

三篇文章差不多，但都涉及了一些其他东西，所以都转了。

第一篇

1. 进入终端

1. 直接执行命令 `emulator -shell;`
2. 在启动 `emulator` 后，通过命令 `adb shell` 进入。

2. 下载busybox以及交叉编译工具

1. 首先去 [busybox](http://busybox.net/) 主页 下载最新版本的 `busybox` 源代码。
2. 下载交叉编译工具：GNU Toolchain for ARM Processors

下载地址：http://www.codesourcery.com/gnu_toolchains/arm/download.html

其中第一项选择 ARM EABI 或 ARM GNU/Linux，第二项选择 IA32 GNU/Linux TAR 即可。

最后得到一个 `arm-2007q3-51-arm-none-linux-gnueabi-i686-pc-linux-gnu.tar.bz2` 文件

将压缩包解压到 `linux` 下某个目录，并设置好环境变量，使之能找到前缀为 `"arm-none-linux-gnueabi-"` 的交叉编译工具。

3. 对busybox进行相应配置

3. 进入 `busybox` 源码目录。

先执行 `make menuconfig` 命令对编译过程进行配置。过程如下：

`Busybox Settings -> Build Options ->`

`[*] Build BusyBox as a static binary (no shared libs)` 这个要选上，因上这样子编译出来的 `busyBox` 才可以独立运行的。

`(/home/jo/tool/arm-2008q3/bin/arm-none-linux-gnueabi-) Cross Compiler prefix` 这是交叉编译器的“路径+前缀”，要根据具体的情况来设置。

`Busybox Settings -> Installation Options->`

`[*] Don't use /usr` 这样子编译出来的 `busybox` 才不会安装到你主机的 `/usr` 目录下。一定要选上。

- 4 执行 `make` 编译源码。如果编译过程中产生无法找到头文件的错误，可指定交叉编译器用到的头文件所在路径。如

```
$make -I ./home/jo/tool/arm-2008q3/arm-none-linux-gnueabi/libc/usr/include/
```

- 5 编译成功后将得到一个名为 `busybox` 的可执行文件。

4. 把busybox移植入android

- 6 在 `Android` 的 `console` 下建立目录 `/data/busybox`，并使用下面的命令将文件 `busybox` 复制该目录中：

```
adb push busybox /data/busybox
```

- 7 从 `Android` 的 `console` 进入 `/data/busybox` 目录，为可执行文件 `busybox` 加上可执行属性：`chmod 775 busybox`

- 8 这时我们就可以使用 `busybox` 工具了，如果希望在任何目录下都可以直接使用 `busybox`，则需要将路径 `/data/busybox` export 到 `path` 环境变量中：

```
export PATH=$PATH:/data/busybox
```

9 设置环境变量之后，在命令行输入busybox，即可看到其用法，如：

在当前目录及其子目录中查找名为init.rc的文件可使用以下命令：

```
$busybox find . -name init.rc
```

10 安装busybox。

通过执行./busybox --install -s 我们会看到出现一些错误，原因是因为Android系统的根目录和/sbin目录是只读的。因此我们无法把一些常用命令安装到/bin目录下，也不能修改/init.rc文件来设置PATH环境变量。

对于如何执行命令busybox --install -s 的默认安装路径，这里我还不清楚，或许跟Makefile和编译busybox的相关设置有关。

尽管我们不可在Android里面通过执行busybox --install -s 命令来安装常用命令，我们却可以通过另外一个方法来把常用命令安装到Android里面。

方法很简单，可以先在外面的linux系统中执行make install命令把相关的命令提取出来。然后再通过命令adb push ./_install/bin /data/busybox/bin 把这些常用命令复制进Android系统中。注意：make install安装命令的所在路径可通过执行make menuconfig进行设置，默认为./_install目录下。

这样，我们就不用每次使用busybox提供的一个命令时，都要在前面加上busybox了。

11 修改/init.rc文件设置环境变量。

要提醒的是，修改PATH环境变量的时候，应该把busybox常用命令的路径"/data/busybox/bin"放在Android的常用命令路径"/system/bin"的前面，否则，先查找得到的将不是busybox的命令，可以这样设置：

```
export PATH /data/busybox/bin:/sbin:/system/sbin:/system/bin:/system/sbin
```

第二篇

大家是否有过这样的经历，在命令行里输入adb shell，然后使用命令操作你的手机或模拟器，但是那些命令都是常见Linux命令的阉割缩水版，用起来很不爽。是否想过在Android上使用较完整的shell呢？用BusyBox吧。不论使用adb连接设备使用命令行还是在手机上直接用terminal emulator都可以。

一、什么是BusyBox ？

BusyBox 是标准 Linux 工具的一个单个可执行实现。BusyBox 包含了一些简单的工具，例如 cat 和 echo，还包含了一些更大、更复杂的工具，例如 grep、find、mount 以及 telnet。有些人将 BusyBox 称为 Linux 工具里的瑞士军刀。简单的说BusyBox就好像是个大工具箱，它集成压缩了 Linux 的许多工具和命令。（摘自百度百科）

二、在Android上安装BusyBox
















准备：

0. 先要把手机给Root了，具体教程这里就不提供了，网上有很多。

1. 下载BusyBox的binary，打开这个地址 <http://www.busybox.net/downloads/binaries>，选择最新版本，然后下载对应你的设备架构的版本，这里我下载了busybox-armv6l，下面将以这个文件名为示例。

Index of /downloads/binaries/1.18.0

Name	Last modified	Size	Description
 Parent Directory		-	
 busybox-armv4eb	31-Dec-2010 04:47	1.0M	
 busybox-armv4l	31-Dec-2010 04:47	1.0M	

	busybox-armv4t1	31-Dec-2010 04:48	1.0M
	busybox-armv5l	31-Dec-2010 04:48	1.0M
	busybox-armv6l	31-Dec-2010 04:48	1.0M
	busybox-i486	31-Dec-2010 04:48	837K
	busybox-i586	31-Dec-2010 04:48	837K
	busybox-i686	31-Dec-2010 04:48	844K
	busybox-m68k	31-Dec-2010 04:48	1.0M
	busybox-mips	31-Dec-2010 04:48	1.6M
	busybox-mips64	31-Dec-2010 04:48	1.5M
	busybox-mipsel	31-Dec-2010 04:48	1.6M
	busybox-powerpc	31-Dec-2010 04:48	1.1M
	busybox-powerpc-440fp	31-Dec-2010 04:48	1.1M
	busybox-sh4	31-Dec-2010 04:49	923K
	busybox-sparc	31-Dec-2010 04:49	1.1M
	busybox-x86_64	31-Dec-2010 04:49	916K

Apache Server at www.busybox.net Port 80

2. 需要有一个命令行的环境，在电脑上使用adb或在手机上使用terminal emulator。
3. 连接手机和电脑，手机的USB Mode设置成None（仅充电），并且开启USB调试模式。

安装：

1. 将busybox-armv6l重命名为busybox
2. 将busybox传入手机的SD卡，可以使用下面的命令或自己想其他办法。

打开terminal（Linux，Mac）或cmd（Windows）

```
adb push ~/Desktop/busybox /mnt/sdcard
```

其中的~/Desktop请根据自己的情况替换成正确的路径

3. 输入以下命令，为了在/system目录写入文件

```
adb shell
su
mount -o remount,rw -t yaffs2 /dev/block/mtdblock3 /system
```

使用 ls 检查一下 /system 里是否有 xbin 目录，没有的话输入 mkdir xbin 创建，因为本示例是要把 busybox安装到 /system/xbin 。

4. 复制 busybox 文件到 /system/xbin，并为其分配“可执行”的权限

```
cp /mnt/sdcard/busybox /system/xbin
chmod 755 busybox
```

5. 这时就可以使用 busybox 的命令了，例如以前没有清屏的clear命令，现在只需输入 busybox clear 就可以实现清屏功能，使用完整版的 ls 只需输入 busybox ls 。

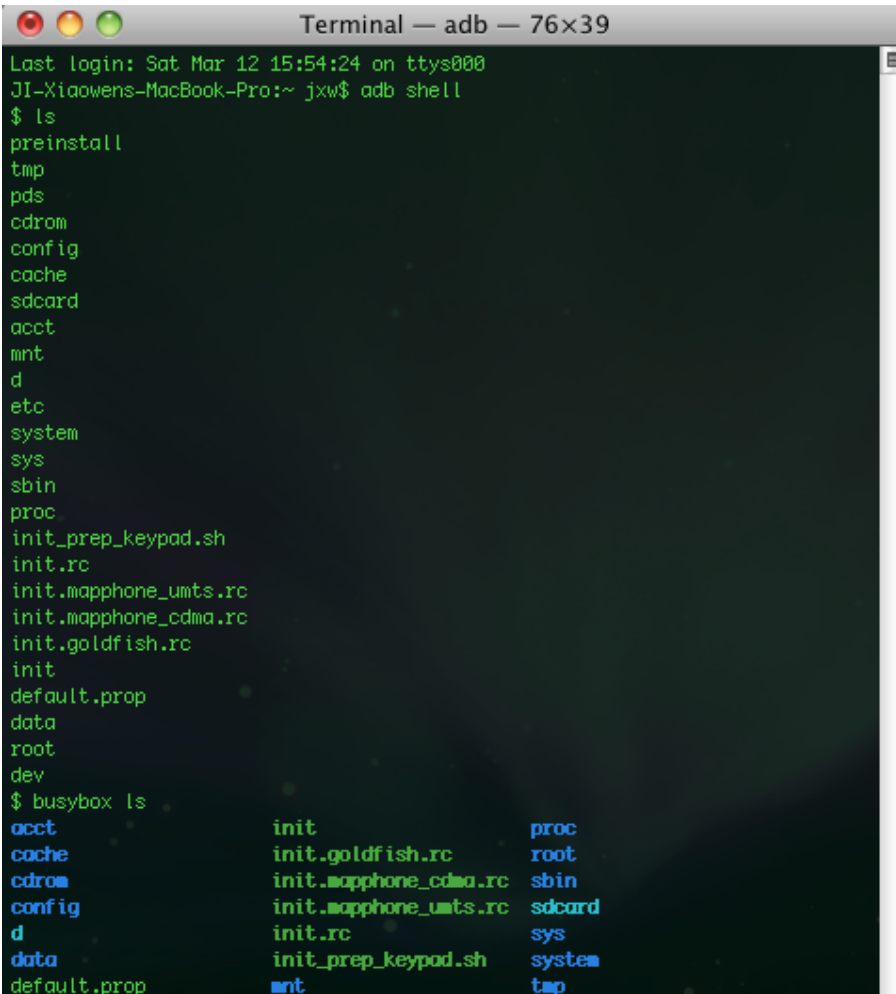
但是每次前面都加上个busybox太麻烦了，所以我们还要继续完成安装。

在 `/system/xbin` 下输入

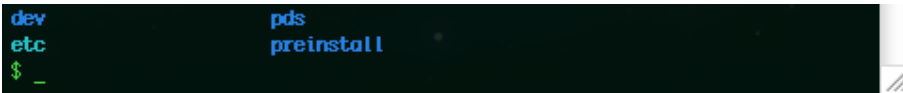
```
busybox --install .
```

如果想安装到别的目录，则把点替换成别的路径。

至此就安装完成了，比较一下原来的 `ls` 命令和 `busybox` 里的 `ls` 命令。



```
Terminal — adb — 76x39
Last login: Sat Mar 12 15:54:24 on ttys000
JI-Xiaowens-MacBook-Pro:~ jxw$ adb shell
$ ls
preinstall
tmp
pds
cdrom
config
cache
sdcard
acct
mnt
d
etc
system
sys
sbin
proc
init_prep_keypad.sh
init.rc
init.mapphone_ums.rc
init.mapphone_cdma.rc
init.goldfish.rc
init
default.prop
data
root
dev
$ busybox ls
acct          init          proc
cache         init.goldfish.rc  root
cdrom         init.mapphone_cdma.rc  sbin
config        init.mapphone_ums.rc  sdcard
d             init.rc          sys
data          init_prep_keypad.sh  system
default.prop  mnt              tmp
```



常见错误：

1. 如果安装时出现这样的错误，

```
busybox: /bin/zcat: No such file or directory
```

```
busybox: /sbin/zcip: Invalid cross-device link
```

说明没有输入安装路径，正确的示例 `busybox --install /system/xbin`

2. 如果出现这样的错误，

```
cp: /system/xbin/busybox: Read-only file system
```

说明没有正确输入上面第三步的mount命令。

小技巧：

1. busybox 里有 ash 和 hush 还有 sh 这几种 shell，在命令行输入 ash 或 hush，可以像在 bash 里那样，通过按上下键选择刚才输入的命令。

2. android系统本身就有ls命令，busybox里也有ls，输入ls时调用的是android的ls，那么想用busybox的ls就要每次都在前面加个busybox吗？不用，使用alias命令可以搞定。

```
alias ls='busybox ls'
```

同样的，cp、mv等二者都有的命令都可以这样搞定。也可以通过修改 /init.rc 来解决。

第三篇

android 自带的toolbox

shell实现分为两部分：

一、shell解释器和内置命令

源码位于system/core/sh目录下，主要完成shell命令的解释查找，对于builtins.c中包含的内置命令，直接执行，对于 toolbox的扩展命令，间接调用toolbox程序完成。

二、toolbox扩展命令

主要完成扩展命令的执行，每一个扩展命令对应一个name_main函数，如ls命令，对应ls_main函数。同时，每一个扩展命令都由一个 system/core/toolbox/目录下面的.c文件实现。toolbox.c会根据这个目录下面的.c文件生成tools.h头文件，并在 system/core/toolbox/[Android.mk](#) 文件中为每个命令生成指向toolbox的连接。toolbox的实现结构使它扩展一个命令很容易。

假设现在我们自己想手工添加一个shell命令mycommand，只要在system/core/toolbox/目录下面新建一个mycommand.c文件，并在里面实现一个mycommand_main函数，然后在system/core/toolbox/[Android.mk](#) 中添加mycommand.c即可。Android.mk会自动把它编译进toolbox程序，并在编译生成的Android系

你加my command, copy。Android, 里面已经编译过toolbox程序，并在编译生成的Android系统/system/bin 目录下为这个命令生成一个指向toolbox的连接。

替换toolbox为busybox

接下来翻译一下网上的一篇文章，借助它，可以把Android自带的toolbox替换成busybox。

[Installing Busybox command line tools](#)

在Android系统中安装busybox命令行工具

本文简单地介绍了怎么把busybox安装到Android的文件系统中去。如果你想直接安装，可以从下面的地址下载我已经预编译好并在Android2.1系统上试验成功的busybox，然后直接跳过下面的安装步骤。

下载在Linux公社的1号 FTP服务器里，下载地址：

FTP地址： <ftp://www.linuxidc.com>

用户名： www.linuxidc.com

密码： www.muui.cc

在 2011年LinuxIDC.com\3月\Android自带的toolbox分析及扩展

下载方法见 <http://www.linuxidc.net/thread-1187-1-1.html>

一、编译busybox

1、下载busybox的最新版本，本文写作时最新版本是1.13.3。

下载地址： <http://www.busybox.net/>

2、解压源码：

```
tar jxf busybox-1.13.3.tar.bz2
```

3、运行menuconfig对busybox进行配置

```
cd busybox-1.13.3/
```

```
make menuconfig
```

4、在menuconfig中设置以下选项

Busybox Settings --> Build Options --> Build Busybox as a static binary (no shared libs) - Enable this option by pressing "Y"

Busybox Settings --> Build Options --> Cross compiler prefix - Set this option equal to "arm-none-linux-gnueabi-"

Busybox Settings --> Installation Options --> Don't use /usr - Enable this option by pressing "Y"

5、把交叉编译器的地址导入到环境变量：

```
export PATH=/opt/arm/arm-2007q3/bin:$PATH
```

6、编译busybox

```
make
```

二、安装busybox

把busybox安装到[Android](#)系统中去，做这几步：

1、在[Android](#)系统根目录下创建一个/bin目录。

```
mkdir /<path-to-Android-fs>/bin
```

2、把编译出来的busybox复制到/bin目录下

```
cp busybox /<path-to-Android-fs>/bin
```

3、把busybox安装到[Android](#)机器中

```
cd /bin
```

```
./busybox --install
```

三、把busybox作为默认shell

需要像下面这样编辑一下init.rc

1、编辑console服务，让它默认运行busybox

```
service console /system/bin/sh -> service console /bin/sh
```

2、把busybox路径加入到环境变量中

```
export PATH /sbin:/system/sbin:/system/bin:/system/xbin --> export PATH  
/bin:/sbin:/system/sbin:/system/bin:/system/xbin
```

注：

我使用busybox时，只是想简单地增加一些命令，把toolbox一些功能不是很全的命令替换掉，所以操作上没有上面说的那么复杂。下面是我的替代方案，可以试一下：

1、把busybox复制到/system/bin目录下。

```
adb push busybox /system/bin
```

2、把要添加的命令通过ln建立到busybox的连接。

比如，[Android](#)自带的toolbox是没有test这个命令的。我们要添加test命令就可以：

```
cd /system/bin
```

```
ln -s busybox test
```



这样，用户通过机器上的shell执行test命令时，就会调用busybox中实现test功能的applet。

对于一些原有的命令，如ls、chown等，如果不想用toolbox，也可以把它们的连接目标指向toolbox，拿chown来举例。

```
cd /system/bin
```

```
rm chown
```

```
ln -s busybox chown
```

这样做，最大的好处就是保证对系统的改动最少，又可以最大限度的扩展shell功能。

英文文档

To build busybox

Download the latest version of busybox from the following website. At the time of writing the latest version was v.1.13.3.

<http://www.busybox.net> Extract the busybox source:

```
tar jxf busybox-1.13.3.tar.bz2
```

Configure busybox by running menuconfig

```
cd busybox-1.13.3/
```

make menuconfig

In menuconfig set the following options

Busybox Settings --> Build Options --> Build Busybox as a static binary (no shared libs) - Enable this option by pressing "Y"

Busybox Settings --> Build Options --> Cross compiler prefix - Set this option equal to "arm-none-linux-gnueabi-"

Busybox Settings --> Installation Options --> Don't use /usr - Enable this option by pressing "Y"

Export path to where the cross-compiler is located on the host, for example:

```
export PATH=/opt/arm/arm-2007q3/bin:$PATH
```

Build busybox

```
make
```

Installing Busybox

To install busybox in the target file-system

Create a /bin directory in the target file-system. For example:

```
mkdir /<path-to-Android-fs>/bin
```

Copy the busybox binary to the /bin directory in the target file-system

cp busybox /<path-to-Android-fs>/bin

Install the busybox command line tools on the target by executing the following commands:

```
cd /bin
```

```
./busybox --install
```

Make the Busybox shell the default shell

To make the busybox shell the default shell, edit the file "init.rc" in the target file-system as follows:

Edit the console service so that it runs the busybox shell and not the default shell by replacing:

```
service console /system/bin/shWith:
```

```
service console /bin/shAdd the path of the busybox command line tools to the system path variable by replacing:
```

```
export PATH /sbin:/system/sbin:/system/bin:/system/xbinWith
```

```
export PATH /bin:/sbin:/system/sbin:/system/bin:/system/xbin
```

```
=====
```