OpenCV 移植到 ARM 全过程

Host:VMware + Ubuntu 10.04
Target:Real6410 with Linux 2.6.28.6
Crossing Compiler:arm-none-linux-gnueabi-4.3.2 with EABI

一 交叉编译链的安装
1 解压，即可得到 arm-none-linux-gnueabi 目录
tar –xjvf arm-none-linux-gnueabi-4.3.2.tar.bz2
2 为了使用方便,
还可以编辑/etc/bash.bashrc 文件添加把编译器路径到环境变量 PATH 中，
只要在这个文件中添加下面这 2 个语句即可:
PATH=/root/ arm-none-linux-gnueabi-4.1.0/bin:$PATH
export PATH
3 编辑完毕后使用 source /etc/bash.bashrc 命令执行以下这个文件,让设置生效,之后再输入:
arm-none-linux-gnueabi-gcc -v
如果输出下面的信息则表面设置成功:
Using built-in specs.
Target: arm-none-linux-gnueabi
Configured with: /scratch/julian/lite-respin/linux/src/gcc-4.3/configure --build=i686-pc-linux-
gnu --host=i686-pc-linux-gnu --target=arm-none-linux-gnueabi --enable-threads --disable-libmudflap --disable-libssp --disable-libstdcxx-pch --with-gnu-as --with-gnu-ld --enable-languages=c,c++ --enable-shared --enable-symvers=gnu --enable-__cxa_atexit --with-pkgversion='Sourcery G++ Lite 2008q3-72' --with-bugurl=https://support.codesourcery.com/GNUToolchain/ --disable-nls --prefix=/opt/codesourcery
--with-sysroot=/opt/codesourcery/arm-none-linux-gnueabi/libc --with-build-sysroot=/scratch/julian/lite-respin/linux/install/arm-none-linux-gnueabi/libc --with-gmp=/scratch/julian/lite-respin/linux/obj/host-libs-2008q3-72-arm-none-linux-gnueabi-i686-pc-
linux-gnu/usr --with-mpfr=/scratch/julian/lite-respin/linux/obj/host-libs-2008q3-72-arm-none-
linux-gnueabi-i686-pc-linux-gnu/usr --disable-libgomp --enable-poison-system-directories --with-
build-time-tools=/scratch/julian/lite-respin/linux/install/arm-none-linux-gnueabi/ bin --with-build-
time-tools=/scratch/julian/lite-respin/linux/install/arm-none-linux-gnueabi/bin
Thread model: posix

gcc version 4.3.2 (Sourcery G++ Lite 2008q3-72)
至此交叉编译链安装完成。

二 交叉编译 libjpeg
为了使 OpenCV 能处理 jpeg 图像，我们必须事先交叉编译好 libjpeg
这里使用的版本是 jpegsrc.v6b
1 下载 libjpeg 源码：ftp://ftp.uu.net/graphics/jpeg/jpegsrc.v6b.tar.gz

2 解压进入目录
3 配置
#./configure --prefix=/root/libjpeg-arm --exec-prefix=/root/libjpeg-arm --enable-shared --enable-static

下面分别介绍这几个参数的作用:

--prefix=/root/libjpeg-arm : 执行 make install 后,会将与体系无关的文件拷贝到此目录下,具体如下:

```
/root/libjpeg-arm.....................................
      |
      +---include.....................................
          |
          ---jconfig.h
          |
          ---jerror.h
          |
          ---jmorecfg.h
          |
          ---jpeglib.h
      +---man.....................................
          |
          +---man1.....................................
              |
              ---cjeg.1
              |
              ---djpeg.1
              |
              ---jpegtran.1
              |
              ---rdjpgcom.1
              |
              ---wrjpgcom.1
```

--exec-prefix=/root/libjpeg-arm : 执行 make install 后,会将与体系无关的文件拷贝到此目录下,即将一些可执行程序、动态链接库和静态链接库拷贝到此目录的相应目录下，具体如下：

```
/root/libjpeg-arm.....................................
      |
      +---bin.....................................
          |
          ---cjeg
          |
          ---djpeg
          |
          ---jpegtran
          |
          ---rdjpgcom
          |
          ---wrjpgcom
      +---lib.....................................
          |
          ---libjpeg.la
          |
          ---libjpeg.so
          |
          ---libjpeg.so.62
          |
          ---libjpeg.so.62.0.0
```

--enable-shared : 用 GNU libtool 编译成动态链接库 。

4 修改生成的 Makefile 文件:

The name of your C compiler:

CC= gcc 该成 CC= /root/arm-none-linux-gnueabi/bin/arm-none-linux-gnueabi-gcc (根据你自己交叉编译器的位置修改)

library (.a) file creation command

AR= ar rc 该成 AR= /root/arm-none-linux-gnueabi/bin/arm-none-linux-gnueabi-ar rc (同上)

second step in .a creation (use "touch" if not needed)

AR2= ranlib 该成 AR2= /root/arm-none-linux-gnueabi/bin/arm-none-linux-gnueabi-ranlib (同上)

5 在/root/libjpeg-arm 目录下建立 man/man1,include,lib,bin 四个目录

6 # make

make install

7 将/root/libjpeg-arm/include/中（jconfig.h, jerror.h, jmorecfg.h, jpeglib.h）四个头文件拷贝到：/root/arm-none-linux-gnueabi/arm-none-linux-gnueabi/include 中。

将/root/libjpeg-arm/lib 中（libjpeg.la, libjpeg.so, libjpeg.so.62, libjpeg.so.62.0.0)四个库文件拷贝到：/root/arm-none-linux-gnueabi/arm-none-linux-gnueabi/lib 中

注意：执行以下命令检查生成的 libjpeg.so 是否为 ARM 版：

file libjpeg.so
以下为正确输出，否则检查交叉编译器路径以及 Makefile 并重新编译。
注意：执行完以上操作后执行以下命令检查库文件是否已正确安装：
arm-linux-gcc -print-file-name=libjpeg.so
如果输出为"libjpeg.so"则说明没有正确安装，重复 7 步骤。
如果输出为"DIR/libjpeg.so"则说明安装正确。
至此 libjpeg 交叉编译完成。

三 交叉编译 x264,xvid,ffmpeg
为了使 OpenCV 能处理视频，我们要事先交叉编译 ffmpeg,而 ffmpeg 又是依赖 x264 和 xvid 的。
1 下载 yasm：
到 http://www.tortall.net/projects/yasm/wiki/Download 下载 yasm0.7.2（x264 需要用到的汇编编译器）
./configure --enable-shared --prefix=/root/arm-none-linux-gnueabi/arm-none-linux-gnueabi/ --host=arm-linux

make
make install

2 交叉编译 x264
到 ftp://ftp.videolan.org/pub/videolan/x264/snapshots/ 下载 x264-snapshot-20060805-2245.tar.bz2,解压进入目录
(1).配置
./configure --prefix=/root/arm-none-linux-gnueabi/arm-none-linux-gnueabi/ --enable-shared

(2).修改配置参数
修改 config.mak：
prefix=/root/arm-none-linux-gnueabi/arm-none-linux-gnueabi/
exec_prefix=${prefix}
bindir=${exec_prefix}/bin
libdir=${exec_prefix}/lib
includedir=${prefix}/include
# 这里改为 ARM
ARCH=ARM
SYS=LINUX
# 这里改为 arm-linux-gcc
CC=arm-linux-gcc
# 这里去掉-DHAVE_MMXEXT -DHAVE_SSE2 -DARCH_X86
CFLAGS=-Wall -I. -O4 -ffast-math -D__X264__ -DHAVE_MALLOC_H -DSYS_LINUX -DHAVE_PTHREAD -s -fomit-frame-pointer
LDFLAGS= -lm -lpthread -s
AS=nasm

```
ASFLAGS=-O2 -f elf
VFW=no
GTK=no
EXE=
VIS=no
HAVE_GETOPT_LONG=1
DEVNULL=/dev/null
CONFIGURE_ARGS= '--enable-shared' '--prefix=/root/arm-none-linux-gnueabi/arm-none-linux-gnueabi/'
SONAME=libx264.so.49
default: $(SONAME)
```

修改 Makefile，将 66~68 行的 ar 和 ranlib 改为 arm 下的：

```
libx264.a: .depend $(OBJS) $(OBJASM)
        arm-linux-ar rc libx264.a $(OBJS) $(OBJASM)
        arm-linux-ranlib libx264.a
```

(3).编译安装

```
# make
# make install
```

这里可以在交叉编译链目录/root/arm-none-linux-gnueabi/arm-none-linux-gnueabi/的 lib 下生成 libx264.so

---

3 交叉编译 xvid

到 http://downloads.xvid.org/downloads/xvidcore-1.1.3.tar.gz 下载 xvid

下载 xvid 解压并进入 build/generic

配置

```
# ./configure --prefix=/root/arm-none-linux-gnueabi/arm-none-linux-gnueabi/ --disable-assembly
```

[解释]--disable-assembly :因为 xvid 没有针对 ARM 的汇编优化，所以编译时必须关掉汇编

修改 Makefile 引用的 platform.inc 文件，将 CC=gcc 改为 CC=arm-linux-gcc

```
# make
# make install
```

成功后进入 example 文件夹

测试，输入

```
arm-linux-gcc -o xvid_encraw xvid_encraw.c  -lc -lm -I../src/ -L../build/generic/=build
```

-lxvidcore
即可生成 xvid_encraw

这里可以在交叉编译链目录/root/arm-none-linux-gnueabi/arm-none-linux-gnueabi/的 include,lib 下生成相应的头文件和库文件

---

4 交叉编译 ffmpeg

到 http://download.chinaunix.net/download.php?id=5532&ResourceID=2990 在这个网址上下载 ffmpeg-0.4.9-p20051120.tar.bz2，然后解压。

修改 configure 文件，要修改的如下

由于 cc、ar、ranlib、strip 都是交叉编译环境中的执行文件，可以这样配置
prefix="/root/arm-none-linux-gnueabi/arm-none-linux-gnueabi/"
cross_prefix="root/arm-none-linux-gnueabi/bin/arm-linux-"
cpu="arm"

配置
./configure --cpu=arm --cc=arm-linux-gcc --enable-shared --disable-ffserver --enable-xvid --enable-x264 --enable-gpl --enable-pthreads --disable-strip

make

make install

这里可以在交叉编译链目录/root/arm-none-linux-gnueabi/arm-none-linux-gnueabi/的 include,lib 下生成相应的头文件和库文件

至此，OpenCV 所依赖的库都交叉编译完成，并在交叉编译链中的 include,bin,share,lib 下有相应的文件

四 交叉编译 OpenCV

下载 OpenCV-1.0.0 源码 http://www.opencv.org.cn/download/opencv-1.0.0.tar.gz

解压进入目录配置

./configure --host=arm-none-linux-gnueabi --without-gtk --without-carbon --without-quicktime --without-1394libs --with-ffmpeg --without-python --without-swig --enable-static --enable-shared --disable-apps CXX=arm-none-linux-gnueabi-g++ CPPFLAGS=-I/root/arm-none-linux-gnueabi/arm-none-linux-gnueabi/include LDFLAGS=-L/root/arm-

none-linux-gnueabi/arm-none-linux-gnueabi/lib --with-v4l --prefix=/root/opencv-arm --libdir=/root/opencv-arm/lib --includedir=/root/opencv-arm/include

说明：

--host=arm-none-linux-gnueabi :指出交叉编译 arm 平台

--without-gtk:忽略 gtk + 2.0 windows

--without-carbon: 不使用 Mac OS 上的 X 库

--without-quicktime

--without-1394libs

--without-ffmpeg

--without-python

--without-swig

--enable-static :生成静态库

--enable-shared :生成动态库

CXX=arm-none-linux-gnueabi-g++ ： 指定编译工具

CPPFLAGS=-I/root/arm-none-linux-gnueabi/arm-none-linux-gnueabi/include ：OpenCV 会用到一些 dev 的包，如 png.h,jpeglib.h，大部分头文件在/usr/include 下

--prefix=/root/opencv-arm ： 指定安装目录

-libdir=/root/opencv-arm/lib: 指定库文件安装位置

--includedir=/root/opencv-arm/include：指定包含文件安装位置

如果配置正确，会有下面信息

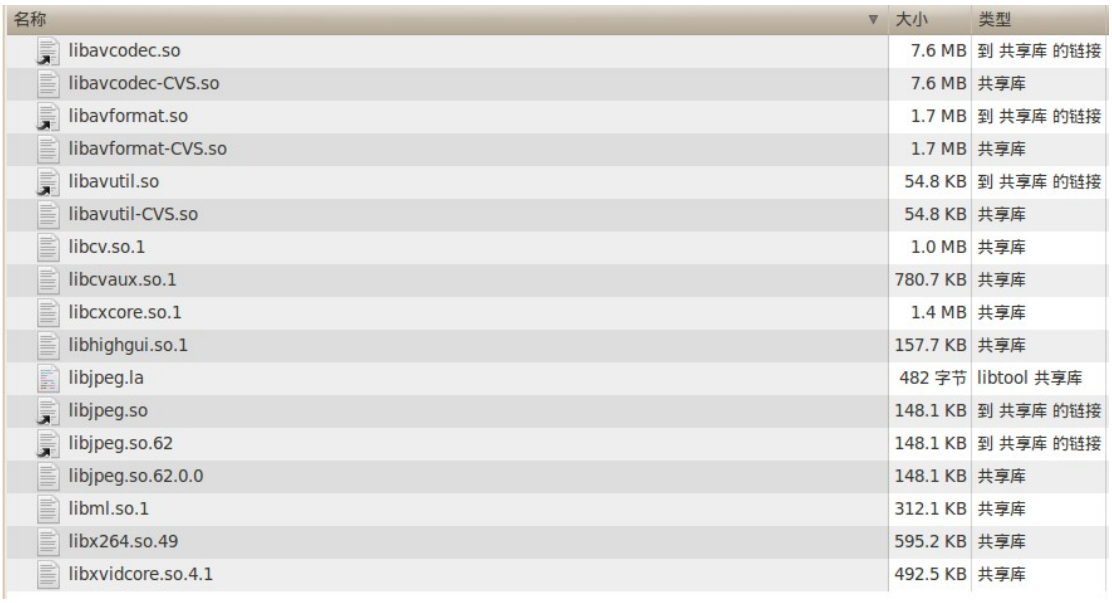
```
General configuration ================================================
   Compiler:              arm-none-linux-gnueabi-g++
   CXXFLAGS:                -Wall -fno-rtti -pipe -O3 -fomit-frame-pointer

   Install path:          /root/opencv-arm

HighGUI configuration ================================================

   Windowing system ----------
   Use Carbon / Mac OS X:    no
   Use gtk+ 2.x:             no
   Use gthread:              no

   Image I/O ----------------------
   Use libjpeg:              yes
```

```
    Use zlib:                yes
    Use libpng:              yes
    Use libtiff:             no
    Use libjasper:           no
    Use libIlmImf:           no

    Video I/O ----------------------
    Use QuickTime / Mac OS X: no
    Use xine:                no
    Use ffmpeg:              yes
    Use dc1394 & raw1394:    no
    Use v4l:                 yes
    Use v4l2:                yes

Wrappers for other languages ========================================
    SWIG
    Python                   no

Additional build settings ===========================================
    Build demo apps          no

Now run make ...
=====================================================================
===
# make
# make install
```

arm 上运行 OpenCV 所需库：
1 将/root/opencv-arm/lib 下生成的库文件
libcvaux.so.1.0.0
libcv.so.1.0.0
libcxcore.so.1.0.0
libhighgui.so.1.0.0
libml.so.1.0.0
拷出来全部重命名
*.so.1

2 加上之前的库文件，将/root/arm-none-linux-gnueabi/arm-none-linux-gnueabi/lib 下的 libjpeg,xvid,x264,ffmpeg 库都拷出来放去板子的/usr/lib 或者/lib 下,也可以复制到板子上一个文件夹，然后#export LD_LIBRARY_PATH=$LD_LIBRARY_PATH:/YOUR/lib/DIR

3 OpenCV 所需库总表:



名称	大小	类型
libavcodec.so	7.6 MB	到 共享库 的链接
libavcodec-CVS.so	7.6 MB	共享库
libavformat.so	1.7 MB	到 共享库 的链接
libavformat-CVS.so	1.7 MB	共享库
libavutil.so	54.8 KB	到 共享库 的链接
libavutil-CVS.so	54.8 KB	共享库
libcv.so.1	1.0 MB	共享库
libcvaux.so.1	780.7 KB	共享库
libcxcore.so.1	1.4 MB	共享库
libhighgui.so.1	157.7 KB	共享库
libjpeg.la	482 字节	libtool 共享库
libjpeg.so	148.1 KB	到 共享库 的链接
libjpeg.so.62	148.1 KB	到 共享库 的链接
libjpeg.so.62.0.0	148.1 KB	共享库
libml.so.1	312.1 KB	共享库
libx264.so.49	595.2 KB	共享库
libxvidcore.so.4.1	492.5 KB	共享库

编译源文件方法

arm-none-linux-gnueabi-g++ demo.c -o demo -I/root/opencv-arm/include/opencv -L/root/opencv-arm/lib -lcv -lcxcore -lpthread -lrt -lcvaux -lm -lpng -ljpeg -lz -lml -lhighgui -ldl

最后，将生成的二进制文件拷到板子上就可运行