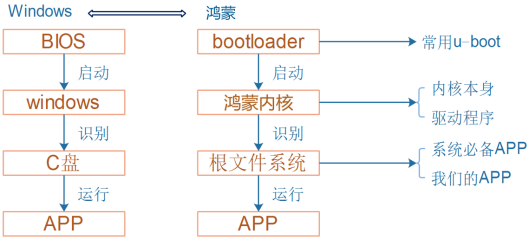
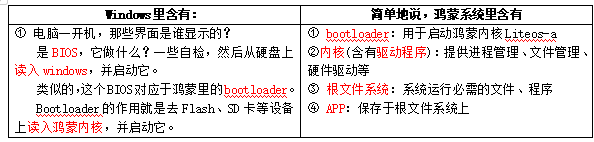
# 实验一环境搭建与鸿蒙LiteOS-a内核体验

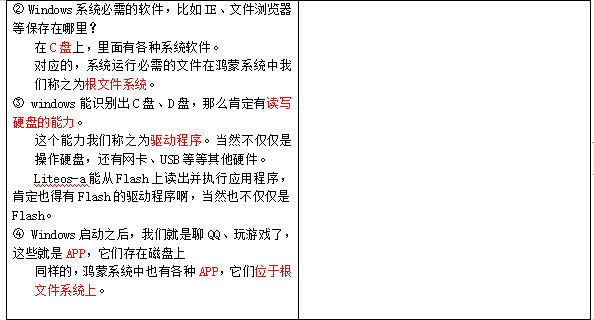
1. 实验目的
2. 实验内容

1.1 鸿蒙系统有哪些内容？

鸿蒙系统，就相当于一套完整的PC软件系统。







1.2 日常工作中开发流程是怎样？

1.2.1 鸿蒙的IDE

鸿蒙的IDE分为南向IDE、北向IDE。上北下南，北向只APP，南向指内核。

南向IDE的下载地址：https://device.harmonyos.com/cn/ide 它的功能将会很强大，但是目前状况为：

可以在Windows/Linux下阅读源码

但是编译的话，只能在Linux下使用命令行来编译

可以使用Jlink来调试，目前只支持在Windows下调试

这个IDE已经支持IMX6ULL的调试了，后面需要的话我会介绍调试功能。

1.2.2 我们的开发环境

Bootloader、内核、APP等等软件，需要在Ubuntu中编译；但是阅读、修改这些源码时，在Windows下会比较方便。

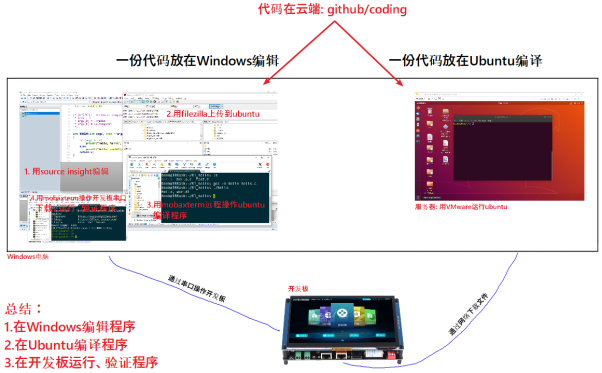
所以，我们需要在Windows、Ubuntu上都存有源码。

① 在Windows上阅读、研究、修改(使用Source insight会很方便)，修改后上传到Ubuntu(使用Filezilla)

②在Ubuntu上编译、制作(使用MobaXterm远程登录Ubuntu会很方便)

③把制作好的可执行程序下载到开发板上运行、测试(使用MobaXterm连接开发板的串口)。

在整个开发过程中，我们会用到Windows、Ubuntu、开发板，如下图所示：



1.3 搭建开发环境需要做哪些事情

这里只是列出要做的事情：（很多是Linux的重复做的，去看i.mx6ull开发中的文章，会讲解到这些环境是如何配置的）

安装VMware、下载Ubuntu映象

安装Windows上各个APP 这些APP有：SouceInsight、FileZilla、MobaXterm、Notepad++

下载源码和工具链 使用repo命令在Ubuntu中下载，后面介绍。

连接开发板

① 连接电源线到开发板， ② 开发板的串口线，接到Windows电脑；并用MobaXterm连接串口。

烧写系统 开发过程会不断编译程序、烧写、测试。

1.4 使用NAT配置Ubuntu网络

1.4.1 设置Ubuntu使用NAT网络

我们配置网络是为了方便后续调试开发板系统或者应用程序时使用tftp协议 nfs协议等拷贝文件或者设置文件系统启动方式为nfs挂载启动。

以前我们在VMware中使用桥接，这需要保证Windows、Ubuntu、开发板三者互通，而很多人的电脑有多个网卡，这涉及的网络设置太复杂了。

现在我们使用NAT方式，这样会简单很多。

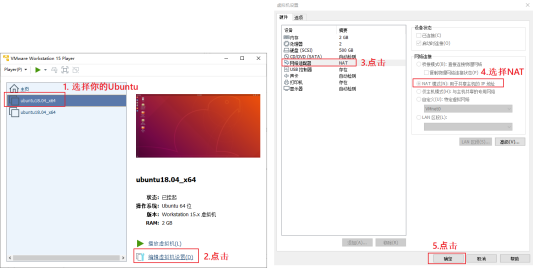
NAT是什么意思？Network Address Translation，网络地址转换。

举个例子，在NAT里，Windows就是一个爱护孩子的父亲，Ubuntu就是受保护的小孩。小孩要买东西，都由他父亲代劳，别人根本不知道这小孩的存在；亲戚想跟小孩谈话，也要经过父亲中转，亲戚不能直接跟小孩接触。

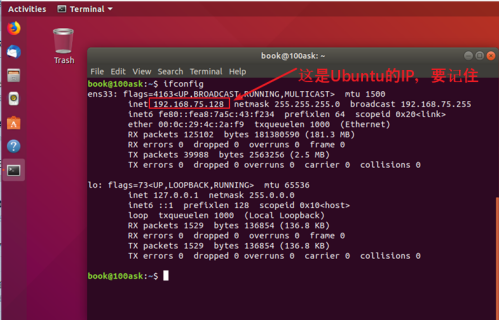
在NAT里，Ubuntu要主动发起网络访问，需要经过Windows代劳，外面的设备只看到Windows发出的网络数据；Windows收到数据的再转发给Ubuntu。

在NAT里，外面的设备比如开发板要访问Ubuntu，也只能通过Windows代劳：开发板是看不到Ubuntu的，开发板是ping不通Ubuntu的。

先关闭VMware中的Ubuntu，在VMware界面设置Ubuntu使用NAT，如下图：



然后启动Ubuntu，打开终端，执行ifconfig命令，确定网卡IP，如下图所示，记住这个IP(192.168.75.128，也许你的跟它不一样)，后面要用：

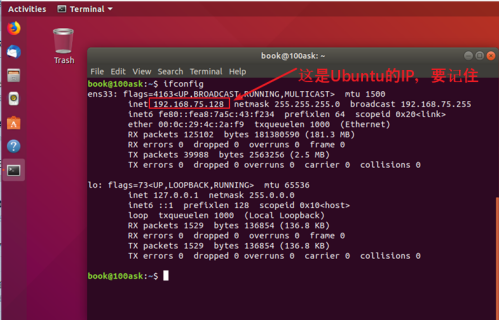


1.4.2 Ubuntu的IP可能突然变化

你用着用着，某次重新开机后，发现无法访问Ubuntu了。可能是Ubuntu的ip变了,这就需要使用新IP来访问Ubuntu。

怎么确定Ubuntu的IP？

启动Ubuntu，打开终端，执行ifconfig命令，确定网卡IP，如下图所示：



1.4.3 验证网络

注意：如果照着本节操作不成功，请确认：

①在VMware里是否设置Ubuntu使用了NAT网络；

②使用的UbuntuIP是否正确

1.4.3.1 几个概念

IP地址127.0.0.1表示的是“本机”，在Windows上使用127.0.0.1表示的是Windows机器；在开发板上使用127.0.0.1表示的是开发板；在Ubuntu里使用127.0.0.1表示的是Ubuntu。ping

127.0.0.1是永远成功的，它不涉及具体网卡。 在本文中，在Windows上安装VMware，在VMware里运行Ubuntu，Ubuntu的网络都是使用NAT(Network

Address Translation，网络地址转换）。 NAT是什么意思？

举个例子，在NAT里，Windows就是一个爱护孩子的父亲，Ubuntu就是受保护的小孩。小孩要买东西，都由他父亲代劳，别人根本不知道这小孩的存在；亲戚想跟小孩谈话，也要经过父亲中转，亲戚不能直接跟小孩接触。

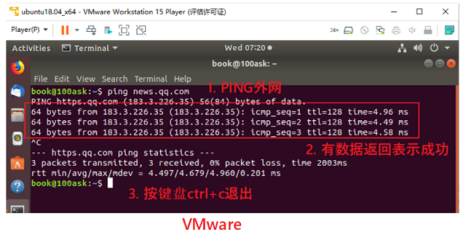
在NAT里，Ubuntu要主动发起网络访问，需要经过Windows代劳，外面的设备只看到Windows发出的网络数据；Windows收到数据的再转发给Ubuntu。

在NAT里，外面的设备比如开发板要访问Ubuntu，也只能通过Windows代劳：开发板是看不到Ubuntu的，开发板是ping不通Ubuntu的。

1.4.3.2 验证Ubuntu的网络

使用NAT时，不需要我们去设置Ubuntu的网络，它会自动获得IP。只要你的Windows电脑可以上网，Ubuntu就可以访问外网。

你可以在Ubuntu中执行“ping news.qq.com”，肯定可以看到有数据返回，同时按键盘“ctrl+c”退出ping命令，如下图。



1.4.3.3 验证Windows和Ubuntu之间的网络

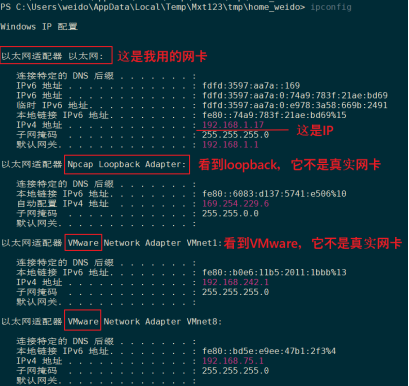
使用VMware时，在Windows命令行可以ping通Ubuntu的IP；Ubuntu也可以ping通Windows的IP。

只要有一个方向能ping通，那证明Windows和Ubuntu之间的网络是通的。

1. Ubuntu ping Windows：

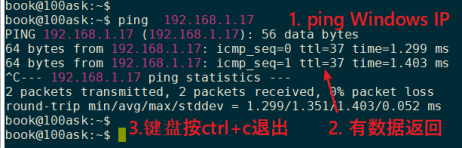
以Ubuntu ping Windows为例，先确定Windows IP，再在Ubuntu中ping Windows(如果Windows开了防火墙，可以无法ping通Windows，这没关系)：

怎么确定Windows的IP？在Powershell命令行或Windows命令行中执行ipconfig，如下图：



这时就

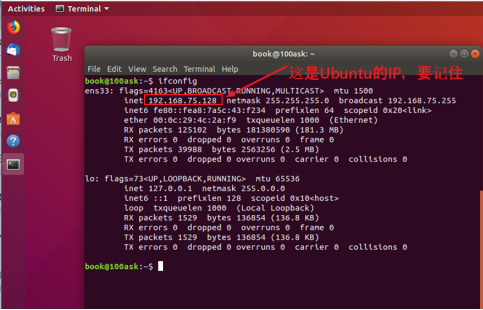
可以在Ubuntu命令行中ping windows了，如下：



2. Windows ping Ubuntu：

注意：如果Windows开了防火墙，Ubuntu也无法ping通Windows，但是没关系，只要Ubuntu能ping通外网就表示网络没问题。

先确定Ubuntu的IP，打开终端，执行ifconfig命令，确定网卡IP，如下图所示，记住这个IP(192.168.75.128，也许你的跟它不一样)，后面要用：



启动Windows命令行，ping Ubuntu的IP即可，如下图：



1.4.4 常见问题

1.4.4.1 Ubuntu不能ping通外网

如果能“ping 8.8.8.8”成功，但是“ping news.qq.com”失败，这就是DNS的问题。一般是不需要手工设置DNS的，但是如果有此现象，就需要设置DNS。

修改Ubuntu的/etc/resolv.conf文件，如下：

nameserver 127.0.0.1(或其他IP)

改为：

nameserver 8.8.8.8

1

2

3

1.4.4.2 Ubuntu无法PING通Windows

如果Ubuntu无法ping通Windows，基本上是Windows安装了杀毒软件或是打开了防火墙。 首先关闭杀毒软件，然后关闭防火墙。

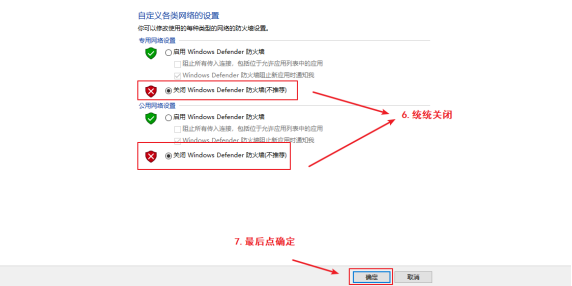
步骤如下图所示：





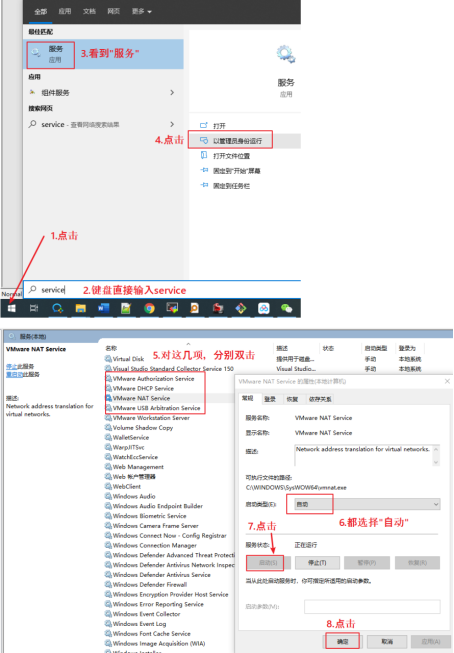






1.4.4.3 VMwre中的Ubuntu网络不可用

如果关闭Windows防火墙，Ubuntu还是有网络问题，比如无法获得IP、无法上网、无法PING通Windows。那么确认一下VMware服务是否启动，如下图所示：

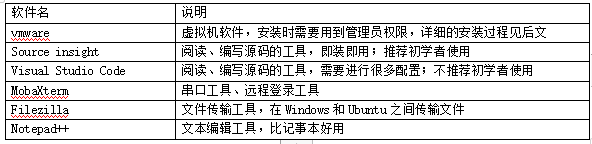


然后关闭Ubuntu、关闭VMware、重启VMware、重启Ubuntu。

1.6 给unbuntu配上鸿蒙的环境

1.6.1 安装Windows软件

在“开发板配套资料\01\_Tools (工具)\01\_Tools.zip”中，解压可以得到一系列的安装软件，建议全部安装。有如下软件：



Visual Studio Code的配置比较麻烦，建议初学者使用Source insight来阅读、编写源码。

注意：上面很多软件是和安装linux是相同的。

1.6.2 安装Ubuntu软件

确保Ubuntu能上网之后，使用下面命令一键配置/初始化开发环境(其实就是安装tftp，nfs，vim等软件，此脚本只支持Ubuntu-16.04 /Ubuntu-18.04)。 注意：为了方便大家复制，这些命令写成了多行，你要把它们复制到记事本，合并成一行，注意空格：

book@100ask: ~ $ wget --no-check-certificate -O Configuring\_ubuntu.sh

https://weidongshan.coding.net/p/DevelopmentEnvConf/d/DevelopmentEnvConf/git/raw/master/Configuring\_ubuntu.sh && sudo chmod +x Configuring\_ubuntu.sh && sudo ./Configuring\_ubuntu.sh

1

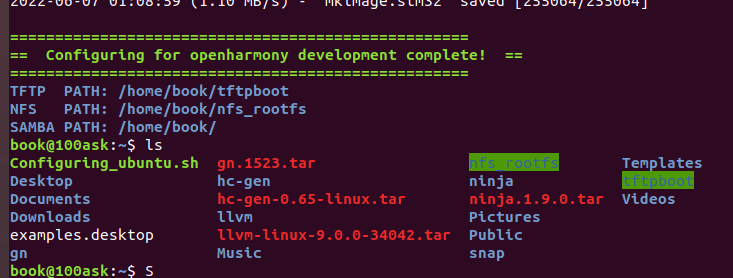
2

上述命令中，wget是下载脚本，chmod是给它添加可执行权限，最后运行它。

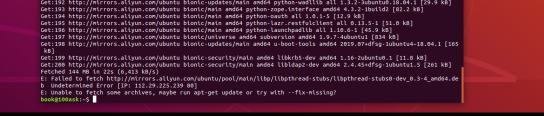
按提示输入book密码123456和选择对应的系统，如下图所示：



如果命令执行到最后，出现类似这样的错误，就重新执行命令：



如果执行该命令出现如下错误：



可以先执行“sudo apt-get update”，再重新执行前面的命令。

1. 在IMX6ULL上体验鸿蒙系统

下载GIT仓库，里面含有烧写软件。

本文先让大家在IMX6ULL上体验一下Liteos-a。

百问网开发了一款烧写软件：100ask\_imx6ull\_flashing\_tool，它的界面如下：

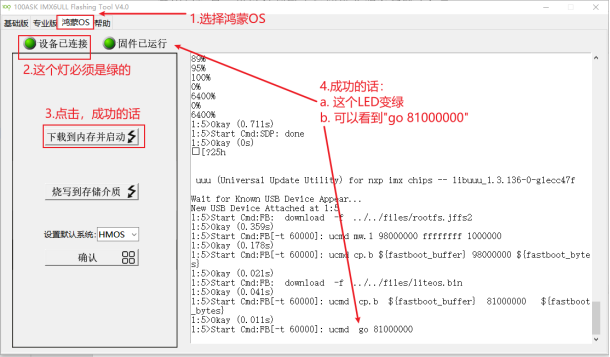


使用这软件，只需要一条USB线连接电脑和开发板USB OTG口，只需要点击一个按钮就可以体验鸿蒙系统。

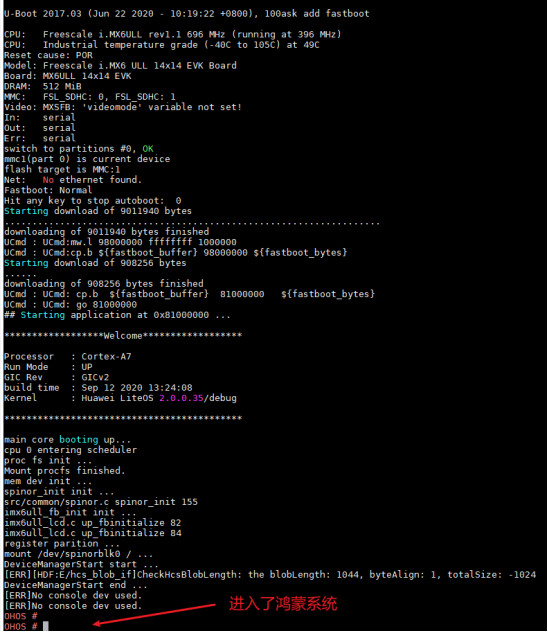
1.1 一键体验鸿蒙：下载到内存运行

1.1.1 一键启动

把开发板设置为USB启动，接好2条USB线，装好驱动程序后，运行烧写工具，点击下图所示按钮，观察串口信息，可以看到板子启动进入鸿蒙系统了：

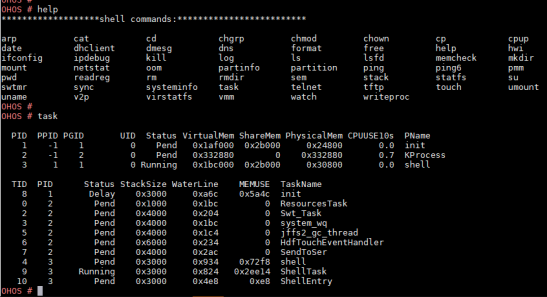


串口信息如下：



1.1.2 执行shell命令

执行help命令，可以看到支持的SHELL命令，如下：



1.1.3 执行数码相框GUI程序

注意：必须用“./bin/digitpic”，不能用绝对路径“/bin/digitpic”

注意：这个GUI程序是我们自己写得，很丑，与鸿蒙无关。

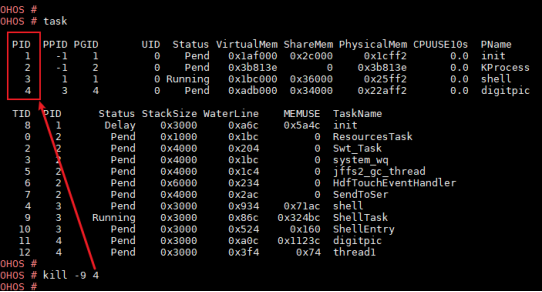
在这里插入图片描述

在板子屏幕上可以看到：



1.1.4 退出程序

执行task命令确定进程号，然后执行“kill -9 PID”杀掉进程，比如：

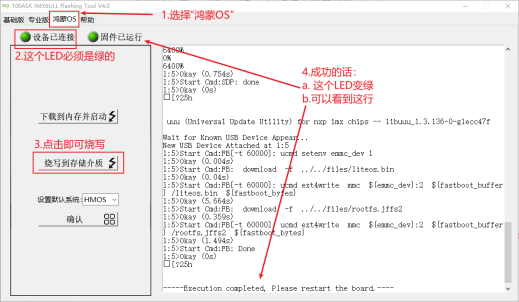


1.1.5 开机自动启动鸿蒙

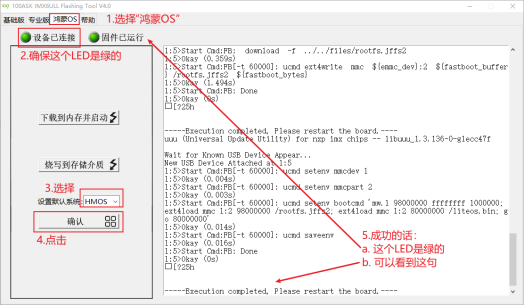
注意：对于IMX6ULL pro板，设置为HMOS启动后，无法再通过烧写工具切回Linux(下一个版本完善这功能)。要想切回Linux，需要重新烧录EMMC，这个烧写工具里没有放emmc.bin(它太大了)。

可以使用百度网盘中“01\_Tools (开发工具)/100ask\_imx6ull\_pro开发板系统烧写工具.7z”来重新烧录EMMC。

把开发板设置为USB启动，接好2条USB线，装好驱动程序后，运行烧写工具. 先烧写，点击下图所示按钮：



然后设置默认系统，如下图所示：



最后，设置为EMMC启动，重新上电后就可以自动进入鸿蒙系统。