



IAC-IMX6UL-Kit LINUX

用户手册

版本号：V1.0

目录

前言.....	2
产品说明.....	2
产品版本说明.....	2
修订记录.....	2
一、阅读前说明.....	3
二、准备工作.....	4
三、功能测试.....	6
四、固件烧写.....	7
五、安装交叉编译器.....	8
5.1 概述	8
5.2 安装步骤	8
六、搭建 NFS 网络文件系统	11
6.1 概述	11
6.2 安装步骤	11
七、源码编译.....	14
7.1 概述	14
7.2 uboot 编译.....	14
7.3 内核的配置及编译.....	18
八、应用程序的开发.....	21
8.1 编写应用程序及交叉编译	21
8.2 运行应用程序	22
九、添加应用程序到文件系统.....	26
9.1 概述	26
9.2 添加到文件系统	26
十、结束语.....	29

前言

产品说明

欢迎使用浙江启扬智能科技有限公司产品 IAC-IMX6UL-Kit，本产品 Linux 部分包含 6 份手册：

- 《IAC-IMX6UL-Kit 用户手册.pdf》
- 《IAC-IMX6UL-Kit 开发板简介.pdf》
- 《IAC-IMX6UL-Kit BSP 开发指南.pdf》
- 《IAC-IMX6UL-Kit 测试手册.pdf》
- 《IAC-IMX6UL-Kit 固件烧写手册.pdf》

本手册主要介绍交叉编译环境的搭建、源代码以及应用例程的编译。
使用之前请仔细阅读《IAC-IMX6UL-Kit 硬件说明手册.pdf》！

读者对象

本手册主要适用于以下工程师：

- 测试工程师
- 技术支持工程师
- 软件开发工程师

产品版本说明

本手册适用的产品版本描述如下：

产品名称	产品版本
IAC-IMX6UL-Kit	V1.0

修订记录

修订记录累积了每次文档更新的说明。最新版本的文档包含以前所有文档版本的更新内容。

修订日期	版本	修订说明
2016/10/10	V1.0	版本发布

一、阅读前说明

- 装有 Linux 系统 (ubuntu 或其它 Linux 发行版), 本手册以 ubuntu 12.04 操作为例, 具体搭建请参照《虚拟机安装 ubuntu 指导手册.pdf》!
- 在 ubuntu 用户目录下创建一个工作目录: `mkdir ~/work①②`, 并将要编译的文件拷贝到此目录下。
- 关于 linux 下的常用命令以及 vi 的操作等在这里都不做详细说明, 请用户自行查阅相关资料!
- 开发板标配光盘目录, 文档说明中用到的所有工具软件以及代码文件全部在光盘的对应目录下, 使用前请确保光盘资料齐全!
- 光盘中有移植好的源码, 用户可直接编译使用, 也可以根据实际情况进行配置、编译。

注:

- ① 这里的~/表示用户目录, 整条命令表示在用户目录下创建 work 目录实际对应的绝对路径为/home/lvmh (此目录为登录账户工作目录, 本手册以 lvmh 账户为例, 操作过程中请以实际登录账户所在目录为准), ~/work 对应的路径是/home/lvmh/work。
- ② 为了统一和陈述方便, 所有文件都拷贝到该目录进行操作, 具体目录用户可以自行创建, 这里只是以~/work 目录为例!

二、准备工作

- 串口连接：通过提供的串口线将开发板的调试串口(J14)与 PC 机的串口接口连接。
- 网络连接：通过网线将开发板的以太网接口(J2)与PC机的网络接口连接。
- USB 连接：通过 USB 连接线将开发板的 USB Device(J6)与 PC 机的 USB 口接。
- 串口设置：打开终端通讯软件 Xshell(迷你终端或 Windows 下的超级终端)，选择所用到的串口并设置如下参数：波特率（115200）、数据位（8 位）、停止位（1 位）、校验位（无）、数据流控制（无），具体操作如图 1-1，图 1-2 所示：

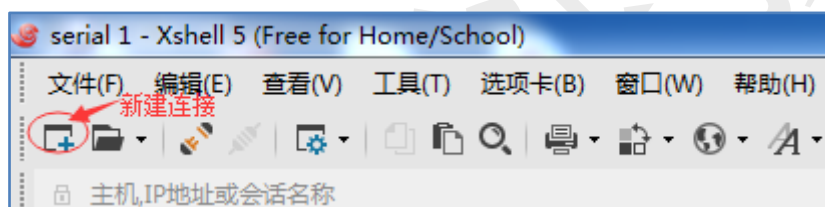


图 1-1

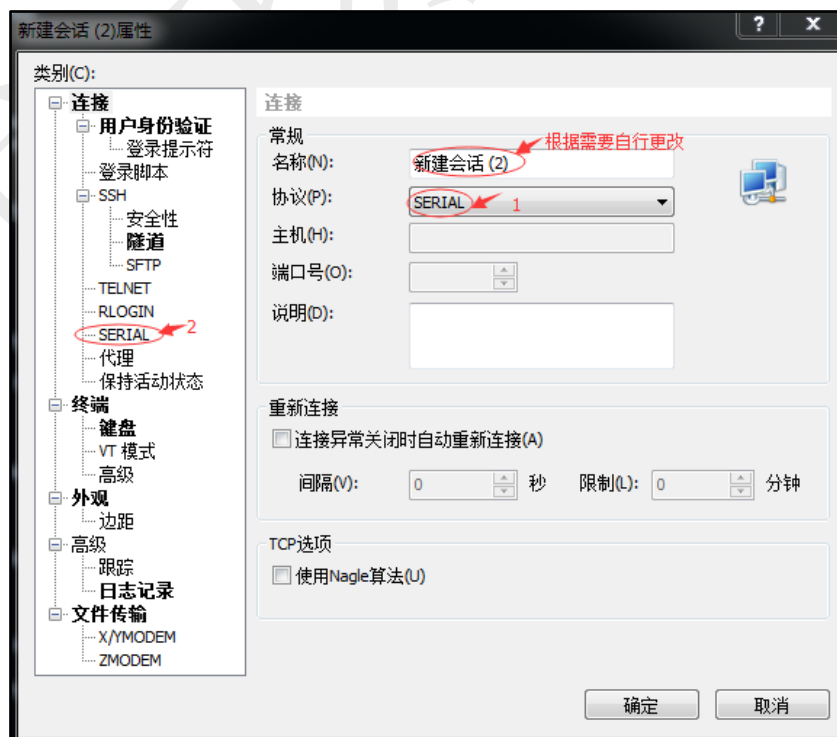


图 1-2

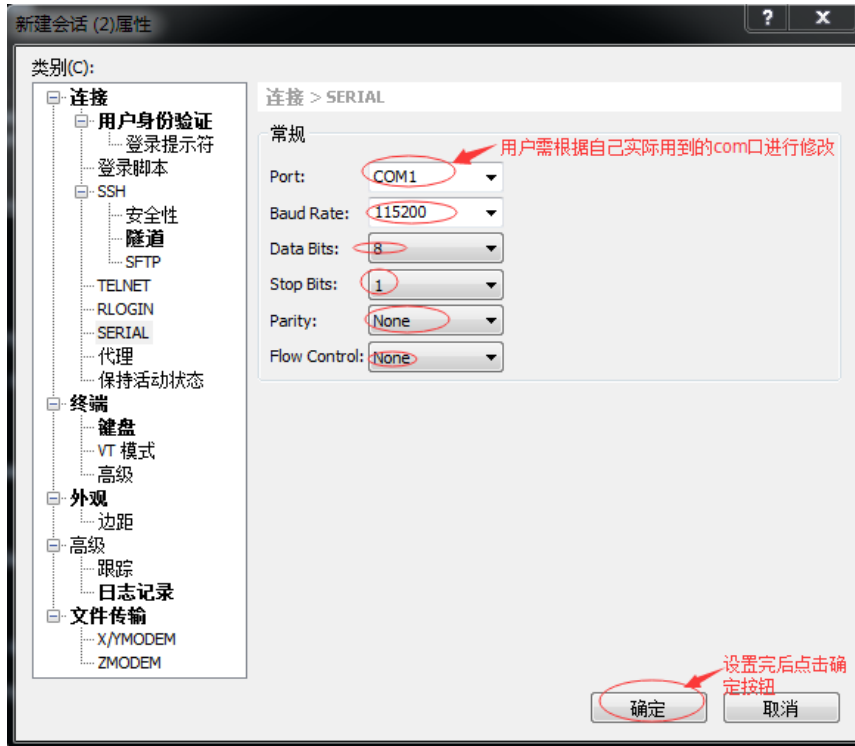


图 1-3

三、功能测试

先进行此步骤是为了检测开发板的功能是否良好，开发板标配文件系统中已经集成了测试程序，启动板子后可在/usr/test 目录下找到相应的测试程序，具体测试方法，请参照手册：《IAC-IMX6UL-Kit 测试手册》。

若测试出现问题，请根据测试手册中列举的问题排查错误原因。

若不能解决问题，请联系：supports@qiyangtech.com。


四、固件烧写

如果需要重新烧写 linux 系统，开发板提供了 EMMC 启动方式，具体烧写方法请参照：《IAC-IMX6UL-Kit 固件烧写手册.pdf》，该手册详细介绍了如何烧写镜像的方法。

五、安装交叉编译器

5.1 概述

bootloader、kernel、fs、所有的应用程序以及库文件的编译都需要用到交叉编译器。所以，在这里需要先安装一下交叉编译工具链，在光盘的编译工具的目录下已经有制作好的版本为交叉编译工具，用户可以直接安装使用，该交叉编译器的 GCC 版本为 4.7，如下图所示：

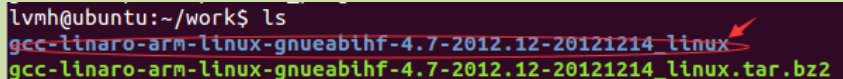
名称	修改日期	类型
 gcc-linaro-arm-linux-gnueabi-hf-4.7-2012.12-20121214_linux.tar	2016/8/19 15:50	好压 BZ2 压缩文件

5.2 安装步骤

进入~/work 目录，将 gcc-linaro-arm-linux-gnueabi-hf-4.7-2012.12-20121214_linux.tar.bz2 这个交叉编译工具链拷贝到此目录下。

具体的步骤如下：

步骤一：

命令	1、\$sudo tar -xjvf gcc-linaro-arm-linux-gnueabi-hf-4.7-2012.12-20121214_linux.tar.bz2 2、\$ ls
相关描述	解压交叉编译器压缩包并列出当前目录下的文件，查看是否解压成功
添加信息	无
参考图	

步骤二：

命令	\$ vi ~/.bashrc
相关描述	把交叉编译器的路径添加到系统环境变量 PATH 里面
添加信息	在文件最后添加路径 ^①
参考图	<pre>export PATH=/home/lvmh/work/gcc-linaro-arm-linux-gnueabi-4.7-2012.12-20121214_linux/bin:\$PATH</pre>

步骤三：

命令	\$ source ~/.bashrc
相关描述	使新的环境变量生效
添加信息	无
参考图	无

步骤四：

命令	\$ arm-linux-gnueabi-gcc -v
相关描述	检查交叉编译器是否安装成功
添加信息	无

注：

① export PATH=/home/lvmh/work/ gcc-linaro-arm-linux-gnueabi-4.7-2012.12-20121214_linux/bin:\$PATH。

参考图

```
lvmh@ubuntu:~/work$ arm-linux-gnueabi-gcc -v
Using built-in specs.
COLLECT_GCC=arm-linux-gnueabi-gcc
COLLECT_LTO_WRAPPER=/home/lvmh/work/gcc-linaro-arm-linux-gnueabi-4.7-2012.12-20121214_linux/bin/./libexec/gcc/arm-linux-gnueabi/4.7.3/lto-wrapper
Target: arm-linux-gnueabi
Configured with: /cbuild/slaves/oorts/crosstool-ng/builds/arm-linux-gnueabi-linux/.build/src/gcc-linaro-4.7-2012.12/configure --build=i686-build_pc-linux-gnu --host=i686-build_pc-linux-gnu --target=arm-linux-gnueabi --prefix=/cbuild/slaves/oorts/crosstool-ng/builds/arm-linux-gnueabi-linux/install --with-sysroot=/cbuild/slaves/oorts/crosstool-ng/builds/arm-linux-gnueabi-linux/install/arm-linux-gnueabi/libc --enable-languages=c,c++,fortran --enable-multilib --with-arch=armv7-a --with-tune=cortex-a9 --with-fpu=vfpv3-d16 --with-float=hard --with-pkgversion='crosstool-NG linaro-1.13.1-4.7-2012.12-20121214 - Linaro GCC 2012.12' --with-bugurl=https://bugs.launchpad.net/gcc-linaro --enable-__cxa_atexit --enable-libmudflap --enable-libgomp --enable-libssp --with-gmp=/cbuild/slaves/oorts/crosstool-ng/builds/arm-linux-gnueabi-linux/.build/arm-linux-gnueabi/build/static --with-mpfr=/cbuild/slaves/oorts/crosstool-ng/builds/arm-linux-gnueabi-linux/.build/arm-linux-gnueabi/build/static --with-mpc=/cbuild/slaves/oorts/crosstool-ng/builds/arm-linux-gnueabi-linux/.build/arm-linux-gnueabi/build/static --with-ppl=/cbuild/slaves/oorts/crosstool-ng/builds/arm-linux-gnueabi-linux/.build/arm-linux-gnueabi/build/static --with-cloog=/cbuild/slaves/oorts/crosstool-ng/builds/arm-linux-gnueabi-linux/.build/arm-linux-gnueabi/build/static --with-libelf=/cbuild/slaves/oorts/crosstool-ng/builds/arm-linux-gnueabi-linux/.build/arm-linux-gnueabi/build/static --with-host-libstdcxx='-L/cbuild/slaves/oorts/crosstool-ng/builds/arm-linux-gnueabi-linux/.build/arm-linux-gnueabi/build/static/lib -lpwl' --enable-threads=posix --disable-libstdcxx-pch --enable-linker-build-id --ena
```

六、搭建 NFS 网络文件系统

6.1 概述

NFS 是 Network FileSystem 的简称，可以让不同的主机通过网络访问远端的 NFS 服务器共享出文件以供开发板使用。这样，将主机当作 NFS 服务器，我们就可以在开发板上通过网络访问主机的文件。

6.2 安装步骤

具体的步骤如下表：

步骤一：

命令	\$ sudo apt-get install nfs-kernel
相关描述	下载并安装
添加信息	无
参考图	无

步骤二：

命令	\$ sudo vi /etc/exports
相关描述	修改/etc/exports文件末尾，创建共享目录
添加信息	在文件末尾加入 ^①

注：

① / nfs *(rw, sync, no_subtree_check, no_root_squash)。

参考图	<pre># /etc/exports: the access control list for filesystems which may be exported # to NFS clients. See exports(5). # # Example for NFSv2 and NFSv3: # /srv/homes hostname1(rw,sync,no_subtree_check) hostname2(ro,sync,no_subtr ee_check) # # Example for NFSv4: # /srv/nfs4 gss/krb5i(rw,sync,fsid=0,crossmnt,no_subtree_check) # /srv/nfs4/homes gss/krb5i(rw,sync,no_subtree_check) # /nfs *(rw,sync,no_root_squash,no_subtree_check) ~</pre>
-----	--

步骤三：

命令	\$ sudo service nfs-kernel-server start
相关描述	启动NFS服务器，nfs-kernel-server服务
添加信息	无
参考图	<pre>lvmh@ubuntu:~/work\$ sudo service nfs-kernel-server start * Exporting directories for NFS kernel daemon... [OK] * Starting NFS kernel daemon [OK]</pre>

步骤四：

命令	\$ sudo service portmap start
相关描述	启动NFS服务器，启动portmap服务
添加信息	无
参考图	<pre>lvmh@ubuntu:~/work\$ sudo service portmap start portmap start/running, process 3453</pre>

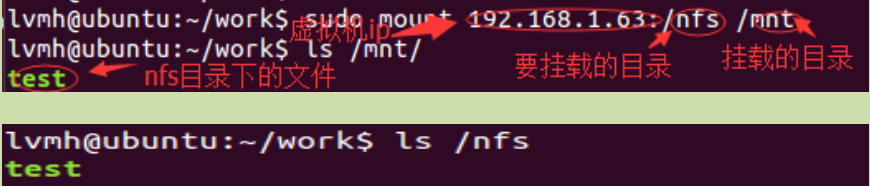
步骤五：

命令	<ol style="list-style-type: none"> 1、\$ sudo mount 192.168.1.63^① :/ /mnt 2、\$ ls /mnt
相关描述	挂载到主机根目录并查看是否挂载成功 ^②

注：

① 此处根据实际情况写虚拟机的 ip 地址。

② 在/mnt 目录下查看到根目录下的内容，则说明 NFS 配置正确。

添加信息	无
参考图	 <p>The screenshot shows a terminal session on an Ubuntu system. The first command is <code>sudo mount 192.168.1.63:/nfs /mnt</code>. Red arrows point to <code>192.168.1.63</code> (labeled '虚拟ip'), <code>/nfs</code> (labeled '要挂载的目录'), and <code>/mnt</code> (labeled '挂载的目录'). The second command is <code>ls /mnt/</code>, which outputs <code>test</code>. A red arrow points to <code>test</code> (labeled 'nfs目录下的文件'). The third command is <code>ls /nfs</code>, which also outputs <code>test</code>.</p>

Note:

如果开发板不能挂载到宿主机可以试下以下方式重新启动 NFS 服务器：

```
sudo exportfs -a
```

```
sudo /etc/init.d/portmap restart
```

```
sudo /etc/init.d/nfs-kernel-server restart
```

七、源码编译

7.1 概述

源码包含 u-boot，内核与文件系统。其中 u-boot 和内核需要进行编译，文件系统需要制作。本章详细介绍如何编译 u-boot 以及内核。如何制作文件系统请参考第九章。

7.2 u-boot 编译

将光盘中如下图所示的 u-boot 源码拷贝到~/work 工作目录中（这里的源码名字只是暂时的别名，源码名称及版本可能更新，以光盘源码为准）：

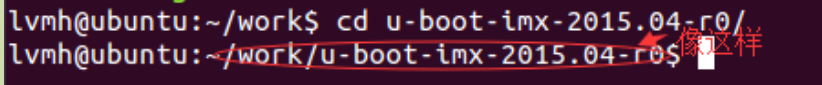
名称	修改日期	类型
 u-boot.tar	2016/8/19 15:50	好压 GZ 压缩文件

具体的步骤如下：

步骤一：

命令	1、\$ tar -xzf u-boot.tar.gz 2、\$ ls
相关描述	解压源代码, 并查看是否解压成功
添加信息	无
参考图	

步骤二：

命令	\$ cd u-boot-imx-2015.04-r0
相关描述	进入源代码目录
添加信息	无
参考图	

步骤三：

命令	\$ make distclean
相关描述	清除已生成文件
添加信息	无
参考图	

步骤四：

命令	\$ make mx6ul_qiyang_defconfig^①
相关描述	配置板子
添加信息	无

注：

① 配置uboot的命令是-make <board_name>_deconfig,我们现在板子的名字是mx6ul_qiyang。

参考图	<pre> lvmh@ubuntu:~/work/u-boot-imx-2015.04-r0\$ make distclean CLEAN scripts/basic CLEAN scripts/kconfig CLEAN include/config include/generated CLEAN .config .config.old include/autoconf.mk include/autoconf.mk.dep include /config.h lvmh@ubuntu:~/work/u-boot-imx-2015.04-r0\$ make mx6ul_qiyang_defconfig HOSTCC scripts/basic/fixdep HOSTCC scripts/kconfig/conf.o SHIPPED scripts/kconfig/zconf.tab.c SHIPPED scripts/kconfig/zconf.lex.c SHIPPED scripts/kconfig/zconf.hash.c HOSTCC scripts/kconfig/zconf.tab.o HOSTLD scripts/kconfig/conf # # configuration written to .config # </pre>
-----	--

步骤五：

命令	\$ make -j8 ^①
相关描述	编译
添加信息	无
参考图	<pre> OBJCOPY u-boot.bin OBJCOPY u-boot.srec CFGS board/freescale/mx6ul_qiyang/imximage.cfg.cfgtmp MKIMAGE <u>u-boot.imx</u> ← 生成的uboot镜像，要用来烧写 </pre>

Note:

若编译不成功，请尝试以下做法：

1、**\$ vi build.sh**

```

sudo make distclean
export ARCH=arm
export CROSS_COMPILE=/home/lvmh/work/gcc-linaro-arm-linux-gnueabi-hf-4.7-2012.12-20121214_linux/bin/arm-linux-gnueabi-hf-
make mx6ul_qiyang_defconfig

make -j4
sleep 2
sudo chmod 777 u-boot.imx

```

把交叉编译器路径改为实际路径

修改完后：

\$./build.sh

2、**\$ vi Makefile**

注：

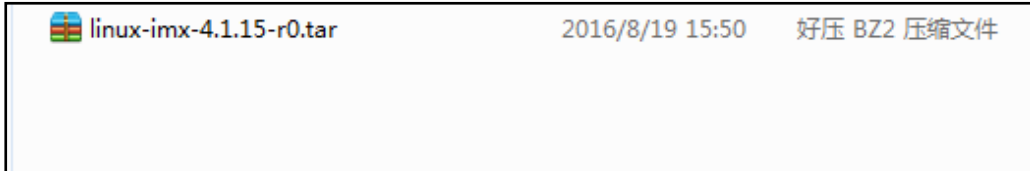
① 执行之后开始编译，编译过程可能持续 1~3 分钟，终端会打印出一大堆信息，最后的一条信息表示生成了可用的 uboot 镜像。

```
ifneq ($HOSTARCH, $ARCH)
ARCH := arm 若没有这句话, 请添加上去
CROSS_COMPILE := /home/lvmh/work/gcc-linaro-arm-linux-gnueabi-4.7-2012.12-20121214_linux/bin/arm-linux-gnueabi-
endif 修改为实际交叉编译器路径
```

修改完后重复 uboot 编译步骤。


7.3 内核的配置及编译

进入~/work目录，将如下图所示的linux-imx-4.1.15-r0.tar.bz2 这个内核源码拷贝到此目录下（这里的源码名字只是暂时的别名，源码名称及版本可能更新，以光盘源码为准）：

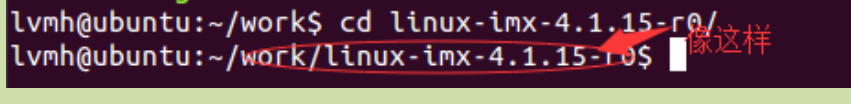


具体配置及编译的步骤如下：

步骤一：

命令	1、\$ sudo tar -xjvf linux-imx-4.1.15-r0.tar.bz2 2、\$ ls
相关描述	解压内核源代码
添加信息	无
参考图	

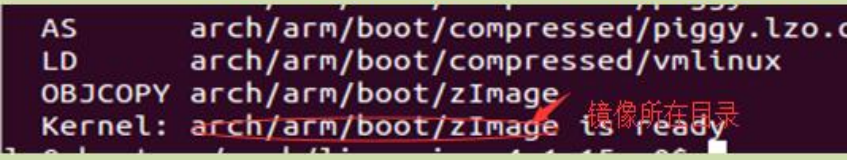
步骤二：

命令	\$ cd
相关描述	进入内核源码目录
添加信息	无
参考图	

步骤三：

命令	\$ make menuconfig ^①
相关描述	配置内核
添加信息	无
参考图	

步骤四：

命令	\$ make zImage dtbs ^②
相关描述	编译内核以及设备树
添加信息	无
参考图	

编译完成后在 arch/arm/boot/ 目录下生成可烧写到开发板的内核镜像文件 zImage, 在 arch/arm/boot/dts 目录下生成内核设备树镜像文件 imx6ul-qiyang.dtb。

注：

① 执行后会出现如图内核配置界面，用户在此界面对内核进行配置配置好之后保存退出。退出的时候，请选择“YES”来保存配置。

② make zImage 是编译内核镜像，make dtbs 编译内核设备树。

Note:

若编译不成功，请进行以下操作：

1、\$ vi Makefile

```
#ARCH           ?= $(SUBARCH)
#CROSS_COMPILE ?= $(CONFIG_CROSS_COMPILE:"%"=%)
ARCH            := arm
CROSS_COMPILE  := arm-linux-gnueabi- ← 修改为实际交叉编译器路径
```

修改完成后重新编译。

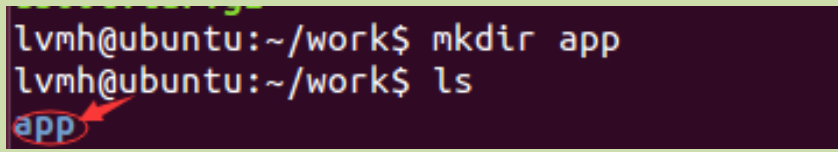
八、应用程序的开发

8.1 编写应用程序及交叉编译

您可以在 linux 上自由地开发应用程序，这里以最简单的 Hello World 为例。
进入~/work 目录。

具体的步骤如下：

步骤一：

命令	<code>\$ mkdir app</code>
相关描述	创建 app 文件夹
添加信息	无
参考图	

步骤二：

命令	<code>\$ cd app</code>
相关描述	进入 app 文件夹
添加信息	无
参考图	

步骤三：

命令	<code>\$ vi hello.c^①</code>
----	--

注：

① 执行后显示空白页面，在此页面中输入程序，如图 5-1 所示。

相关描述	编译应用程序
添加信息	无
参考图	<pre> #include <stdio.h> int main(void) { printf("Hello World ! \n"); return 0; } </pre>

步骤四：

命令	<pre> \$ arm-linux-gnueabi-gcc -o hello hello.c^① \$ ls </pre>
相关描述	编译应用程序并查看是否生成可执行文件
添加信息	无
参考图	<pre> lvmh@ubuntu:~/work/app\$ arm-linux-gnueabi-gcc -o hello hello.c lvmh@ubuntu:~/work/app\$ ls hello hello.c </pre> <p>编译后得到的可执行程序</p>

8.2 运行应用程序

这样我们就得到了可以在开发板上执行的程序了，下面介绍如何在主板运行应用程序，有两种方法：挂载 NFS 服务器和添加到文件系统^②。本小节主要介绍挂载 NFS 服务器方式。在[虚拟终端软件 Xshell](#)上进行。

我们可以使用之前配置好的 NFS 服务器^③，在开发板上挂载主机的 NFS 服务器后，就可以开发板上操作主机上的文件，如复制文件、运行程序等，这种方式十分方便于调试。

具体的步骤如下：

注：

- ① 需要使用之前安装好的交叉编译器对应用程序进行编译。
- ② 如何添加请查看六、添加应用程序到文件系统。
- ③ 确保通过网线将开发板与 PC 机连接好，并在 PC 机上开启了 NFS 服务。

步骤一：

命令	\$ ifconfig eth0 192.168.1.106
相关描述	设置开发板 ip 以及子网掩码
添加信息	无
参考图	<pre>root@qy_mx6ul:~# ifconfig eth0 192.168.1.106 root@qy_mx6ul:~# ifconfig eth0 Link encap:Ethernet HWaddr 08:00:3E:26:0A:5B inet addr:192.168.1.106 Bcast:192.168.1.255 Mask:255.255.255.0 UP BROADCAST MULTICAST MTU:1500 Metric:1 RX packets:0 errors:0 dropped:0 overruns:0 frame:0 TX packets:0 errors:0 dropped:0 overruns:0 carrier:0 collisions:0 txqueuelen:1000 RX bytes:0 (0.0 B) TX bytes:0 (0.0 B)</pre>

步骤二：

命令	\$ route add default gw 192.168.1.1
相关描述	设置网关
添加信息	无
参考图	<pre>root@qy_mx6ul:~# route add default gw 192.168.1.1</pre>

步骤三：

命令	# ping 192.168.1.63
相关描述	Ping 虚拟机 ip，测试网络连接
添加信息	无

参考图	<pre> root@qy_mx6ul:~# ping 192.168.1.63 PING 192.168.1.63 (192.168.1.63): 56 data bytes 64 bytes from 192.168.1.63: seq=0 ttl=64 time=2.226 ms 64 bytes from 192.168.1.63: seq=1 ttl=64 time=0.719 ms 64 bytes from 192.168.1.63: seq=2 ttl=64 time=0.704 ms 64 bytes from 192.168.1.63: seq=3 ttl=64 time=0.833 ms 64 bytes from 192.168.1.63: seq=4 ttl=64 time=0.716 ms 64 bytes from 192.168.1.63: seq=5 ttl=64 time=0.693 ms 64 bytes from 192.168.1.63: seq=6 ttl=64 time=0.728 ms 64 bytes from 192.168.1.63: seq=7 ttl=64 time=0.718 ms 64 bytes from 192.168.1.63: seq=8 ttl=64 time=0.749 ms 64 bytes from 192.168.1.63: seq=9 ttl=64 time=0.747 ms 64 bytes from 192.168.1.63: seq=10 ttl=64 time=0.722 ms --- 192.168.1.63 ping statistics --- 11 packets transmitted, 11 packets received, 0% packet loss round-trip min/avg/max = 0.693/0.868/2.226 ms </pre>
-----	---

步骤四：

命令	# mount -o nolock 192.168.1.63^① /mnt
相关描述	挂载主机 NFS 服务器
添加信息	无
参考图	<pre> root@qy_mx6ul:~# mount -o nolock 192.168.1.63:/nfs /mnt </pre>

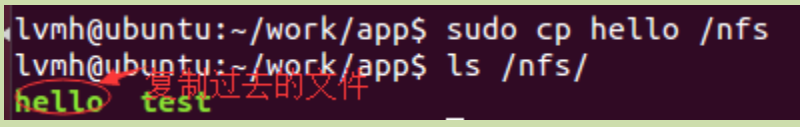
步骤五：

命令	# cd /mnt
相关描述	查看是否挂载成功
添加信息	无
参考图	<pre> root@qy_mx6ul:/mnt# ls test </pre> <p>nfs目录下的文件</p>

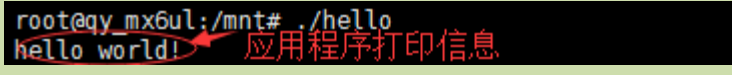
注：

① 192.168.1.63 为虚拟机 IP。

步骤六：

命令	<code>\$ sudo cp hello /</code>
相关描述	在 ubuntu 中 hello.c 所在目录下，将 hello 复制到根目录下
添加信息	无
参考图	

步骤七：

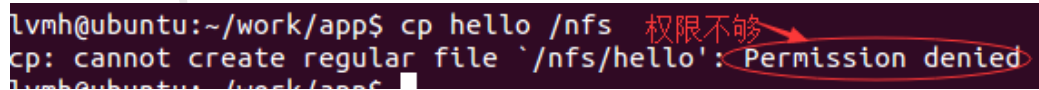
命令	<code># ./hello</code>
相关描述	运行可用程序
添加信息	无
参考图	

通过这种方式，可以方便调试程序，直到程序调试正确后，再将应用程序添加到文件系统中，烧写到 EMMC。这样，避免了不断的烧写程序到 EMMC。

Note:

若出现如下图所示错误，请在命令前加 sudo：

`$ sudo cp hello /`



九、添加应用程序到文件系统

9.1 概述

为了生产方便，一般我们都把编译测试好的应用程序、库以及配置文件文件打包到文件系统当中，这样生产的时候只要烧写文件系统就可以了，不需要手动去添加应用程序、库以及配置文件。

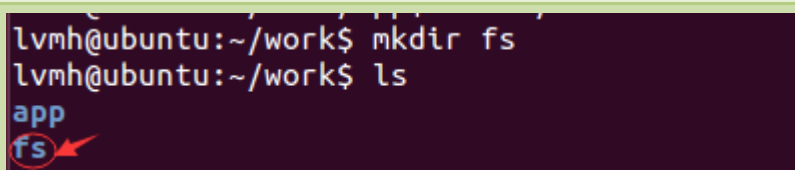
9.2 添加到文件系统

进入~/work 目录，将如下图所示文件系统源码拷贝到此目录下（这里的源码名字只是暂时的别名，源码名称及版本可能更新，以光盘源码为准）。

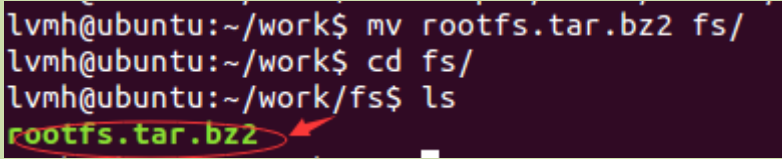
名称	修改日期	类型
 rootfs.tar	2016/8/19 15:50	好压 BZ2 压缩文件

添加步骤如下：

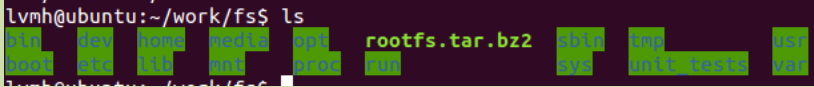
步骤一：

命令	<code>\$ mkdir fs</code>
相关描述	创建 fs 文件夹
添加信息	无
参考图	

步骤二：

命令	<pre>\$ mv rootfs.tar.bz2 fs/ \$ cd fs \$ ls</pre>
相关描述	将文件系统源码 rootfs.tar.bz2 移动到 fs 文件夹中, 进入 fs 目录并查看是否移成功
添加信息	无
参考图	 <pre>lvmh@ubuntu:~/work\$ mv rootfs.tar.bz2 fs/ lvmh@ubuntu:~/work\$ cd fs/ lvmh@ubuntu:~/work/fs\$ ls rootfs.tar.bz2</pre>

步骤三：

命令	<pre>1、\$ sudo tar -xjvf rootfs.tar.bz2^{①②} 2、\$ ls</pre>
相关描述	解压 rootfs 源码, 并查看是否解压成功
添加信息	无
参考图	 <pre>lvmh@ubuntu:~/work/fs\$ ls bin dev home media opt rootfs.tar.bz2 sbin tmp usr boot etc lib mnt proc run sys unit-tests var</pre>

步骤四：

命令	<pre>1、\$ rm rootfs.tar.bz2 2、\$ ls</pre>
相关描述	删除原先的文件系统 rootfs. tar. bz2, 查看是否删除
添加信息	无

注：

- ① 文件系统源码需要root权限才能进行完整解压, 若当前为普通用户, 在解压命令之前加上sudo。
- ② 在 rootfs 目录下各目录添加自己的应用程序、库以及配置文件。

参考图	<pre>lvmh@ubuntu:~/work/fs\$ sudo rm rootfs.tar.bz2 lvmh@ubuntu:~/work/fs\$ ls bin dev home media opt run sys unit_tests var boot etc lib mnt proc sbin tmp usr</pre>
-----	--

步骤五：

命令	<ol style="list-style-type: none"> 1、\$ sudo tar -jcvf rootfs.tar.bz2 * -R 2、\$ ls
相关描述	重新压缩文件系统
添加信息	无
参考图	<pre>lvmh@ubuntu:~/work/fs\$ ls bin dev home media opt rootfs.tar.bz2 sbin tmp usr boot etc lib mnt proc run sys unit_tests var</pre>

十、结束语

IAC-IMX6UL-Kit 的相关操作介绍到这里就结束了。感谢各位用户使用本公司产品，如果有什么问题，请联系技术支持：supports@qiyangtech.com。