

建立针对arm-linux的交叉编译环境 PDF转换可能丢失图片或格式，建议阅读原文

https://www.100test.com/kao_ti2020/144/2021_2022__E5_BB_BA_E7_AB_8B_E9_92_88_E5_c103_144254.htm

交叉编译环境介绍

交叉编译是嵌入式开发过程中的一项重要技术，它的主要特征是某机器中执行的程序代码不是在本机编译生成，而是由另一台机器编译生成，一般把前者称为目标机，后者称为主机。

采用交叉编译的主要原因在于，多数嵌入式目标系统不能提供足够的资源供编译过程使用，因而只好将编译工程转移到高性能的主机中进行。

linux下的交叉编译环境重要包括

以下几个部分：针对目标系统的编译器gcc 针对目标系统的二进制工具binutils 目标系统的标准c库glibc 目标系统的linux内核头文件 交叉编译环境的建立步骤 下载源代码 下载包

包括binutils、gcc、glibc及linux内核的源代码（需要注意的是

，glibc和内核源代码的版本必须与目标机上实际使用的版本保持一致），并设定shell变量PREFIX指定可执行程序的安装路径。

编译binutils 首先运行configure文件，并使

用--prefix=\$PREFIX参数指定安装路径，使

用--target=arm-linux参数指定目标机类型，然后执行make

install。配置linux内核头文件 首先执行make mrproper进行清

理工作，然后执行make config ARCH=arm（或make

menuconfig/xconfig ARCH=arm）进行配置（注意，一定要在

命令行中使用ARCH=arm指定cpu架构，因为缺省架构为主机的cpu架构），这一步需要根据目标机的实际情况进行详细的

配置，笔者进行的实验中目标机为HP的ipaq-hp3630 PDA，因

而设置system type为SA11X0，SA11X0 Implementations中选

择Compaq iPAQ H3600/H3700。配置完成之后，需要将内核头文件拷贝到安装目录：`cp -dR include/asm-arm $PREFIX/arm-linux/include/asm cp -dR include/linux $PREFIX/arm-linux/include/linux` 第一次编译gcc 首先运行configure文件，使用`--prefix=$PREFIX`参数指定安装路径，使用`--target=arm-linux`参数指定目标机类型，并使用`--disable-threads`、`--disable-shared`、`--enable-languages=c`参数，然后执行`make install`。这一步将生成一个最简的gcc。由于编译整个gcc是需要目标机的glibc库的，它现在还不存在，因此需要首先生成一个最简的gcc，它只需要具备编译目标机glibc库的能力即可。交叉编译glibc 这一步骤生成的代码是针对目标机cpu的，因此它属于一个交叉编译过程。该过程要用到linux内核头文件，默认路径为`$PREFIX/arm-linux/sys-linux`，因而需要在`$PREFIX/arm-linux`中建立一个名为`sys-linux`的软连接，使其内核头文件所在的`include`目录；或者，也可以在接下来要执行的configure命令中使用`--with-headers`参数指定linux内核头文件的实际路径。configure的运行参数设置如下（因为是交叉编译，所以要将编译器变量CC设为`arm-linux-gcc`）：`CC=arm-linux-gcc ./configure --prefix=$PREFIX/arm-linux --host=arm-linux --enable-add-ons` 最后，按以上配置执行configure和`make install`，glibc的交叉编译过程就算完成了，这里需要指出的是，glibc的安装路径设置为`$PREFIX/ARCH=arm/arm-linux`，如果此处设置不当，第二次编译gcc时可能找不到glibc的头文件和库。第二次编译gcc 运行configure，参数设置为`--prefix=$PREFIX --target=arm-linux --enable-languages=c,c`。运行`make install`。

到此为止整个交叉编译环境就完全生成了。几点注意事项 第一点、在第一次编译gcc的时候可能会出现找不到stdio.h的错误，解决办法是修改gcc/config/arm/t-linux文件，在TARGET_LIBGCC2_CFLAGS变量的设定中增加-Dinhibit_libc和-D__gthr_posix_h。第二点、对与2.3.2版本的glibc库,编译linuxthread/sysdeps/pthread/sigaction.c时可能出错,需要通过补丁glibc-2.3.2-arm.patch 解决：执行patch -p1 第三点、第二次编译gcc时可能会出现libc.so的错误，这是需要利用文本编辑器手动修改libc.so。 100Test 下载频道开通，各类考试题目直接下载。详细请访问 www.100test.com