

task2

姓名: 吕东航 学号: 3120104529 专业: 计科

课程名称: Embedded System

实验时间: 2015-3-19 实验地点: 曹西-501 指导老师: 翁恺

一、实验目的和要求

查询 PCduino 网卡的物理 MAC 地址;

写一个 PC 程序, Unix 或 windows, 能够能通过对局域网内所有 IP 地址的 ping, 找到 arp 表里的 MAC 地址和 IP 地址 对应关系, 根据 PCduino 的 MAC 段找到局域网内的 RPi 的 IP 地址。

二、主要仪器设备

硬件

实验主板一块 (WRTnode);

5V/1A 电源一个;

microUSB 线一根;

USB-TTL 串口线一根 (FT232RL 芯片或 PL2303 芯片)。

以下为自备 (可选) 器材:

PC (Windows/Mac OS/Linux) 一台;

以太网线一根 (可能还需要路由器等);

HDMI 显示器;

HDMI 线;

USB 键盘/鼠标 (无线);

USB Hub, 推荐自带供电的类型。

软件

PC 上的 USB-TTL 串口线配套的驱动程序;

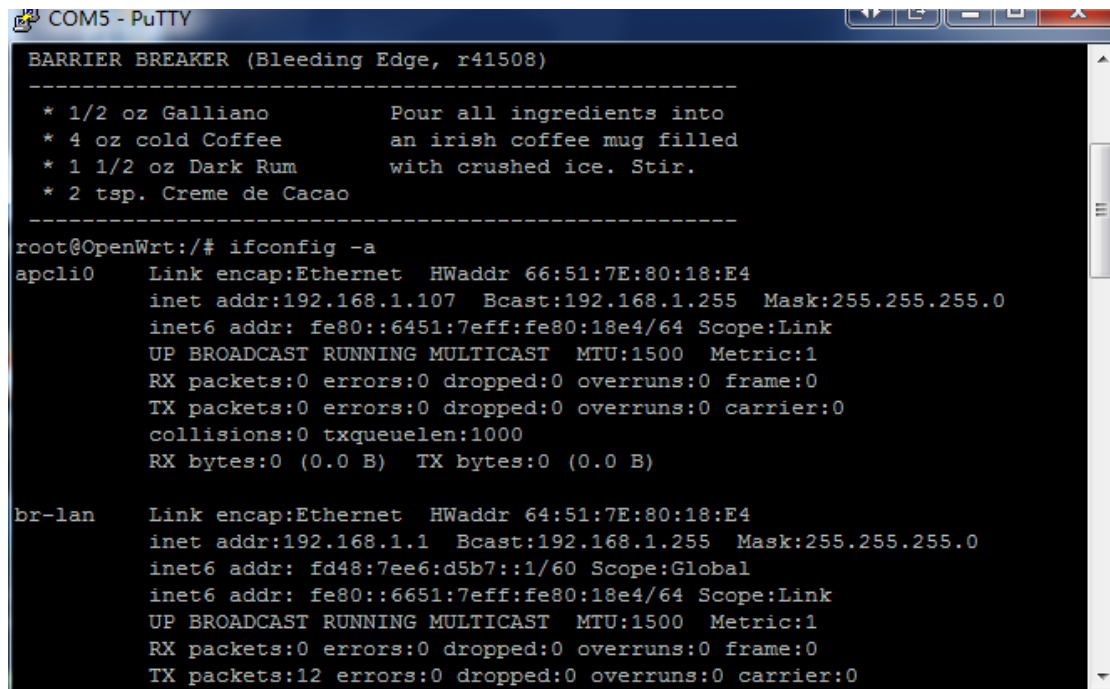
PC 上的串口终端软件, 如 minicom、picocom、putty 等;

PC 上的 SSH 软件, 如 putty 等。

三、操作方法与实验步骤

首先为了验证我们的脚本工作，我们现在 wrtnode 上查询了这块板子的 mac 地址，确认地本次试验采用的是 wrtnode 板子作为路由，pc 连接到 wrtnode 上，所以理论上 ip 应该是 192.168.1.1

首先查询板子的 mac 地址，使用串口登陆板子并查询到地址为 64-51-7e-80-18-e4，如下图所示。



```
BARRIER BREAKER (Bleeding Edge, r41508)
-----
* 1/2 oz Galliano      Pour all ingredients into
* 4 oz cold Coffee    an irish coffee mug filled
* 1 1/2 oz Dark Rum   with crushed ice. Stir.
* 2 tsp. Creme de Cacao
-----

root@OpenWrt:/# ifconfig -a
apcli0      Link encap:Ethernet  HWaddr 66:51:7E:80:18:E4
            inet addr:192.168.1.107  Bcast:192.168.1.255  Mask:255.255.255.0
            inet6 addr: fe80::6451:7eff:fe80:18e4/64 Scope:Link
            UP BROADCAST RUNNING 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)

br-lan      Link encap:Ethernet  HWaddr 64:51:7E:80:18:E4
            inet addr:192.168.1.1    Bcast:192.168.1.255  Mask:255.255.255.0
            inet6 addr: fd48:7ee6:d5b7::1/60 Scope:Global
            inet6 addr: fe80::6651:7eff:fe80:18e4/64 Scope:Link
            UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
            RX packets:0 errors:0 dropped:0 overruns:0 frame:0
            TX packets:12 errors:0 dropped:0 overruns:0 carrier:0
```

接下来编写 python 脚本：

```
import os
import re
x = os.popen("arp -a").read()
#print x
sqliist = x.split("\n")
for sql in sqliist:
    #print sql
    findre = re.compile(r'(.*)64-51-7e-80-18-e4(.*)')
    for x in findre.findall(sql):
        print x[0]
```

该脚本执行了 windows 系统的 cmd 命令 arp -a 来获取 arp 表并存放在 x 中。
数据如下图所示：

接口: 192.168.1.100 --- 0xc

Internet 地址	物理地址	类型
192.168.1.1	64-51-7e-80-18-e4	动态
192.168.1.255	ff-ff-ff-ff-ff-ff	静态
224.0.0.2	01-00-5e-00-00-02	静态
224.0.0.22	01-00-5e-00-00-16	静态
224.0.0.252	01-00-5e-00-00-fc	静态
239.255.255.250	01-00-5e-7f-ff-fa	静态
255.255.255.255	ff-ff-ff-ff-ff-ff	静态

接口: 169.254.58.137 --- 0x1e

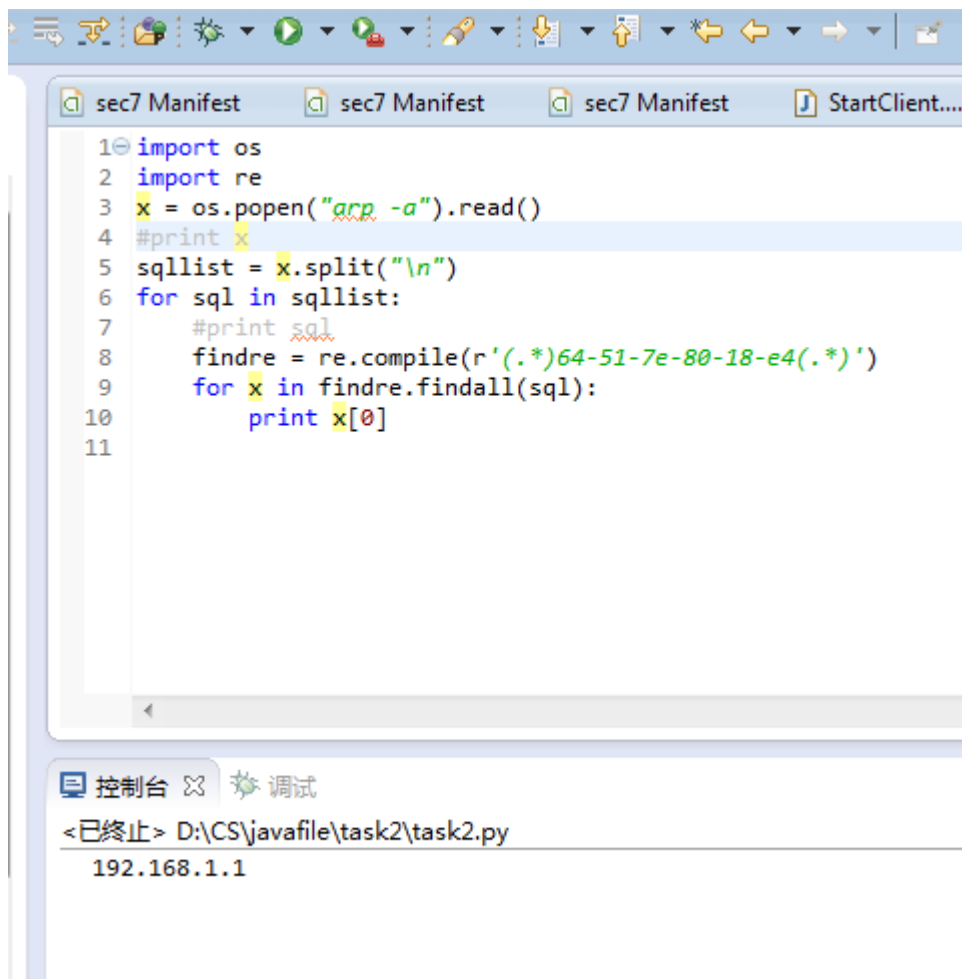
Internet 地址	物理地址	类型
169.254.255.255	ff-ff-ff-ff-ff-ff	静态
224.0.0.2	01-00-5e-00-00-02	静态
224.0.0.22	01-00-5e-00-00-16	静态
224.0.0.252	01-00-5e-00-00-fc	静态
239.255.255.250	01-00-5e-7f-ff-fa	静态
255.255.255.255	ff-ff-ff-ff-ff-ff	静态

接口: 169.254.232.15 --- 0x1f

Internet 地址	物理地址	类型
169.254.255.255	ff-ff-ff-ff-ff-ff	静态
224.0.0.2	01-00-5e-00-00-02	静态
224.0.0.22	01-00-5e-00-00-16	静态
224.0.0.252	01-00-5e-00-00-fc	静态
239.255.255.250	01-00-5e-7f-ff-fa	静态
255.255.255.255	ff-ff-ff-ff-ff-ff	静态

然后将获取的数据按行切分，正则匹配每行是否有对的 mac 地址，如果有，提取出来并输出对应的 ip。

最后程序运行结果如下：



```
1 import os
2 import re
3 x = os.popen("arp -a").read()
4 #print x
5 slllist = x.split("\n")
6 for sql in slllist:
7     #print sql
8     findre = re.compile(r'(.*)64-51-7e-80-18-e4(.*)')
9     for x in findre.findall(sql):
10         print x[0]
11
```

控制台 调试

<已终止> D:\CS\javafile\task2\task2.py

192.168.1.1

本次试验的 arp 表就是上面贴的那张表，我的 pc 连接了 wrtnode 的路由，可见路由的 ip 刚好是 192.168.1.1，结果吻合。

四、实验结果与分析

实验成功，截图见第三部分。

五、讨论、心得

本次试验一开始想用 c++编写，后来发现 C++很难将 cmd 命令的输出得到，所以改为使用了 python，竟然意外的方便。可见 python 在这方面还是很不错的。