

任务 2：找到你的板子

课程名称： 嵌入式 实验类型： 综合

实验项目名称： 任务 2：找到你的板子

学生姓名： 武梦晗 专业： 计科 学号： 3120104689

实验日期： 2015 年 3 月 24 日

任务内容

写一个 PC 端程序，在局域网中找到 WRTnode 的 ip 地址。

实现目的

1. 了解局域网的相关知识；
2. 进一步熟悉 WRTnode。

实验器材

硬件

- WRTnode 板一块；
- 5V/1A 电源一个；
- microUSB 线一根；

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。

实验步骤

1. 查询 WRTnode 物理地址；

```
root@OpenWrt:~# ifconfig
apcli0    Link encap:Ethernet  HWaddr 66:51:7E:80:1B:0C
          inet6 addr: fe80::6451:7eff:fe80:1b0c/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)
```

如图可以得知，物理地址（HWaddr）为 64:51:7E:80:1B:0C。

2. 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 Acadia 或 RPi 或 WRTnode 的 MAC 段找到局域网内的 RPi 的 IP 地址 。

程序源码如下：

```
#include <iostream>
#include <stdlib.h>
#include <sstream>
#include <fstream>
using namespace std;

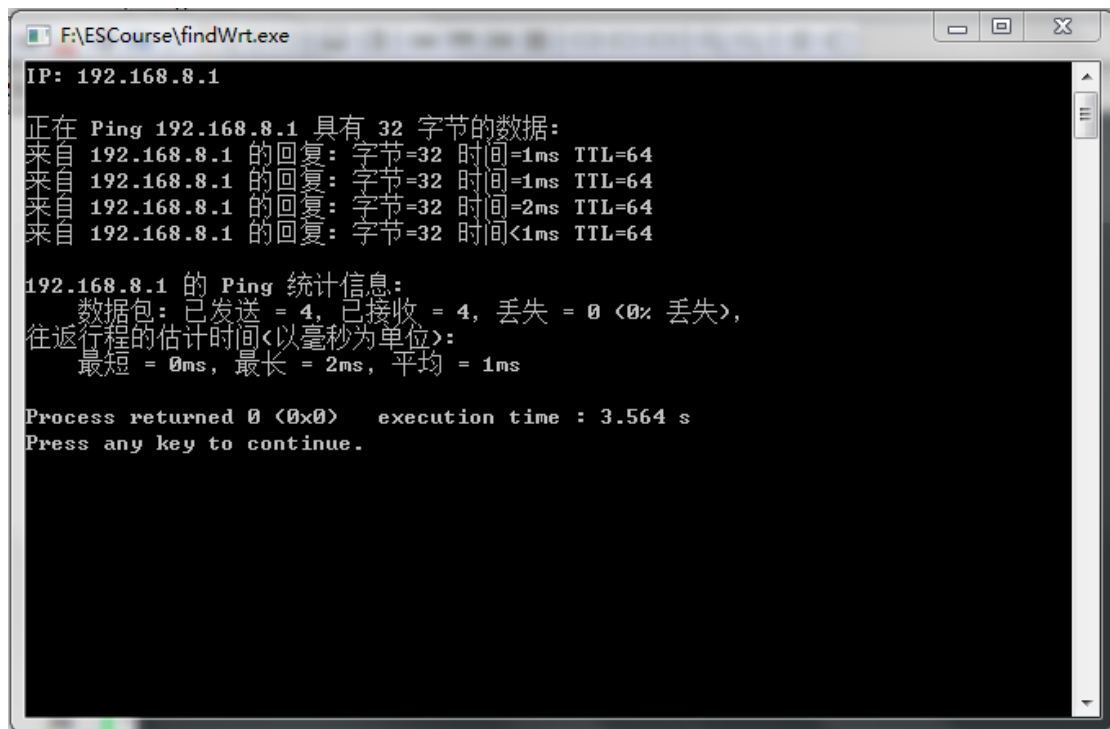
int main() {
    system("arp -a > a.txt");
    string mac="64-51-7e-80-1b-0c";

    ifstream fin("a.txt");
    if(!fin)
    {
        cout<<"Cannot open file!"<<endl;
        exit(-1);
    }
    string ip="";
    string readmac;
    fin >> ip;
    while(!fin.eof())
    {

        fin >> readmac;
        if(readmac==mac)
        {
            cout<<"IP: " << ip <<endl;
            break;
        }
        else
            ip = readmac;
    }
}
```

```
}  
fin.close();  
string command = "ping " + ip;  
system(command.c_str());  
}
```

程序运行结果如图：



```
F:\ESCOURSE\findWrt.exe  
IP: 192.168.8.1  
正在 Ping 192.168.8.1 具有 32 字节的数据:  
来自 192.168.8.1 的回复: 字节=32 时间=1ms TTL=64  
来自 192.168.8.1 的回复: 字节=32 时间=1ms TTL=64  
来自 192.168.8.1 的回复: 字节=32 时间=2ms TTL=64  
来自 192.168.8.1 的回复: 字节=32 时间<1ms TTL=64  
  
192.168.8.1 的 Ping 统计信息:  
    数据包: 已发送 = 4, 已接收 = 4, 丢失 = 0 (0% 丢失),  
    往返行程的估计时间(以毫秒为单位):  
        最短 = 0ms, 最长 = 2ms, 平均 = 1ms  
  
Process returned 0 (0x0)   execution time : 3.564 s  
Press any key to continue.
```