Building from source--想自己编译cyanogenmod ROM吗？

From CyanogenMod Wiki

How do I compile the CyanogenMod kernel from source

The following instructions will enable the build of a CyanogenMod KERNEL, not the build of the CyanogenMod itself. As a matter of fact you will download the Google's Android source without the CyanogenMod modifications. If you want to build the CyanogenMod itself from source you should look at the following section (2.* of this document) "Building for Passion/Nexus One". Even if you don't go for these targets it should be a good help.

BEWARE! INEXPERIENCED USERS MAY BRICK THEIR PHONES!

These instructions have only been tested for **4.2.10.1** on **25 DEC 2009**. **This information may be out-of-date.**

These instructions are for a **Debian/Ubuntu 32bit system**. If you have a different platform, more information can be found here and here.

Install CyanogenMod on your Phone

1. Install CyanogenMod on your phone.

Take a backup

Better safe than sorry, boot into recovery and take a backup of your working phone.

Install Repo

1. Install the various tools you will need.

```
sudo aptitude install git-core gnupg sun-java6-jdk flex bison gperf libsdl-dev libesd0-dev libwxgtk2.6-dev build-essential zip curl libncurses5-dev
```

   And for 64bit only:

```
sudo aptitude install ia32-libs lib32z1-dev lib32ncurses5-dev gcc-multilib g++-multilib
```

2. Make sure you have a /bin directory in your /home directory

```
cd ~/
mkdir bin
cd bin
echo $PATH
```

   You should see something like '...:/home/user_name/bin:...' in a line of text. If you don't see this, try restarting your computer.
3. Download repo and make it executable

```
curl http://android.git.kernel.org/repo >~/bin/repo
chmod a+x ~/bin/repo
```

4. Create a directory to hold you working files:

```
mkdir ~/mydroid
cd ~/mydroid
```

5. Download the latest Repo:

```
repo init -u git://android.git.kernel.org/platform/manifest.git
```

   If everything was successful, you should see a message like 'repo initialized in /mydroid'
6. Next, get the latest files:

```
repo sync
```

   This step may take a while.
7. Create an environment variable denoting the location of the android toolchain as follows:

```
export CCOMPILER=/home/$USER/mydroid/prebuilt/linux-x86/toolchain/arm-eabi-4.4.0/bin/arm-eabi-
```

Prepare Kernel From Source

1. Create a new directory in which to place the kernel source

   ```
   mkdir androkern
   ```

2. Enter the kernel directory

   ```
   cd androkern
   ```

3. Download the kernel

   ```
   git clone git://github.com/cyanogen/cm-kernel.git
   ```

4. Enter the source directory

   ```
   cd cm-kernel
   ```

5. Retrieve a kernel .config file. This can be pulled from the phone or from a boot.img
   1. Retrieve a working kernel config from your phone:

      ```
      adb pull /proc/config.gz /home/user_name/androkern/cm-kernel/config.gz
      ```

      1. Unzip the config.gz and rename it to .config

         ```
         gunzip config.gz && mv config .config
         ```

   2. Alternatively, you can pull the .config from the newest boot.img
      1. `scripts/extract-ikconfig boot.img > .config`

6. Run

   ```
   make ARCH=arm CROSS_COMPILE=$CCOMPILER oldconfig
   ```

   Provide answers to the questions it asks, if any, or just keep pressing enter to accept defaults.
7. Customize the kernel

   ```
   make ARCH=arm CROSS_COMPILE=$CCOMPILER menuconfig
   ```

8. Make the build

   ```
   make ARCH=arm CROSS_COMPILE=$CCOMPILER
   ```

   This step may take from minutes to hours.

At this point you should have a kernel stored in cm-kernel/arch/arm/boot/zImage

Build CyanogenMod

1. `cd ~/mydroid`
2. `source build/envsetup.sh`
3. Now you need choose which product you want to build, You can list all buildable product with

   ```
   sed -n -e "s/^add_lunch_combo//gp" vendor/*/vendorsetup.sh
   ```

4. `lunch cyanogen_dream_us-eng`

   this load cyanogenmod product
5. `make -j4`

   build process is long... take a pizza, a beer and watch TV. The -j flag specifies how many threads the compiler should use. Rule of thumb is # of processors+1 but will vary with who you ask.

Next soon.

Complete the build

You need to merge this file with a working cyanogen ramdisk in order to create a boot image suitable for flashing.

Follow these instructions here in order to do so.

If all goes well, you should now be running your own custom CyanogenMod kernel on your phone.

If you are building for Nexus One then you should use --base 0x20000000 when running mkbootimg.

Building for Passion/Nexus One

This will explain how to build CM, using the code from Cyanogen's github repository.

Obtaining the code

Read the introduction and how-to on obtaining the source from http://source.android.com/ this is quite similar but won't cover the basics.

We use Cyanogens manifest file instead of theirs, it has all the good stuff in it.

Initialize your repository

1.
   ```
   mkdir ~/android-cm
   ```

2.
   ```
   cd ~/android-cm
   ```

3.
   ```
   repo init -u git://github.com/cyanogen/android.git -b eclair
   ```

4.
   ```
   repo sync
   ```

The sync can take quite some time, it pulls about 2gig of files.

Create build environment

1.
   ```
   cd ~/android-cm
   ```

2.
   ```
   source build/envsetup.sh
   ```

Build ADB (only required if you haven't built it yet)

1.
   ```
   make -j4 adb
   ```

Configure build

1. Now you need choose which product you want to build, You can list all buildable products with
   ```
   sed -n -e "s/^add_lunch_combo//gp" vendor/*/vendorsetup.sh
   ```

2.
   ```
   lunch cyanogen_passion-userdebug
   ```
   select a cyanogenmod build type

Get proprietary files from your device

1.
   ```
   cd ~/android-cm/vendor/google/passion
   ```

2.
   ```
   ./extract-files.sh
   ```

3.
   ```
   cd -
   ```

Build it

Looks like we're ready to build

1. start building

```
make -j4
```

Now this takes from 30 min to 2 hours, depending on your computer.

Once ready, all files will be placed in

```
~/android-cm/out/target/product/passion
```

You can now create a rom or just flash them with fastboot like this

1. `cd ~/android-cm/out/target/product/passion`
2. `fastboot flash system system.img`
3. `fastboot flash boot boot.img`
4. `fastboot flash userdata userdata.img`
5. `fastboot reboot`

TODO

- TODO: Explain how to compile wlan.ko in order to enable wifi.
- TODO: Explain how to provide other kernel modules for a chroot debian install on the G1.
- TODO: Explain how to compile the rest of CyanogenMod

Retrieved from "http://wiki.cyanogenmod.com/index.php/Building_from_source"