# 之前能用的3.0版本总结

//////////////开发软件准备工作

LoongIDE软件用来烧写Pmon（也可以用来开发裸机程序）。

VMware虚拟机安装Ubuntu来制作内核与文件系统镜像（此时可建立公共文件夹或安装samba服务构建主机与虚拟机的传输通道）。

Putty串口用于给开发板发送命令接收回显。

Tftpd32用于tftp传输（文件系统与内核导入）

//////////////文件系统与内核制作

gcc-4.3-ls232.tar.gz为交叉编译器安装包，解压到根目录路径/opt/gcc-4.3-ls232/即可完成安装，举例gcc/g++编译器在/opt/gcc-4.3-ls232/bin/mipsel-linux-gcc或者/opt/gcc-4.3-ls232/bin/mipsel-linux-g++，用PATH命令添加路径是为了之后使用交叉编译器时简单快捷。

linux-3.0-ls1b.tar.gz为内核的压缩包文件，解压后在当前路径下产生vmlinux文件夹，进入该文件夹调用交叉编译器编译内核，产生内核文件vmlinux（备用）。

yaffs2-utils.tar.gz是镜像生成工具的压缩包，解压到当前路径产生yaffs2-utils文件夹，进入yaffs2-utils/utils/进行编译，编译完成后在yaffs2-utils/utils/路径下生成可执行文件mkyaffs2image，将此文件复制到/usr/bin路径下即可完成镜像生成工具的安装。

buildroot.tar.gz为文件系统压缩包，解压后在当前路径生成buildroot文件夹，进入该文件夹后调用交叉编译器对文件系统源码进行编译，在该文件夹的output/images/目录下看到编译出来的rootfs.cpio压缩文件。

将刚刚编译出来的rootfs.cpio压缩文件解压到Documents/rootfs/路径下，调用镜像生成工具mkyaffs2image制作文件系统镜像rootfs-yaffs2.img（保存在Documents路径下备用）。

//////////////文件系统与内核导入前准备工作

用公共文件夹或samba服务将内核文件vmlinux与文件系统镜像rootfs-yaffs2.img拷贝到tftp文件夹下，准备传输。

//////////////文件系统内核导入

烧写Pmon，检查网卡，擦除mtd0/mtd1，导入内核，导入文件系统，设置启动参数，重启

<附1>：load命令导入（文件系统+内核），是将其导入到内存中（易失性存储器DDRAM），断电即丢失。而devcp命令分别导入文件系统、内核，是将其导入到FLASH中，断电不丢失。

<附2>：二者的启动参数也有所不同：

Load导入陈工的内核g console=ttyS5,115200 rdinit=/sbin/init

devcp分别导入：set al mtd0

set append “root=/dev/mtdblock1 console=ttyS5,115200 rootfstype=yaffs2”

<附3>：gzrom.bin文件一旦导入相当于重装当前Pmon，Pmon版本不变。

# 陈工提供3.0版本

//////////////龙芯3.0\*文件系统与内核

gcc-4.3-ls232.tar.gz为交叉编译器安装包，解压到根目录路径/opt/gcc-4.3-ls232/即可完成安装，举例gcc/g++编译器在/opt/gcc-4.3-ls232/bin/mipsel-linux-gcc或者/opt/gcc-4.3-ls232/bin/mipsel-linux-g++，用PATH命令添加路径是为了之后使用交叉编译器时简单快捷。

linux-3.0-ls1b.tar.gz为内核的压缩包文件，解压后在当前路径下产生vmlinux文件夹，进入该文件夹调用交叉编译器编译内核，产生内核文件vmlinux（备用）。

yaffs2-utils.tar.gz是镜像生成工具的压缩包，解压到当前路径产生yaffs2-utils文件夹，进入yaffs2-utils/utils/进行编译，编译完成后在yaffs2-utils/utils/路径下生成可执行文件mkyaffs2image，将此文件复制到/usr/bin路径下即可完成镜像生成工具的安装。不再需要制作文件系统镜像了，所以此步骤就用不到了。

<附1>：龙芯给的文件系统在内核之中，ramdisk-stress0.cpio文件就像之前的rootfs.cpio压缩文件，他在vmlinx/路径下。调用交叉编译器的时候就将文件系统编译在了内核之中，所以生成的vmlinux内核自带文件系统。

<附2>：所以在导入开发板的过程中用：

mtd\_erase /dev/mtd0

mtd\_erase /dev/mtd1

devcp tftp://192.168.0.10/vmlinux /devmtd0

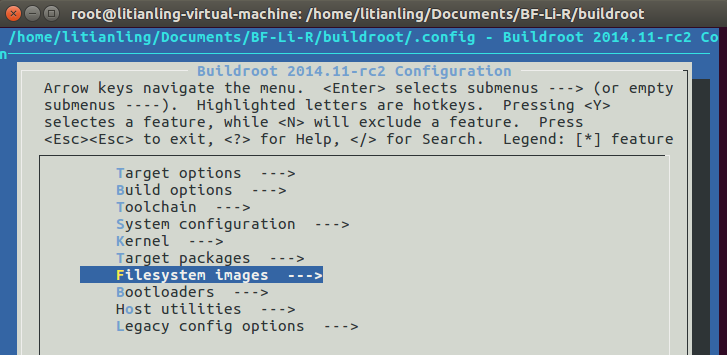
set al mtd0

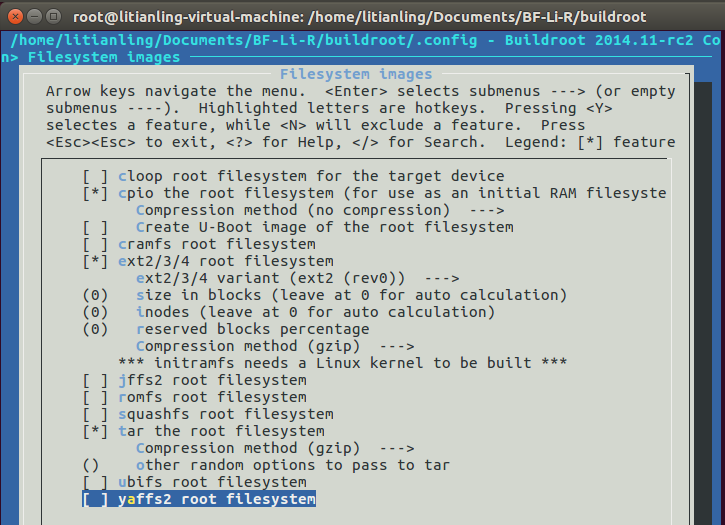
set append “console=ttyS5,115200 rdinit=/sbin/init”

<附3>：也可以将此文件系统（已成功）和内核（需要重新配置交叉编译参数）拆开未完成

# buildroot直接编译生成.img镜像文件

//////////////buildroot直接编译生成.img镜像文件（不用解压CPIO之后再用mkyaffs2image工具压缩成img镜像文件）

C:\Users\李天凌\AppData\Local\Temp\WeChat Files\a4bb2fa4769dacf6490a61ba78402fb.png



这样生成的文件系统镜像为rootfs.yaffs2可以直接烧写到mtd1

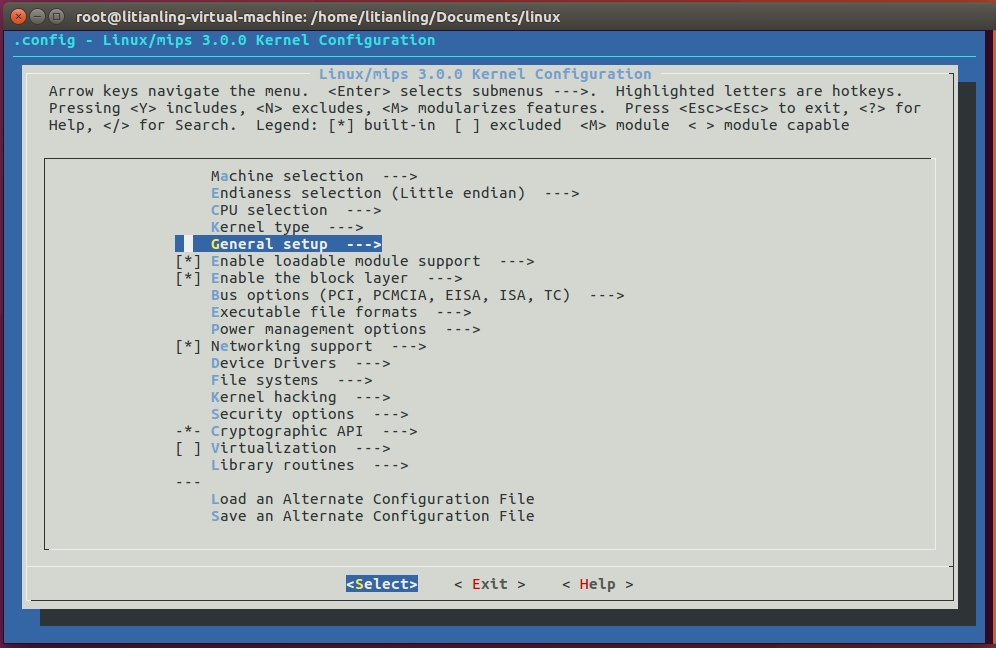
# 内核编译时是否包含文件系统

//////////////

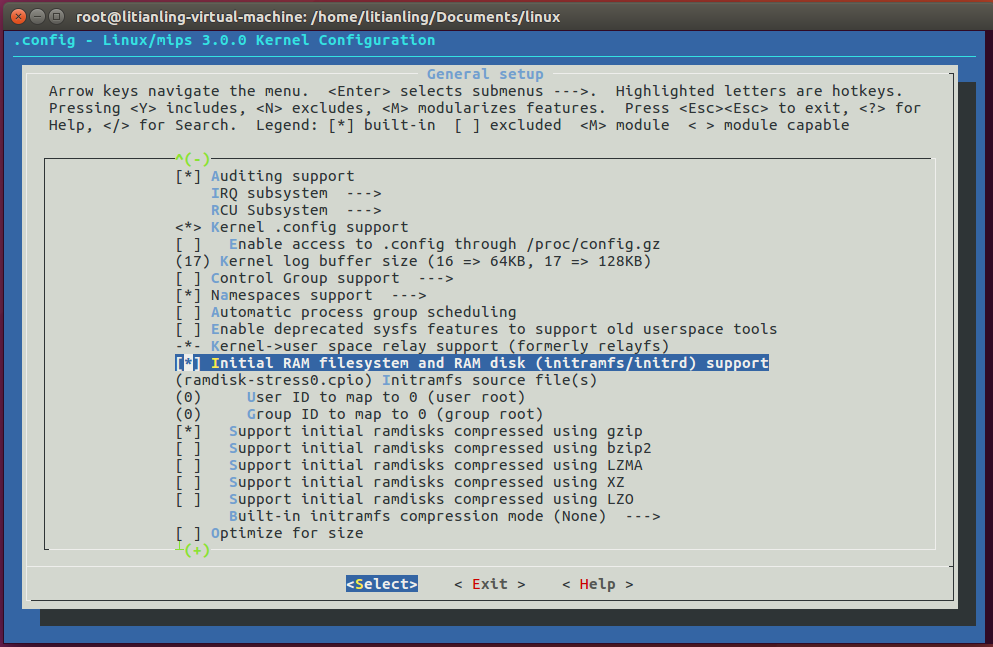
将交叉编译器添加到路径后，执行如下命令行

make ARCH=mips CROSS\_COMPILE=mipsel-linux- menuconfig

如下图内核编译器配置菜单：进入通用设置选项卡



勾选此项（带\*）则将文件系统（.cpio格式）编译进入内核之中，否则单独对内核进行编译



设置完成之后，重新执行make ARCH=mips CROSS\_COMPILE=mipsel-linux-命令行即可进行编译生成内核文件vmlinux

# 总结

最初分开编译：

文件系统buildroot文件夹--（交叉编译）-->.cpio文件---->.yaffs2文件系统可直接烧写

内核vmlinux文件夹—(交叉编译)-->vmlinux内核文件

分开编译改进版：

文件系统buildroot文件夹--（交叉编译修改编译选项）-->.cpio文件

内核vmlinux文件夹—(交叉编译)-->vmlinux内核文件

合在一起编译：

文件系统buildroot文件夹--（交叉编译）-->.cpio文件（拷贝到linux内核文件夹下）

内核vmlinux文件夹—(交叉编译修改选项)-->vmlinux内核文件（带有文件系统）