

开发环境的搭建

准备工作: 客户需要安装虚拟机+Ubuntu14.04 ;

注意: 1.符号 “ // ” 后面的内容是操作解释,不是指令的一部分;

2.有的指令需要运行一定时间才完成,请耐心等待;

第一步,安装 arm-xilinx-linux-gnueabi.bin 交叉编译器。

步骤 1:

打开虚拟机中的 linux 系统 (64 位) , 打开终端输入如下命令

```
sudo apt-get install g++ //安装 g++
```

```
sudo apt-get install lib32z1 lib32ncurses5 lib32bz2-1.0 //安装 32 位库
```

步骤 2 :

(1) 将文件 xilinx-2012.09-104-arm-xilinx-linux-gnueabi.bin 拷贝到新建的文件夹

Software 中 ;

(2) 在终端输入如下命令,注意,输入所有命令均处在用户模式,不要进入 root 用户

1) cd Software

2) sudo dpkg-reconfigure dash //弹出提示 选 NO

3) ls // 查看文件属性 (文件需要可执行权限) (可忽略该步)

4) 修改权限 chmod 755 xilinx-2012.09-104-arm-xilinx-linux-gnueabi.bin

5) ls // 查看 xilinx-2012.09-104-arm-xilinx-linux-gnueabi.bin 是否为可

执行文件 (可执行文件显示为绿色) (可忽略该步)

6)运行 ./xilinx-2012.09-104-arm-xilinx-linux-gnueabi.bin //执行后按照相

关提示引导进行配置,用默认设置的即可

7) cd // 进入用户根目录

8) `sudo gedit .bashrc` 会弹出文件编辑页面，

在.bashrc 最末行添加如下语句：

```
export ARCH=arm
```

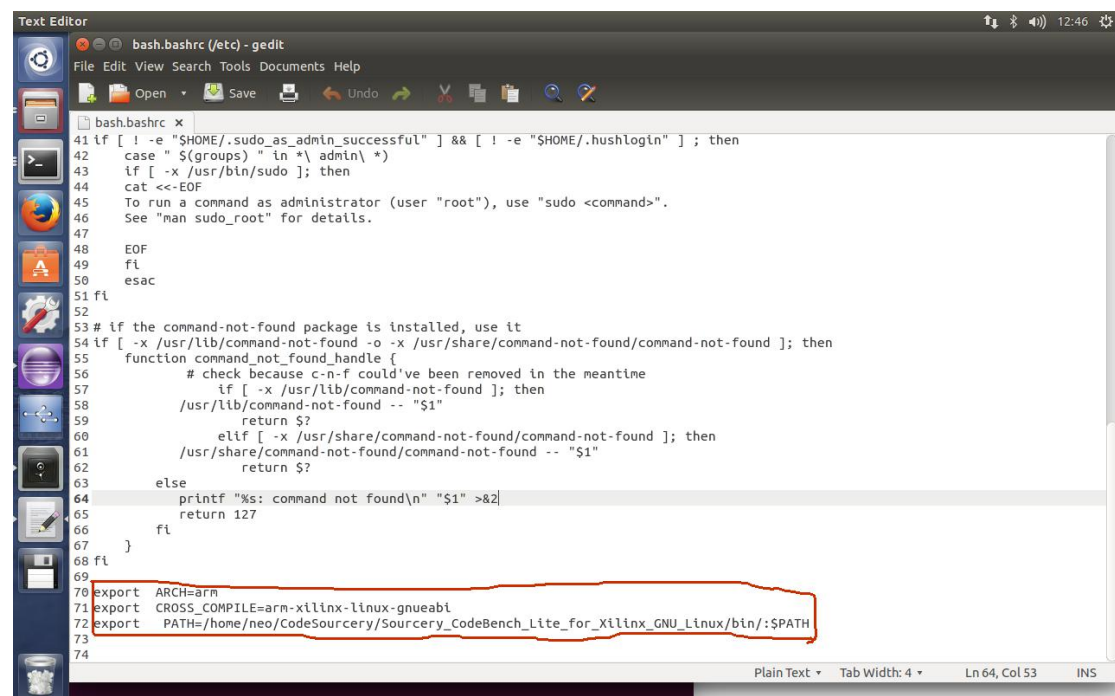
```
export CROSS_COMPILE=arm-xilinx-linux-gnueabi
```

```
export PATH =
```

```
/home/usr/CodeSourcery/Sourcery_CodeBench_Lite_for_Xilinx_GNU_Linux
```

```
/bin/:$PATH
```

//usr 是当前用户的名称



```
Text Editor bash.bashrc (etc) - gedit
File Edit View Search Tools Documents Help
Open Save Undo
bash.bashrc x
41 if [ ! -e "$HOME/.sudo_as_admin_successful" ] && [ ! -e "$HOME/.hushlogin" ]; then
42 case " $(groups) " in *\ admin\ *)
43 if [ -x /usr/bin/sudo ]; then
44 cat <<-EOF
45 To run a command as administrator (user "root"), use "sudo <command>".
46 See "man sudo_root" for details.
47 EOF
48 fi
49 esac
50 fi
51 fi
52
53 # if the command-not-found package is installed, use it
54 if [ -x /usr/lib/command-not-found -o -x /usr/share/command-not-found/command-not-found ]; then
55 function command_not_found_handle {
56 # check because c-n-f could've been removed in the meantime
57 if [ -x /usr/lib/command-not-found ]; then
58 /usr/lib/command-not-found -- "$1"
59 return $?
60 elif [ -x /usr/share/command-not-found/command-not-found ]; then
61 /usr/share/command-not-found/command-not-found -- "$1"
62 return $?
63 else
64 printf "%s: command not found\n" "$1" >&2
65 return 127
66 fi
67 }
68 fi
69
70 export ARCH=arm
71 export CROSS_COMPILE=arm-xilinx-linux-gnueabi
72 export PATH=/home/neo/CodeSourcery/Sourcery_CodeBench_Lite_for_Xilinx_GNU_Linux/bin/:$PATH
73
74
Plain Text Tab Width: 4 Ln 64, Col 53 INS
```

9) `source .bashrc` 然后关闭之前开的终端;

10) 重新打开终端, `echo $PATH` //查看路径是否添加上 (可忽略该步骤)

11) `arm-xilinx-linux-gnueabi-gcc -v` //查看版本信息 (可忽略该步骤), 这一步可能需要重新打开终端或重启系统才能成功。如果能够打印出版本信息, 说明这个交叉编译工具能够使用。

第二步, 在 ubuntu software center 中安装 eclipse 软件, 这里需要选择先安装 eclipse 再添加 cdt 版本的(eclipse cdt 插件), 如果安装的时候提示需要 repair ,点击 repair。

第三步，将 boost_1_58_0.tar.gz 拷贝到 /home/usr/Downloads 文件夹中，将 N_Kunhou_Arm_sdk.tar.gz 拷贝到 /home/usr/arm 文件夹中，在它们所在的文件夹中解压它们。

//usr 指代 ubuntu 用户名称

第四步，先安装 cmake，

1. cd ~

2. sudo apt-get install cmake

然后修改配置文件 N_Kunhou_Arm_sdk/source/CMakeARM_Boost.txt，主要是根据客户的实际路径，修改配置文件；有三处，1.CMAKE_INSTALL_PREFIX 对应 N_Kunhou_Arm_sdk 的绝对路径, 2.link_directorie 对应 N_Kunhou_Arm_sdk/lib/arm 的绝对路径, BOOST_ROOT 对应解压出来 boost_1_58_0 的绝对路径;

下图中是把 N_Kunhou_Arm_sdk.tar.gz 拷贝到 /home/neo/Documents 解压, boost_1_58_0.tar.gz 拷贝到 /home/neo/Downloads 下解压，对应的 CMakeARM_Boost.txt 配置文件。

```
21 message("${CMAKE_C_FLAGS}")
22
23 message("CMAKE_CXX_FLAGS:")
24 message("${CMAKE_CXX_FLAGS}")
25
26 SET(CMAKE_INSTALL_PREFIX "/home/neo/Documents/N_Kunhou_Arm_sdk") 1
27 IF(BUILD_SDK)
28 ELSE()
29 link_directories("/home/neo/Documents/N_Kunhou_Arm_sdk/lib/arm") 2
30 ENDIF(BUILD_SDK)
31 message("${CMAKE_INSTALL_PREFIX}")
32
33 message("-----use arm boost-----")
34
35 SET(BOOST_ROOT "/home/neo/Downloads/boost_1_58_0") 3
36 set(BOOST_USE_STATIC_LIBS ON)
37 set(BOOST_USE_MULTITHREADED ON)
38
39 find_package(Boost 1.58.0 COMPONENTS thread date_time program_options filesystem system signals serialization REQUIRED)
40 #find_package(Boost 1.54.0 COMPONENTS thread date_time program_options filesystem system signals serialization REQUIRED)
41 if(Boost_FOUND)
42     include_directories(${Boost_INCLUDE_DIRS})
43     link_directories(${Boost_LIBRARY_DIRS})
44     #message("${CMAKE_MODULE_PATH}")
45     #message("${Boost_ROOT}")
46     message("${Boost_INCLUDE_DIRS}")
47     message("${Boost_LIBRARY_DIRS}")
48     message("${Boost_LIBRARIES}")
49     message("boost found")
50
51 else()
52     message("boost not found")
53 endif()
54
55
```

配置文件修改并保存后，开启新终端 `cd` 进入 `N_Kunhou_Arm_sdk` 文件夹，运行 `./build_ARM.sh`；如果没有报错，在 `N_Kunhou_Arm_sdk` 中可以看到新增的 `build-ar` 文件夹；

开启新终端，`cd` 进入 `N_Kunhou_Arm_sdk/build-ar` 文件夹，输入 `make` 命令，等待编译完成。

如果能顺利编译，就可以把相应工程添加到 Eclipse 中，具体操作：

Eclipse->file->import, 选则 General> Existing Projects into Workspace ;

下一步, Exist Code Location 目录选择客户 `N_Kunhou_Arm_sdk` 所在的目录即可；

然后在 Eclipse 左侧的项目栏中可以看到相应工程。

这样，整个环境即搭建完成，客户只需要在 Eclipse 编写代码，然后在

`N_Kunhou_Arm_sdk /build-ar` 中 使用 `make` 指令 编译, 即可在 `N_Kunhou_Arm_sdk /bin/arm` 中生成新的可执行文件。