

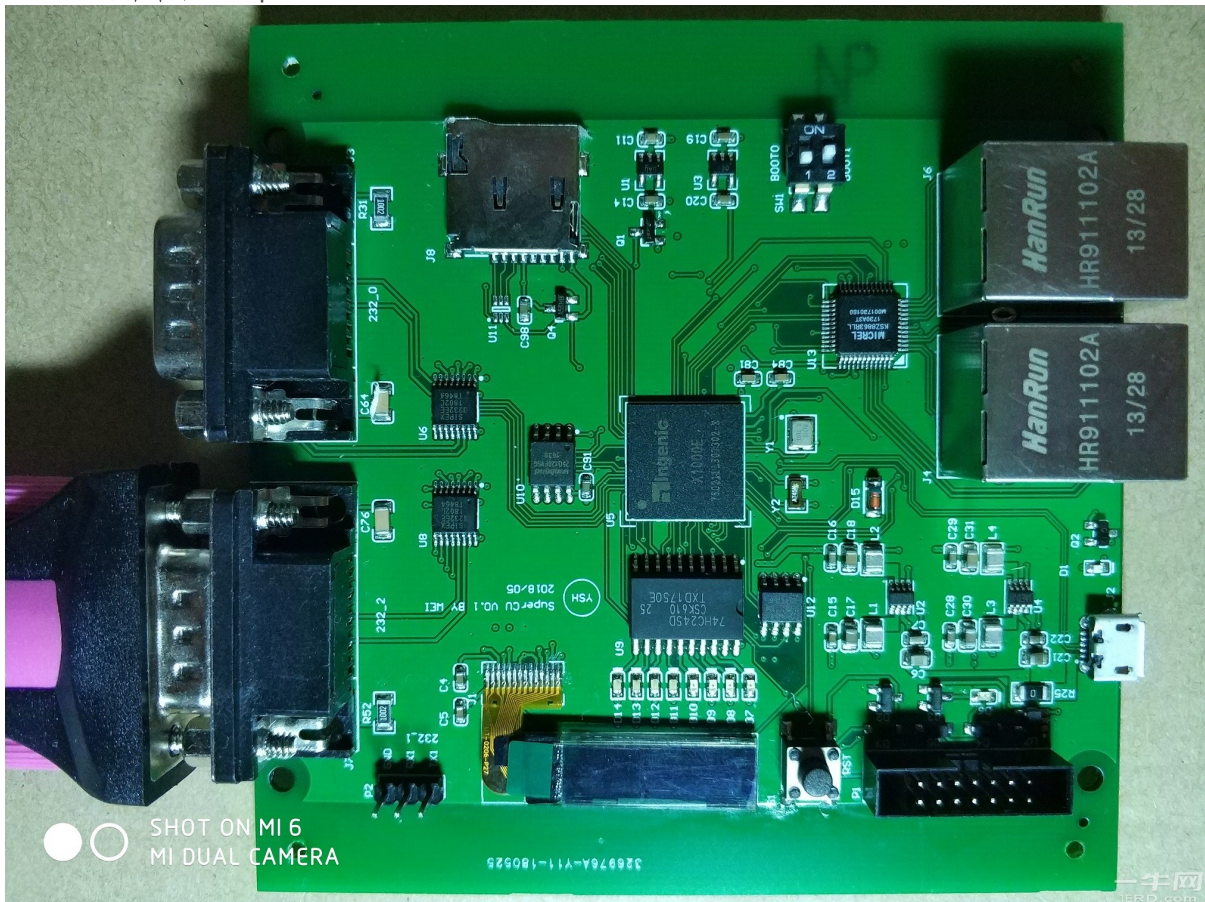
在树莓派和基于君正JZ4780的CI20板子上都跑过Debian8，突发奇想是否能把Debian8裁剪之后在君正X1000 E这一类的轻量化平台上跑起来，先开个帖子，近期准备来折腾折腾。

说一说大致的思路，预计有以下几个步骤：

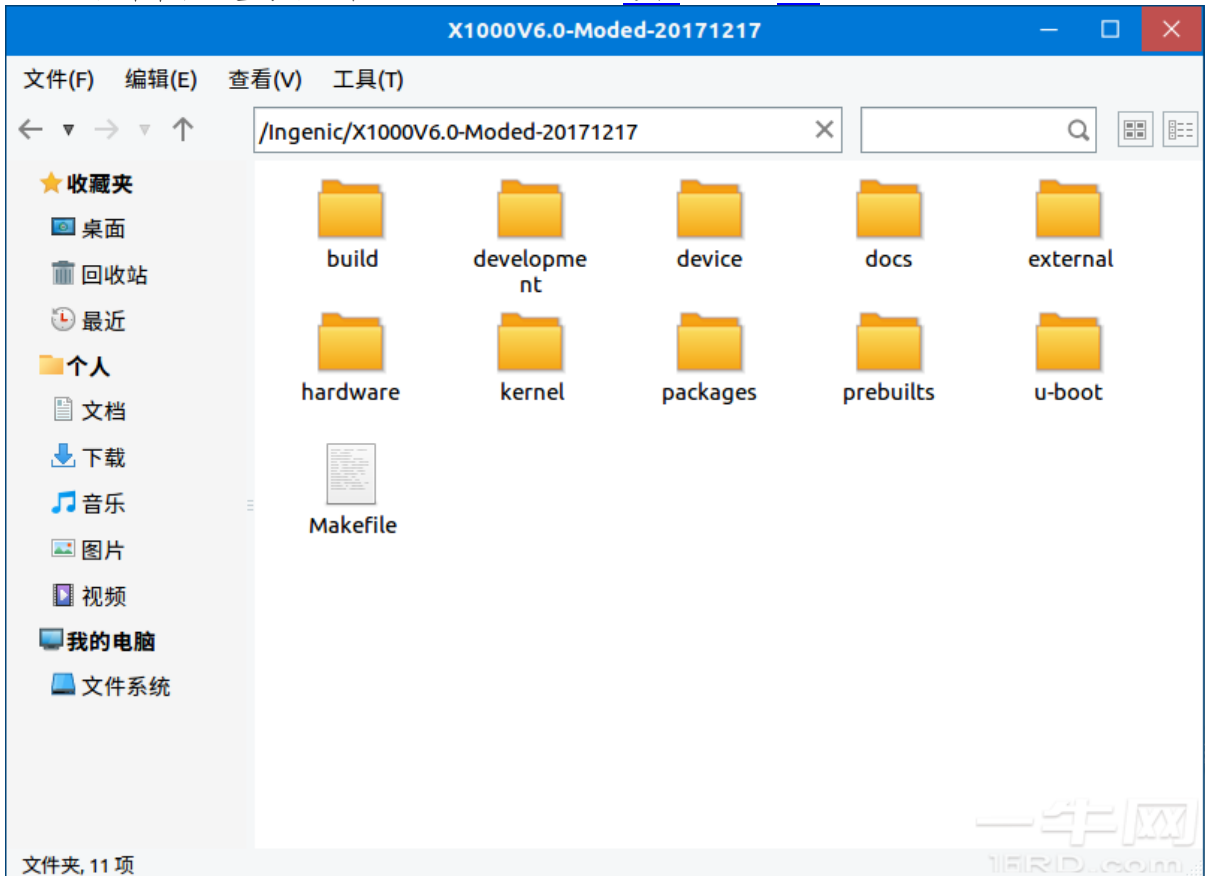
1. 基于已有的BSP，实现由TF卡启动系统。
2. 打包Debian8的最小rootfs，替代原BSP中的Buildroot。
3. 解决各种出现的bug。

基于现有BSP已经成功通过TF卡启动，下一步将尝试引导Debian8的Rootfs，基本情况如下：

- 硬件平台：SuperCU Basic V0.1。



- 软件平台：基于君正官方Phoenix v6.0 BSP移植的RatCharm BSP。



- 成果：已成功通过TF卡启动Build Root系统，并通过命令行查看/proc/cpuinfo信息。

```
#
#
# cat /proc/cpuinfo
system type      : RatCharm
machine         : Unknown
processor        : 0
cpu model       : Ingenic Xburst V4.15  FPU V0.0
BogoMIPS        : 1196.85
wait instruction : yes
microsecond timers : no
tlb_entries     : 32
extra interrupt vector : yes
hardware watchpoint : yes, count: 1, address/irw mask: [0x0fff]
isa             : mips32r1
ASEs implemented :
shadow register sets : 1
kscratch registers : 0
core            : 0
VCED exceptions  : not available
VCEI exceptions  : not available

Hardware        : RatCharm
Serial          : 00000000 00000000 00000000 00000000
#
```

- 以下是移植步骤：

1. 修改并编译用于TF卡启动的uboot。

因为所用硬件平台引出的调试串口是复用于Port D的UART2，因此需要修改打印串口的配置。

将u-boot文件夹中include/configs/RatCharm.h中关于主频的配置进行修改，以下是原始代码：

```
01. #define CONFIG_SYS_APLL_FREQ          1008000000 /*If APLL not use mast be set 0*/
02. #define CONFIG_SYS_MPLL_FREQ          600000000 /*If MPLL not use mast be set 0*/
03. #define CONFIG_CPU_SEL_PLL            APLL
04. #define CONFIG_DDR_SEL_PLL            MPLL
05. #define CONFIG_SYS_CPU_FREQ           1008000000
06. #define CONFIG_SYS_MEM_FREQ           200000000
```

[复制代码](#)

修改后的代码如下：

```
01. #define CONFIG_SYS_APLL_FREQ          1200000000 /*If APLL not use mast be set 0*/
02. #define CONFIG_SYS_MPLL_FREQ          600000000 /*If MPLL not use mast be set 0*/
03. #define CONFIG_CPU_SEL_PLL            APLL
04. #define CONFIG_DDR_SEL_PLL            MPLL
05. #define CONFIG_SYS_CPU_FREQ           1200000000
06. #define CONFIG_SYS_MEM_FREQ           200000000
```

[复制代码](#)

随后对同一文件下的关于打印串口的配置部分进行修改，以下是原始代码：

```
01. #define CONFIG_SYS_UART_INDEX          2
02. #if defined(CONFIG_RatCharm_BARE) || defined(CONFIG_RatCharm_RTTHREAD)
03. #define CONFIG_SYS_UART2_PD
04. #define CONFIG_BAUDRATE                115200
05. #else
06. #define CONFIG_SYS_UART2_PC
07. #define CONFIG_BAUDRATE                115200
08. #endif
```

[复制代码](#)

修改后的代码如下：

```
01. #define CONFIG_SYS_UART_INDEX                2
02. #if defined(CONFIG_RatCharm_BARE) || defined(CONFIG_RatCharm_RTTHREAD)
03. #define CONFIG_SYS_UART2_PD
04. #define CONFIG_BAUDRATE                    115200
05. #else
06. #define CONFIG_SYS_UART2_PD
07. #define CONFIG_BAUDRATE                    115200
08. #endif
```

[复制代码](#)

修改完毕后使用以下命令进行编译，编译耗时大约1分钟左右，待编译完成后，即可在u-boot目录下得到用于TF卡启动的u-boot-with-spl-mbr-[gpt](#).bin文件。

```
01. make RatCharm_uImage_msc0 -j4
```

[复制代码](#)

2. 修改并编译用于TF卡启动的kernel。(以make menuconfig方式为例)

前面提到过，由于调试串口使用的是Port D的UART2，因此首先需要在内核配置中修改相应配置。首先进入到kernel目录，使用make RatCharm [linux_defconfig](#)进行默认配置，然后将原来默认的Port C端口修改为Port D端口，具体如下：

```
01. | Symbol: SERIAL_JZ47XX_UART2_PD [=y] |
02. | Type : boolean |
03. | Prompt: PORT D |
04. | Location: |
05. | -> Device Drivers |
06. | -> Character devices |
07. | -> Serial drivers |
08. | -> ingenic jz47xx serial port support (SERIAL_JZ47XX_UART [=y]) |
09. | -> enable uart2 (SERIAL_JZ47XX_UART2 [=y]) |
10. | -> JZ SERIAL GPIO function pins select (<choice> [=y]) |
11. | Defined at drivers/tty/serial/Kconfig:1575 |
12. | Depends on: <choice> && SOC_X1000 [=y] |
```

[复制代码](#)

修改前后的截图对比，上图为修改前下图为修改后：

```
.config - Linux/mips 3.10.14 Kernel Configuration
> Device Drivers > Character devices > Serial drivers
Serial drivers
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 < >
^(-)
[*] Console on jz47xx and compatible serial port
[*] enable uart0
[ ] enable uart0 wakeup cpu
[ ] enable uart0 dma mode
[ ] enable uart1
[*] enable uart2
[ ] enable uart2 wakeup cpu
[ ] enable uart2 dma mode
[*] JZ SERIAL GPIO function pins select (PORT C) --->
[ ] enable uart3
+ (+)
<Select> < Exit > < Help > < Save > < Load >
```

Serial drivers

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 < >

- ^(-)
- [*] Console on jz47xx and compatible serial port
- [*] enable uart0
- [] enable uart0 wakeup cpu
- [] enable uart0 dma mode
- [] enable uart1
- [*] enable uart2
- [] enable uart2 wakeup cpu
- [] enable uart2 dma mode
- [*] JZ SERIAL GPIO function pins select (PORT D) --->
- [] enable uart3
- +(+)

<Select> < Exit > < Help > < Save > < Load >



现在的平台没有无线通信模块，因此修改相应配置关闭无线网络和蓝牙功能，共需修改三处：第一处，关闭Wireless LAN，具体如下：

- 01. | Symbol: WLAN [=n] |
- 02. | Type : boolean |
- 03. | Prompt: Wireless LAN |
- 04. | Location: |
- 05. | -> Device Drivers |
- 06. | -> Network device support (NETDEVICES [=y]) |
- 07. | Defined at drivers/net/wireless/Kconfig:5 |
- 08. | Depends on: NETDEVICES [=y] && !S390 && NET [=y] |
- 09. | Selects: WIRELESS [=n] |

复制代码

修改前后的截图对比，上图为修改前下图为修改后：

Network device support

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 < >

- ^(-)
- *** CAIF transport drivers ***
- Distributed Switch Architecture drivers --->
- [*] Ethernet driver support --->
- *- PHY Device support and infrastructure --->
- < > PPP (point-to-point protocol) support
- < > SLIP (serial line) support
- USB Network Adapters --->
- [*] Wireless LAN --->
- *** Enable WiMAX (Networking options) to see the WiMAX driv
- [] Wan interfaces support --->
- +(+)

<Select> < Exit > < Help > < Save > < Load >



```
.config - Linux/mips 3.10.14 Kernel Configuration
> Device Drivers > Network device support
```

Network device support

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 < >

^(-)

```
*** CAIF transport drivers ***
Distributed Switch Architecture drivers --->
[*] Ethernet driver support --->
-*- PHY Device support and infrastructure --->
< > PPP (point-to-point protocol) support
< > SLIP (serial line) support
USB Network Adapters --->
[*] Wireless LAN --->
    *** Enable WiMAX (Networking options) to see the WiMAX driv
[ ] Wan interfaces support --->
```

+(+)

<Select> <Exit> <Help> <Save> <Load>



16RD.com

第二处，关闭Wireless，具体如下：

```
01. | Symbol: WIRELESS [=n] |
02. | Type : boolean |
03. | Prompt: Wireless |
04. | Location: |
05. | -> Networking support (NET [=y]) |
06. | Defined at net/Kconfig:334 |
07. | Depends on: NET [=y] && !S390 |
08. | Selected by: WLAN [=n] && NETDEVICES [=y] && !S390 && NET [=y] |
```

复制代码

修改前后的截图对比，上图为修改前下图为修改后：

```
.config - Linux/mips 3.10.14 Kernel Configuration
> Networking support
```

Networking support

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 < >

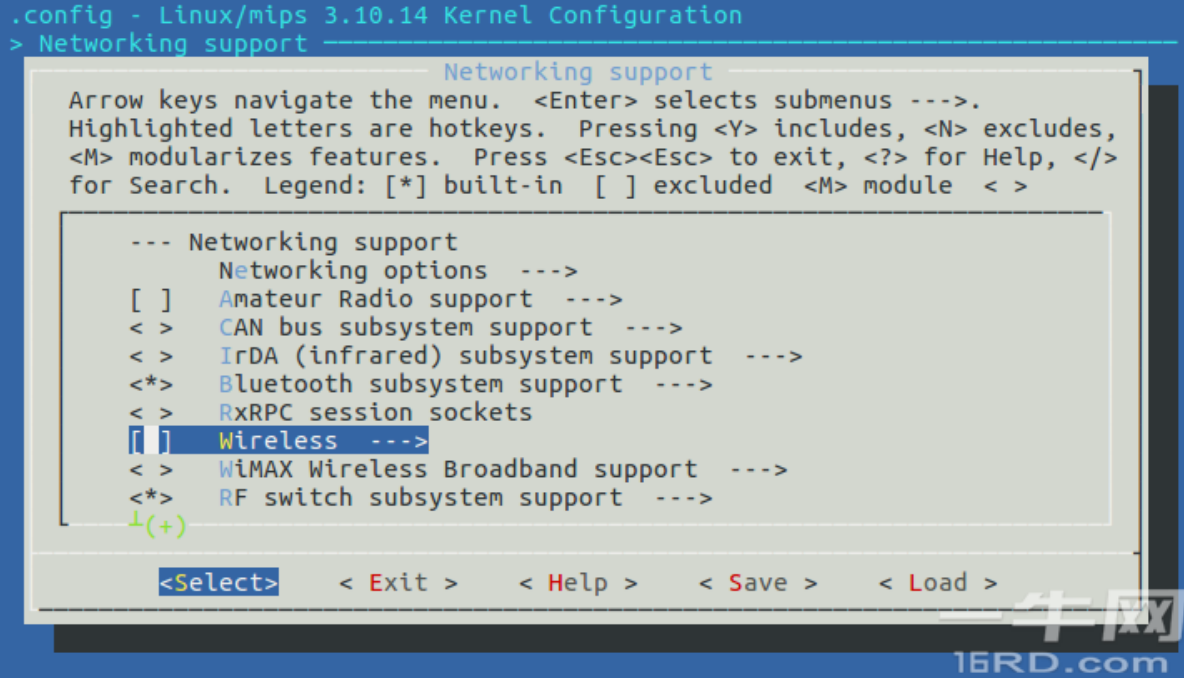
```
--- Networking support
    Networking options --->
[ ] Amateur Radio support --->
< > CAN bus subsystem support --->
< > IrDA (infrared) subsystem support --->
<*> Bluetooth subsystem support --->
< > RxRPC session sockets
[*] Wireless --->
< > WiMAX Wireless Broadband support --->
<*> RF switch subsystem support --->
```

+(+)

<Select> <Exit> <Help> <Save> <Load>



16RD.com



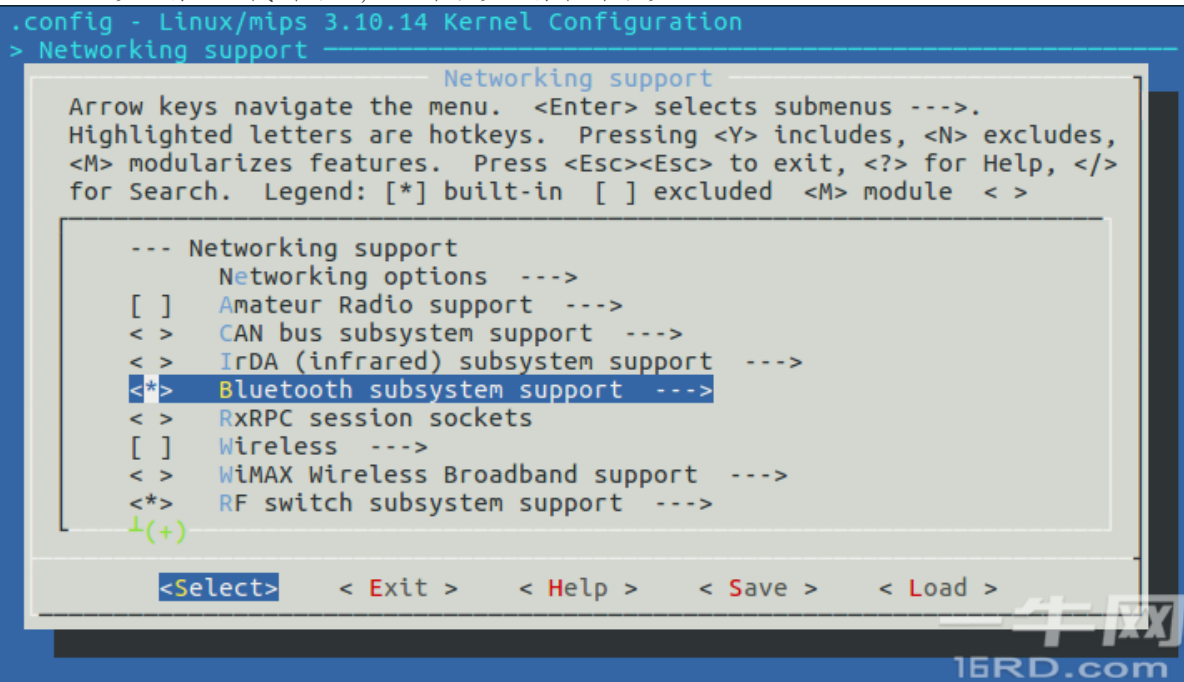
第三处，关闭Bluetooth subsystem support，具体如下：

```

01. | Symbol: BT [=n] |
02. | Type : tristate |
03. | Prompt: Bluetooth subsystem support |
04. | Location: |
05. | -> Networking support (NET [=y]) |
06. | Defined at net/bluetooth/Kconfig:5 |
07. | Depends on: NET [=y] && !S390 && (RFKILL [=n] || !RFKILL [=n]) |
08. | Selects: CRC16 [=y] && CRYPTO [=y] && CRYPTO_BLKCIPHER [=y] && \ |
09. | CRYPTO_AES [=y] && CRYPTO_ECB [=y] && CRYPTO_SHA256 [=y] |
```

复制代码

修改前后的截图对比，上图为修改前下图为修改后：



```
.config - Linux/mips 3.10.14 Kernel Configuration
> Networking support
```

Networking support

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 < >

```
--- Networking support
    Networking options --->
[ ] Amateur Radio support --->
< > CAN bus subsystem support --->
< > IrDA (infrared) subsystem support --->
[*] Bluetooth subsystem support --->
< > RxRPC session sockets
[ ] Wireless --->
< > WiMAX Wireless Broadband support --->
<*> RF switch subsystem support --->
```

↑(+)

<Select> <Exit> <Help> <Save> <Load>

16RD.com

关闭无线网络及蓝牙相关的功能后，相应的功率管制功能也需要关闭，共需修改两处。第一处，关闭RF switch subsystem support，具体如下：

```
01. | Symbol: RFKILL [=n] |
02. | Type : tristate |
03. | Prompt: RF switch subsystem support |
04. | Location: |
05. | -> Networking support (NET [=y]) |
06. | Defined at net/rfkill/Kconfig:4 |
07. | Depends on: NET [=y] |
```

复制代码

修改前后的截图对比，上图为修改前下图为修改后：

```
.config - Linux/mips 3.10.14 Kernel Configuration
> Networking support
```

Networking support

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 < >

```
^(-)
[ ] Amateur Radio support --->
< > CAN bus subsystem support --->
< > IrDA (infrared) subsystem support --->
< > Bluetooth subsystem support --->
< > RxRPC session sockets
[ ] Wireless --->
< > WiMAX Wireless Broadband support --->
[*] RF switch subsystem support --->
< > Plan 9 Resource Sharing Support (9P2000) --->
< > CAIF support --->
```

↑(+)

<Select> <Exit> <Help> <Save> <Load>

16RD.com

```
.config - Linux/mips 3.10.14 Kernel Configuration
> Networking support
```

Networking support

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 < >

```
^(-)
[ ] Amateur Radio support --->
< > CAN bus subsystem support --->
< > IrDA (infrared) subsystem support --->
< > Bluetooth subsystem support --->
< > RxRPC session sockets
[ ] Wireless --->
< > WiMAX Wireless Broadband support --->
< * > RF switch subsystem support --->
< > Plan 9 Resource Sharing Support (9P2000) --->
< > CAIF support --->
```

↓(+)

< Select > < Exit > < Help > < Save > < Load >

第二处，关闭BCM module power control core driver，具体如下：

```
01. | Symbol: BCM_PM_CORE [=n] |
02. | Type : tristate |
03. | Prompt: BCM module power control core driver |
04. | Location: |
05. | -> Device Drivers |
06. | -> Misc devices |
07. | Defined at drivers/misc/Kconfig:545 |
```

复制代码

修改前后的截图对比，上图为修改前下图为修改后：

```
.config - Linux/mips 3.10.14 Kernel Configuration
> Device Drivers > Misc devices
```

Misc devices

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 < >

```
^(-)
< > ROHM BH1780GLI ambient light sensor
< > BH1770GLC / SFH7770 combined ALS - Proximity sensor
< > APDS990X combined als and proximity sensors
< > Honeywell HMC6352 compass
< > Dallas DS1682 Total Elapsed Time Recorder with Alarm
[ ] UID based statistics tracking exported to /proc/uid_stat
< > BMP085 digital pressure sensor on I2C
< > FSA9480 USB Switch
< * > BCM module power control core driver
[ ] Generic on-chip SRAM driver
```

↓(+)

< Select > < Exit > < Help > < Save > < Load >

Misc devices

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 < >

^(-)

- < > ROHM BH1780GLI ambient light sensor
- < > BH1770GLC / SFH7770 combined ALS - Proximity sensor
- < > APDS990X combined als and proximity sensors
- < > Honeywell HMC6352 compass
- < > Dallas DS1682 Total Elapsed Time Recorder with Alarm
- [] UID based statistics tracking exported to /proc/uid_stat
- < > BMP085 digital pressure sensor on I2C
- < > FSA9480 USB Switch
- < > **BCM module power control core driver**
- [] Generic on-chip SRAM driver

+(+)

<Select> <Exit> <Help> <Save> <Load>

完成所有修改后使用以下命令进行编译，编译耗时大约3至5分钟，待编译完成后，即可在kernel/arch/mips/boot目录下得到用于TF卡启动的uImage文件。

01. make uImage -j4

[复制代码](#)

3. 打包ext4格式的Build Root文件系统。

由于是使用TF卡启动，因此文件系统需要制作成ext4格式，具体步骤如下：

- (1) 进入到BSP根目录，选择RatCharm_norflash-user对BSP进行一次完整编译。
- (2) 此时可在out/product/RatCharm路径下找到制作rootfs所需的system目录。
- (3) 进入以上路径（不要进入system目录）后依次执行以下命令：


01. dd if=/dev/zero of=system.ext4 bs=1M count=512
02. mkfs.ext4 system.ext4
03. sudo mount -o loop system.ext4 /mnt
04. sudo cp -a -fr system/* /mnt
05. sync
06. sudo umount /mnt
07. sudo chmod 777 system.ext4

[复制代码](#)

若顺利完成则可得名为system.ext4的目标文件，使用cloner烧录以上步骤得到的三个文件即可实现从TF卡启动。


• 附上RatCharm平台BSP包及已编译好的用于TF卡启动的目标文件：

RatCharm平台BSP包：

 X1000V6.0-Moded-20171217.part1.rar (450 MB, 下载次数：8)

 X1000V6.0-Moded-20171217.part2.rar (430.67 MB, 下载次数：7)

用于TF卡启动的目标文件：

 u-boot-with-spl-mbr-gpt.zip (95.18 KB, 下载次数：2)

 ulmage.zip (2.4 MB, 下载次数：2)

 system.zip (9.35 MB, 下载次数：2)

接下来将尝试引导Debian8的Rootfs，预计会有以下几个步骤：

1. 打包Debian8的最小rootfs并制作成ext4格式镜像。
2. 用Debian8的rootfs镜像替代此前的Buildroot。
3. 尝试引导Debian8并解决出现的各种问题。

- 以下是移植步骤：

1. 修改并使用mkdebianrfs.sh脚本打包Debian8的最小rootfs，该脚本来源于IMG公司的C120项目。因为所用硬件平台使用的调试串口为UART2，与C120平台使用的调试串口不同，因此需要先进行相应修改，以下是脚本中的原始代码：

```
01. # Configure a serial console.
02. echo "T0:23:respawn:/sbin/getty -L ttyS0 115200 vt100" >> \
03.     "${target_dir}/etc/inittab"
```

[复制代码](#)

修改后的代码如下：

```
01. # Configure a serial console.
02. echo "T0:23:respawn:/sbin/getty -L ttyS2 115200 vt100" >> \
03.     "${target_dir}/etc/inittab"
```

[复制代码](#)

因为原脚本默认使用的是国外的源，为了避免因为网速问题导致打包时间过长，因此需要将其更改为国内的源，以下是脚本中的原始代码：

```
01. debian_packages="locales"
02. debian_mirror="http://ftp.uk.debian.org/debian/"
03. debian_path="/usr/local/bin:/usr/bin:/bin:/usr/local/sbin:/usr/sbin:/sbin"
```

[复制代码](#)

修改后的代码如下，可根据自己的实际情况选择其他的国内源：

```
01. debian_packages="locales"
02. debian_mirror="http://mirrors.ustc.edu.cn/debian/"
03. debian_path="/usr/local/bin:/usr/bin:/bin:/usr/local/sbin:/usr/sbin:/sbin"
```

[复制代码](#)

2. 安装进行打包工作所必须的软件包。

因mkdebianrfs.sh调用的是debootstrap，因此需要安装debootstrap。

同时由于X1000E是MIPS指令集的处理器的，为了在X86平台上进行打包，还需安装qemu模拟器。

依次执行以下命令，完成对所需软件包的安装：

```
01. sudo apt-get update
02. sudo apt-get install binfmt-support qemu qemu-user-static debootstrap bzip2 gedit
```

[复制代码](#)

3. 建立所需目录，并完成Debian8最小rootfs的打包及相关设置。

依次执行以下命令创建所需目录并进行打包操作：

```
01. mkdir debian8-rootfs
02. sudo ./mkdebianrfs.sh mipsel jessie debian8-rootfs
```

[复制代码](#)

此时将会开始进行Debian8最小rootfs的打包，根据网络情况耗时约几分钟至十几分钟。待打包完毕后会提示为root账户设置密码，如下图所示，共需输入两次。

文件(F) 编辑(E) 查看(V) 搜索(S) 终端(T) 帮助(H)

```
I: Configuring libp11-kit0:mipsel...
I: Configuring libgnutls-deb0-28:mipsel...
I: Configuring wget...
I: Configuring apt...
I: Configuring man-db...
I: Configuring libnetfilter-acct1:mipsel...
I: Configuring rsyslog...
I: Configuring nfacct...
I: Configuring apt-utils...
I: Configuring libirs-export91...
I: Configuring libgnutls-openssl27:mipsel...
I: Configuring iputils-ping...
I: Configuring isc-dhcp-common...
I: Configuring isc-dhcp-client...
I: Configuring tasksel...
I: Configuring tasksel-data...
I: Configuring libc-bin...
I: Configuring systemd...
I: Base system installed successfully.
```

Configuring target system...

Please enter a password for the root user:
Enter new UNIX password: █

一牛网
16RD.com

设置完毕root密码后会提示选择要安装的语言包，选中下图所示的四项简体中文语言包：

jdi@jdi-virtual-machine: /Ingenic

文件(F) 编辑(E) 查看(V) 搜索(S) 终端(T) 帮助(H)

Package configuration

Configuring locales

Locales are a framework to switch between multiple languages and allow users to use their language, country, characters, collation order, etc.

Please choose which locales to generate. UTF-8 locales should be chosen by default, particularly for new installations. Other character sets may be useful for backwards compatibility with older systems and software.

Locales to be generated:

```
[*] zh_CN GB2312
[*] zh_CN.GB18030 GB18030
[*] zh_CN.GBK GBK
[*] zh_CN.UTF-8 UTF-8
[ ] zh_HK BIG5-HKSCS
```

<Ok>

<Cancel>

一牛网
16RD.com

随后会提示选择系统默认使用的语言包，选中zh_CN.UTF-8，如下图所示。此项步骤耗时较长，根据机器配置不同约需要3至5分钟。

Configuring locales

Many packages in Debian use locales to display text in the correct language for the user. You can choose a default locale for the system from the generated locales.

This will select the default language for the entire system. If this system is a multi-user system where not all users are able to speak the default language, they will experience difficulties.

Default locale for the system environment:

None
C.UTF-8
zh_CN
zh_CN.GB18030
zh_CN.GBK
zh_CN.UTF-8

<Ok>

<Cancel>

待语言配置完成后根据提示设置地理位置，如下图所示选中Asia:

Configuring tzdata

Please select the geographic area in which you live. Subsequent configuration questions will narrow this down by presenting a list of cities, representing the time zones in which they are located.

Geographic area:

Asia
Atlantic Ocean
Europe
Indian Ocean
Pacific Ocean
System V timezones
US
None of the above

<Ok>

<Cancel>

根据提示设置时区，国内可选择上海或重庆，因本人在成都，所以就近选择重庆。

Configuring tzdata

Please select the city or region corresponding to your time zone.

Time zone:

Bahrain
Baku
Bangkok
Barnaul
Beirut
Bishkek
Brunei
Chita
Choibalsan
Chongqing

<Ok>

<Cancel>

完成所有操作后，根据提示输入exit退出打包脚本，如下所示，上图为打包完成时，下图为退出脚本后。

I: Configuring systemd...

I: Base system installed successfully.

Configuring target system...

Please enter a password for the root user:

Enter new UNIX password:

Retype new UNIX password:

passwd: password updated successfully

Generating locales (this might take a while)...

zh_CN.GB2312... done

zh_CN.GB18030... done

zh_CN.GBK... done

zh_CN.UTF-8... done

Generation complete.

Current default time zone: 'Asia/Chongqing'

Local time is now: Thu Jun 28 21:37:12 CST 2018.

Universal Time is now: Thu Jun 28 13:37:12 UTC 2018.

Entering target system for additional configuration. Type 'exit' when done.

root@jdi-virtual-machine:/# exit

```
Please enter a password for the root user:
Enter new UNIX password:
Retype new UNIX password:
passwd: password updated successfully
Generating locales (this might take a while)...
 zh_CN.GB2312... done
 zh_CN.GB18030... done
 zh_CN.GBK... done
 zh_CN.UTF-8... done
Generation complete.

Current default time zone: 'Asia/Chongqing'
Local time is now:      Thu Jun 28 21:37:12 CST 2018.
Universal Time is now:  Thu Jun 28 13:37:12 UTC 2018.

Entering target system for additional configuration. Type 'exit' when done.

root@jdi-virtual-machine:/# exit
exit

Done!
jdi@jdi-virtual-machine:/Ingenic$
```

接下来就需要将打包完毕后的Debian8 Rootfs制作成ext4格式的镜像。
和制作Buildroot的ext4镜像类似，依次执行以下命令：

```
01. dd if=/dev/zero of=debian8-rootfs.ext4 bs=1M count=512
02. mkfs.ext4 debian8-rootfs.ext4
03. sudo mount -o loop debian8-rootfs.ext4 /mnt
04. sudo cp -a -fr debian8-rootfs/* /mnt
05. sync
06. sudo umount /mnt
07. sudo chmod 777 debian8-rootfs.ext4
```

[复制代码](#)

若顺利完成则可得到名为debian8-rootfs.ext4的目标文件，
用该文件替换掉此前的system.ext4，使用cloner烧录到目标板。

接下来将尝试引导Debian8并解决相关问题

- 第一次尝试：

很遗憾以失败告终，可以看到系统最终卡在挂载rootfs的步骤。

```
COM5 - PuTTY
[ 0.578228] usbcore: registered new interface driver uvcvideo
[ 0.584243] USB Video Class driver (1.1.1)
[ 0.593437] jzmmc_v1.2 jzmmc_v1.2.0: card inserted, state=0
[ 0.623398] jzmmc_v1.2 jzmmc_v1.2.0: register success!
[ 0.664452] jzmmc_v1.2 jzmmc_v1.2.1: register success!
[ 0.670123] usbcore: registered new interface driver usbhid
[ 0.675971] usbhid: USB HID core driver
[ 0.680976] jz-asoc-aic jz-asoc-aic: Aic core probe success
[ 0.687062] jz-asoc-aic-i2s jz-asoc-aic-i2s: i2s platform probe success
[ 0.694112] icdc-d3 icdc-d3: codec icdc-d3 platfrom probe success
[ 0.700535] jz-asoc-pcm jz-asoc-pcm: pcm platform probe success
[ 0.707365] icdc-d3 icdc-d3: codec icdc-d3 probe enter
[ 0.712714] write extend : sreg: 3 [0 - 4], creg: 10 sdata: 180
[ 0.720088] gpio speaker enable 0
[ 0.723899] dma dma0chan4: Channel 4 have been requested.(phy id 3,type 0x06 desc a3eb0000)
[ 0.732777] dma dma0chan5: Channel 5 have been requested.(phy id 2,type 0x06 desc a3eb1000)
[ 0.741719] ingenic-alsa ingenic-alsa.0: icdc-d3-hifi <-> jz-asoc-aic-i2s mapping ok
[ 0.750366] dma dma0chan0: Channel 0 have been requested.(phy id 7,type 0x20 desc a3eb2000)
[ 0.759297] dma dma0chan1: Channel 1 have been requested.(phy id 6,type 0x20 desc a3eb3000)
[ 0.768238] ingenic-alsa ingenic-alsa.0: pcm dump dai <-> jz-asoc-pcm mapping ok
[ 0.776517] dma dma0chan7: Channel 7 have been requested.(phy id 0,type 0x04 desc a3eb4000)
[ 0.785419] ingenic-alsa ingenic-alsa.0: dmic dump dai <-> jz-asoc-dmic mapping ok
[ 0.794706] input: RatCharm Headset Jack as /devices/platform/ingenic-alsa.0/sound/card0/input0
[ 0.804343] Netfilter messages via NETLINK v0.30.
[ 0.809235] nfnl_acct: registering with nfnetlink.
[ 0.814429] ip_set: protocol 6
[ 0.817759] ip_tables: (C) 2000-2006 Netfilter Core Team
[ 0.823472] TCP: cubic registered
[ 0.826910] NET: Registered protocol family 17
[ 0.831657] Bridge firewalling registered
[ 0.836861] file system registered
[ 0.841615] using random self ethernet address
[ 0.846350] using random host ethernet address
[ 0.851287] android_usb gadget: Mass Storage Function, version: 2009/09/11
[ 0.858485] android_usb gadget: Number of LUNs=1
[ 0.863265] lun0: LUN: removable file: (no medium)
[ 0.868639] android_usb gadget: android_usb ready
[ 0.873989] input: gpio-keys as /devices/platform/gpio-keys/input/input1
[ 0.881259] jz-rtc jz-rtc.0: setting system clock to 2018-06-28 17:25:36 UTC (1530206736)
[ 0.889893] ALSA device list:
[ 0.892964] #0: RatCharm
[ 0.896173] Waiting 1sec before mounting root device...
[ 1.633922] mmc0: new high speed SDHC card at address 1234
[ 1.640062] mmcblk0: mmc0:1234 SA09G 7.41 GiB
[ 1.647844] Alternate GPT is invalid, using primary GPT.
[ 1.653482] mmcblk0: p1 p2 p3 p4 p5 p6 p7 p8
[ 6.967369] EXT4-fs (mmcblk0p7): recovery complete
[ 6.974704] EXT4-fs (mmcblk0p7): mounted filesystem with ordered data mode. Opts: (null)
[ 6.983156] VFS: Mounted root (ext4 filesystem) on device 179:7.
[ 6.990477] Freeing unused kernel memory: 220K (80519000 - 80550000)
```

经分析，判断问题是因为内核中没有开启devtmpfs造成的，因此需要对kernel进行如下配置：

```
01. | Symbol: DEVTMPFS_MOUNT [=y] |
02. | Type : boolean |
03. | Prompt: Automount devtmpfs at /dev, after the kernel mounted the rootfs |
04. | Location: |
05. | -> Device Drivers |
06. | -> Generic Driver Options |
07. | -> Maintain a devtmpfs filesystem to mount at /dev (DEVTMPFS [=y]) |
08. | Defined at drivers/base/Kconfig:44 |
09. | Depends on: DEVTMPFS [=y] |
```

复制代码

修改前后的截图对比，上图为修改前下图为修改后：

jdi@jdi-virtual-machine: /Ingenic/X1000V6.0-Moded-20171217/kernel

文件(F) 编辑(E) 查看(V) 搜索(S) 终端(T) 帮助(H)

.config - Linux/mips 3.10.14 Kernel Configuration

> Device Drivers > Generic Driver Options

Generic Driver Options

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 < >

```
( ) path to uevent helper
[*] Maintain a devtmpfs filesystem to mount at /dev
[*] Select only drivers that don't need compile-time external fir
[*] Prevent firmware from being built
<*> Userspace firmware loading support
[*] Include in-kernel firmware blobs in kernel binary
() External firmware blobs to build into the kernel binary
[*] Fallback user-helper invocation for firmware loading
[ ] Driver Core verbose debug messages
[ ] Managed device resources verbose debug messages
```

<Select> < Exit > < Help > < Save > < Load >

jdi@jdi-virtual-machine: /Ingenic/X1000V6.0-Moded-20171217/kernel

文件(F) 编辑(E) 查看(V) 搜索(S) 终端(T) 帮助(H)

.config - Linux/mips 3.10.14 Kernel Configuration

> Device Drivers > Generic Driver Options

Generic Driver Options

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 < >

```
( ) path to uevent helper
[*] Maintain a devtmpfs filesystem to mount at /dev
[*] Automount devtmpfs at /dev, after the kernel mounted the ro
[*] Select only drivers that don't need compile-time external fir
[*] Prevent firmware from being built
<*> Userspace firmware loading support
[*] Include in-kernel firmware blobs in kernel binary
() External firmware blobs to build into the kernel binary
[*] Fallback user-helper invocation for firmware loading
[ ] Driver Core verbose debug messages
```

<Select> < Exit > < Help > < Save > < Load >

完成修改后重新编译kernel并烧写进目标板，准备进行第二次尝试。

- 第二次尝试：

令人兴奋的是此时已经能够进入Debian8, 但同时也能看到在打印信息中提示了很多错误信息。

```
COM5 - PuTTY
[ 4.072036] VFS: Mounted root (ext4 filesystem) on device 179:7.
[ 4.086019] devtmpfs: mounted
[ 4.090233] Freeing unused kernel memory: 220K (80519000 - 80550000)
[ 4.455118] systemd[1]: systemd 215 running in system mode. (+PAM +AUDIT +SELINUX +IMA +SYSVINIT +LIBCRYPTSETUP +G
CRYPT +ACL +XZ -SECCOMP -APPARMOR)
[ 4.469520] systemd[1]: Detected architecture 'mips'.

Welcome to Debian GNU/Linux 8 (jessie)!

[ 4.518184] systemd[1]: Failed to insert module 'autofs4'
[ 4.524050] systemd[1]: Failed to insert module 'ipv6'
[ 4.532983] systemd[1]: Set hostname to <debian>.
[ 5.098539] systemd[1]: Cannot add dependency job for unit dbus.socket, ignoring: Unit dbus.socket failed to load:
No such file or directory.
[ 5.112041] systemd[1]: Cannot add dependency job for unit display-manager.service, ignoring: Unit display-manager
.service failed to load: No such file or directory.
[ 5.130439] systemd[1]: Expecting device dev-ttyS2.device...
[ 5.136455] systemd[1]: inotify_init1() failed: Function not implemented
Expecting device dev-ttyS2.device...
[ 5.157339] systemd[1]: Starting Forward Password Requests to Wall Directory Watch.
[ 5.165679] systemd[1]: systemd-ask-password-wall.path failed to enter waiting state: Function not implemented
[ 5.176287] systemd[1]: Failed to start Forward Password Requests to Wall Directory Watch.
[ 5.185036] systemd[1]: Unit systemd-ask-password-wall.path entered failed state.
[ 5.192989] systemd[1]: Starting Remote File Systems (Pre).
[ 5.198966] systemd[1]: inotify_init1() failed: Function not implemented
[ OK ] Reached target Remote File Systems (Pre).
[ 5.227261] systemd[1]: Reached target Remote File Systems (Pre).
[ 5.233789] systemd[1]: Starting Encrypted Volumes.
[ 5.239081] systemd[1]: inotify_init1() failed: Function not implemented
[ OK ] Reached target Encrypted Volumes.
[ 5.267272] systemd[1]: Reached target Encrypted Volumes.
[ 5.273108] systemd[1]: Starting Dispatch Password Requests to Console Directory Watch.
[ 5.281714] systemd[1]: systemd-ask-password-console.path failed to enter waiting state: Function not implemented
[ 5.292587] systemd[1]: Failed to start Dispatch Password Requests to Console Directory Watch.
[ 5.301673] systemd[1]: Unit systemd-ask-password-console.path entered failed state.
[ 5.308957] systemd[1]: Starting Paths.
[ 5.313958] systemd[1]: inotify_init1() failed: Function not implemented
[ OK ] Reached target Paths.
[ 5.337263] systemd[1]: Reached target Paths.
[ 5.342086] systemd[1]: Set up automount Arbitrary Executable File Formats File System Automount Point.
[ 5.352035] systemd[1]: Starting Swap.
[ 5.356053] systemd[1]: inotify_init1() failed: Function not implemented
[ OK ] Reached target Swap.
[ 5.377281] systemd[1]: Reached target Swap.
[ 5.381867] systemd[1]: Starting Root Slice.
[ 5.386564] systemd[1]: inotify_init1() failed: Function not implemented
[ OK ] Created slice Root Slice.
[ 5.407285] systemd[1]: Created slice Root Slice.
[ 5.412324] systemd[1]: Starting User and Session Slice.
[ 5.418373] systemd[1]: inotify_init1() failed: Function not implemented
[ OK ] Created slice User and Session Slice.
```

根据打印记录, 梳理一下存在的问题:

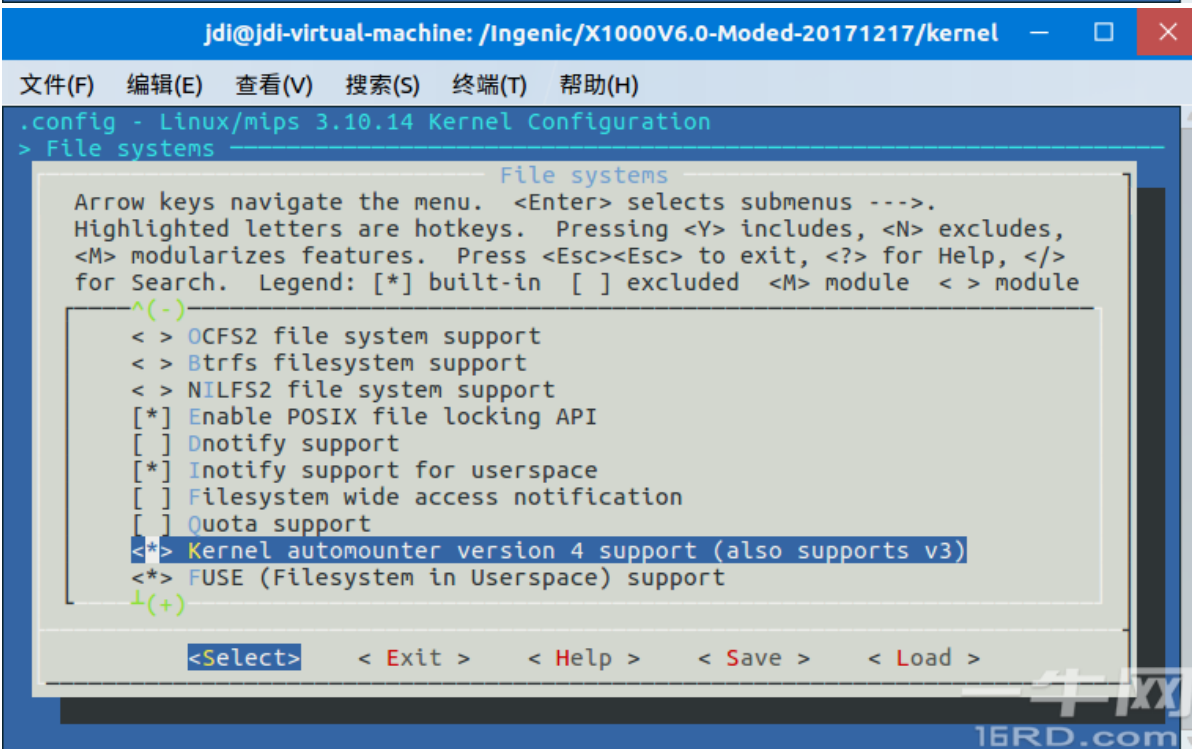
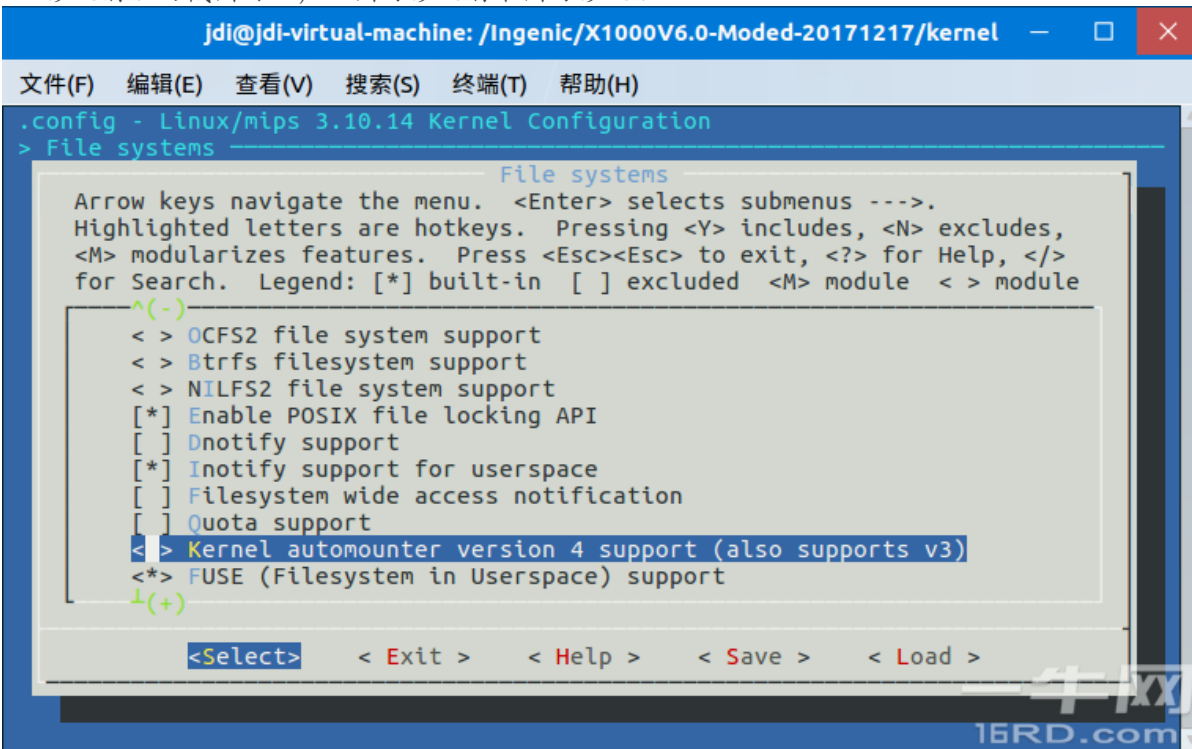
1. 与autofs4相关的问题;
2. 与ipv6相关的问题;
3. 与dbus相关的问题;
4. 与inotify相关的问题;
5. 与getty相关的问题(包括90秒超时和无法登陆两个问题)。

经过分析判定, 与autofs4、ipv6、inotify相关的问题是因为kernel里没有开启相应的选项, 需要对kernel进行对应配置。与dbus及getty相关的问题与rootfs有关, 需要通过重新打包rootfs才能修复。本着先易后难的原则, 先对kernel进行如下配置以便使能autofs4:

```
01. | Symbol: AUTOFS4_FS [=y] |
02. | Type : tristate |
03. | Prompt: Kernel automounter version 4 support (also supports V3) |
04. | Location: |
05. | -> File systems |
06. | Defined at fs/autofs4/Kconfig:1 |
```

复制代码

修改前后的截图对比，上图为修改前下图为修改后：



然后使能ipv6，需如下配置kernel：

```
01. | Symbol: IPV6 [=y] |
02. | Type : tristate |
03. | Prompt: The IPv6 protocol |
04. | Location: |
05. | -> Networking support (NET [=y]) |
06. | -> Networking options |
07. | -> TCP/IP networking (INET [=y]) |
08. | Defined at net/ipv6/Kconfig:6 |
09. | Depends on: NET [=y] && INET [=y] |
```

[复制代码](#)

修改前后的截图对比，上图为修改前下图为修改后：

```
jdi@jdi-virtual-machine: /Ingenic/X1000V6.0-Moded-20171217/kernel
文件(F) 编辑(E) 查看(V) 搜索(S) 终端(T) 帮助(H)
.config - Linux/mips 3.10.14 Kernel Configuration
> Networking support > Networking options
Networking options
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 < > module
^(-)
< > IP: IPsec transport mode
< > IP: IPsec tunnel mode
< > IP: IPsec BEET mode
< > Large Receive Offload (ipv4/tcp)
<*> INET: socket monitoring interface
< > UDP: socket monitoring interface
[ ] TCP: advanced congestion control --->
[ ] TCP: MD5 Signature Option support (RFC2385)
<*> The IPv6 protocol --->
[*] Only allow certain groups to create sockets
I(+)
<Select> < Exit > < Help > < Save > < Load >
```

```
jdi@jdi-virtual-machine: /Ingenic/X1000V6.0-Moded-20171217/kernel
文件(F) 编辑(E) 查看(V) 搜索(S) 终端(T) 帮助(H)
.config - Linux/mips 3.10.14 Kernel Configuration
> Networking support > Networking options
Networking options
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 < > module
^(-)
< > IP: IPsec transport mode
< > IP: IPsec tunnel mode
< > IP: IPsec BEET mode
< > Large Receive Offload (ipv4/tcp)
<*> INET: socket monitoring interface
< > UDP: socket monitoring interface
[ ] TCP: advanced congestion control --->
[ ] TCP: MD5 Signature Option support (RFC2385)
<*> The IPv6 protocol --->
[*] Only allow certain groups to create sockets
I(+)
<Select> < Exit > < Help > < Save > < Load >
```

最后如下配置，以便解决inotify相关问题：

```
01. | Symbol: INOTIFY_USER [=y] |
02. | Type : boolean |
03. | Prompt: Inotify support for userspace |
04. | Location: |
05. | -> File systems |
06. | Defined at fs/notify/inotify/Kconfig:1 |
07. | Selects: ANON_INODES [=y] && FSNOTIFY [=y] |
```

复制代码

修改前后的截图对比，上图为修改前下图为修改后：

```
jdi@jdi-virtual-machine: /Ingenic/X1000V6.0-Moded-20171217/kernel
文件(F) 编辑(E) 查看(V) 搜索(S) 终端(T) 帮助(H)
.config - Linux/mips 3.10.14 Kernel Configuration
> File systems
File systems
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 < > module
^(-)
< > JFS filesystem support
< > XFS filesystem support
< > GFS2 file system support
< > OCFS2 file system support
< > Btrfs filesystem support
< > NILFS2 file system support
[*] Enable POSIX file locking API
[ ] Dnotify support
[ ] Inotify support for userspace
[ ] Filesystem wide access notification
I(+)
<Select> < Exit > < Help > < Save > < Load >
```

```
jdi@jdi-virtual-machine: /Ingenic/X1000V6.0-Moded-20171217/kernel
文件(F) 编辑(E) 查看(V) 搜索(S) 终端(T) 帮助(H)
.config - Linux/mips 3.10.14 Kernel Configuration
> File systems
File systems
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 < > module
^(-)
< > JFS filesystem support
< > XFS filesystem support
< > GFS2 file system support
< > OCFS2 file system support
< > Btrfs filesystem support
< > NILFS2 file system support
[*] Enable POSIX file locking API
[ ] Dnotify support
[*] Inotify support for userspace
[ ] Filesystem wide access notification
I(+)
<Select> < Exit > < Help > < Save > < Load >
```

重新编译kernel并将其烧写到目标板，然后准备进行第三次引导尝试。

• 第三次尝试：

从打印log可以看到已经没有报与autofs4、ipv6及inotify相关的错误了，但又出现了一个新问题：除了此前的getty存在90秒超时的问题，现在有线网络ifup也出现了90秒的超时。

```
COM5 - PuTTY
[ OK ] Started Load/Save Random Seed.
[ OK ] Started Create Volatile Files and Directories.
[ OK ] Started Apply Kernel Variables.
[ OK ] Started Trigger Flushing of Journal to Persistent Storage.
[ 5.699079] systemd-journald[98]: Received request to flush runtime journal from PID 1
[ OK ] Created slice system-ifup.slice.
Expecting device sys-subsystem-net-devices-eth0.device...
Starting Update UTMP about System Boot/Shutdown...
Starting LSB: Raise network interfaces....
Starting Copy rules generated while the root was ro...
[ OK ] Started Copy rules generated while the root was ro.
[ OK ] Started Update UTMP about System Boot/Shutdown.
[ 6.410423] Bus Mode Reg after reset: 0x00020101, cnt=0
[ 9.529522] libphy: 0:01 - Link is Up - 100/Full
[ OK ] Started LSB: Raise network interfaces..
[ OK ] Reached target System Initialization.
[ OK ] Reached target Timers.
[ OK ] Reached target Basic System.
Starting Regular background program processing daemon...
[ OK ] Started Regular background program processing daemon.
Starting getty on tty2-tty6 if dbus and logind are not available...
Starting System Logging Service...
[ OK ] Started getty on tty2-tty6 if dbus and logind are not available.
[ OK ] Started System Logging Service.
[ TIME ] Timed out waiting for device dev-ttyS2.device.
[DEPEND] Dependency failed for Serial Getty on ttyS2.
[ TIME ] Timed out waiting for device sys-subsystem-net-devices-eth0.device.
[DEPEND] Dependency failed for ifup for eth0.
[ OK ] Reached target Network.
Starting /etc/rc.local Compatibility...
Starting Permit User Sessions...
[ OK ] Started /etc/rc.local Compatibility.
[ OK ] Started Permit User Sessions.
Starting Getty on tty6...
[ OK ] Started Getty on tty6.
Starting Getty on tty5...
[ OK ] Started Getty on tty5.
Starting Getty on tty4...
[ OK ] Started Getty on tty4.
Starting Getty on tty3...
[ OK ] Started Getty on tty3.
Starting Getty on tty2...
[ OK ] Started Getty on tty2.
Starting Getty on tty1...
[ OK ] Started Getty on tty1.
[ OK ] Reached target Login Prompts.
[ OK ] Reached target Multi-User System.
[ OK ] Reached target Graphical Interface.
Starting Update UTMP about System Runlevel Changes...
[ OK ] Started Update UTMP about System Runlevel Changes.
```

新出现的ifup超时问题同此前的getty超时非常相似，分析后判断问题也出来rootfs中。

接下来将重新对Debian8的rootfs进行打包以便修复相关问题。操作步骤如下：

1. 重复此前的打包步骤，但在打包完毕后暂不要使用exit退出mkdebianrfs.sh脚本，如下图所示：

```
jdi@jdi-virtual-machine: /Ingenic
文件(F) 编辑(E) 查看(V) 搜索(S) 终端(T) 帮助(H)
I: Configuring systemd...
I: Base system installed successfully.
Configuring target system...
Please enter a password for the root user:
Enter new UNIX password:
Retype new UNIX password:
passwd: password updated successfully
Generating locales (this might take a while)...
 zh_CN.GB2312... done
 zh_CN.GB18030... done
 zh_CN.GBK... done
 zh_CN.UTF-8... done
Generation complete.
Current default time zone: 'Asia/Chongqing'
Local time is now: Thu Jun 28 21:37:12 CST 2018.
Universal Time is now: Thu Jun 28 13:37:12 UTC 2018.
Entering target system for additional configuration. Type 'exit' when done.
root@jdi-virtual-machine:/#
```

2. 新开启一个终端(以下称终端2)并进入到mkdebianrfs.sh脚本所在目录，执行以下命令挂载相应目录。

01. sudo mount -t proc proc debian8-rootfs/proc
02. sudo mount --bind /dev/pts/ debian8-rootfs/dev/pts/

[复制代码](#)

执行完以上两条命令后，若无错误提示则挂载成功，否则可根据提示排除问题后重新挂载。

3. 切换回之前打包的终端(以下称终端1)，依次输入以下命令安装所需软件包到打包好的rootfs中。

```
01. apt-get update
02. apt-get upgrade
03. apt-get install dbus vim
04. apt-get clean
```

[复制代码](#)

4. 创建getty所需的ttyS2配置文件解决getty超时和无法从串口登录的问题，在终端1中执行以下命令：

```
01. cp /lib/systemd/system/serial-getty\@.service /etc/systemd/system/getty.target.wants/serial-getty\@ttyS2.serv
ice
```

[复制代码](#)

执行完以上命令后，切换至终端2执行以下命令：

```
01. sudo gedit debian8-rootfs/etc/systemd/system/getty.target.wants/serial-getty@ttyS2.service
```

[复制代码](#)

在弹出的窗口中修改相关配置，以下是原始代码：

```
01. [Unit]
02. Description=Serial Getty on %I
03. Documentation=man:agetty(8) man:systemd-getty-generator(8)
04. Documentation=http://0pointer.de/blog/projects/serial-console.html
05. BindsTo=dev-%i.device
06. After=dev-%i.device systemd-user-sessions.service plymouth-quit-wait.service
07. After=rc-local.service
```

[复制代码](#)

修改后的代码如下：

```
01. [Unit]
02. Description=Serial Getty on %I
03. Documentation=man:agetty(8) man:systemd-getty-generator(8)
04. Documentation=http://0pointer.de/blog/projects/serial-console.html
05. After=systemd-user-sessions.service plymouth-quit-wait.service
06. After=rc-local.service
```

[复制代码](#)

完成修改并保存后，即可关闭窗口，至此即已完成对getty所需的ttyS2配置文件的创建和编辑。

5. 创建ifup所需的eth0配置文件解决ifup超时的问题，在终端1中依次执行以下命令：

```
01. mkdir /etc/systemd/system/network.target.wants
02. cp /lib/systemd/system/ifup\@.service /etc/systemd/system/network.target.wants/ifup\@eth0.service
```

[复制代码](#)

执行完以上命令后，切换至终端2执行以下命令：

```
01. sudo gedit debian8-rootfs/etc/systemd/system/network.target.wants/ifup@eth0.service
```

[复制代码](#)

在弹出的窗口中修改相关配置，以下是原始代码：

```
01. [Unit]
```

```
02. Description=ifup for %I
03. After=local-fs.target network-pre.target networking.service systemd-sysctl.service
04. Before=network.target
05. BindsTo=sys-subsystem-net-devices-%i.device
06. After=sys-subsystem-net-devices-%i.device
07. ConditionPathIsDirectory=/run/network
08. DefaultDependencies=no
```

[复制代码](#)

修改后的代码如下：

```
01. [Unit]
02. Description=ifup for %I
03. After=local-fs.target network-pre.target networking.service systemd-sysctl.service
04. Before=network.target
05. ConditionPathIsDirectory=/run/network
06. DefaultDependencies=no
```

[复制代码](#)

完成修改并保存后，即可关闭窗口，至此即已完成对ifup所需的eth0配置文件的创建和编辑。

6. 在终端1中使用exit退出脚本然后切换至终端2，依次输入以下命令卸载掉之前挂载的目录。

```
01. sudo umount debian8-rootfs/proc
02. sudo umount debian8-rootfs/dev/pts/
```

[复制代码](#)

若在卸载debian8-rootfs/proc时出现类似以下的错误：

```
01. umount: /Ingenic/debian8-rootfs/proc: target is busy
02.      (In some cases useful info about processes that
03.      use the device is found by lsof(8) or fuser(1).)
```

[复制代码](#)

则首先确保没有打开与要卸载目录有关的文件夹或文件，然后使用以下命令强制卸载：

```
01. sudo umount -lf debian8-rootfs/proc
```

[复制代码](#)

完成卸载后关掉终端2，按照此前制作ext4格式镜像的步骤重新制作镜像，然后将新镜像烧录到目标板，进行新一轮的引导尝试。

• 第四次尝试：

从打印log可以看到前面的问题都已经解决，并且已经出现了Debian8的登录提示，并且使用root账户和此前打包时设置的密码，可以成功登录到系统。

```
COM5 - PuTTY
Starting Update UTMP about System Boot/Shutdown...
Starting LSB: Raise network interfaces...
Starting Copy rules generated while the root was ro...
[ OK ] Started Copy rules generated while the root was ro.
[ OK ] Started Update UTMP about System Boot/Shutdown.
[ 6.374511] Bus Mode Reg after reset: 0x00020101, cnt=0
[ 8.529501] libphy: 0:01 - Link is Up - 100/Full
[ OK ] Started LSB: Raise network interfaces..
Starting ifup for eth0...
[ OK ] Started ifup for eth0.
[ OK ] Reached target Network.
[ OK ] Reached target System Initialization.
[ OK ] Listening on D-Bus System Message Bus Socket.
[ OK ] Reached target Sockets.
[ OK ] Reached target Timers.
[ OK ] Reached target Basic System.
Starting Regular background program processing daemon...
[ OK ] Started Regular background program processing daemon.
Starting /etc/rc.local Compatibility...
Starting Login Service...
Starting D-Bus System Message Bus...
[ OK ] Started D-Bus System Message Bus.
Starting System Logging Service...
Starting Permit User Sessions...
[ OK ] Started /etc/rc.local Compatibility.
[ OK ] Started Permit User Sessions.
[ OK ] Started System Logging Service.
[ OK ] Started Login Service.
Starting Getty on tty1...
[ OK ] Started Getty on tty1.
Starting Serial Getty on ttyS2...
[ OK ] Started Serial Getty on ttyS2.
[ OK ] Reached target Login Prompts.
[ OK ] Reached target Multi-User System.
[ OK ] Reached target Graphical Interface.
Starting Update UTMP about System Runlevel Changes...
[ OK ] Started Update UTMP about System Runlevel Changes.

Debian GNU/Linux 8 debian ttyS2

debian login: root
密码:
Linux debian 3.10.14 #3 PREEMPT Fri Jun 29 23:31:36 CST 2018 mips

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
root@debian:~#
```

登录系统后可以使用apt-get安装软件，此处安装了screenfetch和htop。

```
COM5 - PuTTY
[ OK ] Started Regular background program processing daemon.
Starting /etc/rc.local Compatibility...
Starting Login Service...
Starting D-Bus System Message Bus...
[ OK ] Started D-Bus System Message Bus.
Starting System Logging Service...
Starting Permit User Sessions...
[ OK ] Started /etc/rc.local Compatibility.
[ OK ] Started Permit User Sessions.
[ OK ] Started System Logging Service.
[ OK ] Started Login Service.
Starting Getty on tty1...
[ OK ] Started Getty on tty1.
Starting Serial Getty on ttyS2...
[ OK ] Started Serial Getty on ttyS2.
[ OK ] Reached target Login Prompts.
[ OK ] Reached target Multi-User System.
[ OK ] Reached target Graphical Interface.
Starting Update UTMP about System Runlevel Changes...
[ OK ] Started Update UTMP about System Runlevel Changes.

Debian GNU/Linux 8 debian ttyS2

debian login: root
密码:
Linux debian 3.10.14 #3 PREEMPT Fri Jun 29 23:31:36 CST 2018 mips

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
root@debian:~# apt-get install screenfetch htop
正在读取软件包列表... 完成
正在分析软件包的依赖关系树
正在读取状态信息... 完成
将会安装下列额外的软件包:
  glib1lib1 libfreetype6 libgif4 libid3tag0 libimlib2 libjbig0 libjpeg62-turbo
  libpng12-0 libtiff5 libx11-6 libx11-data libxau6 libxcb1 libxdmcp6 libxext6
  scrot
建议安装的软件包:
  strace ltrace
下列【新】软件包将被安装:
  glib1lib1 htop libfreetype6 libgif4 libid3tag0 libimlib2 libjbig0
  libjpeg62-turbo libpng12-0 libtiff5 libx11-6 libx11-data libxau6 libxcb1
  libxdmcp6 libxext6 screenfetch scrot
升级了 0 个软件包, 新安装了 18 个软件包, 要卸载 0 个软件包, 有 0 个软件包未被升级。
需要下载 2,256 kB 的软件包。
解压缩后会消耗掉 7,078 kB 的额外空间。
您希望继续执行吗? [Y/n]
```

安装完毕后运行screenfetch，可以识别出各种系统信息。
比较有意思的是处理器被识别成了IBM的PowerPC。

```
COM5 - PuTTY
正在解包 .../glib1.2.4-10_mipsel.deb ...
正在解包 glib1.2.4-10_mipsel (1.2.4-10) ...
正在选中未选择的软件包 htop。
正在解包 .../htop 1.0.3-1_mipsel.deb ...
正在解包 htop (1.0.3-1) ...
正在选中未选择的软件包 screenfetch。
正在解包 .../screenfetch 3.6.5-1_all.deb ...
正在解包 screenfetch (3.6.5-1) ...
正在选中未选择的软件包 scrot。
正在解包 .../scrot 0.8-13_mipsel.deb ...
正在解包 scrot (0.8-13) ...
正在处理用于 man-db (2.7.0.2-5) 的触发器 ...
正在设置 libpng12-0:mipsel (1.2.50-2+deb8u3) ...
正在设置 libfreetype6:mipsel (2.5.2-3+deb8u2) ...
正在设置 libgif4:mipsel (4.1.6-11+deb8u1) ...
正在设置 libid3tag0 (0.15.1b-11) ...
正在设置 libjpeg62-turbo:mipsel (1:1.3.1-12) ...
正在设置 libjpeg62:mipsel (2.1-3.1) ...
正在设置 libtiff5:mipsel (4.0.3-12.3+deb8u5) ...
正在设置 libxau6:mipsel (1:1.0.8-1) ...
正在设置 libxdmcp6:mipsel (1:1.1.1-1+b1) ...
正在设置 libxcb1:mipsel (1.10-3+b1) ...
正在设置 libx11-data (2:1.6.2-3+deb8u1) ...
正在设置 libx11-6:mipsel (2:1.6.2-3+deb8u1) ...
正在设置 libxext6:mipsel (2:1.3.3-1) ...
正在设置 libimlib2 (1.4.6-2+deb8u2) ...
正在设置 glib1.2.4-10_mipsel (1.2.4-10) ...
正在设置 htop (1.0.3-1) ...
正在设置 screenfetch (3.6.5-1) ...
正在设置 scrot (0.8-13) ...
正在处理用于 libc-bin (2.19-18+deb8u10) 的触发器 ...
root@debian:~# screenfetch
root@debian
,met$$$$$gg,          root@debian
,g$$$$$$$$$$$$$$$$P.  OS: Debian
,g$$$"      ""Y$.     Kernel: mips Linux 3.10.14
,$$P'      `$$$      Uptime: 37m
',$$P      ,ggs.     `$$b: Packages: 188
`d$$'      ,p"      $$$  Shell: bash 4.3.30
$$P      d$'      $$$   CPU: IBM PowerPC G3 Ingenic Xburst V4.15 FPU V0.0
$$:      $$-      ,d$$'  RAM: 19MB / 57MB
$$;      Y$b.     ,d$$P'
Y$$      `."Y$$$$P"
`$$b      "-_
`Y$$
`Y$$
`$$b.
`Y$$b.
`"Y$b.
`"mm
root@debian:~#
```

运行htop，可以查看CPU占用率、内存占用率、SWAP交换文件占用率、当前运行的进程等信息。

```
COM5 - PuTTY
CPU[#* 0.9%] Tasks: 12, 3 thr; 1 running
Mem[|||||||#####17/57MB] Load average: 1.04 1.03 0.96
Swp[ 0/0MB] Uptime: 00:42:56

PID USER PRI NI VIRT RES SHR S CPU% MEM% TIME+ Command
744 root 20 0 7724 1716 1348 R 0.5 2.9 0:00.17 htop
1 root 20 0 5924 1824 940 S 0.0 3.1 0:02.33 /sbin/init
82 root 20 0 10880 1188 788 S 0.0 2.0 0:00.30 /lib/systemd/syst
101 root 20 0 8616 1420 1184 S 0.0 2.4 0:00.40 /lib/systemd/syst
165 root 20 0 9192 4840 136 S 0.0 8.2 0:00.00 dhclient -v -pf /
189 root 20 0 7684 1080 876 S 0.0 1.8 0:00.03 /usr/sbin/cron -f
191 root 20 0 3736 980 768 S 0.0 1.7 0:00.06 /lib/systemd/syst
193 messagebu 20 0 6056 1172 852 S 0.0 2.0 0:00.09 /usr/bin/dbus-dae
200 root 20 0 31948 1552 816 S 0.0 2.6 0:00.01 /usr/sbin/rsyslog
201 root 20 0 31948 1552 816 S 0.0 2.6 0:00.00 /usr/sbin/rsyslog
202 root 20 0 31948 1552 816 S 0.0 2.6 0:00.02 /usr/sbin/rsyslog
196 root 20 0 31948 1552 816 S 0.0 2.6 0:00.09 /usr/sbin/rsyslog
207 root 20 0 4740 816 696 S 0.0 1.4 0:00.01 /sbin/agetty --no
208 root 20 0 9388 1388 968 S 0.0 2.3 0:00.25 /bin/login --
226 root 20 0 7788 1508 1172 S 0.0 2.5 0:00.06 -bash

F1 Help F2 Setup F3 Search F4 Filter F5 Tree F6 SortBy F7 Nice - F8 Nice + F9 Kill F10 Quit
```



由于X1000E仅有64M内存，设置一个较大的SWAP交换区就显得非常重要。但打包好的rootfs默认是没有SWAP交换区的，这一点从htop中SWAP为0也可以证实。因此接下来将尝试为系统添加SWAP分区，首先使用df命令查看当前可用的存储空间。

```
COM5 - PuTTY
root@debian:~# df
文件系统      1K-块    已用    可用    已用%  挂载点
/dev/root      499656  455208  7752    99%    /
devtmpfs       29584     0  29584     0%    /dev
tmpfs          29680     0  29680     0%    /dev/shm
tmpfs          29680  4384  25296    15%    /run
tmpfs           5120     0   5120     0%    /run/lock
tmpfs          29680     0  29680     0%    /sys/fs/cgroup
root@debian:~#
```



从返回的信息可以看见，rootfs空间占用已达到99%，仅剩约7MB的剩余空间。所以需要打包一份更大的rootfs镜像，考虑到使用的TF卡容量足够大，决定将镜像大小从512MB扩大到4GB，依次执行以下命令：

```
01. dd if=/dev/zero of=debian8-rootfs.ext4 bs=1M count=4096
02. mkfs.ext4 debian8-rootfs.ext4
03. sudo mount -o loop debian8-rootfs.ext4 /mnt
04. sudo cp -a -fr debian8-rootfs/* /mnt
05. sync
06. sudo umount /mnt
07. sudo chmod 777 debian8-rootfs.ext4
```

[复制代码](#)

将新制作好的ext4镜像烧录到目标板，然而很遗憾，出现了kernel panic的错误：

```
COM5 - PuTTY
[ 0.723489] icdc-d3 icdc-d3: codec icdc-d3 probe enter
[ 0.728809] write extend : sreg: 3 [0 - 4], creg: 10 sdata: 216
[ 0.736290] gpio speaker enable 0
[ 0.740087] dma dma0chan4: Channel 4 have been requested.(phy id 3,type 0x06 desc a3ecd000)
[ 0.748914] dma dma0chan5: Channel 5 have been requested.(phy id 2,type 0x06 desc a3ece000)
[ 0.757807] ingenic-alsa ingenic-alsa.0: icdc-d3-hifi <-> jz-asoc-aic-i2s mapping ok
[ 0.766457] dma dma0chan0: Channel 0 have been requested.(phy id 7,type 0x20 desc a3ee0000)
[ 0.775350] dma dma0chan1: Channel 1 have been requested.(phy id 6,type 0x20 desc a3ee1000)
[ 0.784243] ingenic-alsa ingenic-alsa.0: pcm dump dai <-> jz-asoc-pcm mapping ok
[ 0.792505] dma dma0chan7: Channel 7 have been requested.(phy id 0,type 0x04 desc a3ee2000)
[ 0.801359] ingenic-alsa ingenic-alsa.0: dmic dump dai <-> jz-asoc-dmic mapping ok
[ 0.811241] input: RatCharm Headset Jack as /devices/platform/ingenic-alsa.0/sound/card0/input0
[ 0.820979] Netfilter messages via NETLINK v0.30.
[ 0.825844] nfnl_acct: registering with nfnetlink.
[ 0.830999] ip_set: protocol 6
[ 0.834329] ip_tables: (C) 2000-2006 Netfilter Core Team
[ 0.840012] TCP: cubic registered
[ 0.843431] NET: Registered protocol family 17
[ 0.848145] Bridge firewalling registered
[ 0.853657] file system registered
[ 0.859043] using random self ethernet address
[ 0.863756] using random host ethernet address
[ 0.868671] android_usb gadget: Mass Storage Function, version: 2009/09/11
[ 0.875831] android_usb gadget: Number of LUNs=1
[ 0.880605] lun0: LUN: removable file: (no medium)
[ 0.886052] android_usb gadget: android usb ready
[ 0.891399] input: gpio-keys as /devices/platform/gpio-keys/input/input1
[ 0.898623] jz-rtc jz-rtc.0: setting system clock to 2018-07-02 15:22:27 UTC (1530544947)
[ 0.907214] ALSA device list:
[ 0.910308] #0: RatCharm
[ 0.913469] Waiting 1sec before mounting root device...
[ 1.649081] mmc0: new high speed SDHC card at address 59b4
[ 1.655271] mmcblk0: mmc0:59b4 RUNSD 29.5 GiB
[ 1.663074] Alternate GPT is invalid, using primary GPT.
[ 1.668626] mmcblk0: p1 p2 p3 p4 p5 p6 p7 p8
[ 1.921772] List of all partitions:
[ 1.925392] lf00          256 mtddblock0 (driver?)
[ 1.930764] lf01          3072 mtddblock1 (driver?)
[ 1.935974] lf02         12928 mtddblock2 (driver?)
[ 1.941238] b300         30981120 mmcblk0 driver: mmcblk
[ 1.946716] b301          8192 mmcblk0p1 B7 .r<t>#
[ 1.952681] b302         16384 mmcblk0p2 B6 &e$ w
[ 1.958602] b303         16384 mmcblk0p3 N1P0-8-P(26
[ 1.964555] b304         53248 mmcblk0p4 5-1Mrde<F
[ 1.970498] b305          4096 mmcblk0p5 M-g-r3F!
[ 1.976423] b306        102400 mmcblk0p6 *v-S xp
[ 1.982364] b307        716800 mmcblk0p7 M-b Q/d :
[ 1.988286] b308        103:00000 2097152 mmcblk0p8 Uik-#t{
[ 1.994223] No filesystem could mount root, tried: ext4
[ 1.999724] Kernel panic - not syncing: VFS: Unable to mount root fs on unknown-block(179,7)
[ 2.008422] Rebooting in 3 seconds..
```

根据kernel panic的提示，经分析应该是4GB的rootfs大小超过了分区表允许的大小。
为了解决这个问题，需要对u-boot中涉及到分区配置的文件进行修改。
对u-boot目录中的board/ingenic/RatCharm/partitions.tab进行修改，以下是原始代码：

```
01. property:
02.     disk_size = 4096m
03.     gpt_header_lba = 512
04.     custom_signature = 0
05.
06. partition:
07.     #name      = start,   size, fstype
08.     xboot     =    0m,     3m,
09.     boot      =    3m,     8m, EMPTY
10.     recovery  =   12m,    16m, EMPTY
11.     pretest   =   28m,    16m, EMPTY
12.     reserved  =   44m,    52m, EMPTY
13.     misc      =   96m,     4m, EMPTY
14.     cache     =  100m,   100m, LINUX_FS
15.     system    =  200m,   700m, LINUX_FS
16.     data      =  900m,  2048m, LINUX_FS
17.
18. #fstype could be: LINUX_FS, FAT_FS, EMPTY
```

复制代码

修改后的代码如下：

```
01. property:
02.     disk_size = 8192m
03.     gpt_header_lba = 512
```

```

04.     custom_signature = 0
05.
06. partition:
07.     #name      =  start,   size, fstype
08.     xboot      =    0m,     3m,
09.     boot       =    3m,     8m, EMPTY
10.     recovery   =   12m,    16m, EMPTY
11.     pretest    =   28m,    16m, EMPTY
12.     reserved   =   44m,    52m, EMPTY
13.     misc       =   96m,     4m, EMPTY
14.     cache      =  100m,   100m, LINUX_FS
15.     system     =  200m,  5000m, LINUX_FS
16.     data       =  5200m, 2048m, LINUX_FS
17.
18. #fstype could be: LINUX_FS, FAT_FS, EMPTY

```

[复制代码](#)

修改完毕后重新编译u-boot(编译前务必用make distclean清理)并下载至目标板, 然后再次尝试引导系统。

- **第五次尝试:**

登录系统后使用df命令查看磁盘使用情况, 可看到当前使用率仅12%, 剩余空间还有3GB左右。

```

COM5 - PuTTY
Starting ifup for eth0...
[ OK ] Started ifup for eth0.
[ OK ] Reached target Network.
[ OK ] Reached target System Initialization.
[ OK ] Listening on D-Bus System Message Bus Socket.
[ OK ] Reached target Sockets.
[ OK ] Reached target Timers.
[ OK ] Reached target Basic System.
Starting Regular background program processing daemon...
[ OK ] Started Regular background program processing daemon.
Starting /etc/rc.local Compatibility...
Starting Login Service...
Starting D-Bus System Message Bus...
[ OK ] Started D-Bus System Message Bus.
Starting System Logging Service...
Starting Permit User Sessions...
[ OK ] Started /etc/rc.local Compatibility.
[ OK ] Started Permit User Sessions.
[ OK ] Started System Logging Service.
[ OK ] Started Login Service.
Starting Getty on ttty...
[ OK ] Started Getty on ttty.
Starting Serial Getty on tttyS2...
[ OK ] Started Serial Getty on tttyS2.
[ OK ] Reached target Login Prompts.
[ OK ] Reached target Multi-User System.
[ OK ] Reached target Graphical Interface.
Starting Update UTMP about System Runlevel Changes...
[ OK ] Started Update UTMP about System Runlevel Changes.

Debian GNU/Linux 8 debian tttyS2

debian login: root
密码:
Linux debian 3.10.14 #4 PREEMPT Wed Jul 4 00:31:47 CST 2018 mips

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
root@debian:~# df
文件系统      1K-块   已用   可用   已用% 挂载点
/dev/root      3997376 417572 3353708   12% /
devtmpfs       29552    0   29552    0% /dev
tmpfs          29672    0   29672    0% /dev/shm
tmpfs          29672  4372   25300   15% /run
tmpfs           5120    0    5120    0% /run/lock
tmpfs          29672    0   29672    0% /sys/fs/cgroup
root@debian:~#

```

接下来准备设置SWAP交换区, 依次输入以下命令:

```

01. mkdir /opt/swap
02. dd if=/dev/zero of=/opt/swap/swapfile bs=1M count=256
03. mkswap /opt/swap/swapfile
04. chmod 0600 /opt/swap/swapfile

```

[复制代码](#)

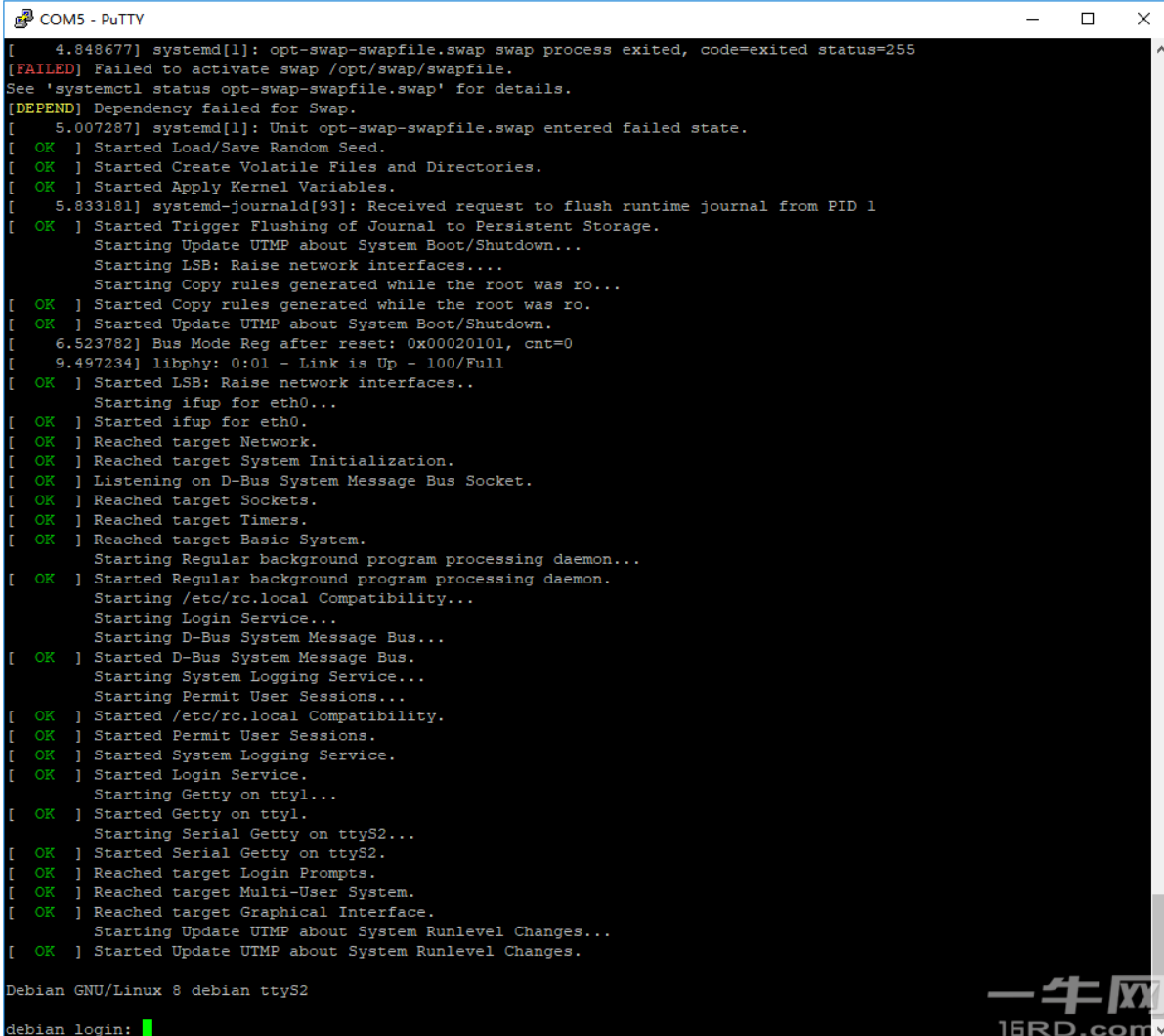
配置好swapfile文件后，我们还需要在/etc/fstab中添加相应命令，输入以下命令：

```
01. vim /etc/fstab  
复制代码
```

在编辑框的末尾添加以下命令，然后保存退出并重启目标板：

```
01. /opt/swap/swapfile swap swap defaults 0 0  
复制代码
```

重启目标板后在启动过程中出现了如下错误：



```
COM5 - PuTTY  
[ 4.848677] systemd[1]: opt-swap-swapfile.swap swap process exited, code=exited status=255  
[FAILED] Failed to activate swap /opt/swap/swapfile.  
See 'systemctl status opt-swap-swapfile.swap' for details.  
[DEPEND] Dependency failed for Swap.  
[ 5.007287] systemd[1]: Unit opt-swap-swapfile.swap entered failed state.  
[ OK ] Started Load/Save Random Seed.  
[ OK ] Started Create Volatile Files and Directories.  
[ OK ] Started Apply Kernel Variables.  
[ 5.833181] systemd-journald[93]: Received request to flush runtime journal from PID 1  
[ OK ] Started Trigger Flushing of Journal to Persistent Storage.  
Starting Update UTMP about System Boot/Shutdown...  
Starting LSB: Raise network interfaces...  
Starting Copy rules generated while the root was ro...  
[ OK ] Started Copy rules generated while the root was ro.  
[ OK ] Started Update UTMP about System Boot/Shutdown.  
[ 6.523782] Bus Mode Reg after reset: 0x00020101, cnt=0  
[ 9.497234] libphy: 0:01 - Link is Up - 100/Full  
[ OK ] Started LSB: Raise network interfaces..  
Starting ifup for eth0...  
[ OK ] Started ifup for eth0.  
[ OK ] Reached target Network.  
[ OK ] Reached target System Initialization.  
[ OK ] Listening on D-Bus System Message Bus Socket.  
[ OK ] Reached target Sockets.  
[ OK ] Reached target Timers.  
[ OK ] Reached target Basic System.  
Starting Regular background program processing daemon...  
[ OK ] Started Regular background program processing daemon.  
Starting /etc/rc.local Compatibility...  
Starting Login Service...  
Starting D-Bus System Message Bus...  
[ OK ] Started D-Bus System Message Bus.  
Starting System Logging Service...  
Starting Permit User Sessions...  
[ OK ] Started /etc/rc.local Compatibility.  
[ OK ] Started Permit User Sessions.  
[ OK ] Started System Logging Service.  
[ OK ] Started Login Service.  
Starting Getty on tty1...  
[ OK ] Started Getty on tty1.  
Starting Serial Getty on ttyS2...  
[ OK ] Started Serial Getty on ttyS2.  
[ OK ] Reached target Login Prompts.  
[ OK ] Reached target Multi-User System.  
[ OK ] Reached target Graphical Interface.  
Starting Update UTMP about System Runlevel Changes...  
[ OK ] Started Update UTMP about System Runlevel Changes.  
Debian GNU/Linux 8 debian ttyS2  
debian login: █
```

登录系统后使用htop命令，可以看到SWAP交换区大小仍然为0。

```
COM5 - PuTTY
CPU[*] 0.9% Tasks: 12, 3 thr; 1 running
Mem[#####|#####*15/57MB] Load average: 0.91 0.37 0.14
Swp[ 0/0MB] Uptime: 00:01:53

PID USER PRI NI VIRT RES SHR S CPU% MEM% TIME+ Command
236 root 20 0 7724 1716 1348 R 0.5 2.9 0:00.17 htop
1 root 20 0 5792 2948 2064 S 0.0 5.0 0:02.16 /sbin/init
79 root 20 0 10892 1488 1108 S 0.0 2.5 0:00.30 /lib/systemd/syst
93 root 20 0 8616 1632 1396 S 0.0 2.7 0:00.37 /lib/systemd/syst
169 root 20 0 9192 5000 300 S 0.0 8.4 0:00.00 dhclient -v -pf /
193 root 20 0 7684 1168 964 S 0.0 2.0 0:00.02 /usr/sbin/cron -f
195 root 20 0 3736 1352 1140 S 0.0 2.3 0:00.04 /lib/systemd/syst
197 messagebu 20 0 6056 1900 1588 S 0.0 3.2 0:00.08 /usr/bin/dbus-dae
204 root 20 0 31964 2328 1604 S 0.0 3.9 0:00.01 /usr/sbin/rsyslog
205 root 20 0 31964 2328 1604 S 0.0 3.9 0:00.00 /usr/sbin/rsyslog
206 root 20 0 31964 2328 1604 S 0.0 3.9 0:00.02 /usr/sbin/rsyslog
200 root 20 0 31964 2328 1604 S 0.0 3.9 0:00.08 /usr/sbin/rsyslog
213 root 20 0 4740 884 760 S 0.0 1.5 0:00.01 /sbin/agetty --no
214 root 20 0 9388 1868 1444 S 0.0 3.1 0:00.22 /bin/login --
232 root 20 0 7788 1996 1668 S 0.0 3.4 0:00.04 -bash

F1 Help F2 Setup F3 Search F4 Filter F5 Tree F6 SortBy F7 Nice - F8 Nice + F9 Kill F10 Quit
```



分析原因后判断是kernel内没有配置对swap的支持，需进行如下配置：

- 01. | Symbol: SWAP [=y] |
- 02. | Type : boolean |
- 03. | Prompt: Support for paging of anonymous memory (swap) |
- 04. | Location: |
- 05. | -> General setup |
- 06. | Defined at init/Kconfig:182 |
- 07. | Depends on: MMU [=y] && BLOCK [=y] |

[复制代码](#)

修改前后的截图对比，上图为修改前下图为修改后：

```
jdi@jdi-virtual-machine: /Ingenic/X1000V6.0-Moded-20171217/kernel
文件(F) 编辑(E) 查看(V) 搜索(S) 终端(T) 帮助(H)
.config - Linux/mips 3.10.14 Kernel Configuration
> General setup
      General setup
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 < > module

(mips-linux-gnu-) Cross-compiler tool prefix
() Local version - append to kernel release
[*] Automatically append version information to the version string
(RatCharm) Default hostname
[ ] Support for paging of anonymous memory (swap)
[*] System V IPC
[ ] POSIX Message Queues
[ ] open by fhandle syscalls
[ ] Auditing support
    IRQ subsystem --->
↑(+)
```

```
jdi@jdi-virtual-machine: /Ingenic/X1000V6.0-Moded-20171217/kernel
文件(F) 编辑(E) 查看(V) 搜索(S) 终端(T) 帮助(H)
.config - Linux/mips 3.10.14 Kernel Configuration
> General setup
      General setup
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 < > module

(mips-linux-gnu-) Cross-compiler tool prefix
() Local version - append to kernel release
[*] Automatically append version information to the version string
(RatCharm) Default hostname
[*] Support for paging of anonymous memory (swap)
[*] System V IPC
[ ] POSIX Message Queues
[ ] open by fhandle syscalls
[ ] Auditing support
    IRQ subsystem --->
↑(+)
```

重新编译好kernel并将其烧写到目标板，准备再次进行引导。

• 第六次尝试：

SWAP交换区挂载出错的问题已经解决，登录系统后使用htop命令，可见SWAP已经正常挂载，总容量为255MB，当前使用量为0。

```
COM5 - PuTTY
CPU[#* 0.9%] Tasks: 12, 3 thr; 1 running
Mem[#####15/57MB] Load average: 0.98 0.50 0.20
Swp[ 0/255MB] Uptime: 00:02:51

PID USER PRI NI VIRT RES SHR S CPU% MEM% TIME+ Command
235 root 20 0 7724 1716 1348 R 0.9 2.9 0:00.48 htop
1 root 20 0 5956 2968 2072 S 0.0 5.0 0:02.14 /sbin/init
79 root 20 0 10880 1488 1108 S 0.0 2.5 0:00.28 /lib/systemd/syst
99 root 20 0 8616 1648 1404 S 0.0 2.8 0:00.38 /lib/systemd/syst
169 root 20 0 9192 5000 300 S 0.0 8.4 0:00.00 dhclient -v -pf /
193 root 20 0 7684 1168 964 S 0.0 2.0 0:00.02 /usr/sbin/cron -f
195 root 20 0 3736 1352 1140 S 0.0 2.3 0:00.04 /lib/systemd/syst
197 messagebu 20 0 6056 1904 1588 S 0.0 3.2 0:00.08 /usr/bin/dbus-dae
204 root 20 0 31948 2332 1604 S 0.0 3.9 0:00.01 /usr/sbin/rsyslog
205 root 20 0 31948 2332 1604 S 0.0 3.9 0:00.00 /usr/sbin/rsyslog
206 root 20 0 31948 2332 1604 S 0.0 3.9 0:00.02 /usr/sbin/rsyslog
200 root 20 0 31948 2332 1604 S 0.0 3.9 0:00.08 /usr/sbin/rsyslog
212 root 20 0 4740 884 760 S 0.0 1.5 0:00.02 /sbin/agetty --no
213 root 20 0 9388 1860 1444 S 0.0 3.1 0:00.22 /bin/login --
231 root 20 0 7788 2000 1668 S 0.0 3.4 0:00.04 -bash

F1 Help F2 Setup F3 Search F4 Filter F5 Tree F6 SortBy F7 Nice - F8 Nice + F9 Kill F10 Quit
```

以上设置SWAP分区操作也可以放到打包rootfs的环节进行，在使用exit退出打包脚本前，使用同样的方法逐个完成前面的各项操作，然后使用exit退出脚本并制作ext4镜像即可。目前为止，Debian8的基本功能都已经可以正常运行，剩余的就是一些零碎的善后工作。

最后，还有一些杂七杂八的善后工作，列了一个清单：

- 添加普通用户：不少操作不适合在root账户下进行，因此需要建立一个普通账户。
- 设置sudo相关配置：在普通用户下很多操作需要sudo来临时提升权限，因此需要进行相关配置。
- 安装常用的软件包：安装日常使用中经常会用到的一些软件。

接下来就来逐步完善以上内容，首先是添加普通用户，使用串口登录root账户，然后执行以下命令：

```
01. adduser rat
复制代码
```

然后根据提示输入登录密码、个人信息等(可不填)，顺利完成后，就会建立起一个名为rat的普通账户。然后执行以下命令：

```
01. su rat
02. nano /home/rat/.bashrc
复制代码
```

然后在编辑框末尾添加以下内容：

```
01. export PATH=$PATH:/sbin
复制代码
```

保存并退出，此时我们就完成了对rat账户的所有配置工作，下次登录时rat账户就已经处于可用状态。

配置完rat这个普通账户后，自然就是配置sudo这个在普通账户中经常会使用到的命令。配置过程仍然需要在root账户下进行，登录root账户后依次执行以下命令：

```
01. apt-get install sudo
02. chown root:root /usr/bin/sudo
```

```
03. chmod 4755 /usr/bin/sudo
```

[复制代码](#)

接下来还需要对/etc/sudoers和/etc/hosts两个文件进行修改。首先修改/etc/sudoers，执行以下命令：

```
01. vim /etc/sudoers
```

[复制代码](#)

然后在编辑框中进行修改，以下是原始代码：

```
01. # User privilege specification
02. root    ALL=(ALL:ALL) ALL
```

[复制代码](#)

修改后的代码如下：

```
01. # User privilege specification
02. root        ALL=(ALL:ALL) ALL
03. rat        ALL=(ALL:ALL) ALL
```

[复制代码](#)

完成以上修改后对文件进行保存。需要注意的是因为是只读文件，所以保存时需要使用wq!命令强制保存。紧接着再来对/etc/hosts进行修改，执行以下命令：

```
01. vim /etc/hosts
```

[复制代码](#)

在编辑框中进行修改，以下是原始代码

```
01. 127.0.0.1    localhost
02. ::1         localhost ip6-localhost ip6-loopback
03. ff02::1     ip6-allnodes
04. ff02::2     ip6-allrouters
```

[复制代码](#)

修改后的代码如下：

```
01. 127.0.0.1    localhost debian
02. ::1         localhost ip6-localhost ip6-loopback
03. ff02::1     ip6-allnodes
04. ff02::2     ip6-allrouters
```

[复制代码](#)

完成修改后保存并退出编辑框。至此，对sudo的所有配置均已完成。

既然运行了Debian8，那么自然要利用好它的软件生态，可以在root账户下使用以下命令安装常用软件包：

```
01. apt-get install ssh build-essential gfortran openjdk-7-jdk
```

[复制代码](#)

完成安装后，可以在目标板对一些简单的程序进行本地编译(复杂的程序也可编译，但耗时会很长)，并且可以通过网络使用ssh登录到目标板进行操作或使用scp传输文件，增加了灵活性。以上所有善后工作也可以放到打包rootfs的环节进行，在使用exit退出打包脚本前，使用同样的方法逐个完成以上各项操作，然后使用exit退出脚本并制作ext4镜像，如此，得到的就是配置好了普通用户、sudo、常用软件包的镜像文件。

- 附上修改好的打包rootfs所需的脚本，以及最终完成的kernel镜像和rootfs镜像：
已修改好的用于打包Debian8 rootfs的脚本：




mkdebianrfs.zip (2.65 KB, 下载次数: 1)

最终版本的u-boot、kernel和rootfs镜像：

 u-boot-with-spl-mbr-gpt.zip (95.18 KB, 下载次数: 2)

 uImage.zip (2.58 MB, 下载次数: 2)

 debian8-rootfs.zip (309.78 MB, 下载次数: 2)