop.

```
416 config %config: scripts basic outputmakefile FORCE
这里需要修改为
```

416 %config %config: scripts_basic outputmakefile FORCE

linux-2.6.22.6内核根目录下,输入 **vi Makefile** 命令,输入 **416gg** 快速定位到416行,看到如下代码:

Makefile:1449: *** mixed implicit and normal rules: deprecated syntax

然后是提示的1449行

1449 %/: prepare scripts FORC■

这里需要修改为

1449 🖊 %/: prepare scripts FORCE

完成后继续执行 make menuconfig 命令即可看到配置顺利进行。

2.1 直接执行 make menuconfig 命令,从头到尾每一项都自己去配置

2.2 使用默认配置,在默认配置基础上进行修改

./arch/blackfin/configs/BF561-EZKIT_defconfig ./arch/blackfin/configs/BF533-STAMP_defconfig ./arch/blackfin/configs/PNAV-10_defconfig ./arch/blackfin/configs/BF537-STAMP_defconfig /arch/blackfin/configs/BF533-EZKIT_defconfig

CRC-CCITT functions (CRC_CCITT) [N/m/y/?] n

CRC ITU-T V.41 functions (CRC_ITU_T) [N/m/y/?] n

book@www.100ask.org:/work/hardware/linux-2.6.22.6\$

CRC16 functions (CRC16) [N/m/y/?] n

configuration written to .config

2.4 执行make menuconfig 配置菜单

g*" 命令实现查找操作,这条命令查找所有带"defconfig"字符串的文件夹路径。查找结果如下,可以看到带"def_config"字 符串的配置文件都在arch架构目录下存放。

先用cd命令切换到linux-2.3.22.6内核存放根目录下,再进行默认配置文件的查找。这里我们可以通过 find -name "*defconfi

执行make menuconfig命令后,里面有成千上万种配置选项,需要一个一个去配置,显然操作十分复杂。

book@www.100ask.org:/work/hardware/linux-2.6.22.6\$ find -name "*defconfig*" ./arch/x86_64/defconfig

```
./arch/arm/configs/corgi_defconfig
./arch/arm/configs/onearm_defconfig
./arch/arm/configs/pxa255-idp_defconfig
./arch/arm/configs/shannon_defconfig
./arch/arm/configs/ateb9200_defconfig
./arch/arm/configs/at9lsam9rlek_defconfig
./arch/arm/configs/realview_defconfig
./arch/arm/configs/kb9202_defconfig
```

```
./arch/arm/configs/shark_defconfig
./arch/arm/configs/lubbock_defconfig
./arch/arm/configs/iop33x_defconfig
./arch/arm/configs/ixp23xx_defconfig
   ./arch/arm/configs/clps7500_defconfig
./arch/arm/configs/lart_defconfig
   /arch/arm/configs/pnx4008_defconfig
   /arch/arm/configs/versatile_defconfig
   /arch/arm/configs/iop32x_defconfig
  ./arch/arm/configs/at91sam9263ek_defconfig
./arch/arm/configs/csb637_defconfig
./arch/arm/configs/hackkit_defconfig
   ./arch/arm/configs/omap_h2_1610_defconfig
   ./arch/arm/configs/carmeva_defconfig
   ./arch/arm/configs/lpd7a404_defconfig
   /arch/arm/configs/mainstone_defconfig
   ./arch/arm/configs/pleb_defconfig
    /arch/arm/configs/ep93xx_defconfig
   ./arch/arm/configs/at91rm9200ek defconfig
我们的开发板使用的芯片属于arm架构,所以通过cd ./arch/arm/configs 进入configs目录寻找和开发板使用芯片相似的配置
文件。进入目标文件夹后使用 Is 命令来列出该文价夹下包含的配置文件,该目录下包含的配置文件如下图所示:
book@wnw.100ask.org:/work/hardware/linux-2.6.22.6$ cd ./arch/arm/cobook@wnw.100ask.org:/work/hardware/linux-2.6.22.6$/arch/arm/configs$ assabet_defconfig clps7500_defconfig h7201_defconfig at91rm9200dk_defconfig corgi_defconfig hackkit_defconfig at91sam9260ek_defconfig csb337_defconfig integrator_defconfig at91sam9263ek_defconfig csb337_defconfig inp13xx_defconfig at91sam9263ek_defconfig ebsa10_defconfig iop32x_defconfig at91sam97lek_defconfig edb7211_defconfig iop33x_defconfig ate9200_defconfig ep93xx_defconfig ixp200_defconfig ep93xx_defconfig ixp200_defconfig fortunet_defconfig ixp23xx_defconfig carmeva_defconfig h3600_defconfig iprnada720_defconfig
                                                                                                                    hnigs
kafa_defconfig
kb9202_defconfig
ks8695_defconfig
lart_defconfig
lpd270_defconfig
lpd78400_defconfig
lpd78404_defconfig
                                                                                                                                                         neponset_defconfig
netwinder_defconfig
netx_defconfig
ns9xxx_defconfig
omap_h2_1610_defconfig
onearm_defconfig
picotux200_defconfig
pleb_defconfig
pnx4008_defconfig
pxa255-idp_defconfig
realview_defconfig
                                                                                                                                                                                                  realview-smp_defconfig
rpc_defconfig
s3c2410_defconfig
shannon_defconfig
shark_defconfig
simpad_defconfig
spitz_defconfig
trizeps4_defconfig
versatile_defconfig
                                                                                                                      lubbock_defconfig
lusl7200_defconfig
```

根目录,然后运行 $make s3c2410_defconfig$ 命令来进行配置,配置结果如下图所示:

通过上面查找结果可以看到,与2440相似的配置只有s3c2410_defconfig文件了,这里我们先返回存放linux-2.6.22.6内核的

```
配置完后打印结果"configuration written to .config",表示所有的配置信息都写入到了.config文件中,执行make menuco
nfig 命令就是回去读取.config文件内容。
2.3 使用厂家提供的配置文件
jz2440出厂时,厂家提供了相应的配置文件config_ok,在打补丁时已经存放到内核的目录下。由上面的介绍,我们知道执
行make menuconfig 命令就是去读取.config内容,若想使用厂家提供配置文件,只需要把厂家提供配置文件内容复制到.
config文件中。使用cd命令切换到linux-2.3.22.6内核存放目录下,使用 cp config_ok .config 命令就能把实现前面所叙述
操作。
```

CRC32 functions (CRC32) [Y/?] y
CRC32c (Castagnoli, et al) Cyclic Redundancy-Check (LIBCRC32C) [N/m/y/?] n

letters are hotkeys. Pressing <Y> includes, <N> excludes, <M> modularizes <features. Press <Esc><Esc> to exit, <?> for Help, </>> for Search. Legend: [*] < built-in [] excluded <M> module < > module capable @^< Code maturity level options General setup --->

Linux Kernel Configuration Arrow keys navigate the menu. <Enter> selects submenus --->. Highlighted

< Exit >

< Help >

o^o^o^o^o^o^o^o^o

```
<Select>
```

<

Loadable module support --->
Block layer --->
System Type --->

---> Floating point emulation --->

Userspace binary formats --->

Bus support ---> Kernel Features --->

Boot options

```
菜单操作命令如下表所述:
 Arrow keys navigate the menu
                           |键盘上下键可以控制菜单上下移动,左右键可以控制水平三个菜单移动
  <Enter> selects submenus --->
                           |按下键盘Enter(回车键)可以进入该项菜单子目录
                           开头为粗体字母代表是该项菜单热键,在键盘上按下相应字母,光标定位到该处
 Highlighted letters are hotkeys
 Pressing <Y> includes
                           Y键,编译到内核中
                           N键,不编译到内核中
  <N> excludes
                           M键,进行模块化
  <M> modularizes features
 Press <Esc> <Esc> to exit
                           Esc键退出
  <?> for Help
                           ?键帮助
                           /键收索
  </> for Search
 Legend:
                           显示
 [*] built-in
                           已经内置到内核中
 excluded
                           不编译到内核中
  <M> module
                           已经模块化
                           表示该选项可以模块化
  < > module capable
```

3.2 输入 make uImage命令生成uImage形式内核文件,uImage文件包含头部和真正内核部分,可以 烧写进去让uboot直接使用。编译uImage文件时必须把编译uboot得到的mkimage工具复制到/user/bi

3.1 输入 make 命令直接进行编译

3 编译

4 启动内核

n目录中。 3.3 运行uboot,按下命令键k可以通过dnw软件把uImage文件传给uboot,uboot再清除flash,把uI mage文件写入到flash中。

4.1 通过uboot烧写uImage形式内核文件完成后,可以通过菜单键中b键重启内核,注意flash中存储有

完整文件系统时内核才能启动成功,否则内核启动会卡在某一位置。 Linux公社的RSS地址: https://www.linuxidc.com/rssFeed.aspx

本文永久更新链接地址: https://www.linuxidc.com/Linux/2018-07/153064.htm

Q.