建立 Linux 交叉编译工具链（S3C2440）

Author: Guotongbin

Nickname: Panda.Guo

Create: 2010.12.10

Change log:

- 2011.11.01. -> 重新整理文档，并在 ubuntu10.04 下测试.
- 2011-12-07. -> 更新 crosstool-ng 脚本重新制作交叉编译器。

简介

用 crosstool-ng 建立 Linux 交叉编译环境（以 S3C2440（armv4t）为例）

之前用 crosstool 来创建交叉编译工具链，网址 http://www.kegel.com/crosstool，只是由于 crosstool，只支持 gcc-4.1.1 版本。而 Linux-2.6.29 之后的内核版本需要更高版本的 gcc 支持。

本次使用 crosstool-ng 这个脚本，来生成新的交叉编译工具链。以下是操作笔记：

主机环境

- Ubuntu: Ubuntu 10.04
- Linux: Linux 2.6.32-27
- GCC: gcc 4.4.3

基本概念

要做嵌入式开发，那你需要首先建立你的开发环境，首先需要建立交叉编译器，那么什么是交叉编译器呢：

- 交叉编译器的概念

所谓交叉编译器就是编译器本身运行在一种平台上， 而编译生成的代码的运行在另外一种平台上。举个例子： arm-linux-gcc, 本身运行在 X86 平台上， 而编译生成的 a.out, 运行在 arm 平台上，仅此而已。

不管交叉编译器完成什么功能，它总是个程序。他也是由源码经过编译而生成的。 编译生成交叉编译器的过程， 我们称之为制作交叉编译器。

制作交叉编译器，首先要弄清楚 3 个概念：host, build, target：

- build — 你在什么平台上建立的这个交叉编译器
- host — 这个交叉编译器将来要在什么平台上运行
- target — 交叉编译器最终会生成在哪个平台上执行的可执行代码

这里我可以给个例子 build=i386 host=mips target=arm 表示我们在 x86 平台上编译了一个在 mips 平台上运行的编译器，它将源码编译生成 a.out, 需要在 arm 平台上运行。

- 制作交叉编译器的基本流程

  通过编译源码，制作交叉编译器的基本步骤如下：

  - 编译 binutils：生成 二进制工具
  - 编译 bootstrap 的 gcc：这个也比较容易，这一步是为编译 glibc 准备的
  - 编译 glibc：这个最容易出错，因为你需要为它准备 kernel 的头文件，而且有时候需要打 patch
  - 重新编译 gcc：告诉配置文件刚才编译的 glibc 的位置，这样在最终编译一个能够自动找到库和头文件的交叉编译器。

安装相关包

在交叉编译器的制作过程中，还需要如下的安装包来支持。故在需要首先安装如下包：

- texinfo: makeinfo
- aotumake: aotumake
- g++
- libncurses5-dev
- bison
- flex
- libtool
- patch
- gcj (好像是支持 Java 的，如果不需要 Java 可以不用)
- cvs
- cvsd
- gawk
- Panda.Guo $ sudo apt-get install sed bash dpkg-dev bison flex patch texinfo automake m4 libtool websvn tar gzip bzip2 lzma libncurses5-dev bison flex texinfo automake libtool patch gcj cvs cvsd gawk -y curl

  其实，在下面的 crosstool-ng 脚本配置（ ./configure ）时，会自动检查编译所需的环境。到时提示缺什么就安装 什么即可，为此，我也书写了一个脚本 [install.sh], 用来安装相关包。 如下命令

- Panda.Guo $./install.sh

---

crosstool-ng 脚本下载、配置、安装

crosstool-ng 是新的用来建立交叉工具链的脚本工具，它是 crosstool 的升级替代者.是基于 crosstool 写出来的，使用 crosstool 最多只能编译 gcc 4.1.1　glibc 2.x 的版本，而使用低于 gcc4.1.1 版本的 交叉编译器编译 LinuxKernel2.6.29 版本的内核时，有可能报编译器版本的低等相关错误。 而 crosstool-ng 一直保持着更新,现在能够建立最新的 gcc 版本和 glibc 版本...

脚本下载：

crosstool-ng 的下载地址是：http://ymorin.is-a-geek.org/download/crosstool-ng/

值得注意的是，下载 crosstool-ng 以后，记得在到 http://ymorin.is-a-geek.org/download/crosstool-ng/01-fixes/ 看看有没有相应的补丁，有得话一起下载下来。

这次我选用的是 crosstool-ng1.12.3 版本， 截止到目前（2011-11-01）坦白的说，版本已经有很多了。 不过我也没有兴趣，追求最新的了。本着够用就可的原则。

关于补丁，在你创建的时候，需要根据你当时的情况，去看看还有没有补丁。到目前为止，有四个补丁。

在下载之前呢，我们先创建以下我们的工作目录

在用户目录下创建工作目录

- Panda.Guo $ cd
- 目录： < ~ />
- Panda.Guo $ mkdir crosstool
- Panda.Guo $ cd crosstool/
- 目录： < ~ /crosstool>
- Panda.Guo $mkdir build install

  build 目录作为我们的未来创建交叉工具链的工作目录，

  install 目录， 作为我们安装 crosstool-ng 的目录

  下载脚本，
- 目录： < ~ /crosstool>

- Panda.Guo $ wget -c http://crosstool-ng.org/download/crosstool-ng/crosstool-ng-1.12.3.tar.bz2

  下载补丁：下载时，看是否有更新补丁
- Panda.Guo $ wget -c http://crosstool-ng.org/download/crosstool-ng/01-fixes/1.12.3/000-Pass_CXXFLAGS_to_binutils_gold_.patch
- Panda.Guo $ wget -c http://crosstool-ng.org/download/crosstool-ng/01-fixes/1.12.3/001-binutils_binutils_ensure_gold_is_staticaly_linked_if_needed.patch
- Panda.Guo $ wget -c http://crosstool-ng.org/download/crosstool-ng/01-

fixes/1.12.3/002-complibs_cloog_catch_autogen_sh_s_output.patch

- Panda.Guo $ wget -c http://crosstool-ng.org/download/crosstool-ng/01-fixes/1.12.3/003-libc_glibc_run_configure_in_CONFIG_SHELL.patch

---

打补丁

crosstool-ng 和 crosstool 不同的地方之一就是：她并不是一下载下来就可以使用了，必须先配置、安装。

接下来我们，解包、配置、和安装 crosstool-ng.

- 目录： < ~ /crosstool>
- Panda.Guo $ tar xvf crosstool-ng-1.12.3.tar.bz2
- Panda.Guo $ cd crosstool-ng-1.12.3/
- 目录： < ~ /crosstool/crosstool-ng-1.12.3>
- Panda.Guo $ patch -p1 < ../000-Pass_CXXFLAGS_to_binutils_gold_.patch
- Panda.Guo $ patch -p1 < ../001-binutils_binutils_ensure_gold_is_staticaly_linked_if_needed.patch
- Panda.Guo $ patch -p1 < ../002-complibs_cloog_catch_autogen_sh_s_output.patch
- Panda.Guo $ patch -p1 < ../003-libc_glibc_run_configure_in_CONFIG_SHELL.patch

---

配置、安装

接下来，对已经打好补丁的 crosstool-ng 进行配置、安装

- 目录： < ~ /crosstool/crosstool-ng-1.12.3>
- Panda.Guo $./configure - -prefix=../install

```
Computing version string... 1.13.0
configure: 'BINDIR' is not an absolute path: '../install/bin'
```

configure 时，指定相对路径则出错：应该指定绝对路径

- Panda.Guo $ ./configure - -prefix=/home/panda/crosstool/install

  - - prefix 指定 ct-ng 的安装目录，

  在这个配置过程中，会检查当前主机的相关包的安装情况， 如果上述包没有安装，则配置不过去。安装即可

  配置成功后，编译、安装即可

- Panda.Guo $ make
- Panda.Guo $ make install

配置、编译交叉工具链

crosstool-ng 安装好之后。接下来就可以配置交叉工具链，并编译之

在 crosstool-ng 已很多已经做好的默认配置（位于 crosstool-ng-X.Y.Z/samples 目录下），你只要针对 其进行修改就好了。对于编译器组件部分的版本最好不要修改，因为那个配搭应该是经过测试后的最高本版了， 但内核版本可以修改。

- 目录： <~/crosstool/crosstool-ng-1.12.3>
- Panda.Guo $ cd ../build
- 目录： <~/crosstool/build>
- mkdir src tools

src 目录是下载的源码包（用于生成库的）的路径，

tools 是未来制作好的交叉工具链的安装路径

首先，拷贝 crosstool-ng 做好的默认配置:

- Panda.Guo $ cp ../crosstool-ng-1.12.3/samples/arm-unknown-linux-gnueabi/* ./
- Panda.Guo $ mv crosstool.config .config
- Panda.Guo $ ../install/bin/ct-ng menuconfig

进入 menuconfig，开始修改配置。这次主要针对 S3C2440(armv4t)。

1、已下载好的源码包路径和交叉编译器的安装路径。

```
Paths and misc options  --->
   *** Paths ***
  ($PWD/src) Local tarballs directory          （源码包目录）
  ($PWD/.build) Working directory      （编译过程中，工作目录）
  ($PWD/tools) Prefix directory                （安装路径）
```

2、增加编译时的并行进程数，以增加运行效率，加快编译。

```
Paths and misc options  --->
  *** Build behavior ***
  (4) Number of parallel jobs                        （并行进程数目）
```

这个数值不宜过大，应该为 CPU 数量的两倍。由于我的 CPU 是双核的，所以我填了 4.

3、修改交叉编译器针对的构架

```
Target options  --->
 *** Target optimisations ***
 (armv4t) Architecture level
 (arm9tdmi) Emit assembly for CPU
 (arm920t) Tune for CPU
```

以上这几个参数，你可以参考，s3c2440 手册。

4、 添加编译器标识，和定义符号链接

```
Toolchain options  --->
 *** Tuple completion and aliasing ***
 (akae) Tuple's vendor string          （arm-akae/linux-gnueabi-gcc）
 (arm-linux) Tuple's alias             （生成: arm-linux-gcc 符号链接）
```

上述更改后，产生的编译器前缀就是：arm-akae-linux-gnueabi-gcc...， 并且在同一目录下， 自动生成 arm-linux-gcc ... 的符号链接

5、更改 Linux 内核版本，（不改也行，反正之后需要调整.config 文件中的内核版本）

```
Operating System  --->
  Linux kernel version (2.6.33.16)  --->    （默认 3.0，太新了）
```

不改也行，反正我们用的是 2.6.33.7， 稍候需要更改配置文件，这样改的话，待会少替换几个数字 ^_^。

6、更改二进制工具（binutils）版本：

```
Binary utilities  --->
  *** GNU binutils ***
  binutils version (2.19.1a)  --->   （默认 2.20.1a）
```

7、 更改标准 C 库与内核的版本匹配

```
C-library  --->
  Minimum supported kernel version (Same as kernel headers (default)) --->
  (X) Specific kernel version   （选择指定内核版本）
```

更改为如下:

```
  Minimum supported kernel version (Specific kernel version)  --->
    Specific kernel version
  (2.6.16) Minimum kernel version to support
```

如果你未来编译的内核版本比当前高，或者持平，不改也罢。否则建议你降低内核号。

其他不需要调整，建议你退出前，对其他包的版本看一下，看是否和我提供的源码包一致。 防止脚本更新带来的新问题。

检查完毕后，保存退出即可

8、更改当前目录下的 .config 文件。有如下几个内核版本号需要调整：

```
CT_KERNEL_VERSION="2.6.33.7"
CT_KERNEL_V_2_6_33_7=y
```

如果再次../install/bin/ct-ng menuconfig，这个修改又会复原，必须再次手工修改。

你也可以选择修改 build/config/kernel/linux.in 等文件，只是比较麻烦，但这可以彻底解决， 实现在界面中增加内核版本。

配置完后， 执行下载源码包，编译

- 目录： <~/crosstool/build>
- Panda.Guo $ ../install/bin/ct-ng build

当然，在之前可以拷贝我给的源码，到 ~/crosstool/build/src 目录下，如果你的 网络好的话，也可以自己下载。

编译好后的交叉编译器。

编译器位于：~/crosstool/build/tools/bin

库文件位于：~/crosstool/build/tools/arm-akae-linux-gnueabi/lib

源码包位于： ~/crosstool/build/src
