

Orange pi PC(H3)移植 linux 总结

By BobLiang/QQ: 106030169 on 2017-5-11

一、编译前的准备

- 1) 安装 VMware Workstation
- 2) 安装 32 位 Ubuntu1604
- 3) 安装 VMTools
- 4) 安装特定版本的两种交叉编译器 arm-linux-gnueabi 和 arm-linux-gnueabihf, 记住不能使用版本 : arm-linux-gnueabihf-gcc-5 (Ubuntu/Linaro 5.4.0-6ubuntu1~16.04.4) 5.4.0 20160609

参考小撸的博客: https://0w0.pw/Mainline_U-Boot.html

新手都喜欢用最新版本的软件, 如果已经有了这个 5.4.0 版本改怎么办?

那就先看 apt-cache 有哪些版本, 然后在指定安装非 5.4.0 版本

命令: apt-cache showpkg gcc-arm-linux-gnueabihf

安装指定版本的编译器 (以下指定安装 4.8.2 版本, 根据本地 apt-cache 选择不同版本来安装)

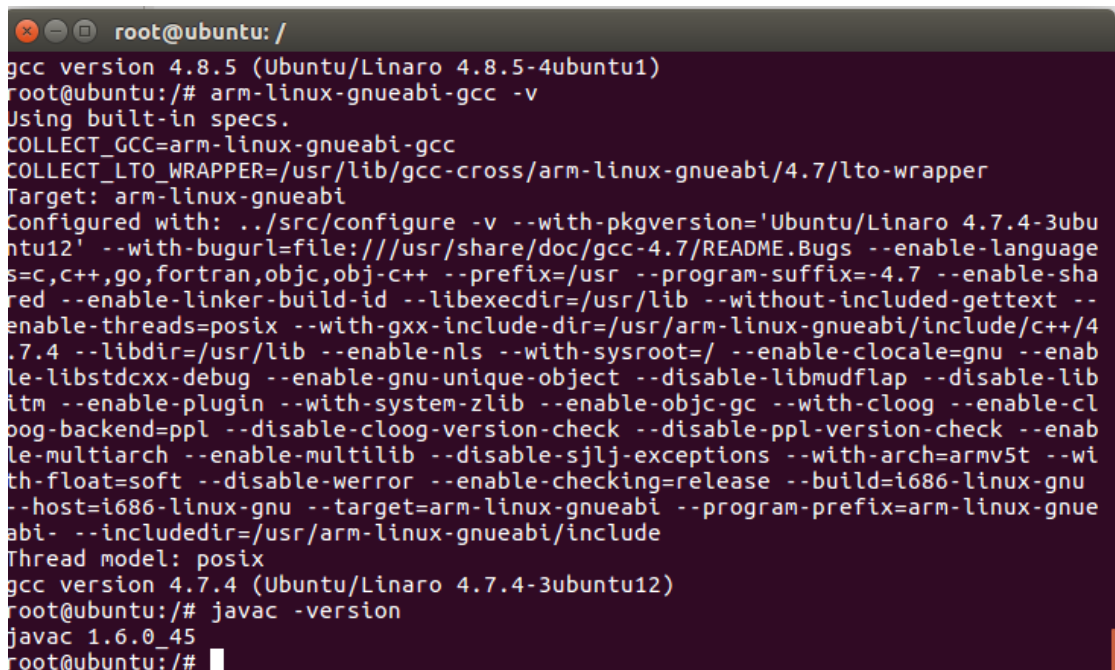
命令: apt-get install gcc-arm-linux-gnueabihf=4:4.8.2-1

另外, 这个交叉编译器 gcc-arm-linux-gnueabi 也不能用 5.4.0

命令: apt-get install gcc-arm-linux-gnueabi=4:4.7.2-1

不知道什么原因, 交叉编译有时候会自动升级到最新版本。

- 5) 安装 jdk, 安装过程网上很多, 下面是使用的版本



```
root@ubuntu: /
gcc version 4.8.5 (Ubuntu/Linaro 4.8.5-4ubuntu1)
root@ubuntu: /# arm-linux-gnueabi-gcc -v
Using built-in specs.
COLLECT_GCC=arm-linux-gnueabi-gcc
COLLECT_LTO_WRAPPER=/usr/lib/gcc-cross/arm-linux-gnueabi/4.7/lto-wrapper
Target: arm-linux-gnueabi
Configured with: ../src/configure -v --with-pkgversion='Ubuntu/Linaro 4.7.4-3ubuntu12' --with-bugurl=file:///usr/share/doc/gcc-4.7/README.Bugs --enable-language=s,c,c++,go,fortran,objc,obj-c++ --prefix=/usr --program-suffix=-4.7 --enable-shared --enable-linker-build-id --libexecdir=/usr/lib --without-included-gettext --enable-threads=posix --with-gxx-include-dir=/usr/arm-linux-gnueabi/include/c++/4.7.4 --libdir=/usr/lib --enable-nls --with-sysroot=/ --enable-clocale=gnu --enable-libstdcxx-debug --enable-gnu-unique-object --disable-libmudflap --disable-libitm --enable-plugin --with-system-zlib --enable-objc-gc --with-cloog --enable-clog-backend=ppl --disable-cloog-version-check --disable-ppl-version-check --enable-multiarch --enable-multilib --disable-sjlj-exceptions --with-arch=armv5t --with-float=soft --disable-werror --enable-checking=release --build=i686-linux-gnu --host=i686-linux-gnu --target=arm-linux-gnueabi --program-prefix=arm-linux-gnueabi- --includedir=/usr/arm-linux-gnueabi/include
Thread model: posix
gcc version 4.7.4 (Ubuntu/Linaro 4.7.4-3ubuntu12)
root@ubuntu: /# javac -version
javac 1.6.0_45
root@ubuntu: /#
```

二、获取内核源码

git clone https://github.com/orangepi-xunlong/orangepi_h3_linux.git

这个是最新的内核源码, 包含两个内核在里面, 当然 u-boot 也在里面了。

```
root@ubuntu: /home/bob/bakup/orangepi_h3_linux/OrangePi-Kernel
root@ubuntu: /home/bob/bakup#
root@ubuntu: /home/bob/bakup#
root@ubuntu: /home/bob/bakup#
root@ubuntu: /home/bob/bakup#
root@ubuntu: /home/bob/bakup#
root@ubuntu: /home/bob/bakup#
root@ubuntu: /home/bob/bakup#
root@ubuntu: /home/bob/bakup#
root@ubuntu: /home/bob/bakup#
root@ubuntu: /home/bob/bakup#
root@ubuntu: /home/bob/bakup#
root@ubuntu: /home/bob/bakup#
root@ubuntu: /home/bob/bakup#
root@ubuntu: /home/bob/bakup#
root@ubuntu: /home/bob/bakup#
root@ubuntu: /home/bob/bakup# cd orangepi_h3_linux/
root@ubuntu: /home/bob/bakup/orangepi_h3_linux# cd OrangePi-Kernel/
root@ubuntu: /home/bob/bakup/orangepi_h3_linux/OrangePi-Kernel# ls
brandy                build_mali_driver.sh  linux-4.9
build                 build_scripts.sh      pctools
build_linux_kernel.sh chips                 README.md
build_mainline_kernel.sh config_linux_kernel.sh u-boot-2017.03
build_mainline_uboot.sh linux-3.4.113
root@ubuntu: /home/bob/bakup/orangepi_h3_linux/OrangePi-Kernel#
```

里面有 u-boot 编译脚本，主线内核编译脚本和非主线内核编译脚本

三、编译 u-boot

`./build_mainline_uboot.sh pc`

其他板子参考 README.md 文件，不过是英文的，幸亏我在蓝翔的时候学过一点英语。

```
README.md (/home/bob/bakup/orangepi_h3_linux/OrangePi-Kernel) - gedit
File Edit View Search Tools Documents Help
Open [icon] Save

Building kernel 3.4.113
-----

Kernel config files and the files specific to OPI board are placed in **build** directory.

The included build script *build_linux_kernel.sh* can be used to build the kernel<br />
`./build_linux_kernel.sh [clean | all | 2 | plus] [clean]`

**clean** as 1st parameter cleans the kernel tree and build directories<br />
**clean** as 2nd parameter cleans the kernel tree before build<br />
**all** builds the uImage for OPI-2 & OPI-PLUS<br />
**2** builds the uImage for OPI-2 OPI-PCPLUS OPI-ONE OPI-PC<br />
**plus** builds the uImage for OPI-PLUS OPI-PLUS2E<br />

After the build the resulting kernel files (uImage and kernel modules) are placed into **build** directory.

To build **script.bin** for all OPI boards and resolutions run:<br />
`./build_scripts [clean]`<br />
**clean** as 1st parameter cleans the scripts and logs<br />

After the build the *script.bin* are placed into **build** directory.

To **configure kernel** run:<br />
`./config_linux_kernel.sh`

-----
Building u-boot-2017.03 and kernel 4.9
-----

The included build script *build_mainline_uboot.sh and build_mainline_kernel.sh* can be used to
build u-boot and kernel.

`./build_mainline_uboot.sh [2 | one | pc | pc-plus | plus | lite | plus2e]`
`./build_mainline_kernel.sh [opi | clean]`

Markdown ▾ Tab Width: 8 ▾ Ln 1, Col 1 ▾ INS
```

u-boot 编译后界面:

```
root@ubuntu: /home/bob/bakup/orangepi_h3_linux/OrangePi-Kernel
CC      cmd/test.o
CC      cmd/usb.o
CC      cmd/disk.o
CC      cmd/ximg.o
CC      cmd/nvedit.o
LD      cmd/built-in.o
Thu May 11 22:57:39 CST 2017

/home/bob/bakup/orangepi_h3_linux/OrangePi-Kernel/build/uboot

cp: cannot stat '../u-boot-2017.03/u-boot-sunxi-with-spl.bin': No such file or directory
*****Compile uboot OK*****
chmod: cannot access 'u-boot-sunxi-with-spl.bin': No such file or directory
Image Name:
Created:   Thu May 11 22:57:39 2017
Image Type: ARM Linux Script (uncompressed)
Data Size: 501 Bytes = 0.49 kB = 0.00 MB
Load Address: 00000000
Entry Point: 00000000
Contents:
  Image 0: 493 Bytes = 0.48 kB = 0.00 MB
cp: cannot stat 'u-boot-sunxi-with-spl.bin': No such file or directory
root@ubuntu: /home/bob/bakup/orangepi_h3_linux/OrangePi-Kernel#
```

提示没有文件: u-boot-sunxi-with-spl.bin, 真是要命, u-boot-sunxi-with-spl.bin 文件才是我们要写入 SD 卡的文件。粗略看了下 u-boot 目录下的 Makefile 文件, 发现在转换为 u-boot-sunxi-with-spl.bin 的时候出错了。

Makefile 摘取:

```
.....
ifneq ($(CONFIG_ARCH_SUNXI),)
u-boot-sunxi-with-spl.bin: spl/sunxi-spl.bin u-boot.img u-boot.dtb FORCE
    $(call if_changed,binman)
endif
.....
```

其中 u-boot-sunxi-with-spl.bin 是另外一个文件 u-boot-sunxi-with-spl.bin.cmd, 主要是用 binman 工具转换 xx_spl.bin 文件, 其中的错误也不想深究了, 于是不用脚本编译了, 直接下发命令编译。

1) 配置单板类型

命令: make CROSS_COMPILE=arm-linux-gnueabi- orangepi_pc_defconfig
主要是目的是将默认配置文件 orangepi_pc_defconfig 去覆盖 config

2) 编译 u-boot

命令: make CROSS_COMPILE=arm-linux-gnueabi-
用 arm-linux-gnueabi 也可以, 两者只是浮点方案不一样。不用纠结这个了

3) 编译结果

```
root@ubuntu: /home/bob/bakup/orangepi_h3_linux/OrangePi-Kernel/u-boot-2017.03
CC      spl/drivers/mmc/mmc.o
CC      spl/drivers/mmc/sunxi_mmc.o
LD      spl/drivers/mmc/built-in.o
CC      spl/drivers/power/sy8106a.o
LD      spl/drivers/power/built-in.o
LD      spl/drivers/power/pmic/built-in.o
LD      spl/drivers/power/regulator/built-in.o
CC      spl/drivers/serial/serial.o
CC      spl/drivers/serial/serial_ns16550.o
CC      spl/drivers/serial/ns16550.o
LD      spl/drivers/serial/built-in.o
LD      spl/drivers/built-in.o
LD      spl/dts/built-in.o
LD      spl/fs/built-in.o
LDS     spl/u-boot-spl.lds
LD      spl/u-boot-spl
OBJCOPY spl/u-boot-spl-nodtb.bin
COPY    spl/u-boot-spl.bin
MKSUNXI spl/sunxi-spl.bin
OBJCOPY u-boot-nodtb.bin
CAT      u-boot-dtb.bin
COPY    u-boot.bin
MKIMAGE u-boot.img
COPY    u-boot.dtb
BINMAN   u-boot-sunxi-with-spl.bin
OBJCOPY u-boot.srec
SYM      u-boot.sym
MKIMAGE u-boot-dtb.img
CFGCHK   u-boot.cfg
root@ubuntu: /home/bob/bakup/orangepi_h3_linux/OrangePi-Kernel/u-boot-2017.03#
```

于是得到了我们想要的文件 u-boot-sunxi-with-spl.bin

四、 烧写 u-boot-sunxi-with-spl.bin 到 SD 启动分区

1) 烧写之前，先将 SD 卡分区（其实烧写不一定要分区，只是我认为这个是一个好习惯）

a) 查看设备： **dmesg|tail -20**

```
root@ubuntu: /# dmesg|tail -20
[ 7235.720769] usb 1-1: Manufacturer: Generic
[ 7235.720771] usb 1-1: SerialNumber: 000000000272
[ 7236.091617] usb-storage 1-1:1.0: USB Mass Storage device detected
[ 7236.094449] scsi host33: usb-storage 1-1:1.0
[ 7236.094821] usbcore: registered new interface driver usb-storage
[ 7236.131568] usbcore: registered new interface driver uas
[ 7237.100152] scsi 33:0:0:0: Direct-Access     Generic STORAGE DEVICE    0272 PQ: 0 ANSI: 0
[ 7237.106766] sd 33:0:0:0: Attached scsi generic sg2 type 0
[ 7237.412517] sd 33:0:0:0: [sdb] 15728640 512-byte logical blocks: (8.05 GB/7.50 GiB)
[ 7237.420500] sd 33:0:0:0: [sdb] Write Protect is off
[ 7237.420504] sd 33:0:0:0: [sdb] Mode Sense: 0b 00 00 08
[ 7237.428520] sd 33:0:0:0: [sdb] No Caching mode page found
[ 7237.428526] sd 33:0:0:0: [sdb] Assuming drive cache: write through
[ 7237.462511] sdb: sdb1
[ 7237.493529] sd 33:0:0:0: [sdb] Attached SCSI removable disk
[ 7238.331331] EXT4-fs (sdb1): recovery complete
[ 7238.331338] EXT4-fs (sdb1): mounted filesystem with writeback data mode. Opts: (null)
[ 7785.958277] sdb: sdb1
[ 7858.667717] sdb: sdb1 sdb2
[ 7858.709676] sdb: sdb1 sdb2
root@ubuntu: /#
```

b) 看挂载点： **df -l**

```
root@ubuntu: /# df -l
Filesystem      1K-blocks      Used Available Use% Mounted on
udev            1012804          0    1012804   0% /dev
tmpfs            206176      6460     199716   4% /run
/dev/sda1       60763196 20824216  36829348  37% /
tmpfs           1030864        212    1030652   1% /dev/shm
tmpfs            5120           4       5116   1% /run/lock
tmpfs           1030864          0    1030864   0% /sys/fs/cgroup
tmpfs           206176         52     206124   1% /run/user/1000
/dev/sdb1       1377276    1181284    150960  89% /media/bob/7582dc74-8ced-4b05-aebc-136e279105cb
root@ubuntu: /#
```

c) 卸载挂载

```

root@ubuntu:/# df -l
Filesystem      1K-blocks      Used Available Use% Mounted on
udev             1012804          0    1012804   0% /dev
tmpfs            206176          6460    199716    4% /run
/dev/sda1        60763196 20824216    36829348   37% /
tmpfs            1030864          212    1030652   1% /dev/shm
tmpfs             5120           4      5116    1% /run/lock
tmpfs            1030864          0    1030864   0% /sys/fs/cgroup
tmpfs            206176          52     206124   1% /run/user/1000
/dev/sdb1        1377276    1181284    150960    89% /media/bob/7582dc74-8ced-4b05-aebc-136e279105cb
root@ubuntu:/# umount /media/bob/7582dc74-8ced-4b05-aebc-136e279105cb
root@ubuntu:/# df -l
Filesystem      1K-blocks      Used Available Use% Mounted on
udev             1012804          0    1012804   0% /dev
tmpfs            206176          6452    199724    4% /run
/dev/sda1        60763196 20824176    36829388   37% /
tmpfs            1030864          212    1030652   1% /dev/shm
tmpfs             5120           4      5116    1% /run/lock
tmpfs            1030864          0    1030864   0% /sys/fs/cgroup
tmpfs            206176          52     206124   1% /run/user/1000
root@ubuntu:/#

```

注意一定要卸载挂载，因为在 ubuntu1604 一旦连接 U 盘它就自动给你挂载了，如果没有卸载挂载点，那么后面的分区无法保存成功，最后导致分区失败。我刚刚开始的时候就踩到这个坑了。

- d) 分区： `sudo fdisk /dev/sdb`

```

root@ubuntu:/# sudo fdisk /dev/sdb

Welcome to fdisk (util-linux 2.27.1).
Changes will remain in memory only, until you decide to write them.
Be careful before using the write command.


Command (m for help): p
Disk /dev/sdb: 7.5 GiB, 8053063680 bytes, 15728640 sectors
Units: sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disklabel type: dos
Disk identifier: 0x4f2c6c4d


Device            Boot Start        End    Sectors  Size Id Type
/dev/sdb1          2048 15414047 15412000   7.4G 83 Linux

Command (m for help): d
Selected partition 1
Partition 1 has been deleted.

Command (m for help): p
Disk /dev/sdb: 7.5 GiB, 8053063680 bytes, 15728640 sectors
Units: sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disklabel type: dos
Disk identifier: 0x4f2c6c4d

```



```

Disk identifier: 0x4f2c6c4d

Command (m for help): n
Partition type
  p   primary (0 primary, 0 extended, 4 free)
  e   extended (container for logical partitions)
Select (default p): p
Partition number (1-4, default 1): 1
First sector (2048-15728639, default 2048):
Last sector, +sectors or +size{K,M,G,T,P} (2048-15728639, default 15728639): +100MB

Created a new partition 1 of type 'Linux' and of size 95 MiB.

Command (m for help): n
Partition type
  p   primary (1 primary, 0 extended, 3 free)
  e   extended (container for logical partitions)
Select (default p):

Using default response p.
Partition number (2-4, default 2):
First sector (196608-15728639, default 196608):
Last sector, +sectors or +size{K,M,G,T,P} (196608-15728639, default 15728639):

Created a new partition 2 of type 'Linux' and of size 7.4 GiB.

Command (m for help): p
Disk /dev/sdb: 7.5 GiB, 8053063680 bytes, 15728640 sectors
Units: sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disklabel type: dos
Disk identifier: 0x4f2c6c4d

Device      Boot  Start      End  Sectors  Size Id Type
/dev/sdb1                2048    196607    194560    95M 83 Linux
/dev/sdb2            196608 15728639 15532032   7.4G 83 Linux

```

```

Disk identifier: 0x4f2c6c4d

Device      Boot  Start      End  Sectors  Size Id Type
/dev/sdb1                2048    196607    194560    95M 83 Linux
/dev/sdb2            196608 15728639 15532032   7.4G 83 Linux

Command (m for help): w
The partition table has been altered.
Calling ioctl() to re-read partition table.
Syncing disks.

root@ubuntu:/# ^C

```

- e) 修改分区格式：启动分区 100MB 为 FAT32,另外一个分区默认 ext，记得 w 保存退出

```

root@ubuntu:/# sudo fdisk /dev/sdb

Welcome to fdisk (util-linux 2.27.1).
Changes will remain in memory only, until you decide to write them.
Be careful before using the write command.

Command (m for help): t
Partition number (1,2, default 2): 1
Partition type (type L to list all types): L

   0  Empty                24  NEC DOS               81  Minix / old Lin   bf  Solaris
   1  FAT12                 27  Hidden NTFS Win    82  Linux swap / So  c1  DRDOS/sec (FAT-
   2  XENIX root            39  Plan 9             83  Linux             c4  DRDOS/sec (FAT-
   3  XENIX usr              3c  PartitionMagic     84  OS/2 hidden or   c6  DRDOS/sec (FAT-
   4  FAT16 <32M            40  Venix 80286        85  Linux extended   c7  Syrix
   5  Extended              41  PPC PReP Boot     86  NTFS volume set  da  Non-FS data
   6  FAT16                 42  SFS                87  NTFS volume set  db  CP/M / CTOS / .
   7  HPFS/NTFS/exFAT       4d  QNX4.x             88  Linux plaintext  de  Dell Utility
   8  AIX                   4e  QNX4.x 2nd part   8e  Linux LVM         df  BootIt
   9  AIX bootable          4f  QNX4.x 3rd part   93  Amoeba           e1  DOS access
  a  OS/2 Boot Manag       50  OnTrack DM        94  Amoeba BBT       e3  DOS R/O
  b  W95 FAT32              51  OnTrack DM6 Aux   9f  BSD/OS           e4  SpeedStor
  c  W95 FAT32 (LBA)        52  CP/M              a0  IBM Thinkpad hi  ea  Rufus alignment
  e  W95 FAT16 (LBA)        53  OnTrack DM6 Aux   a5  FreeBSD          eb  BeOS fs
  f  W95 Ext'd (LBA)       54  OnTrackDM6        a6  OpenBSD          ee  GPT
10  OPUS                   55  EZ-Drive          a7  NeXTSTEP         ef  EFI (FAT-12/16/
11  Hidden FAT12           56  Golden Bow        a8  Darwin UFS       f0  Linux/PA-RISC b
12  Compaq diagnost       5c  Priam Edisk        a9  NetBSD           f1  SpeedStor
14  Hidden FAT16 <3       61  SpeedStor         ab  Darwin boot      f4  SpeedStor
16  Hidden FAT16           63  GNU HURD or Sys   af  HFS / HFS+       f2  DOS secondary
17  Hidden HPFS/NTF       64  Novell Netware    b7  BSDI fs          fb  VMware VMFS
18  AST SmartSleep        65  Novell Netware    b8  BSDI swap        fc  VMware VMKCORE
1b  Hidden W95 FAT3       70  DiskSecure Mult   bb  Boot Wizard hid  fd  Linux raid auto
1c  Hidden W95 FAT3       75  PC/IX             bc  Acronis FAT32 L  fe  LANstep
1e  Hidden W95 FAT1       80  Old Minix         be  Solaris boot     ff  BBT

Partition type (type L to list all types): b

Partition type (type L to list all types): b
Changed type of partition 'W95 FAT32' to 'W95 FAT32'.

Command (m for help): p
Disk /dev/sdb: 7.5 GiB, 8053063680 bytes, 15728640 sectors
Units: sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disklabel type: dos
Disk identifier: 0x4f2c6c4d

Device      Boot  Start      End  Sectors  Size Id Type
/dev/sdb1             2048   196607   194560    95M  b W95 FAT32
/dev/sdb2          196608 15728639 15532032   7.4G 83 Linux

Command (m for help): w
The partition table has been altered.
Calling ioctl() to re-read partition table.
Syncing disks.

root@ubuntu:/#

```

f) 格式化分区: `sudo mkfs.vfat /dev/sdb1` 和 `sudo mkfs.ext3 /dev/sdb2`

```

root@ubuntu:/home/bob# sudo mkfs.vfat /dev/sdb1
mkfs.fat 3.0.28 (2015-05-16)
root@ubuntu:/home/bob# sudo mkfs.ext3 /dev/sdb2
mke2fs 1.42.13 (17-May-2015)
Creating filesystem with 1941504 4k blocks and 485760 inodes
Filesystem UUID: a6420928-03da-406c-961b-5102e3948dfc
Superblock backups stored on blocks:
    32768, 98304, 163840, 229376, 294912, 819200, 884736, 1605632

Allocating group tables: done
Writing inode tables: done
Creating journal (32768 blocks): done
Writing superblocks and filesystem accounting information: done

root@ubuntu:/home/bob#

```

需要稍等，有点慢。。。

- 2) 用 dd 命令烧写 u-boot-sunxi-with-spl.bin

```

root@ubuntu:/home/bob# find ./ -name u-boot-sunxi-with-spl.bin
./u-boot/u-boot-sunxi-with-spl.bin
./bakup/orangepi_h3_linux/OrangePi-Kernel/u-boot-2017.03/u-boot-sunxi-with-spl.bin
./orangepi_h3_linux/OrangePi-Kernel/u-boot-2017.03/u-boot-sunxi-with-spl.bin
root@ubuntu:/home/bob# sudo dd if=./orangepi_h3_linux/OrangePi-Kernel/u-boot-2017.03/u-boot-sunxi-with-spl.bin of=/dev/sdb bs=1024 seek=8
458+1 records in
458+1 records out
469423 bytes (469 kB, 458 KiB) copied, 1.17054 s, 401 kB/s
root@ubuntu:/home/bob#

```

- 3) 测试 u-boot-sunxi-with-spl.bin

把 SD 卡安装在 H3 的板子上并上电，看看串口(波特率 115200)终端打印的信息：

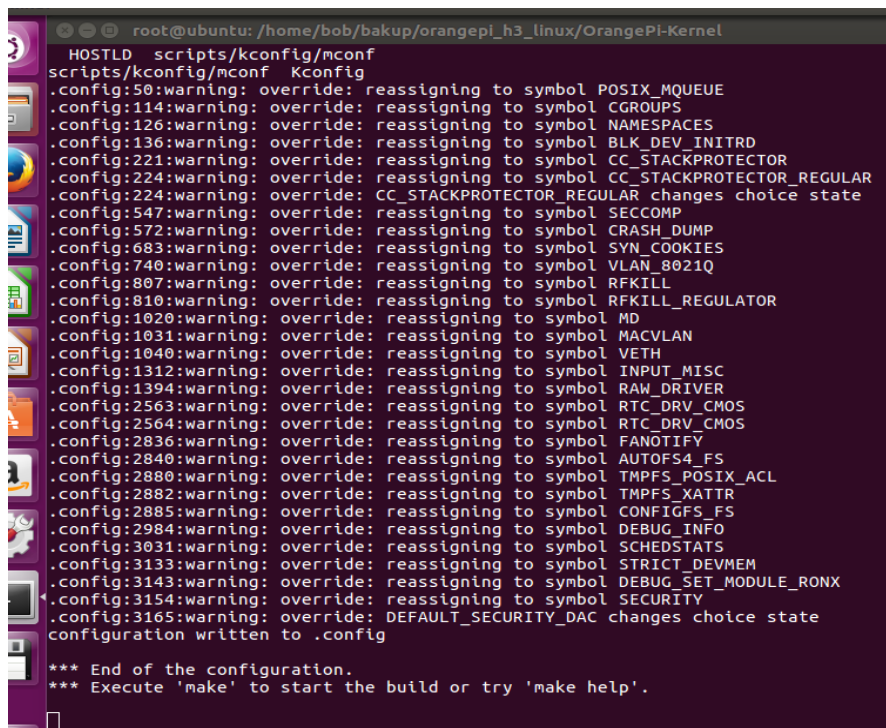
```

(COM3,115200) - PuTTY 打开成功
missing environment variable: bootfile
Retrieving file: pxelinux.cfg/00000000
*** ERROR: `serverip` not set
missing environment variable: bootfile
Retrieving file: pxelinux.cfg/00000000
*** ERROR: `serverip` not set
missing environment variable: bootfile
Retrieving file: pxelinux.cfg/00000000
*** ERROR: `serverip` not set
missing environment variable: bootfile
Retrieving file: pxelinux.cfg/000000
*** ERROR: `serverip` not set
missing environment variable: bootfile
Retrieving file: pxelinux.cfg/000000
*** ERROR: `serverip` not set
missing environment variable: bootfile
Retrieving file: pxelinux.cfg/0000
*** ERROR: `serverip` not set
missing environment variable: bootfile
Retrieving file: pxelinux.cfg/000
*** ERROR: `serverip` not set
missing environment variable: bootfile
Retrieving file: pxelinux.cfg/00
*** ERROR: `serverip` not set
missing environment variable: bootfile
Retrieving file: pxelinux.cfg/0
*** ERROR: `serverip` not set
missing environment variable: bootfile
Retrieving file: pxelinux.cfg/default-arm-sunxi
*** ERROR: `serverip` not set
missing environment variable: bootfile
Retrieving file: pxelinux.cfg/default-arm
*** ERROR: `serverip` not set
missing environment variable: bootfile
Retrieving file: pxelinux.cfg/default
*** ERROR: `serverip` not set
Config file not found
BOOTP broadcast 1
BOOTP broadcast 2
BOOTP broadcast 3

```

五、编译内核（主线内核）

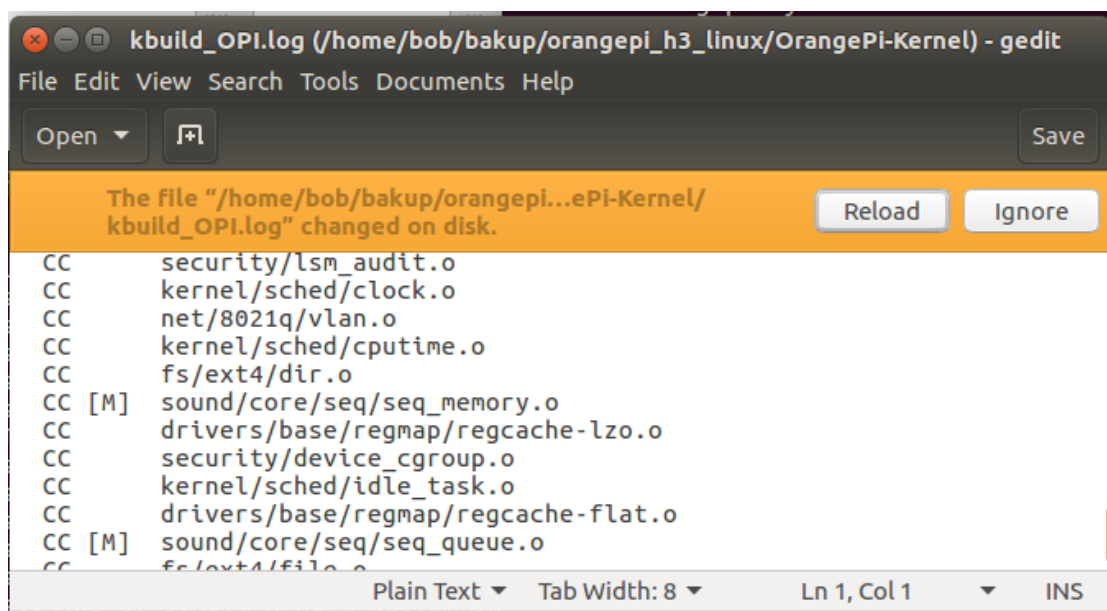
- 1) `sudo ./build_mainline_kernel.sh clean`
 - 2) `sudo ./build_mainline_kernel.sh opi`
- 保存内核配置后就会开始编译，如下图：



```
root@ubuntu: /home/bob/bakup/orangepi_h3_linux/OrangePI-Kernel
HOSTLD scripts/kconfig/mconf
scripts/kconfig/mconf Kconfig
.config:50:warning: override: reassigning to symbol POSIX_MQUEUE
.config:114:warning: override: reassigning to symbol CGROUPS
.config:126:warning: override: reassigning to symbol NAMESPACES
.config:136:warning: override: reassigning to symbol BLK_DEV_INITRD
.config:221:warning: override: reassigning to symbol CC_STACKPROTECTOR
.config:224:warning: override: reassigning to symbol CC_STACKPROTECTOR_REGULAR
.config:224:warning: override: CC_STACKPROTECTOR_REGULAR changes choice state
.config:547:warning: override: reassigning to symbol SECCOMP
.config:572:warning: override: reassigning to symbol CRASH_DUMP
.config:683:warning: override: reassigning to symbol SYN_COOKIES
.config:740:warning: override: reassigning to symbol VLAN_8021Q
.config:807:warning: override: reassigning to symbol RFKILL
.config:810:warning: override: reassigning to symbol RFKILL_REGULATOR
.config:1020:warning: override: reassigning to symbol MD
.config:1031:warning: override: reassigning to symbol MACVLAN
.config:1040:warning: override: reassigning to symbol VETH
.config:1312:warning: override: reassigning to symbol INPUT_MISC
.config:1394:warning: override: reassigning to symbol RAW_DRIVER
.config:2563:warning: override: reassigning to symbol RTC_DRV_CMOS
.config:2564:warning: override: reassigning to symbol RTC_DRV_CMOS
.config:2836:warning: override: reassigning to symbol FANOTIFY
.config:2840:warning: override: reassigning to symbol AUTOFS4_FS
.config:2880:warning: override: reassigning to symbol TMPFS_POSIX_ACL
.config:2882:warning: override: reassigning to symbol TMPFS_XATTR
.config:2885:warning: override: reassigning to symbol CONFIGFS_FS
.config:2984:warning: override: reassigning to symbol DEBUG_INFO
.config:3031:warning: override: reassigning to symbol SCHEDSTATS
.config:3133:warning: override: reassigning to symbol STRICT_DEVMEM
.config:3143:warning: override: reassigning to symbol DEBUG_SET_MODULE_RONX
.config:3154:warning: override: reassigning to symbol SECURITY
.config:3165:warning: override: DEFAULT_SECURITY_DAC changes choice state
configuration written to .config

*** End of the configuration.
*** Execute 'make' to start the build or try 'make help'.
```

编译过程打印的信息已全部写入日志 `kbuild_OPI.log`，可以查看这个日志了解编译进度。



```
kbuild_OPI.log (/home/bob/bakup/orangepi_h3_linux/OrangePI-Kernel) - gedit
File Edit View Search Tools Documents Help
Open Save

The file "/home/bob/bakup/orangepi...ePi-Kernel/
kbuild_OPI.log" changed on disk. Reload Ignore

CC security/lsm_audit.o
CC kernel/sched/clock.o
CC net/8021q/vlan.o
CC kernel/sched/cputime.o
CC fs/ext4/dir.o
CC [M] sound/core/seq/seq_memory.o
CC drivers/base/regmap/regcache-lzo.o
CC security/device_cgroup.o
CC kernel/sched/idle_task.o
CC drivers/base/regmap/regcache-flat.o
CC [M] sound/core/seq/seq_queue.o
CC fs/ext4/file.o

Plain Text Tab Width: 8 Ln 1, Col 1 INS
```

当然，如果不爽，也可以用命令行不要脚本编译内核，自己决定。

编译后得到文件 `zImage` 和 `ulmage`，将 `ulmage` 拷贝到 SD 卡的启动分区

编译完成信息：

```

root@ubuntu: /home/bob/bakup/orangepi_h3_linux/OrangePi-Kernel
.config:3133:warning: override: reassigning to symbol STRICT_DEVMEM
.config:3143:warning: override: reassigning to symbol DEBUG_SET_MODULE_RONX
.config:3154:warning: override: reassigning to symbol SECURITY
.config:3165:warning: override: DEFAULT_SECURITY_DAC changes choice state
configuration written to .config

*** End of the configuration.
*** Execute 'make' to start the build or try 'make help'.

Building kernel & modules ...
Image Name:      linux-4.9
Created:         Fri May 12 16:08:54 2017
Image Type:      ARM Linux Kernel Image (uncompressed)
Data Size:       3563504 Bytes = 3479.98 kB = 3.40 MB
Load Address:    48000000
Entry Point:     48000000
Exporting modules ...
Exporting firmware ...
cp: cannot create regular file '../build/lib/uImage': No such file or directory
***OK***
root@ubuntu: /home/bob/bakup/orangepi_h3_linux/OrangePi-Kernel#

```

提示创建 uImage 失败，但是实际上在这路径上已经创建成功

orangepi_h3_linux/OrangePi-Kernel/linux-4.9/output

六、用 busybox 制作 rootfs

- 1) 建一个目录 myrootfs
- 2) 下载和解压 busybox (busybox-1.23.2.tar.bz2)
- 3) 建一个目录 myrootfs/rootfs
- 4) 在目录 myrootfs/rootfs 下建一下空目录

以上步骤结果：

```

root@ubuntu: /home/bob/bakup/myrootfs# ls
busybox-1.23.2 busybox-1.23.2.tar.bz2 rootfs
root@ubuntu: /home/bob/bakup/myrootfs# cd rootfs/
root@ubuntu: /home/bob/bakup/myrootfs/rootfs# ls
bin dev etc lib mnt proc sbin sys tmp usr var
root@ubuntu: /home/bob/bakup/myrootfs/rootfs#

```

在继续建立下一级空目录

usr/bin

usr/sbin

usr/lib

lib/moudules

```

root@ubuntu: /home/bob/bakup/myrootfs/rootfs/usr# ls
bin lib sbin
root@ubuntu: /home/bob/bakup/myrootfs/rootfs/usr# cd ..
root@ubuntu: /home/bob/bakup/myrootfs/rootfs# cd lib/
root@ubuntu: /home/bob/bakup/myrootfs/rootfs/lib# ls
modules
root@ubuntu: /home/bob/bakup/myrootfs/rootfs/lib#

```

- 5) 在空目录 dev 下创建设备文件

mknod -m 666 console c 5 1

mknod -m 666 null c 1 3

创建设备文件结果：

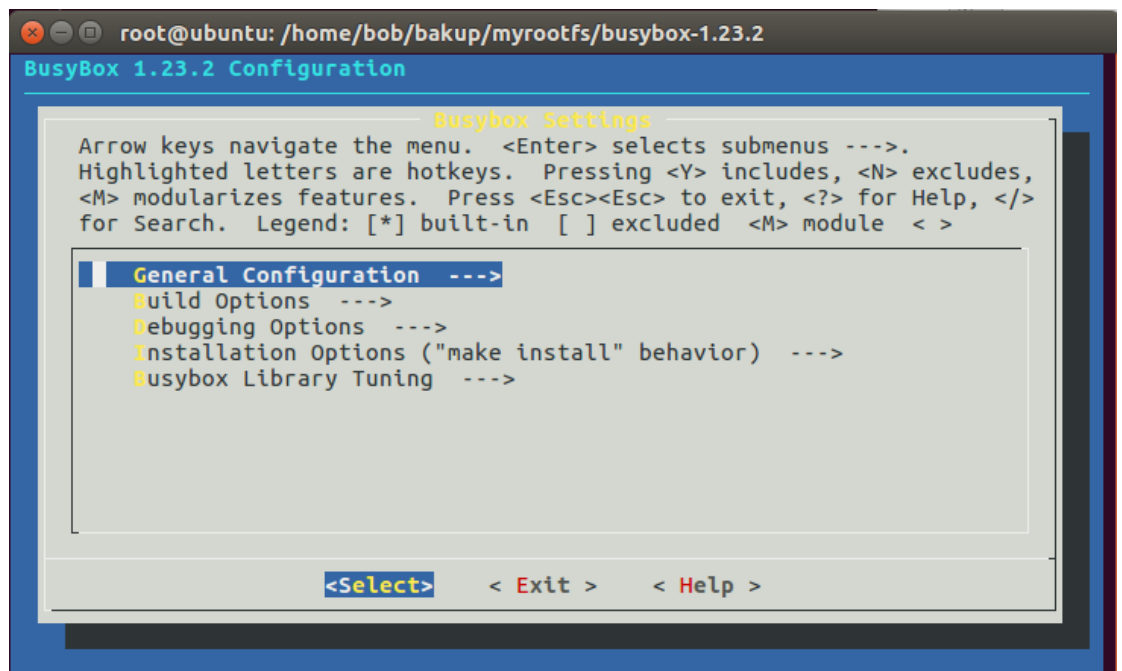
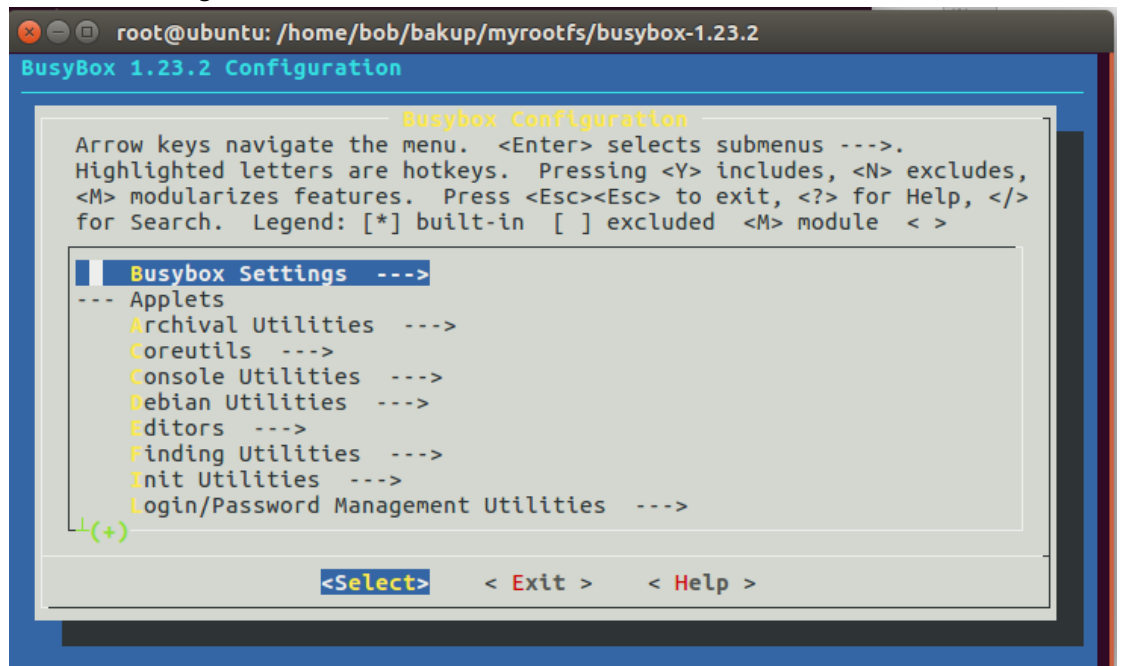
```
root@ubuntu:/home/bob/bakup/myrootfs/rootfs/lib# cd ..
root@ubuntu:/home/bob/bakup/myrootfs/rootfs# cd dev
root@ubuntu:/home/bob/bakup/myrootfs/rootfs/dev# ls
console null
root@ubuntu:/home/bob/bakup/myrootfs/rootfs/dev#
```

6) etc 目录

这个目录下的文件不好写，直接从 [orange pi](#) 官网上下载的 armbian 镜像系统 etc 下面的内容全部拷贝过来就可以了。

7) 配置和编译 busybox(和配置内核一样，不过这里是 busybox)

1) make menuconfig



```
root@ubuntu: /home/bob/bakup/myrootfs/busybox-1.23.2
BusyBox 1.23.2 Configuration

Installation Options ("make install" behavior)
Arrow keys navigate the menu. <Enter> selects submenus --->.
Highlighted 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 < >

What kind of applet links to install (as script wrappers) --->
/bin/sh applet link (as soft-link) --->
(/home/bob/bakup/myrootfs/rootfs) BusyBox installation prefix

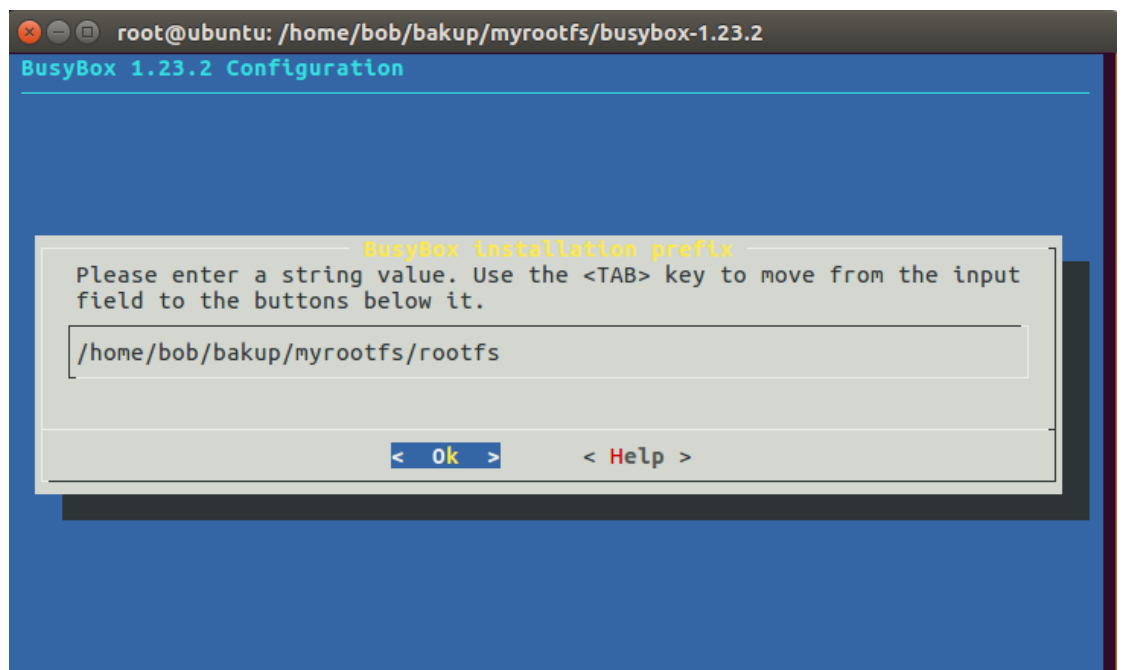
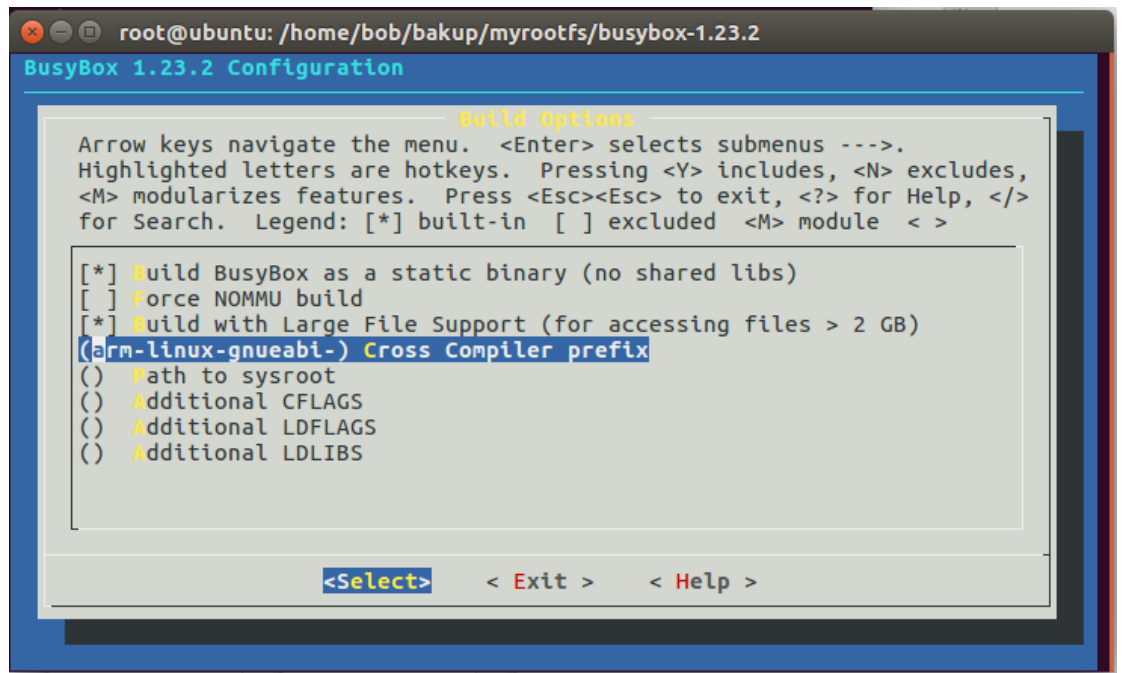
<Select> < Exit > < Help >
```

```
root@ubuntu: /home/bob/bakup/myrootfs/busybox-1.23.2
BusyBox 1.23.2 Configuration

Busybox Settings
Arrow keys navigate the menu. <Enter> selects submenus --->.
Highlighted 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 < >

General Configuration --->
Build Options --->
Debugging Options --->
Installation Options ("make install" behavior) --->
Busybox Library Tuning --->

<Select> < Exit > < Help >
```



然后保存配置并退出

2) 编译 busybox

make ARCH=arm CROSS_COMPILE=arm-linux-gnueabi-


```

root@ubuntu: /home/bob/bakup/myrootfs/busybox-1.23.2
ed with attribute warn_unused_result [-Wunused-result]
    fgets(line_old, sizeof(line_old), fp);
    ^
    GEN    include/applet_tables.h
    CC     applets/applets.o
    LD     applets/built-in.o
    HOSTCC applets/usage_pod
applets/usage_pod.c: In function 'main':
applets/usage_pod.c:74:3: warning: format not a string literal and no format arguments [-Wformat-security]
    printf(usage_array[i].aname);
    ^
    CC     libbb/appletlib.o
    AR     libbb/lib.a
    LINK   busybox_unstripped
Trying libraries: crypt m
Library crypt is not needed, excluding it
Library m is needed, can't exclude it (yet)
Final link with: m
DOC      busybox.pod
DOC      BusyBox.txt
DOC      busybox.1
DOC      BusyBox.html
root@ubuntu: /home/bob/bakup/myrootfs/busybox-1.23.2#

```

3) 安装 busybox 到我们的 rootfs

命令: make install

安装完毕结果:

```

root@ubuntu: /home/bob/bakup/myrootfs/busybox-1.23.2
/home/bob/bakup/myrootfs/rootfs//usr/sbin/remove-shell
/home/bob/bakup/myrootfs/rootfs//usr/sbin/rtcwake
/home/bob/bakup/myrootfs/rootfs//usr/sbin/sendmail
/home/bob/bakup/myrootfs/rootfs//usr/sbin/setfont
/home/bob/bakup/myrootfs/rootfs//usr/sbin/setlogcons
/home/bob/bakup/myrootfs/rootfs//usr/sbin/svlogd
/home/bob/bakup/myrootfs/rootfs//usr/sbin/telnetd
/home/bob/bakup/myrootfs/rootfs//usr/sbin/tftpd
/home/bob/bakup/myrootfs/rootfs//usr/sbin/ubiattach
/home/bob/bakup/myrootfs/rootfs//usr/sbin/ubidetach
/home/bob/bakup/myrootfs/rootfs//usr/sbin/ubimkvol
/home/bob/bakup/myrootfs/rootfs//usr/sbin/ubirmvol
/home/bob/bakup/myrootfs/rootfs//usr/sbin/ubirsvol
/home/bob/bakup/myrootfs/rootfs//usr/sbin/ubiupdatevol
/home/bob/bakup/myrootfs/rootfs//usr/sbin/udhcpd

-----
You will probably need to make your busybox binary
setuid root to ensure all configured applets will
work properly.
-----
root@ubuntu: /home/bob/bakup/myrootfs/busybox-1.23.2#

```

这样，我们之前创建的 sbin bin usr 空目录被拷贝很多文件进来了：

```

root@ubuntu: /home/bob/bakup/myrootfs/rootfs/usr
chown      fatattr      linux32      mt           rpm           umount
conspy     fdflush      linux64      mv           run-parts    uname
cp         fgrep        ln           netstat      scriptreplay  usleep
cpio       fsync        login        nice         sed           vi
cttyhack   getopt       ls           pidof        setarch       watch
date       grep         lsattr      ping         setserial     zcat
dd         gunzip       lzop        ping6        sh

root@ubuntu: /home/bob/bakup/myrootfs/rootfs/bin# cd ..
root@ubuntu: /home/bob/bakup/myrootfs/rootfs# cd sbin/
root@ubuntu: /home/bob/bakup/myrootfs/rootfs/sbin# ls
acpid      fsck        insmod      makedevs     poweroff      switch_root
adjtimex   fsck.minix  ip          mdev          raidautorun   sysctl
arp        fstrim      ipaddr      mkdosfs      reboot        syslogd
blkid      getty       iplink      mke2fs        rmmod         tuncctl
blockdev   halt        iproute     mkfs.ext2     route         udhcpc
bootchartd hdparm      iprule      mkfs.minix    runlevel      vconfig
depmod     hwclock     iptunnel    mkfs.vfat     setconsole    watchdog
devmem     ifconfig    klogd       mkswap        slattach      zcip
fb splash   ifdown      loadkmap    modinfo       start-stop-daemon
fdisk      ifenslave   logread     modprobe      sulogin
findfs     ifup        losetup     nameif        swapoff
freeramdisk init         lsmod       pivot_root    swapon

root@ubuntu: /home/bob/bakup/myrootfs/rootfs/sbin# cd ..

```

- 4) 建内核 init 进程

命令: `ln -s ./bin/busybox init`

```

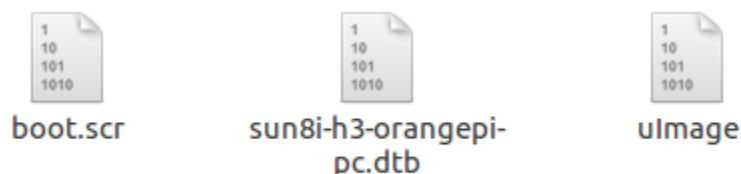
root@ubuntu: /home/bob/bakup/myrootfs/rootfs# ls
bin dev etc lib linuxrc mnt proc sbin sys tmp usr var
root@ubuntu: /home/bob/bakup/myrootfs/rootfs# ln -s ./bin/busybox init
root@ubuntu: /home/bob/bakup/myrootfs/rootfs# ls
bin dev etc init lib linuxrc mnt proc sbin sys tmp usr var
root@ubuntu: /home/bob/bakup/myrootfs/rootfs#

```

看见产生了一个连接文件 init，这个文件连接到 bin/busybox

- 5) 把 rootfs 下的所有文件拷贝到 SD 卡的第二个分区(ext4)
- 6) 拷贝文件 sun8i-h3-orangepi-pc.dtb 到 SD 卡启动分区

SD 卡启动分区(第一个分区 100MB)的文件如下:



七、 运行移植的系统

```
(COM3,115200) - PuTTY 打开成功
[ 1.201036] ehci-platform 1c1d000.usb: USB 2.0 started, EHCI 1.00
[ 1.207947] hub 4-0:1.0: USB hub found
[ 1.211786] hub 4-0:1.0: 1 port detected
[ 1.216191] sun6i-rtc 1f00000.rtc: setting system clock to 1970-01-01 00:00:05 UTC
(5)
[ 1.224384] vcc3v0: disabling
[ 1.227366] vcc5v0: disabling
[ 1.241502] EXT4-fs (mmcblk0p2): mounted filesystem with ordered data mode. Opts: (
null)
[ 1.249665] VFS: Mounted root (ext4 filesystem) readonly on device 179:2.
[ 1.266067] devtmpfs: mounted
[ 1.270148] Freeing unused kernel memory: 1024K (c0800000 - c0900000)
/etc/rcS.d/S02fake-hwclock: .: line 18: can't open '/lib/lsb/init-functions'
/etc/rcS.d/S02hostname.sh: .: line 19: can't open '/lib/init/vars.sh'
/etc/rcS.d/S02mountkernfs.sh: .: line 15: can't open '/lib/lsb/init-functions'
/etc/rcS.d/S02setserial: .: line 35: can't open '/lib/lsb/init-functions'
/etc/rcS.d/S05mountdevsubfs.sh: .: line 24: can't open '/lib/lsb/init-functions'
/etc/rcS.d/S05procps: .: line 24: can't open '/lib/lsb/init-functions'
/etc/rcS.d/S06hwclock.sh: .: line 25: can't open '/lib/lsb/init-functions'
/etc/rcS.d/S07checkroot.sh: .: line 20: can't open '/lib/init/vars.sh'
/etc/rcS.d/S08checkfs.sh: .: line 18: can't open '/lib/init/vars.sh'
/etc/rcS.d/S09checkroot-bootclean.sh: .: line 16: can't open '/lib/lsb/init-functions'
/etc/rcS.d/S09etc-setserial: .: line 22: can't open '/lib/lsb/init-functions'
/etc/rcS.d/S09urandom: .: line 31: can't open '/lib/init/vars.sh'
/etc/rcS.d/S10mountall.sh: .: line 13: can't open '/lib/lsb/init-functions'
/etc/rcS.d/S10networking: .: line 20: can't open '/lib/lsb/init-functions'
/etc/rcS.d/S11mountall-bootclean.sh: .: line 14: can't open '/lib/lsb/init-functions'
/etc/rcS.d/S11mountnfs.sh: .: line 16: can't open '/lib/lsb/init-functions'
/etc/rcS.d/S12mountnfs-bootclean.sh: .: line 14: can't open '/lib/lsb/init-functions'
/etc/rcS.d/S13bootmisc.sh: .: line 13: can't open '/lib/lsb/init-functions'

Please press Enter to activate this console.
#
# ls
bin          etc          lib          lost+found  proc        sys          usr
dev          init         linuxrc      mnt         sbin        tmp          var
#
```

鸣谢:

女神 sunxi 不是人名!(860223860);

小撸 孙晞(1034825603)

蓝翔校友:

小王子与木头人(846863428);

✓ ws 鹏(858780646)

明月心惜(891845397)

Orange Pi(香橙派)创客群 2 (592292216) 的所有同学