君正 GCC 交叉工具链手册

1、君正 GCC 编译器的安装和使用

本节介绍君正 GCC 交叉编译工具链的安装和使用方法。

为了方便用户基于 Linux 进行开发，君正集成电路提供以下几个 GCC 交叉编译器：

Linux 2.6 系统：

- GCC-4.1.2 + GLIBC-2.3.6

Linux 2.4 系统：

- GCC-4.1.2 + GLIBC-2.3.2
- GCC-3.3.1 + GLIBC-2.3.2

这些交叉编译工具运行在Linux主机环境下，用来交叉编译生成可以运行在君正32位处理器上的代码。

用户可根据下面情况选择编译器版本：

1. 编译 U-Boot-1.1.6：可选择 GCC-4.1.2 或者 GCC-3.3.1
2. 编译 Linux 2.4 内核：可选择 GCC-3.3.1
3. 编译 Linux 2.6 内核：可选择 GCC-4.1.2
4. 编译 busybox 和应用程序：可选择 GCC-4.1.2 或者 GCC-3.3.1

需要注意的是，用户在选择某一版本的编译器编译应用程序时，应用程序运行所依赖的根文件系统 GLIBC 动态库版本必须与编译器的 GLIBC 版本相一致。对于君正 Linux 系统来说，Linux 2.4 系统使用 glibc 2.3.2 动态库，Linux 2.6 系统使用 glibc 2.3.6 动态库。

从君正公司网站（http://www.ingenic.cn）上可以下载到下面几个文件：

- mipseltools-gcc412-lnx26.tar.gz: Linux 2.6 系统编译器，Linux 主机版本
- mipseltools-gcc412-lnx24.tar.gz: Linux 2.4 系统编译器，Linux 主机版本
- mipseltools-gcc412-lnx24-cygwin.tar.gz: Linux 2.4 系统编译器，Windows Cygwin 版本
- mipseltools-gcc331-lnx24.tar.gz: Linux 2.4 系统编译器，Linux 主机版本
- mipseltools-gcc331-lnx24-cygwin.tar.gz: Linux 2.4 系统编译器，Windows Cygwin 版本

建议用户使用 Linux 主机做为开发环境，在基于 Linux 2.4 内核进行开发时请使用 GCC-3.3.1 编译器，在基于 Linux 2.6 内核行开发时请使用 GCC-4.1.2 编译器。

如果你已经从君正公司主页下载了 GCC 编译器，现在就可以安装和使用它了。

安装 Linux 2.6 编译器的步骤如下：

1、把 GCC 编译器安装到工作目录下，举例如下：

```
# cd /opt
# tar xzf mipseltools-gcc412-lnx26.tar.gz
```

2、设置 GCC 编译器的路径：

```
# export PATH=/opt/mipseltools-gcc412-lnx26/bin:$PATH
```

安装 Linux 2.4 编译器的步骤如下：

1、把 GCC 编译器安装到工作目录下，举例如下：

```
# cd /opt
# tar xzf mipseltools-gcc331-lnx24.tar.gz
```

2、设置 GCC 编译器的路径：

```
# export PATH=/opt/mipseltools-gcc331-lnx24/bin:$PATH
```

这时编译器就算安装好了。MIPS 交叉编译器的前缀为”mipsel-linux-”，安装好编译器后，您就可以使用 mipsel-linux-gcc 和 mipsel-linux-g++工具编译程序了。

君正 GCC 交叉编译工具链的语法和使用方法与 GNU GCC 兼容，有关 GCC 编译器更多编译选项和使用方法请参考 GNU GCC 的用户手册。

下面举例编译 helloworld.c:

```
#include <stdio.h>
int main(void)
{
    printf(“Hello world!\n”);
    return 0;
}
```

编译生成 helloworld:

```
# mipsel-linux-gcc -O2 -o helloworld helloworld.c
```

如果能够正确生成 helloworld，说明编译器已经正确安装可以正常使用了。

2、君正 Linux 2.6 编译器

君正 Linux 2.6 编译器使用了 GCC 4.1.2 和 GLIBC 2.3.6，下面将简单介绍制作该编译器需要的源码包和制作步骤。用户可以参考本节介绍的步骤自己制作交叉编译器。

需要的源码包：

- linux-2.6.24.3 headers
- binutils-2.17
- gcc-4.1.2
- glibc-2.3.6
- glibc-ports-2.3.6
- glibc-linuxthreads-2.3.6
- gdb-6.0
- e2fsprogs-1.40.4
- zlib-1.2.3
- jpeg-6b
- lcms_1.16
- libpng-1.2.24
- libmng-1.0.10

以上源码包和相关补丁可以到下面地址直接下载：

ftp://ftp.ingenic.cn/3sw/01linux/00toolchain/jz-gcc412-glib236-src.tar.gz

制作交叉编译器的基本过程分四步：

- 编译 binutils
- 编译 bootstrap 的 gcc，生成用来编译 glibc 的工具
- 编译 glibc，需要指定内核头文件
- 编译完整的 gcc 和 g++

下面是制作编译器的脚本：

```
#!/bin/sh

#############################################################
# Modify following variables to yours
TARBALL_PATH=/home/jlwei/work-gcc/tarball
PATCHES_PATH=/home/jlwei/work-gcc/patches
LINUX_HEADERS_PATH=/home/jlwei/work-gcc/linux-headers-2.6.24.3
INSTALL_PATH=/opt/mipseltools-gcc412-lnx26
BUILD_PATH=`pwd`
```

```
#############################################################

BINUTILS_VER=binutils-2.17
GCC_VER=gcc-4.1.2
GLIBC_VER=glibc-2.3.6
GLIBC_PORTS_VER=glibc-ports-2.3.6
GLIBC_LINUXTHREADS_VER=glibc-linuxthreads-2.3.6
GDB_VER=gdb-6.0

unset CFLAGS
unset CXXFLAGS

export PATH=${INSTALL_PATH}/bin:$PATH

echo "----------------------------------"
echo "@@ Building binutils ..."
echo "----------------------------------"

# prepare binutils source
cd ${BUILD_PATH}
rm -rf ${BINUTILS_VER} binutils-build
tar jxf ${TARBALL_PATH}/${BINUTILS_VER}.tar.bz2

# configure and build binutils
mkdir -v binutils-build
cd binutils-build

../${BINUTILS_VER}/configure --target=mipsel-linux
--prefix=${INSTALL_PATH}
make CFLAGS="-O2"

# install binutils
make install

echo "----------------------------------"
echo "@@ Building bootstrap gcc ..."
echo "----------------------------------"

# prepare gcc source
cd ${BUILD_PATH}
rm -rf ${GCC_VER} gcc-build

tar jxf ${TARBALL_PATH}/${GCC_VER}.tar.bz2
cd ${BUILD_PATH}/${GCC_VER}/libiberty
```

```
cat strsignal.c | sed -e 's/#ifndef HAVE_PSIGNAL/#if 0/g' >junk.c
cp -f strsignal.c strsignal.c.fixed; mv -f junk.c strsignal.c

# configure and build gcc
cd ${BUILD_PATH}
mkdir -v gcc-build
cd gcc-build

../${GCC_VER}/configure --target=mipsel-linux \
      --host=i686-pc-linux-gnu --prefix=${INSTALL_PATH} \
      --disable-shared --disable-threads --disable-multilib \
      --enable-languages=c

make CFLAGS="-O2" all-gcc

# install gcc
make install-gcc

echo "----------------------------------"
echo "@@ Building glibc ..."
echo "----------------------------------"

# prepare glibc source
cd ${BUILD_PATH}
rm -rf ${GLIBC_VER} glibc-build

tar jxf ${TARBALL_PATH}/${GLIBC_VER}.tar.bz2
cd ${GLIBC_VER}
tar jxf ${TARBALL_PATH}/${GLIBC_LINUXTHREADS_VER}.tar.bz2
tar jxf ${TARBALL_PATH}/${GLIBC_PORTS_VER}.tar.bz2
mv ${GLIBC_PORTS_VER} ports
# apply patch
patch -Np1 -i ${PATCHES_PATH}/${GLIBC_VER}-jz.patch
# remove nptl
rm -rf nptl nptl_db

# configure and build glibc
cd ${BUILD_PATH}
mkdir -v glibc-build
cd glibc-build

export CC="mipsel-linux-gcc"
export libc_cv_forced_unwind=yes
export libc_cv_c_cleanup=yes
```

```
../${GLIBC_VER}/configure \
      --host=mips-linux \
      --build=i686-pc-linux-gnu \
      --enable-add-ons \
      --enable-shared \
      --with-cpu=mips32 \
      --prefix=/usr \
      --with-headers=${LINUX_HEADERS_PATH}

make CFLAGS="-O2"

# install glibc
export GLIBC_INSTALL=${BUILD_PATH}/glibc-inst

cd $BUILD_PATH
rm -rf glibc-inst
mkdir -v glibc-inst
cd glibc-build

make install_root=${GLIBC_INSTALL} install

cd ${GLIBC_INSTALL}; tar zcf $BUILD_PATH/glibc-build/glibc-lib.tgz lib
cd ${INSTALL_PATH}; tar zxf $BUILD_PATH/glibc-build/glibc-lib.tgz
cd ${GLIBC_INSTALL}/usr; tar cfz $BUILD_PATH/glibc-build/glibc-usr.tgz
.
cd ${INSTALL_PATH}/mipsel-linux; tar xzf
$BUILD_PATH/glibc-build/glibc-usr.tgz
tar xfz $BUILD_PATH/glibc-build/glibc-lib.tgz

# install linux kernel headers
cd ${LINUX_HEADERS_PATH}
cp -afr {asm,asm-mips,asm-generic,linux,mtd,scsi,sound}
${INSTALL_PATH}/mipsel-linux/include

# fixed libc.so and libpthread.so
sed -i -e 's/\/usr\/lib\///g'
${INSTALL_PATH}/mipsel-linux/lib/libpthread.so
sed -i -e 's/\/usr\/lib\///g' ${INSTALL_PATH}/mipsel-linux/lib/libc.so

sed -i -e 's/\/lib\///g'
${INSTALL_PATH}/mipsel-linux/lib/libpthread.so
sed -i -e 's/\/lib\///g' ${INSTALL_PATH}/mipsel-linux/lib/libc.so
```

```
# install localedata
cd ${BUILD_PATH}/glibc-build
cp -v ${PATCHES_PATH}/${GLIBC_VER}-localedata-Makefile
${BUILD_PATH}/${GLIBC_VER}/localedata/Makefile
cp -v ${PATCHES_PATH}/${GLIBC_VER}-localedata-SUPPORTED
${BUILD_PATH}/${GLIBC_VER}/localedata/SUPPORTED
make localedata/install-locales install_root=${INSTALL_PATH}

echo "---------------------------------"
echo "@@ Building final gcc ..."
echo "---------------------------------"

# prepare gcc source
cd $BUILD_PATH
rm -rf ${GCC_VER} gcc-build

tar jxf ${TARBALL_PATH}/${GCC_VER}.tar.bz2
cd ${BUILD_PATH}/${GCC_VER}/libiberty
cat strsignal.c | sed -e 's/#ifndef HAVE_PSIGNAL/#if 0/g' >junk.c
cp -f strsignal.c strsignal.c.fixed; mv -f junk.c strsignal.c

# apply patches
cd ${BUILD_PATH}/${GCC_VER}
patch -p0 < ${PATCHES_PATH}/gcc-4.1_bug27067.patch

# configure and build gcc
cd ${BUILD_PATH}
mkdir -v gcc-build
cd gcc-build

export CC="gcc"

../${GCC_VER}/configure --target=mipsel-linux \
      --host=i686-pc-linux-gnu --prefix=${INSTALL_PATH} \
      --disable-multilib --enable-shared --enable-languages=c,c++ \
      --with-headers=${INSTALL_PATH}/mipsel-linux/include

make CFLAGS="-O2"

# install gcc
make install

# remove sys-include
rm -rf ${INSTALL_PATH}/mipsel-linux/sys-include
```

```
cd $BUILD_PATH

# fix symlink
echo "Fix symlink ..."

cd ${INSTALL_PATH}/mipsel-linux/lib

rm libanl.so
ln -s libanl.so.1 libanl.so

rm libBrokenLocale.so
ln -s libBrokenLocale.so.1 libBrokenLocale.so

rm libcrypt.so
ln -s libcrypt.so.1 libcrypt.so

rm libdl.so
ln -s libdl.so.2 libdl.so

rm libm.so
ln -s libm.so.6 libm.so

rm libnsl.so
ln -s libnsl.so.1 libnsl.so

rm libnss_compat.so
ln -s libnss_compat.so.2 libnss_compat.so

rm libnss_dns.so
ln -s libnss_dns.so.2 libnss_dns.so

rm libnss_files.so
ln -s libnss_files.so.2 libnss_files.so

rm libnss_hesiod.so
ln -s libnss_hesiod.so.2 libnss_hesiod.so

rm libnss_nisplus.so
ln -s libnss_nisplus.so.2 libnss_nisplus.so

rm libnss_nis.so
ln -s libnss_nis.so.2 libnss_nis.so
```

```
rm libresolv.so
ln -s libresolv.so.2 libresolv.so

rm librt.so
ln -s librt.so.1 librt.so

rm libthread_db.so
ln -s libthread_db.so.1 libthread_db.so

rm libutil.so
ln -s libutil.so.1 libutil.so

# install mxu_as and jz_mxu.h
cp -v ${PATCHES_PATH}/mxu_as ${INSTALL_PATH}/bin
chmod +x ${INSTALL_PATH}/bin/mxu_as
cp -v ${PATCHES_PATH}/jz_mxu.h ${INSTALL_PATH}/mipsel-linux/include

echo "-----------------------------------"
echo "@@ Building GDB ..."
echo "-----------------------------------"

cd $BUILD_PATH
rm -rf ${GDB_VER} gdb-build

tar jxf ${TARBALL_PATH}/${GDB_VER}.tar.bz2
mkdir -v gdb-build
cd gdb-build

export CC="gcc"

../${GDB_VER}/configure --target=mipsel-linux
--prefix=${INSTALL_PATH}
make
make install


echo "@@ Building binutils, glibc, gcc and gdb OK"
```

下面是编译第三方动态库的脚本：

```
#!/bin/sh

#############################################################################
# Modify these variables to yours
THIRDPARTY_PATH=/home/jlwei/work-gcc/thirdparty
INSTALL_PATH=/opt/mipseltools-gcc412-lnx26/mipsel-linux
BUILDDIR=`pwd`
export PATH=/opt/mipseltools-gcc412-lnx26/bin:$PATH # mipsel-linux-gcc
#############################################################################


echo "@@ Building e2fsprogs ..."

cd ${BUILDDIR}
tar xzf ${THIRDPARTY_PATH}/e2fsprogs-1.40.4.tar.gz
cd e2fsprogs-1.40.4

./configure --prefix=${INSTALL_PATH} --host=mipsel-linux --enable-elf-shlibs
--disable-tls CC=mipsel-linux-gcc LD=mipsel-linux-ld
make
make install-libs

# fix symbol links
cd ${INSTALL_PATH}/lib

rm libblkid.so
ln -s libblkid.so.1 libblkid.so

rm libcom_err.so
ln -s libcom_err.so.2 libcom_err.so

rm libe2p.so
ln -s libe2p.so.2 libe2p.so

rm libext2fs.so
ln -s libext2fs.so.2 libext2fs.so

rm libss.so
ln -s libss.so.2 libss.so

rm libuuid.so
ln -s libuuid.so.1 libuuid.so
```

```
echo "@@ Building zlib ..."

cd ${BUILDDIR}
tar xzf ${THIRDPARTY_PATH}/zlib-1.2.3.tar.gz
cd zlib-1.2.3

CC="mipsel-linux-gcc" AR="mipsel-linux-ar cr" RANLIB="mipsel-linux-ranlib"
./configure --prefix=${INSTALL_PATH}
make
make install

CC="mipsel-linux-gcc" AR="mipsel-linux-ar cr" RANLIB="mipsel-linux-ranlib"
./configure --shared --prefix=${INSTALL_PATH}
make
make install


echo "@@ Building libjpeg ..."

cd ${BUILDDIR}
tar xzf ${THIRDPARTY_PATH}/jpegsrc.v6b.tar.gz
cd jpeg-6b

CC=gcc ./configure --prefix=${INSTALL_PATH} --enable-shared --enable-static

sed -i -e 's/CC= gcc/CC= mipsel-linux-gcc/g' Makefile
sed -i -e 's/AR= ar rc/AR= mipsel-linux-ar rc/g' Makefile
sed -i -e 's/AR2= ranlib/AR2= mipsel-linux-ranlib/g' Makefile

make
mkdir -p ${INSTALL_PATH}/man/man1
make install


echo "@@ Building lcms ..."

cd ${BUILDDIR}
tar xzf ${THIRDPARTY_PATH}/lcms_1.16.tar.gz
cd lcms-1.16

./configure --prefix=${INSTALL_PATH} --target=mipsel-linux --host=mipsel-linux
CC=mipsel-linux-gcc
```

```
make
make install


echo "@@ Building libpng ..."

cd ${BUILDDIR}
tar xzf ${THIRDPARTY_PATH}/libpng-1.2.24.tar.gz
cd libpng-1.2.24
./configure --prefix=${INSTALL_PATH} --host=mipsel-linux CC=mipsel-linux-gcc
CFLAGS="-I${INSTALL_PATH}/include" LDFLAGS="-L${INSTALL_PATH}/lib"
make
make install


echo "@@ Building libmng ..."

cd ${BUILDDIR}
tar xzf ${THIRDPARTY_PATH}/libmng-1.0.10.tar.gz
cd libmng-1.0.10
cp makefiles/makefile.linux makefile

# replace gcc to mipsel-linux-gcc
sed -i -e 's/CC=gcc/CC=mipsel-linux-gcc/g' makefile

# replace ar to mipsel-linux-ar
sed -i -e 's/ar rc/mipsel-linux-ar rc/g' makefile

# replace ranlib to mipsel-linux-ranlib
sed -i -e 's/RANLIB=ranlib/RANLIB=mipsel-linux-ranlib/g' makefile

# replace /usr/local to ${INSTALL_PATH}
sed -i -e 's/\/usr\/local/\/opt\/mipseltools-gcc412-lnx26\/mipsel-linux/g' makefile

make
make install


echo "@@ Building tslib ..."

cd ${BUILDDIR}
tar xzf ${THIRDPARTY_PATH}/tslib-jz.tar.gz
cd tslib-jz/tslib-0.1.1
./autogen.sh
```

```
echo "ac_cv_func_malloc_0_nonnull=yes" > config.cache
./configure --prefix=${INSTALL_PATH} --host=mipsel-linux --cache-file=config.cache

make
make install


echo "@@ Building alsa ..."

cd ${BUILDDIR}
tar xzf ${THIRDPARTY_PATH}/alsa-tools.tar.gz
cd alsa-tools/alsa-lib-1.0.15
./configure --prefix=${INSTALL_PATH} AR=mipsel-linux-ar CC=mipsel-linux-gcc
CXX=mipsel-linux-g++ CXX=mipsel-linux-g++ --host=mipsel-linux --enable-shared=yes
--enable-static=no --target=mips-linux --with-debug=no --with-alsa-devdir=/dev
--with-softfloat  LDFLAGS="-lm"

make
make install

echo "Build thirdparty libs done."
```

网络资源：

http://www.gnu.org/：GNU主页
http://ftp.gnu.org/gnu/binutils/：binutils下载网址
http://gcc.gnu.org/：GCC主页
http://ftp.gnu.org/gnu/glibc/：glibc下载网址
http://e2fsprogs.sourceforge.net/：e2fsprogs主页
http://www.zlib.net/：zlib主页
http://www.ijg.org/：libjpeg主页
http://www.littlecms.com/downloads.htm：lcms下载网址
http://www.libpng.org/：libpng主页
http://www.libmng.com/：libmng主页