本文原创作者:[Cloud Chou.](http://weibo.com/muguachou) 欢迎转载，请注明出处和[本文链接](http://www.cloudchou.com/android/post-701.html)

概述

本篇讲述如何在Linux下使用Ndk[编译busybox](http://www.cloudchou.com/tag/%e7%bc%96%e8%af%91busybox)源码，生成一个静态链接的兼容性好的可支持中文的busybox。busybox是静态链接的，所以busybox可以既在recovery模式下运行，也可以在系统模式下运行，解决busybox not found的问题。busybox兼容性好，使用ndk编译，可以在armv7和armv5的cpu上运行，解决了[busybox illegal instruction](http://www.cloudchou.com/tag/busybox-illegal-instruction)的问题。另外busybox还支持中文，查看中文文件名的文件时，不会出现’?'。

接下来为大家介绍编译的步骤。

环境准备

* 1) 准备源代码

使用CM 10.1源码目录下external/busybox的源代码即可，为了不影响其它代码的编译，可以将其拷贝至1个单独的目录，如~/work/busybox

* 2) 准备编译环境

下载Ndk

<http://developer.android.com/tools/sdk/ndk/index.html>

解压Ndk包

|  |  |
| --- | --- |
| 1  2 | $chmod 755 android-ndk-r10d-linux-x86\_64.bin  $.**/**android-ndk-r10d-linux-x86\_64.bin |

* 3) 将编译工具链目录添加至PATH环境变量

|  |  |
| --- | --- |
| 1 | **export** PATH=$PATH:**/**home**/**cloud**/**softwares**/**android-ndk-r10d**/**toolchains**/**arm-linux-androideabi-4.9**/**prebuilt**/**linux-x86\_64**/**bin |

准备config文件

拷贝 configs/android\_ndk\_defconfig 至busybox目录.config

修改.config里的CONFIG\_SYSROOT变量和CONFIG\_EXTRA\_CFLAGS变量

* 1) CONFIG\_SYSROOT变量

CONFIG\_SYSROOT变量指定ndk系统目录，例如我们可以配制成:

CONFIG\_SYSROOT=”/home/cloud/softwares/android-ndk-r10d/platforms/android-15/arch-arm”

当然我们也可以指定成别的Android版本，比如android-14

* 2) CONFIG\_EXTRA\_CFLAGS变量

CONFIG\_EXTRA\_CFLAGS变量指定编译变量，原来.config文件的设置如下所示：

|  |  |
| --- | --- |
| 1  2  3  4 | CONFIG\_EXTRA\_CFLAGS="-DANDROID -D\_\_ANDROID\_\_ -DSK\_RELEASE -nostdlib  -march=armv7-a -msoft-float -mfloat-abi=softfp -mfpu=neon -mthumb  -mthumb-interwork -fpic -fno-short-enums -fgcse-after-reload  -frename-registers" |

-march=armv7-a 指定支持armv7-a指令架构，但这样编译出来的busybox必须在armv7-a以上架构的cpu执行，否则运行会出错，提示illegal instructions。因此我们必须去掉选项-march=armv7-a。更改后如下所示：

|  |  |
| --- | --- |
| 1  2  3 | CONFIG\_EXTRA\_CFLAGS="-DANDROID -D\_\_ANDROID\_\_ -DSK\_RELEASE -nostdlib  -msoft-float -mfloat-abi=softfp -mfpu=neon -mthumb -mthumb-interwork  -fpic -fno-short-enums -fgcse-after-reload -frename-registers" |

为了让busybox运行时不依赖于任何动态库，这样在Recovery模式下运行busybox也无问题。只需将busybox设置为静态链接的即可，这时需在.config文件里设置CONFIG\_STATIC变量，设置如下所示即可：

|  |  |
| --- | --- |
| 1 | CONFIG\_STATIC=y |

但是.config文件的初始设置时指定链接方式为动态链接，需将其改为静态链接，其原始的设置如下所示：

|  |  |
| --- | --- |
| 1  2  3  4 | CONFIG\_EXTRA\_LDFLAGS="-Xlinker -z -Xlinker muldefs -nostdlib -Bdynamic -Xlinker  -dynamic-linker -Xlinker /system/bin/linker -Xlinker -z -Xlinker nocopyreloc  -Xlinker --no-undefined ${SYSROOT}/usr/lib/crtbegin\_dynamic.o  ${SYSROOT}/usr/lib/crtend\_android.o" |

修改后如下所示：

|  |  |
| --- | --- |
| 1  2  3  4 | CONFIG\_EXTRA\_LDFLAGS="-Xlinker -z -Xlinker muldefs  -nostdlib -Xlinker -z -Xlinker nocopyreloc -Xlinker  --no-undefined ${SYSROOT}/usr/lib/crtbegin\_dynamic.o  ${SYSROOT}/usr/lib/crtend\_android.o" |

去掉了-Bdynamic -Xlinker -dynamic-linker -Xlinker /system/bin/linker。

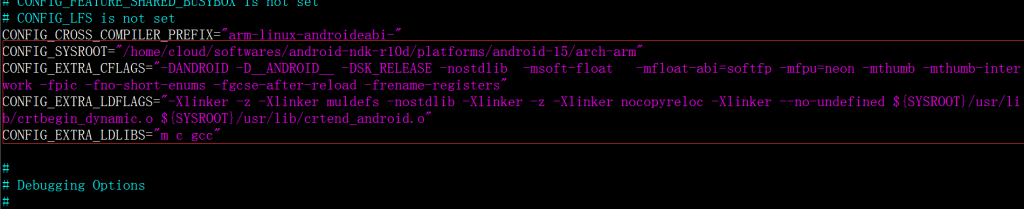
此时如果编译还是会出错，提示找不到dl动态库，还需修改CONFIG\_EXTRA\_LDLIBS变量，去掉dl库。修改前如下所示：

|  |  |
| --- | --- |
| 1 | CONFIG\_EXTRA\_LDLIBS="dl m c gcc" |

修改后如下所示：

|  |  |
| --- | --- |
| 1 | CONFIG\_EXTRA\_LDLIBS="m c gcc" |

设置如下图所示：

[](http://www.cloudchou.com/wp-content/uploads/2015/02/config.png)

支持unicode

需要让busybox支持unicode编码，否则使用ls查看中文文件名的文件时会得到?,而不能正常显示中文文件名。

使用make menuconfig命令调出选择界面，然后进行选择：

Busybox Settting -> General Configuration -> Support Unicode

按空格键勾选Support Unicode

Charcter code to substitue unprintable characters with 填63，即0x3F

Range of supported Unicode charaters 填1个较大数值，如200000

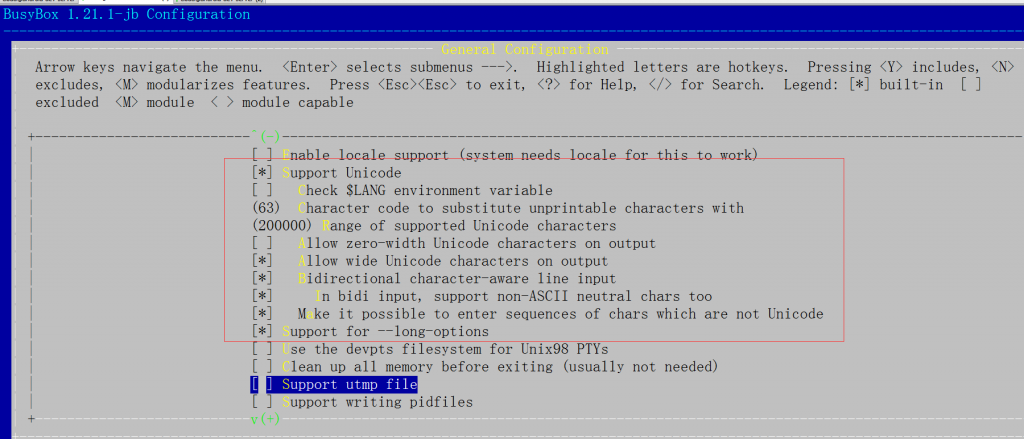
勾选Allow wide Unicode characters on output

勾选 Bidirectional character-aware line input

勾选 Make it possible to enter sequences of chars wich are not unicode

这样就可支持unicode了，可以解析utf8编码的中文文件名

支持unicode的菜单设置如下图所示：

[](http://www.cloudchou.com/wp-content/uploads/2015/02/unicode.png)

选择busybox带的工具

make menuconfig调出勾选菜单，在Applets目录选择需要带的工具，

Archival Utilties 主要是 打包工具

Coreutils 主要是 核心工具，比如basename，cat，date

Console Utilities 主要是 控制台工具

Debian Utilities 是Debian系列Linux用的工具

Editors 文本编辑工具例如vi,awk,sed

Finding Utilties 查找工具

Init Utilities Init相关工具

Networking Utilities 网络相关工具，如ping

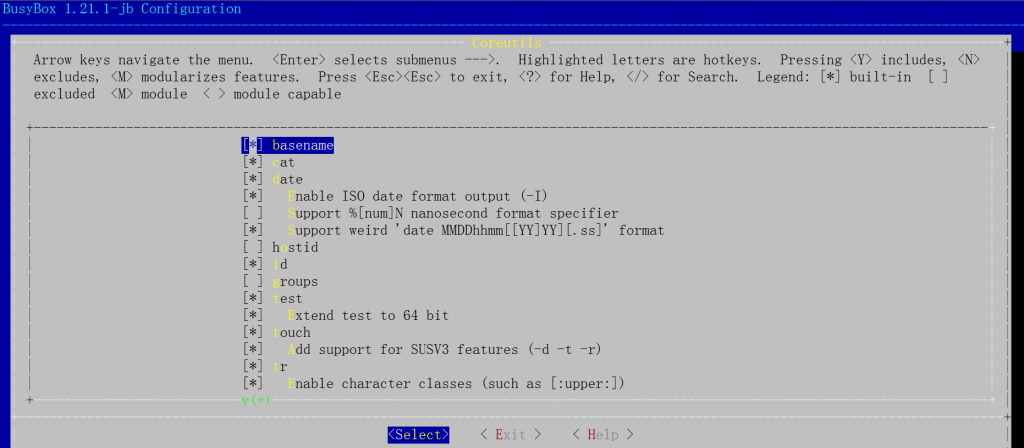
如果选择了mount工具，在进行编译时可能会提示出错，

此时需要修改文件libbb/Kbuild：

添加：

lib-$(CONFIG\_MOUNT) += android.o

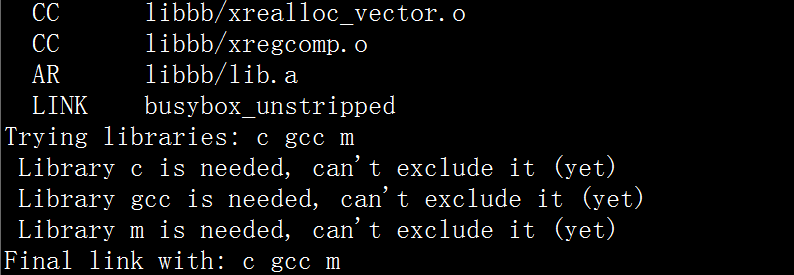
菜单选择如下图所示：

[](http://www.cloudchou.com/wp-content/uploads/2015/02/coretools.png)

编译

make –j8

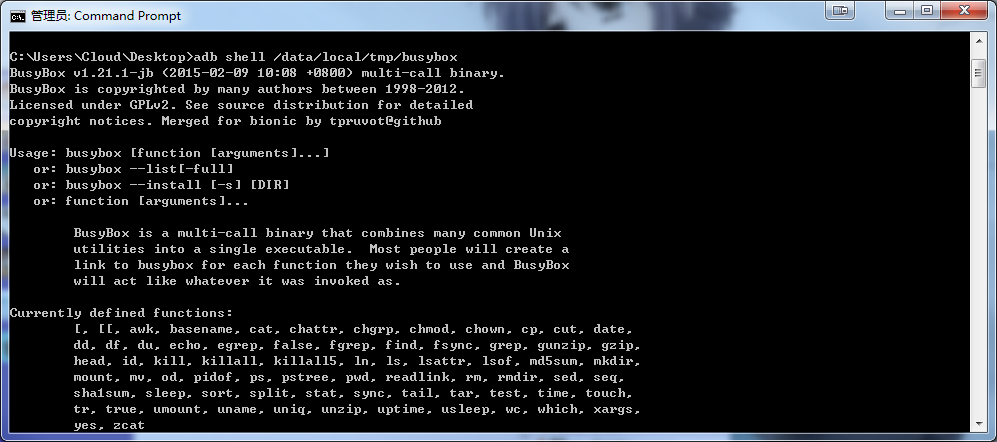
编译成功示意图如下图所示：

[](http://www.cloudchou.com/wp-content/uploads/2015/02/make.png)

测试并执行

|  |  |
| --- | --- |
| 1  2  3 | adb push busybox **/**data**/**local**/**tmp  adb shell **chmod** 755 **/**data**/**local**/**tmp**/**busybox  adb shell **/**data**/**local**/**tmp**/**busybox |

成功输出如下图所示：

[](http://www.cloudchou.com/wp-content/uploads/2015/02/execute.png)