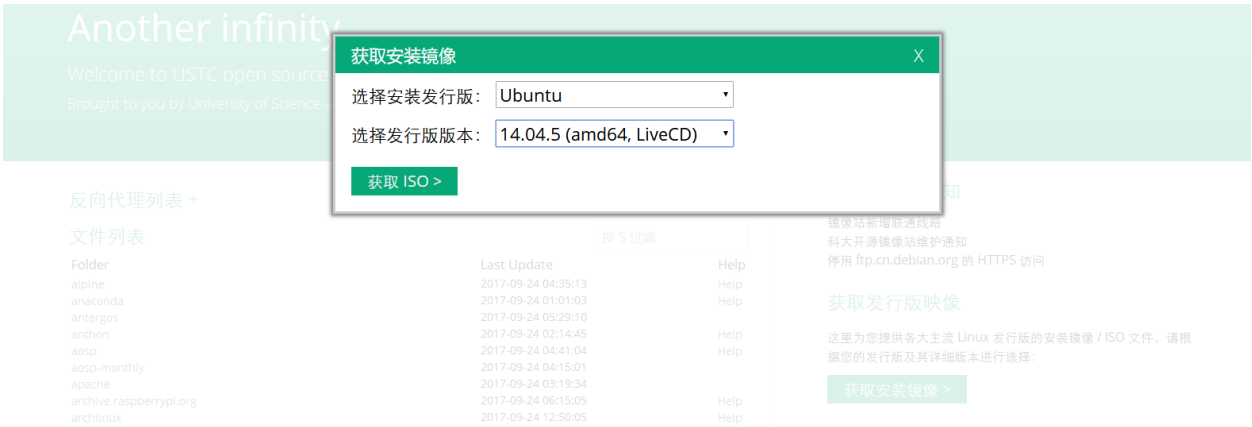
Ubuntu 环境构建

2017.9.24 更新

以下是使用纯净版镜像自行配置环境，非强制实验内容。

针对部分同学机器不支持虚拟化，以及希望自行配置完整环境，现给出独立配置完整环境简明步骤，如有兴趣可按照文档尝试配置。

本教程使用的 linux 版本，可在科大镜像 http://mirrors.ustc.edu.cn/下载。



推荐增加功能：

右键添加终端

C++编程：Kdevelop

输入法：fcitx+搜狗 linux 版

请事先熟悉 vim 的基本操作

华清远见
FARSIGHT
始于2004

0

Ubuntu 安装及配置

用户手册

华清远见教育集团 ● 研发中心


4007061880

farsight2013


http://dev.hqyj.com


support@farsight.com.cn

目 录

第 1 章　系统概述

华清远见所挑选的 Ubuntu 系统是由华清远见研发中心专为培训教学和项目研发选择的适用于嵌入式学习和开发的系统。Ubuntu 是一个以桌面应用为主的 Linux 操作系统，其名称来自非洲南部祖鲁语或豪萨语的“ubuntu”一词（译为友帮拓或乌班图），意思是“人性”、“我的存在是因为大家的存在”，是非洲传统的一种价值观，类似华人社会的“仁爱”思想。Ubuntu 基于 Debian 发行版和 GNOME 桌面环境，与 Debian 的不同在于它每 6 个月会发布一个新版本。Ubuntu 的目标在于为一般用户提供一个最新的、同时又相当稳定的主要由自由软件构建而成的操作系统。Ubuntu 具有庞大的社区力量，用户可以方便地从社区获得帮助。

Ubuntu 的版本号是根据其发布版本的日期而定。版本号由该次发布的年份和月份组成，并未反映其实际版本。Ubuntu 的首次发布(Warty Warthog)是在 2004 年 10 月，因此该版本为 4.10。每六个月发布一个新版本，而每两年发布一个长期支持版本（LTS）。

1.1　Ubuntu12.04 安装介绍

1.1.1　安装过程

打开 VMware，选择新建虚拟机，在“安装程序光盘映像文件”处找到 Ubuntu 12.04 系统 ISO 镜像文件，将其安装在你想要的位置，可在安装过程中对虚拟机的硬件进行配置。

1.1.2　配置过程

在 VMware 中运行 Ubuntu 12.04，等待系统自动执行安装配置等操作，由于 12.04 使用了简单安装的方式，所以该过程无须我们的参与。(请牢记自己的用户名和密码。该密码为 superuser 密码，即我们 sudo 操作需要输入的密码，切勿忘记密码)

Ubuntu14.04 安装介绍

1.1.3 安装过程

打开 VMware，选择新建虚拟机，在“安装程序光盘映像文件”处找到 Ubuntu 14.04 系统 ISO 镜像文件，将其安装在你想要的位置，可在安装过程中对虚拟机的硬件进行配置。

1.1.2 配置过程

在 VMware 中运行 Ubuntu 14.04，等待系统自动执行安装配置等操作，由于 14.04 使用了简单安装的方式，所以该过程无须我们的参与。（请牢记自己的用户名和密码。该密码为 superuser 密码，即我们 sudo 操作需要输入的密码，切勿忘记密码）

第 2 章 vim 操作

2.1 vim 安装与配置

Vim 是一个类似于 Vi 的著名的功能强大、高度可定制的文本编辑器，在 Vi 的基础上改进和增加了很多特性。VIM 是纯粹的自由软件，普遍被推崇为类 Vi 编辑器中最好的一个，它的强大之处在于编辑能力中很大部分是来自于其普通模式命令。

Vim 是我们以后学习和开发过程中所一直使用的重要工具。

2.1.1 vim 的安装

首先我们需要对我们的系统安装 vim，安装命令为：

$ sudo apt-get install vim

我们也把 ctags 和 cscope 装上

$ sudo apt-get install ctags
$ sudo apt-get install cscope

需联网进行该操作

2.1.2 vim 的配置

我们可以使用配置文件对 vim 进行配置工作，使其能具有更多更人性化的功能。

配置步骤：

1) 将我们附带的 vim 配置文件“vim-master.zip”复制入 ubuntu 内
2) 解压该压缩包，解压命令为：***$ unzip vim-master.zip***
3) 进入到刚解压出的文件夹下，运行 setup.sh ： ***$ ./ setup***
4) 安装完成后会提示“安装完成”

该步需要联网完成

第 3 章　在 Ubuntu 中安装 tftp、nfs

tftp 是用来下载远程文件的最简单网络协议，它基于 UDP 协议而实现。嵌入式 linux 的 tftp 开发环境包括两个方面：一是 linux 开发主机端的 tftp-server 支持，二是嵌入式目标系统的 tftp-client 支持。tftp 具有传输速度快、不需要将编译好的内核下载到目标机，提供开发效率的优点。

Ubuntu 上默认是没有安装 NFS 服务器的，首先要安装 NFS。NFS 服务主要的任务是把本地的一个目录通过网络输出，其他计算机可以远程挂接这个目录并进行访问。NFS 有自己的协议和端口，但是在文件传输或者其他相关信息传递的时候，NFS 则使用远程过程调用协（RPC）议。RPC 负责管理端口号的对应与服务相关的工作。NFS 本身的服务并没有传输文件的协议，它通过 RPC 的功能负责。RPC 有 portmap 服务完成。NFS 具有根文件系统和 ap 等都不需要写入到目标机，提供开发效率的优点。

3.1　安装 tftp

安装 tftp 需要网络连接、主机端需要 tftp 服务器软件、目标机需要 tftp 客户端软件支持。

3.1.1 安装检查

检查是否需要安装 tftp sever

检查命令：　***$ dpkg　-s　tftpd-hpa***

3.2　安装 tftp-server

如果未安装，安装 tftp-server

安装命令：　***$ sudo　apt-get　install　tftpd-hpa　tftp-hpa***

3.2.1　开启 tftp 服务

修改文件/etc/default/tftpd-hpa，开启 tftp 服务

打开文件： *$ sudo vi /etc/default/tftpd-hpa*

文件内容修改部分：

```
TFTP_USERNAME="tftp"
TFTP_DIRECTORY="/tftpboot"
TFTP_ADDRESS="0.0.0.0:69"
TFTP_OPTIONS="-c -s -l"
```

```
1 # /etc/default/tftpd-hpa
2
3 TFTP_USERNAME="tftp"
4 TFTP_DIRECTORY="tftpboot"
5 TFTP_ADDRESS="0.0.0.0:69"
6 TFTP_OPTIONS="-c -s -l"
```

3.2.2 启动 tftp-server

创建 **tftpboot** 目录，启动 tftp-server

所需命令：

```
$ sudo mkdir /tftpboot
$ sudo chmod a+w /tftpboot
$ sudo service tftpd-hpa restart
```

```
jcz@ubuntu:~$ sudo service tftpd-hpa restart
tftpd-hpa stop/waiting
tftpd-hpa start/running, process 13969
```

3.2.3 测试

可以先在**/tftpboot** 下创建一个文件： *$ touch text.c*

然后 cd 到其他一个文件夹，创建一个文件： *$ touch text2.c*

在该文件夹下执行命令：

```
$ tftp 127.0.0.1
tftp>get text.c
tftp>put text2.c
tftp>q
```

如成功则当前文件夹和 **tftpboot** 文件夹里都会有 **text.c** 和 **text2.c** 这两个文件

3.3 安装 nfs

安装 NFS 的限制条件是：需要网络连接、主机端需要 nfs 服务器软件支持、目标机同样需要支持 nfs。

3.3.1 安装 NFS

安装命令： *$ sudo apt-get install nfs-kernel-server*

3.3.2 修改 /etc/exports 文件

```
# /etc/exports: the access control list for filesystems which may be exported
#               to NFS clients.  See exports(5).
#
# Example for NFSv2 and NFSv3:
# /srv/homes            hostname1(rw,sync) hostname2(ro,sync)
#
# Example for NFSv4:
# /srv/nfs4             gss/krb5i(rw,sync,fsid=0,crossmnt)
#/srv/nfs4/homes    gss/krb5i(rw,sync)
/source/rootfs    *(rw,sync,no_root_squash,no_subtree_check)
```

格式说明：

共享目录　主机名称或者 IP（参数 1,参数 2,...）

其中共享目录指的是主机上要导出的目录，主机名称或者 IP 是允许按照指定权限访问这个目录的远程主机（如：开发板），参数为各种访问权限

参数说明：

ro　具有只读权限

rw　具有读写权限

no_root_squash　如果客户端是 root 的话，那么他对这个目录具有 root 的权限

root_squash	如果客户端是 root 的话，那么他的权限被限制为匿名使用者
all_squash	不论客户端是什么身份，他的权限都将被限制为匿名使用者
sync	文件同步写入到内存和硬盘
async	文件先写入到内存，而不是直接写入到硬盘

3.3.3 启动 nfs 服务

$ sudo /etc/init.d/nfs-kernel-server restart

```
jcz@ubuntu:/tftpboot$ sudo /etc/init.d/nfs-kernel-server restart
 * Stopping NFS kernel daemon                                              [ OK ]
 * Unexporting directories for NFS kernel daemon...                        [ OK ]
 * Exporting directories for NFS kernel daemon...
exportfs: Failed to stat /source/rootfs: No such file or directory
                                                                           [ OK ]
 * Starting NFS kernel daemon                                             [ OK ]
```

第 4 章　交叉工具链编译安装

交叉编译是在一种平台上编译出能运行在体系结构不同的另一种平台上的程序，比如在 PC 平台（X86 CPU）上编译出能运行在以 ARM 为内核的 CPU 平台上的程序，编译得到的程序在 X86 CPU 平台上是不能运行的，必须放到 ARM CPU 平台上才能运行，虽然两个平台用的都是 Linux 系统。这种方法在异平台移植和嵌入式开发时非常有用。相对与交叉编译，平常做的编译叫本地编译，也 就是在当前平台编译，编译得到的程序也是在本地执行。用来编译这种跨平台程序的编译器就叫交叉编译器，相对来说，用来做本地编译的工具就叫本地编译器。所以要生成在目标机上运行的程序，必须要用交叉编译工具链来完成。在裁减和定制 Linux 内核用于嵌入式系统之前，由于一般嵌入式开发系统存储大小有限，通常都要在性能优越的 PC 上建立一个用于目标机的交叉编译工具链，用该交叉编译工具链在 PC 上编译目标机上要运行的程序。

交叉编译工具链是一个由编译器、连接器和解释器组成的综合开发环境，交叉编译工具链主要由 binutils、gcc 和 glibc 3 个部分组成。

。

4.1　crosstool-ng 编译，安装

4.1.1　解压文件

先将 **crosstool-ng-1.22.0.tar.bz2** 文件放入到我们的 Linux 系统之中，进入到压缩包所在的文件夹，执行解压，这样**/opt** 下会生成一个目录 **crosstool-ng**

解压命令：　***$ sudo tar xjf crosstool-ng-1.22.0.tar.bz2 –C /opt***

（直接解压后就可以，-C 会让解压的文件自动放在根目录下指定路径，不用管）

4.1.2 安装编译必要程序

进入到 **/opt/crosstool-ng/samples** ，可以看到有很多中选择，针对不同架构的，带有不同编译功能、针对的对象不同。本次选择的是 **arm-unknown-Linux-gnueabi**

我们在目录**/opt/crosstool-ng** 下创建两个文件夹，后面会用到。

$ mkdir crosstool-ng_buildl
$ mkdir src

进入到 **/opt/crosstool-ng**

```
jcz@ubuntu:/opt/crosstool-ng$ ls
bootstrap  configure.ac  ct-ng.comp  kconfig     Makefile.in  samples   TODO
config     contrib       ct-ng.in    LICENSES    patches      scripts
configure  COPYING       docs        licenses.d  README.md    steps.mk
```

执行文件中的 **bootstrap** 文件，会提示你没有安装编译必要的程序，所以我们要执行安装编译必要的文件

$ sudo apt-get install autoconf

接下来执行

$ ./configure --prefix=/opt/crosstool-ng/crosstool-ng_install

该操作会在我们的 **/opt/crosstool-ng** 下新建一个 **crosstool-ng_install** 文件夹，这个文件夹用于安装 **crosstool-ng-1.22.0** 这个工具的目录

在配置中会出错，需要安装必要工具

$ sudo apt-get install gperf bison flex texinfo gawk libtool libncurses5-dev

```
configure: error: missing required tool: help2man
```

如果报出以上错误，则需 ***apt-get*** 相应的必要工具，如缺少 **help2man** 则执行***$ sudo apt-get install help2man***

生成配置文件后会显示

```
config.status: creating Makefile
```

然后先执行***$ sudo make*** 操作，接着再执行***$ sudo make install***

```
jcz@ubuntu:/opt/crosstool-ng$ sudo make install
  GEN     'config/configure.in'
  GEN     'paths.mk'
  GEN     'paths.sh'
  INST    'ct-ng'
  MKDIR   '/usr/local/lib/crosstool-ng-1.22.0/'
  INSTDIR 'config/'
  INSTDIR 'contrib/'
  INSTDIR 'patches/'
  INSTDIR 'scripts/'
  INST    'steps.mk'
  INST    'paths'
  INSTDIR 'samples/'
  INST    'kconfig/'
  MKDIR   '/usr/local/share/doc/crosstool-ng/crosstool-ng-1.22.0/'
  INST    'docs/*.txt'
  MKDIR   '/usr/local/share/man/man1/'
  INST    'ct-ng.1.gz'

For auto-completion, do not forget to install 'ct-ng.comp' into
your bash completion directory (usually /etc/bash_completion.d)
```

我们还需要将 ct-ng 命令加入到环境变量中

执行 *$ echo "PATH=$PATH:/opt/crosstool-ng/crosstool-ng_install/bin" >> ~/.bashrc*

然后 *$ source ~/.bashrc*

执行 *$ ct-ng help*

4.1.3 安装测试

安装完成，测试一下***$ ct-ng***，成功安装

```
jcz@ubuntu:/opt/crosstool-ng$ ct-ng
This is crosstool-NG version crosstool-ng-1.22.0

Copyright (C) 2008  Yann E. MORIN <yann.morin.1998@free.fr>
This is free software; see the source for copying conditions.
There is NO warranty; not even for MERCHANTABILITY or FITNESS FOR A
PARTICULAR PURPOSE.

See below for a list of available actions, listed by category:

Configuration actions:
  menuconfig          - Update current config using a menu based program
```

4.2 生成交叉编译工具

4.2.1 修改 config 文件配置

根据要建立的工具链，拷贝 **sample/arm-unknown-linux-gnueabi** 到 **build**

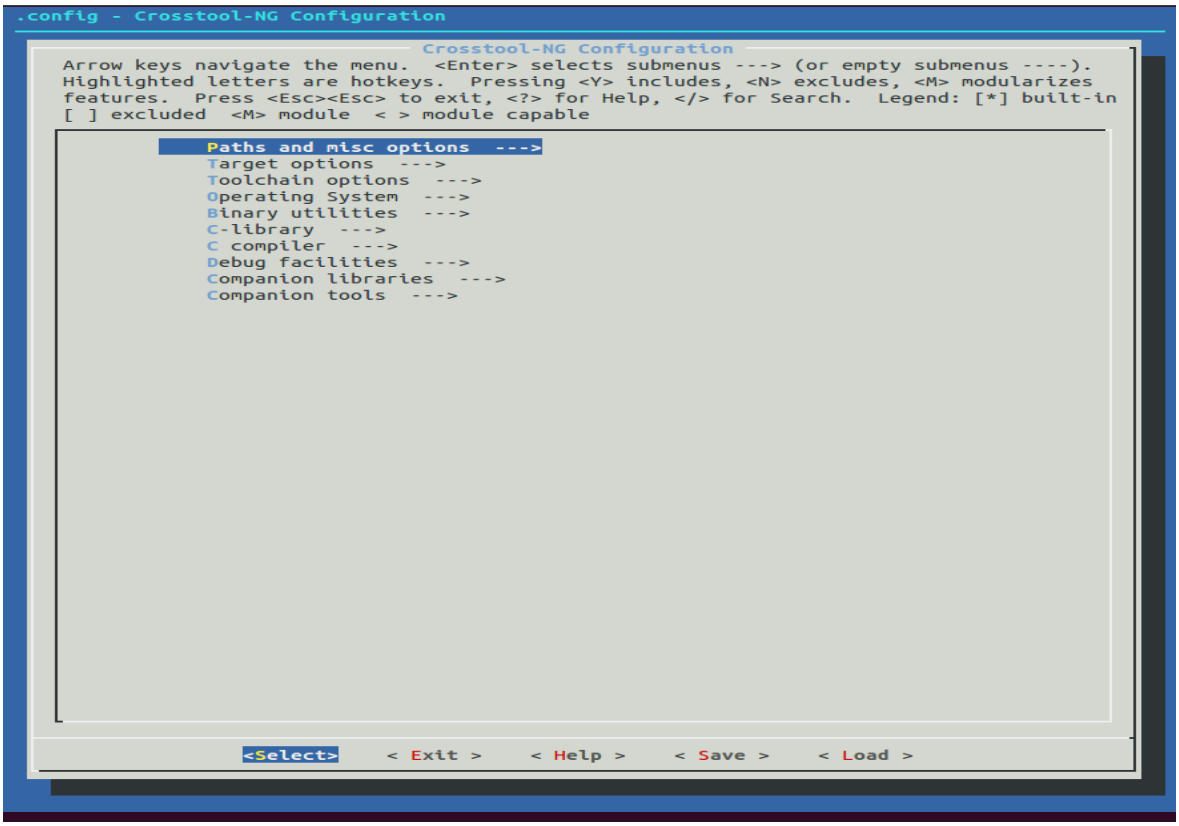
$ cp –Rf /opt/crosstool-ng/sample/arm-unknown-linux-gnueabi /opt/crosstool-ng/crosstool-ng_build

将 bulid 里面文件 crosstool.config 的文件名修改成 .config

$ mv crosstool.config .config

在文件 config 所在目录内执行*$ ct-ng menuconfig*

出现配置画面，进行配置修改（关键）



需要修改以下几项：

1) 已下载源码包存放路径和交叉编译器的安装路径

Paths and misc options --->

***** Paths *****

(/opt/crosstool-ng/src) Local tarballs directory

此为保存源码包路径

2) 增加编译时的并行进程数，增加运行效率，加快编译

Paths and misc options --->

***** Build behavior *****

(2) Number of parallel jobs

括号内数字为选择的并行进程数，根据自己机器情况选择，可保持原配置

3) 修改处理器核心配置（谨慎操作）

Target options --->

Target Architecture(arm) 选择成 arm 体系

***** Target optimisations *****

(armv7) Architecture level 指令集的架构

(cortex-a9) Emit assembly for CPU CPU 的核心类型

（支持架构，和核心类型可以用 man gcc 来查询）

4) 修改 Linux 内核版本

OPERATING system -->

Linux kernel version() -->

设置为 2.6.32.68，这是因为我们的板子一般内核版本都不会太高，考虑到兼容性问题，所以我们选低内核版本。

5) 一些个性化的修改（可以不修改）

Toolchain options --->

***** Tuple completion and aliasing *****

(unknown) Tuple's vendor string

原来的的编译器前缀就是：**arm-unknown-linux-gnueabi-** 我们可以把这内容删去，从而精简编译器前缀长度

4.2.2 编译工具链

crosstool-ng 会根据你所设置的源码版本下载相应的源码包，如果你想加速编译过程，你可以根据配置的源码包版本自行下载，然后放到你指定的源码存放路径。

执行命令***$ ct-ng build***

等待编译完成，crosstool-ng 会自动下载相应的包，内核文件，然后解压，配置，编译，链接。

```
[INFO ]  ============================================================
[INFO ]  Installing native gdb
[EXTRA]    Configuring native gdb
[EXTRA]    Building native gdb
[EXTRA]    Installing native gdb
[INFO ]  Installing native gdb: done in 214.41s (at 99:44)
[INFO ]  ============================================================
[INFO ]  Installing gdbserver
[EXTRA]    Configuring gdbserver
[EXTRA]    Building gdbserver
[EXTRA]    Installing gdbserver
[INFO ]  Installing gdbserver: done in 76.01s (at 101:00)
[INFO ]  ============================================================
[INFO ]  Installing ltrace
[EXTRA]    Copying sources to build dir
[EXTRA]    Configuring ltrace
[EXTRA]    Building ltrace
[EXTRA]    Installing ltrace
[INFO ]  Installing ltrace: done in 29.55s (at 101:29)
[INFO ]  ============================================================
[INFO ]  Installing strace
[EXTRA]    Configuring strace
[EXTRA]    Building strace
[EXTRA]    Installing strace
[INFO ]  Installing strace: done in 51.25s (at 102:20)
[INFO ]  ============================================================
[INFO ]  Cleaning-up the toolchain's directory
[INFO ]    Stripping all toolchain executables
[EXTRA]    Installing the populate helper
[EXTRA]    Installing a cross-ldd helper
[EXTRA]    Creating toolchain aliases
[EXTRA]    Removing access to the build system tools
[INFO ]  Cleaning-up the toolchain's directory: done in 4.28s (at 102:25)
[INFO ]  Build completed at 20161024.013407
[INFO ]  (elapsed: 102:23.96)
[INFO ]  Finishing installation (may take a few seconds)...
[102:25] / jcz@ubuntu:/opt/crosstool-ng/crosstool-ng_build/arm-unknown-linux-gnueabi$
```

完成后*$ cd /home/jcz/x-tools/arm-unknown-linux-gnueabi/bin* 可以看到

```
jcz@ubuntu:~/x-tools/arm-unknown-linux-gnueabi/bin$ ls
arm-unknown-linux-gnueabi-addr2line     arm-unknown-linux-gnueabi-gcov-tool
arm-unknown-linux-gnueabi-ar            arm-unknown-linux-gnueabi-gdb
arm-unknown-linux-gnueabi-as            arm-unknown-linux-gnueabi-gprof
arm-unknown-linux-gnueabi-c++           arm-unknown-linux-gnueabi-ld
arm-unknown-linux-gnueabi-cc            arm-unknown-linux-gnueabi-ld.bfd
arm-unknown-linux-gnueabi-c++filt       arm-unknown-linux-gnueabi-ldd
arm-unknown-linux-gnueabi-cpp           arm-unknown-linux-gnueabi-ld.gold
arm-unknown-linux-gnueabi-ct-ng.config  arm-unknown-linux-gnueabi-nm
arm-unknown-linux-gnueabi-dwp           arm-unknown-linux-gnueabi-objcopy
arm-unknown-linux-gnueabi-elfedit       arm-unknown-linux-gnueabi-objdump
arm-unknown-linux-gnueabi-g++           arm-unknown-linux-gnueabi-populate
arm-unknown-linux-gnueabi-gcc           arm-unknown-linux-gnueabi-ranlib
arm-unknown-linux-gnueabi-gcc-5.2.0     arm-unknown-linux-gnueabi-readelf
arm-unknown-linux-gnueabi-gcc-ar        arm-unknown-linux-gnueabi-size
arm-unknown-linux-gnueabi-gcc-nm        arm-unknown-linux-gnueabi-strings
arm-unknown-linux-gnueabi-gcc-ranlib    arm-unknown-linux-gnueabi-strip
arm-unknown-linux-gnueabi-gcov
```

编译过程中可能会出现缺少一些必须工具而导致编译失败的错误，我们可以用 ***apt-get*** 的命令去下载缺失的工具；也可能因为网络问题而出现由代码源包下载失败导致得编译失败，我们可以通过手动下载相应的包然后将其放入**/opt/ccrosstool-ng/src** 文件夹中来解决问题，如果链接失效可在我们提供的工具链代码源包内寻找

各包下载地址（x 为版本号，每个 x 代表一个数字）：

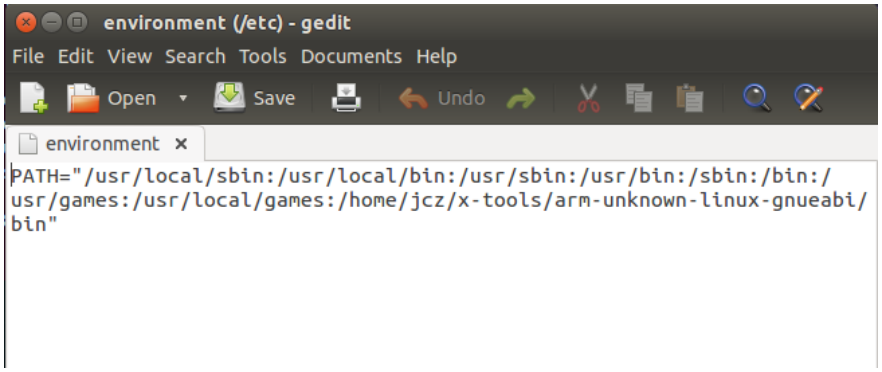
linux-x.xx.x__http://www.kernel.org/pub/linux/kernel/v2.6/
gmp-x.x.x__http://ftp.gnu.org/gnu/gmp/

mpfr-x.x.x__http://ftp.gnu.org/gnu/mpfr/
ppl-x.xx.x__http://www.cs.unipr.it/ppl/Download/ftp/releases/
cloog-ppl-x.xx.x http://gcc-uk.internet.bs/infrastructure/
libelf-x.x.xx http://www.mr511.de/software/
binutils-x.xx.x http://ftp.gnu.org/gnu/binutils/
gcc-x.x.x http://ftp.gnu.org/gnu/gcc/
uClibc-x.x.xx.x http://www.uclibc.org/downloads/?ref=darwinports.com
dmalloc-x.x.x http://dmalloc.com/releases/
duma_x_x_xx http://sourceforge.NET/projects/duma/files/duma/
gdb-x.x http://ftp.gnu.org/gnu/gdb/
ncurses-x.x http://ftp.gnu.org/pub/gnu/ncurses/
expat-x.x.x http://sourceforge.net/projects/expat/files/expat/
ltrace-x.x.x ftp://ftp.debian.org/debian/pool/main/l/ltrace/
strace-x.x.xx http://sourceforge.net/projects/strace/files/strace/

4.2.3 建立软连接

建立一些软链接，建立环境变量，使我们在其他目录下可以不用键入绝对路径而使用命令

1) ***$sudo gedit /etc/environment***，在其中引号内部的末尾添加
: /home/jcz/x-tools/arm-unknown-linux-gnueabi/bin



冒号后面的其实是你的工具路径，也就是你编译出来的工具链的路径，需要根据你自己的实际情况进行编辑

2) ***$ source /etc/ environment*** 使修改生效
3) 检查是否将路径加入到 PATH： ***$ echo $PATH***
显示内容
/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin:/usr/games:/usr/local/games:/home/jcz/x-tools/arm-unknown-linux-gnueabi/bin:/home/jcz/x-tools/arm-unknown-linux-gnueabi/bin
说明已经将交叉编译器的路径加入 PATH。

```
jcz@ubuntu:~$ echo $PATH
/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin:/usr/games:/usr/loc
al/games:/home/jcz/x-tools/arm-unknown-linux-gnueabi/bin:/home/jcz/x-tools/arm-u
nknown-linux-gnueabi/bin
```

4) 新开一个终端，输入***$ arm-unknown-linux-gnueabi-gcc -v*** 会有如下信息，如果没有信息，则重启系统后再试

```
jcz@ubuntu:~$ arm-unknown-linux-gnueabi-gcc -v
Using built-in specs.
COLLECT_GCC=arm-unknown-linux-gnueabi-gcc
COLLECT_LTO_WRAPPER=/home/jcz/x-tools/arm-unknown-linux-gnueabi/libexec/gcc/arm-
unknown-linux-gnueabi/5.2.0/lto-wrapper
Target: arm-unknown-linux-gnueabi
Configured with: /opt/crosstool-ng/crosstool-ng_build/arm-unknown-linux-gnueabi/
.build/src/gcc-5.2.0/configure --build=x86_64-build_pc-linux-gnu --host=x86_64-b
uild_pc-linux-gnu --target=arm-unknown-linux-gnueabi --prefix=/home/jcz/x-tools/
arm-unknown-linux-gnueabi --with-sysroot=/home/jcz/x-tools/arm-unknown-linux-gnu
eabi/arm-unknown-linux-gnueabi/sysroot --enable-languages=c,c++ --with-cpu=corte
x-a9 --with-float=soft --with-pkgversion='crosstool-NG crosstool-ng-1.22.0' --di
sable-sjlj-exceptions --enable-__cxa_atexit --disable-libmudflap --disable-libgo
mp --disable-libssp --disable-libquadmath --disable-libquadmath-support --disabl
e-libsanitizer --with-gmp=/opt/crosstool-ng/crosstool-ng_build/arm-unknown-linux
-gnueabi/.build/arm-unknown-linux-gnueabi/buildtools --with-mpfr=/opt/crosstool-
ng/crosstool-ng_build/arm-unknown-linux-gnueabi/.build/arm-unknown-linux-gnueabi
/buildtools --with-mpc=/opt/crosstool-ng/crosstool-ng_build/arm-unknown-linux-gn
ueabi/.build/arm-unknown-linux-gnueabi/buildtools --with-isl=/opt/crosstool-ng/c
rosstool-ng_build/arm-unknown-linux-gnueabi/.build/arm-unknown-linux-gnueabi/bui
ldtools --with-cloog=/opt/crosstool-ng/crosstool-ng_build/arm-unknown-linux-gnue
abi/.build/arm-unknown-linux-gnueabi/buildtools --with-libelf=/opt/crosstool-ng/
crosstool-ng_build/arm-unknown-linux-gnueabi/.build/arm-unknown-linux-gnueabi/bu
ildtools --enable-lto --with-host-libstdcxx='-static-libgcc -Wl,-Bstatic,-lstdc+
+,-Bdynamic -lm' --enable-threads=posix --enable-target-optspace --enable-plugin
 --enable-gold --disable-nls --disable-multilib --with-local-prefix=/home/jcz/x-
tools/arm-unknown-linux-gnueabi/arm-unknown-linux-gnueabi/sysroot --enable-long-
long
Thread model: posix
gcc version 5.2.0 (crosstool-NG crosstool-ng-1.22.0)
```

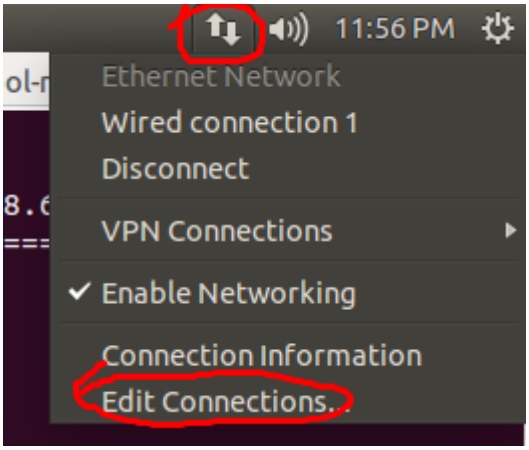
第 5 章　网络配置---图形界面配置

进行开发工作时经常需要手动进行网络配置，Linux 桌面端为我们提供了 Network-Manager 这个带界面的软件，方便我们进行操作。

5.1　Network-Manager 使用介绍

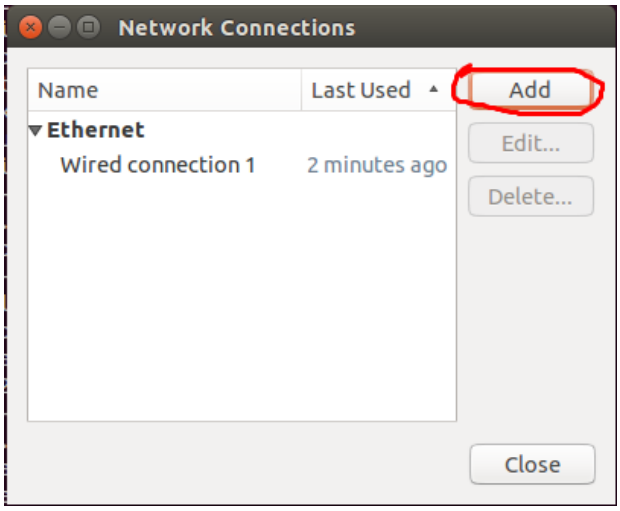
5.1.1　找到 Network-Manager

我们可以通过点击右上角的信号箭头图标，选择 Edit Connections 来进入到 Network-Manager 界面

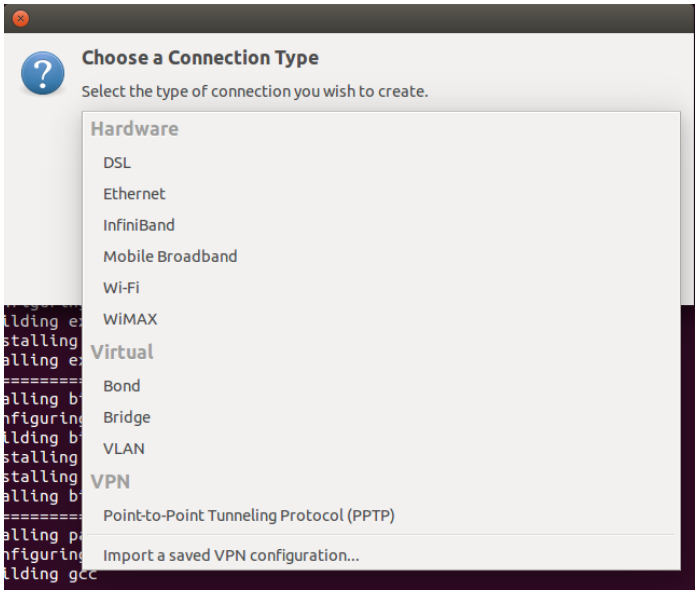


5.1.2　配置 Network-Manager

1)　点击 add 添加一个网络连接



2) 根据当前状态选择网络制式



3) 进入到 IPv4 选项，修改连接名称（选做），方法选择手动，点击添加，输入适当参数，输入 DNS 服务器地址，点击保存完成操作