

# 湖南大学

HUNAN UNIVERSITY



小组成员： 计科 1802 谢正宇 张继伟

---

# 实验 006 软件二层交换机的实现

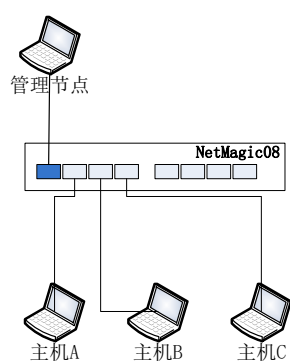
## 一.实验目的

- 1、熟悉基于 NetMagic08 的 Fast 架构编程；
- 2、了解在 linux 环境下的 c 程序的编写、编译与调试；
- 3、了解软件实现交换功能的代码设计；
- 4、理解自学习二层交换机的工作原理；

## 二.实验内容

- 1、基础要求：虚拟机环境下编写一个基于 fast 架构工作的软件自学习二层交换机。
- 2、扩展要求：通过 fast 函数，实现交换表的硬件卸载以及通过读取硬件计数器，实现带宽统计以及链路状态的判断，并且制作界面展示。

## 三.实验环境



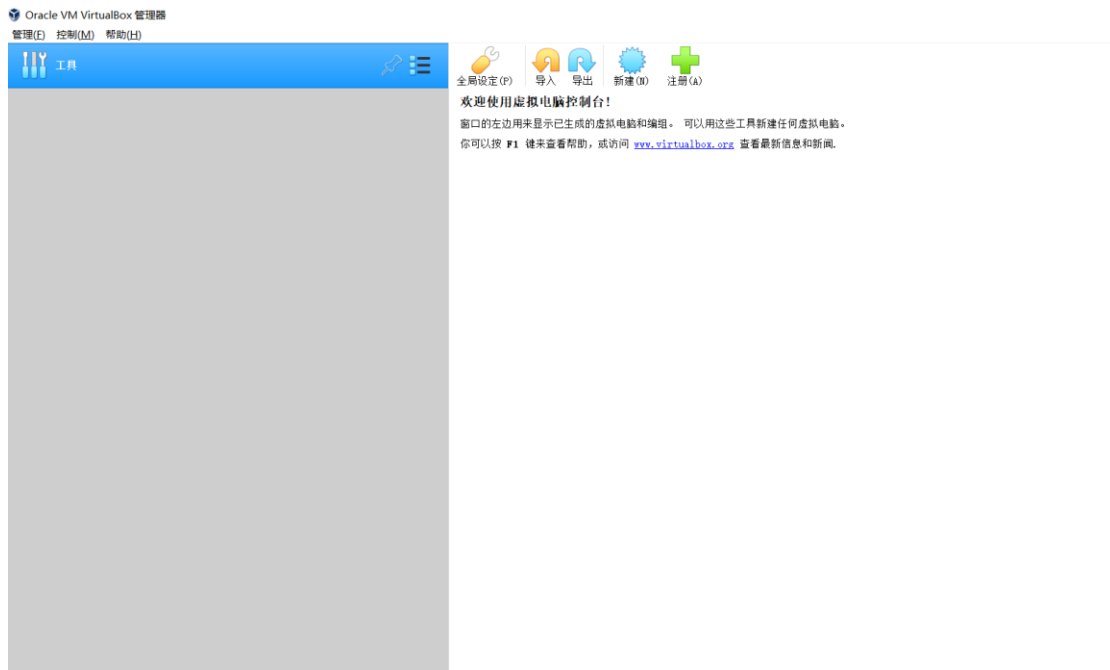
1. 1 台管理节点主机；3 台主机 A、B 以及 C；（分别连接到 2 口和 3 口）
  2. 4 根网线；
  3. NetMagic08 开发平台；
  4. 软件 Quartus 16。
  5. NetEXP 虚拟机运行环境
- 主机及网络详细配置参照附带的实验环境拓扑及软件配置文档。

## 四.实验步骤

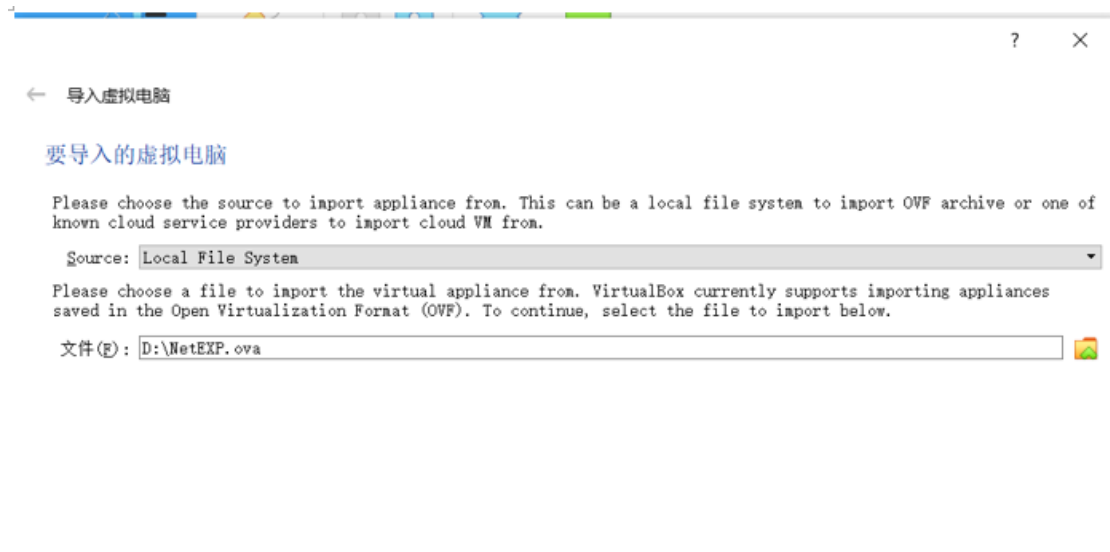
### 1.先将老师发的 sof 文件烧录到路由器交换机中



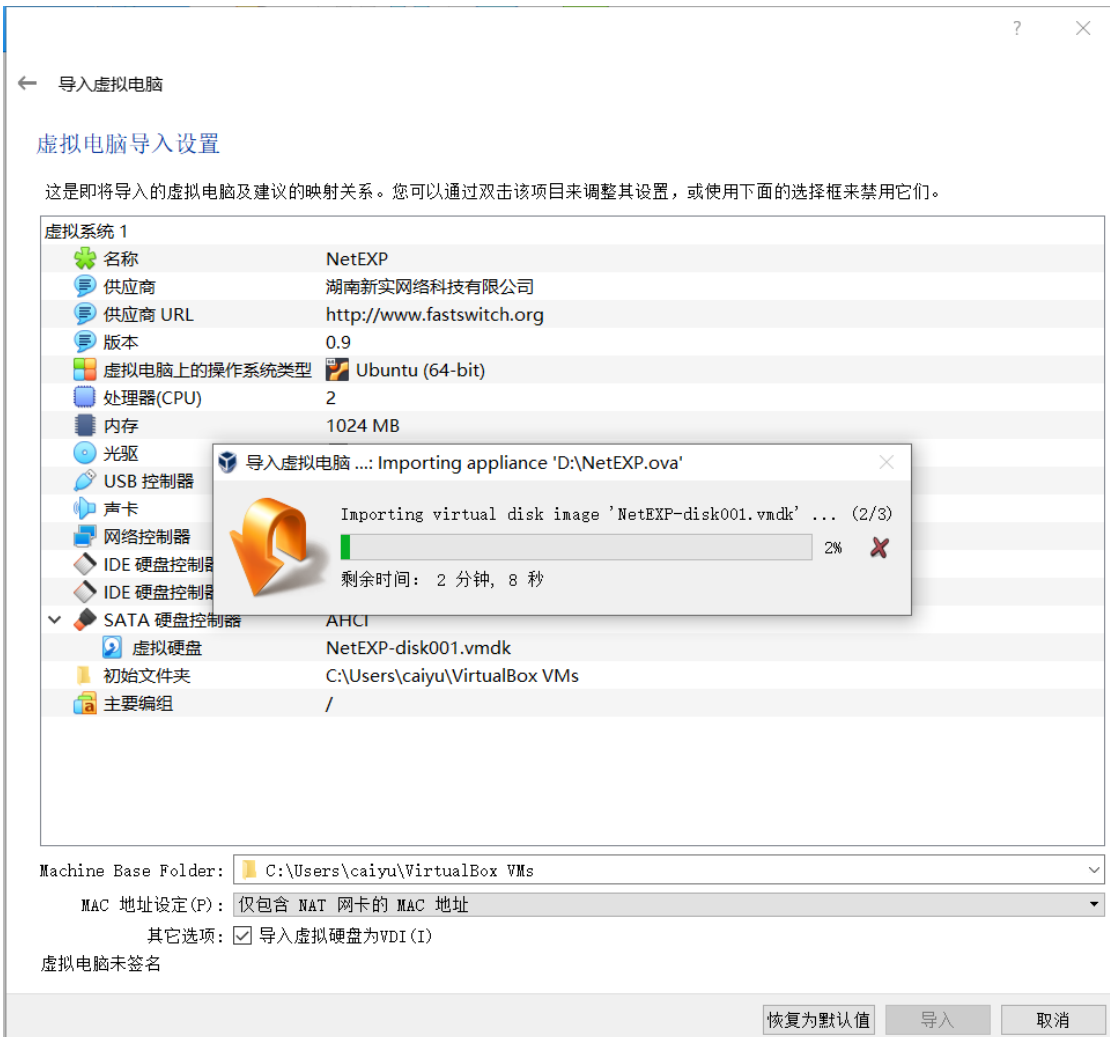
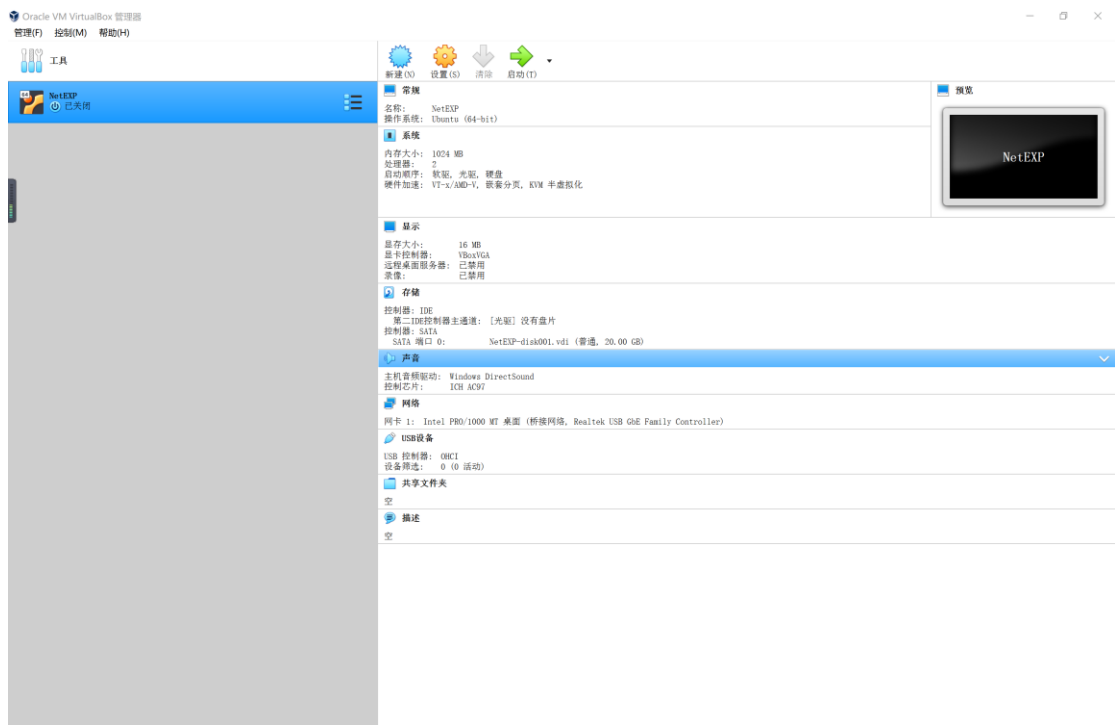
### 2.打开 virtual box 管理器



### 3.点击管理，选择导入现有系统，将我们的 netesp.ova 文件导入



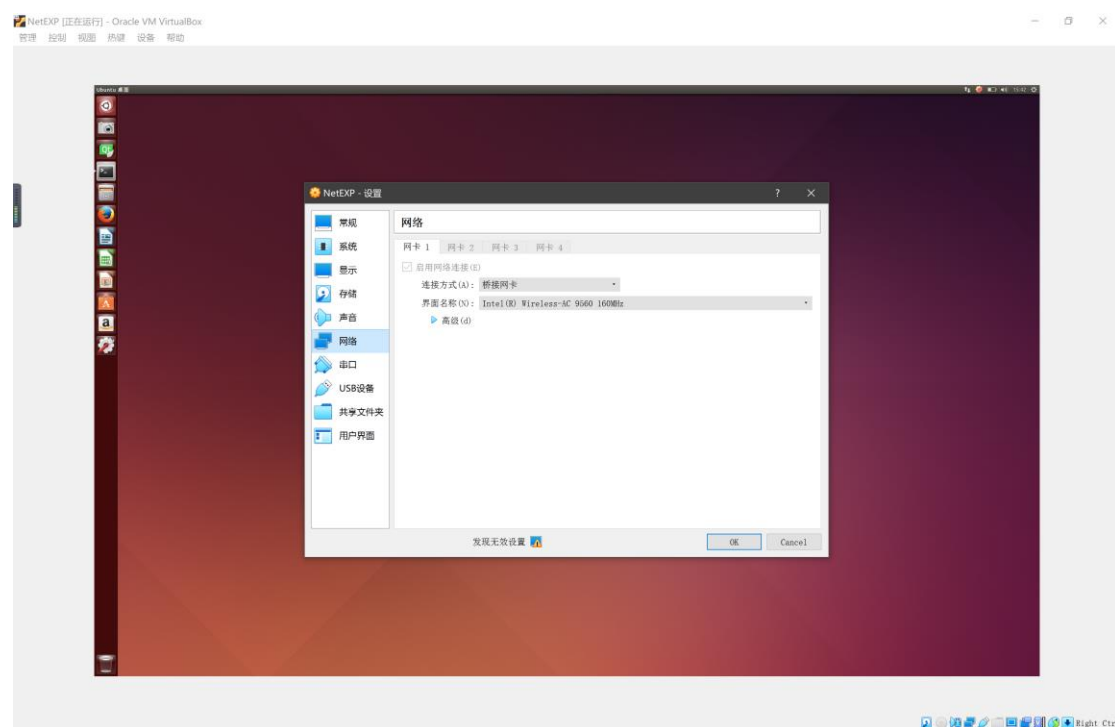
## 4.导入并等待



5.配置网络，在设置中将网络连接方式修改为内部网络

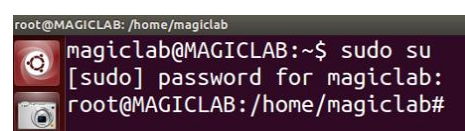


5.启动虚拟环境（这里要把网络连接方式改为桥接模式，否则无法正常进行实验）



Step.1 打开终端，输入命令：sudo su。

Step.2 输入密码：123123，即可进入 root 账户模式。



Step.3 随后输入命令：ifconfig。

查看当前的以太网接口名称，如图为“enp0s25”。

```
root@MAGICLAB:/home/magiclab# ifconfig
eth0      Link encap:以太网  硬件地址 08:00:27:10:b1:b5
          inet6 地址: fe80::a00:27ff:fe10:b1b5/64 Scope:Link
          inet6 地址: 2001:250:4402:1113:d6a1:2745:7718:73ce/128 Scope:Global
          UP BROADCAST RUNNING MULTICAST  MTU:1500  跃点数:1
          接收数据包:464 错误:0 丢弃:0 过载:0 帧数:0
          发送数据包:66 错误:0 丢弃:0 过载:0 载波:0
          碰撞:0 发送队列长度:1000
          接收字节:37905 (37.9 KB)  发送字节:11396 (11.3 KB)

lo        Link encap:本地环回
          inet 地址:127.0.0.1 掩码:255.0.0.0
          inet6 地址: ::1/128 Scope:Host
          UP LOOPBACK RUNNING  MTU:65536 跃点数:1
          接收数据包:96 错误:0 丢弃:0 过载:0 帧数:0
          发送数据包:96 错误:0 丢弃:0 过载:0 载波:0
          碰撞:0 发送队列长度:0
          接收字节:7312 (7.3 KB)  发送字节:7312 (7.3 KB)

root@MAGICLAB:/home/magiclab#
```

Step.4 配置以太网接口的 IP 地址以及子网掩码。

Ifconfig enp0s25 136.136.136.137/24 up

```
magiclab@MAGICLAB:~$ sudo su
[sudo] password for magiclab:
root@MAGICLAB:/home/magiclab# ifconfig
eth0      Link encap:以太网  硬件地址 08:00:27:10:b1:b5
          inet6 地址: fe80::a00:27ff:fe10:b1b5/64 Scope:Link
          inet6 地址: 2001:250:4402:1113:d6a1:2745:7718:73ce/128 Scope:Global
          UP BROADCAST RUNNING MULTICAST  MTU:1500  跃点数:1
          接收数据包:464 错误:0 丢弃:0 过载:0 帧数:0
          发送数据包:66 错误:0 丢弃:0 过载:0 载波:0
          碰撞:0 发送队列长度:1000
          接收字节:37905 (37.9 KB)  发送字节:11396 (11.3 KB)

lo        Link encap:本地环回
          inet 地址:127.0.0.1 掩码:255.0.0.0
          inet6 地址: ::1/128 Scope:Host
          UP LOOPBACK RUNNING  MTU:65536 跃点数:1
          接收数据包:96 错误:0 丢弃:0 过载:0 帧数:0
          发送数据包:96 错误:0 丢弃:0 过载:0 载波:0
          碰撞:0 发送队列长度:0
          接收字节:7312 (7.3 KB)  发送字节:7312 (7.3 KB)

root@MAGICLAB:/home/magiclab# ifconfig eth0 136.136.136.137/24 up
root@MAGICLAB:/home/magiclab#
```

Step.5 将 enp0s25 的 IP 地址设置为 136.136.136.137 之后，输入如下命令、切换至驱动程序目录：cd fast/driver/netmagic08。



```

magiclab@MAGICLAB:~$ sudo su
[sudo] password for magiclab:
root@MAGICLAB:/home/magiclab# ifconfig
eth0      Link encap:以太网  硬件地址 08:00:27:10:b1:b5
          inet6 地址: fe80::a00:27ff:fe10:b1b5/64 Scope:Link
          inet6 地址: 2001:250:4402:1113:d6a1:2745:7718:73ce/128 Scope:Global
          UP BROADCAST RUNNING MULTICAST  MTU:1500  跃点数:1
          接收数据包:464 错误:0  丢弃:0  过载:0  帧数:0
          发送数据包:66 错误:0  丢弃:0  过载:0  载波:0
          碰撞:0  发送队列长度:1000
          接收字节:37905 (37.9 KB)  发送字节:11396 (11.3 KB)

lo        Link encap:本地环回
          inet 地址:127.0.0.1  掩码:255.0.0.0
          inet6 地址: ::1/128 Scope:Host
          UP LOOPBACK RUNNING  MTU:65536  跃点数:1
          接收数据包:96 错误:0  丢弃:0  过载:0  帧数:0
          发送数据包:96 错误:0  丢弃:0  过载:0  载波:0
          碰撞:0  发送队列长度:0
          接收字节:7312 (7.3 KB)  发送字节:7312 (7.3 KB)

root@MAGICLAB:/home/magiclab# ifconfig eth0 136.136.136.137/24 up
root@MAGICLAB:/home/magiclab# cd fast-l2switch-08/fast/driver/netmagic08
root@MAGICLAB:/home/magiclab/fast-l2switch-08/fast/driver/netmagic08#

```

Step.6 按下回车执行命令，切换至 netmagic08 目录下，然后输入并执行命令开始编译驱动程序：make clean && make。

```

root@MAGICLAB:/home/magiclab# ifconfig eth0 136.136.136.137/24 up
root@MAGICLAB:/home/magiclab# cd fast-l2switch-08/fast/driver/netmagic08
root@MAGICLAB:/home/magiclab/fast-l2switch-08/fast/driver/netmagic08# make clean && make
make -C /lib/modules/4.2.0-27-generic/build M=/home/magiclab/fast-l2switch-08/fast/driver/netmagic08 clean
make[1]: 正在进入目录 `/usr/src/linux-headers-4.2.0-27-generic'
CLEAN /home/magiclab/fast-l2switch-08/fast/driver/netmagic08/.tmp_versions
CLEAN /home/magiclab/fast-l2switch-08/fast/driver/netmagic08/Module.symvers
make[1]: 正在离开目录 `/usr/src/linux-headers-4.2.0-27-generic'
make -C /lib/modules/4.2.0-27-generic/build M=/home/magiclab/fast-l2switch-08/fast/driver/netmagic08 clean
make[1]: 正在进入目录 `/usr/src/linux-headers-4.2.0-27-generic'
make[1]: 正在离开目录 `/usr/src/linux-headers-4.2.0-27-generic'
make -C /lib/modules/4.2.0-27-generic/build M=/home/magiclab/fast-l2switch-08/fast/driver/netmagic08 modules
make[1]: 正在进入目录 `/usr/src/linux-headers-4.2.0-27-generic'
CC [M] /home/magiclab/fast-l2switch-08/fast/driver/netmagic08/main_nm08.o
LD [M] /home/magiclab/fast-l2switch-08/fast/driver/netmagic08/nm08_nl.o
LD [M] /home/magiclab/fast-l2switch-08/fast/driver/netmagic08/nm08.o
Building modules, stage 2.
MODPOST 1 modules
CC /home/magiclab/fast-l2switch-08/fast/driver/netmagic08/nm08.mod.o
LD [M] /home/magiclab/fast-l2switch-08/fast/driver/netmagic08/nm08.ko
make[1]: 正在离开目录 `/usr/src/linux-headers-4.2.0-27-generic'
root@MAGICLAB:/home/magiclab/fast-l2switch-08/fast/driver/netmagic08#

```

Step.7 编译完成之后，输入并执行命令（其中 ctlif\_name=后接的参数为以太网接口的名称）：insmod nm08.ko ctlif\_name=enp0s25。

```

root@MAGICLAB:/home/magiclab/fast-l2switch-08/fast/driver/netmagic08# insmod nm08.ko ctlif_name=eth0
root@MAGICLAB:/home/magiclab/fast-l2switch-08/fast/driver/netmagic08#

```

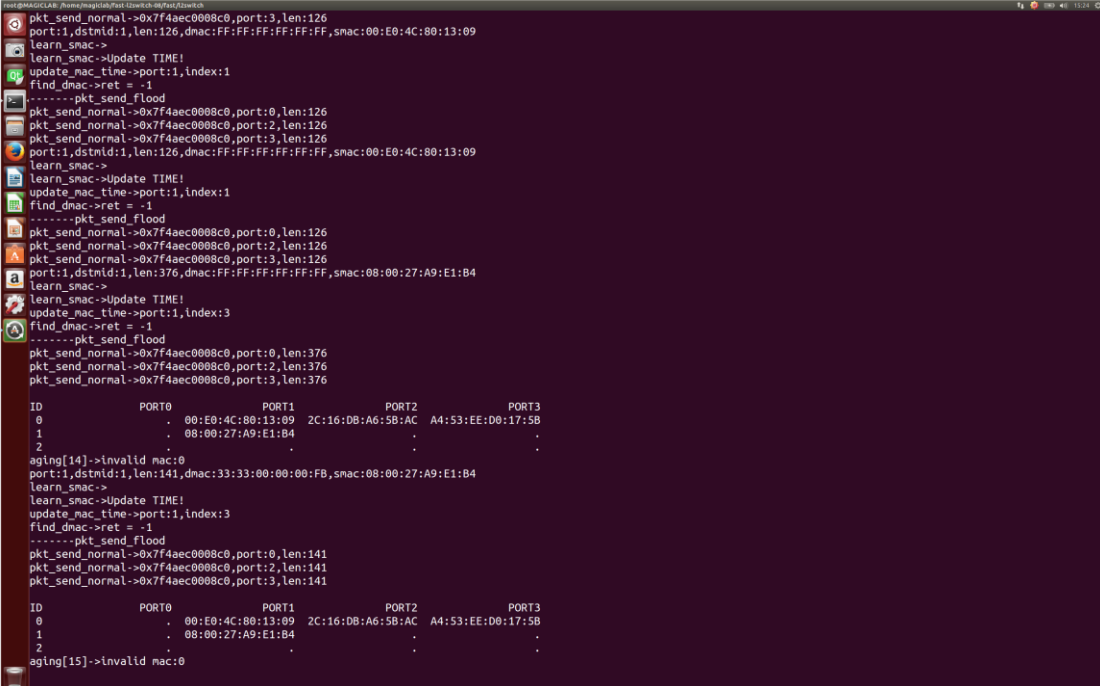
Step.8 加载驱动之后，确保已经连接了 netmagic08，输入执行如下命令：cd ../../l2switch。

```

make -C /lib/modules/4.2.0-27-generic/build M=/home/magiclab/fast-l2switch-08/fast/driver/netmagic08 clean
make[1]: 正在进入目录 `/usr/src/linux-headers-4.2.0-27-generic'
CLEAN /home/magiclab/fast-l2switch-08/fast/driver/netmagic08/.tmp_versions
CLEAN /home/magiclab/fast-l2switch-08/fast/driver/netmagic08/Module.symvers
make[1]:正在离开目录 `/usr/src/linux-headers-4.2.0-27-generic'
make -C /lib/modules/4.2.0-27-generic/build M=/home/magiclab/fast-l2switch-08/fast/driver/netmagic08 clean
make[1]: 正在进入目录 `/usr/src/linux-headers-4.2.0-27-generic'
make[1]:正在离开目录 `/usr/src/linux-headers-4.2.0-27-generic'
make -C /lib/modules/4.2.0-27-generic/build M=/home/magiclab/fast-l2switch-08/fast/driver/netmagic08 modules
make[1]: 正在进入目录 `/usr/src/linux-headers-4.2.0-27-generic'
CC [M] /home/magiclab/fast-l2switch-08/fast/driver/netmagic08/main_nm08.o
CC [M] /home/magiclab/fast-l2switch-08/fast/driver/netmagic08/nm08_nl.o
LD [M] /home/magiclab/fast-l2switch-08/fast/driver/netmagic08/nm08.o
Building modules, stage 2.
MODPOST 1 modules
CC /home/magiclab/fast-l2switch-08/fast/driver/netmagic08/nm08.mod.o
LD [M] /home/magiclab/fast-l2switch-08/fast/driver/netmagic08/nm08.ko
make[1]:正在离开目录 `/usr/src/linux-headers-4.2.0-27-generic'
root@MAGICLAB:/home/magiclab/fast-l2switch-08/fast/driver/netmagic08# insmod nm08.ko ctliif_name=eth0
root@MAGICLAB:/home/magiclab/fast-l2switch-08/fast/driver/netmagic08# cd ../../l2switch
root@MAGICLAB:/home/magiclab/fast-l2switch-08/fast/l2switch#

```

Step.9 输入如下命令执行基本 io 输出功能：./fastio。或者输入如下命令执行 2 层交换功能：./l2switch。



```

pkt_send_normal->0x7f4aec0008c0,port:3,len:126
port:1,dstmid:1,len:126,dmac:FF:FF:FF:FF:FF:FF,smac:08:E0:4C:80:13:09
learn_smac->
update_mac_time->port:1,index:1
find_dmac->ret = -1
-----pkt_send_flood
pkt_send_normal->0x7f4aec0008c0,port:0,len:126
pkt_send_normal->0x7f4aec0008c0,port:2,len:126
pkt_send_normal->0x7f4aec0008c0,port:3,len:126
port:1,dstmid:1,len:126,dmac:FF:FF:FF:FF:FF:FF,smac:08:E0:4C:80:13:09
learn_smac->
update_mac_time->port:1,index:1
find_dmac->ret = -1
-----pkt_send_flood
pkt_send_normal->0x7f4aec0008c0,port:0,len:126
pkt_send_normal->0x7f4aec0008c0,port:2,len:126
pkt_send_normal->0x7f4aec0008c0,port:3,len:126
port:1,dstmid:1,len:376,dmac:FF:FF:FF:FF:FF:FF,smac:08:00:27:A9:E1:B4
learn_smac->
update_mac_time->port:1,index:3
find_dmac->ret = -1
-----pkt_send_flood
pkt_send_normal->0x7f4aec0008c0,port:0,len:376
pkt_send_normal->0x7f4aec0008c0,port:2,len:376
pkt_send_normal->0x7f4aec0008c0,port:3,len:376

ID      PORT0      PORT1      PORT2      PORT3
0      .  00:E0:4C:80:13:09  2C:16:DB:A6:5B:AC  A4:53:EE:D0:17:5B
1      .  08:00:27:A9:E1:B4      .      .
2      .      .      .      .
aging[14]->invalid mac:0
port:1,dstmid:1,len:141,dmac:33:33:00:00:00:FB,smac:08:00:27:A9:E1:B4
learn_smac->
update_mac_time->port:1,index:3
find_dmac->ret = -1
-----pkt_send_flood
pkt_send_normal->0x7f4aec0008c0,port:0,len:141
pkt_send_normal->0x7f4aec0008c0,port:2,len:141
pkt_send_normal->0x7f4aec0008c0,port:3,len:141

ID      PORT0      PORT1      PORT2      PORT3
0      .  00:E0:4C:80:13:09  2C:16:DB:A6:5B:AC  A4:53:EE:D0:17:5B
1      .  08:00:27:A9:E1:B4      .      .
2      .      .      .      .
aging[15]->invalid mac:0

```



```

-----pkt_send_flood
pkt_send_normal->0x7f4aec0008c0,port:0,len:213
pkt_send_normal->0x7f4aec0008c0,port:2,len:213
pkt_send_normal->0x7f4aec0008c0,port:3,len:213
port:1,dstnId:1,len:213,dmac:01:00:5E:7F:FF:FA,smac:00:E0:4C:80:13:09
learn_smac->
learn_smac->Update TIME!
update_mac_time->port:1,index:1
find_dmac->ret = -1
-----pkt_send_flood
pkt_send_normal->0x7f4aec0008c0,port:0,len:213
pkt_send_normal->0x7f4aec0008c0,port:2,len:213
pkt_send_normal->0x7f4aec0008c0,port:3,len:213

ID          PORT0          PORT1          PORT2          PORT3
0            . 00:E0:4C:80:13:09 2C:16:DB:A6:5B:AC A4:53:EE:D0:17:5B
1            . 08:00:27:A9:E1:B4            .
2            .            .            .
aging[38]->invalid mac:0
port:3,dstnId:1,len:191,dmac:33:33:00:01:00:02,smac:A4:53:EE:D0:17:5B
learn_smac->
learn_smac->Update TIME!
update_mac_time->port:3,index:0
find_dmac->ret = -1
-----pkt_send_flood
pkt_send_normal->0x7f4aec0008c0,port:0,len:191
pkt_send_normal->0x7f4aec0008c0,port:1,len:191
pkt_send_normal->0x7f4aec0008c0,port:2,len:191
port:1,dstnId:1,len:376,dmac:FF:FF:FF:FF:FF:FF,smac:00:E0:4C:80:13:09
learn_smac->
learn_smac->Update TIME!
update_mac_time->port:1,index:1
find_dmac->ret = -1
-----pkt_send_flood
pkt_send_normal->0x7f4aec0008c0,port:0,len:376
pkt_send_normal->0x7f4aec0008c0,port:2,len:376
pkt_send_normal->0x7f4aec0008c0,port:3,len:376
port:1,dstnId:1,len:213,dmac:01:00:5E:7F:FF:FA,smac:00:E0:4C:80:13:09
learn_smac->
learn_smac->Update TIME!
update_mac_time->port:1,index:1
find_dmac->ret = -1
-----pkt_send_flood
pkt_send_normal->0x7f4aec0008c0,port:0,len:213
pkt_send_normal->0x7f4aec0008c0,port:2,len:213
pkt_send_normal->0x7f4aec0008c0,port:3,len:213
port:1,dstnId:1,len:213,dmac:01:00:5E:7F:FF:FA,smac:00:E0:4C:80:13:09
learn_smac->
learn_smac->Update TIME!
update_mac_time->port:1,index:1

```

```

pkt_send_normal->0x7f4aec0008c0,port:3,len:98
port:1,dstnId:1,len:118,dmac:33:33:00:01:00:03,smac:00:E0:4C:80:13:09
learn_smac->
learn_smac->Update TIME!
update_mac_time->port:1,index:1
find_dmac->ret = -1
-----pkt_send_flood
pkt_send_normal->0x7f4aec0008c0,port:0,len:118
pkt_send_normal->0x7f4aec0008c0,port:2,len:118
pkt_send_normal->0x7f4aec0008c0,port:3,len:118
port:1,dstnId:1,len:98,dmac:01:00:5E:00:00:FC,smac:00:E0:4C:80:13:09
learn_smac->
learn_smac->Update TIME!
update_mac_time->port:1,index:1
find_dmac->ret = -1
-----pkt_send_flood
pkt_send_normal->0x7f4aec0008c0,port:0,len:98
pkt_send_normal->0x7f4aec0008c0,port:2,len:98
pkt_send_normal->0x7f4aec0008c0,port:3,len:98
port:1,dstnId:1,len:118,dmac:33:33:00:01:00:03,smac:00:E0:4C:80:13:09
learn_smac->
learn_smac->Update TIME!
update_mac_time->port:1,index:1
find_dmac->ret = -1
-----pkt_send_flood
pkt_send_normal->0x7f4aec0008c0,port:0,len:118
pkt_send_normal->0x7f4aec0008c0,port:2,len:118
pkt_send_normal->0x7f4aec0008c0,port:3,len:118
port:1,dstnId:1,len:98,dmac:01:00:5E:00:00:FC,smac:00:E0:4C:80:13:09
learn_smac->
learn_smac->Update TIME!
update_mac_time->port:1,index:1
find_dmac->ret = -1
-----pkt_send_flood
pkt_send_normal->0x7f4aec0008c0,port:0,len:98
pkt_send_normal->0x7f4aec0008c0,port:2,len:98
pkt_send_normal->0x7f4aec0008c0,port:3,len:98
port:1,dstnId:1,len:118,dmac:33:33:00:01:00:03,smac:00:E0:4C:80:13:09
learn_smac->
learn_smac->Update TIME!
update_mac_time->port:1,index:1
find_dmac->ret = -1
-----pkt_send_flood
pkt_send_normal->0x7f4aec0008c0,port:0,len:118
pkt_send_normal->0x7f4aec0008c0,port:2,len:118
pkt_send_normal->0x7f4aec0008c0,port:3,len:118
port:1,dstnId:1,len:98,dmac:01:00:5E:00:00:FC,smac:00:E0:4C:80:13:09
learn_smac->
learn_smac->Update TIME!
update_mac_time->port:1,index:1

```

两台机子互相 ping(ping 后面跟的是虚拟机上的 ipv4 地址)

```
C:\Windows\System32>ping 192.168.1.2 -t

正在 Ping 192.168.1.2 具有 32 字节的数据:
来自 192.168.1.1 的回复: 无法访问目标主机。
来自 192.168.1.1 的回复: 无法访问目标主机。
来自 192.168.1.1 的回复: 无法访问目标主机。
来自 192.168.1.1 的回复: 无法访问目标主机。
来自 192.168.1.1 的回复: 无法访问目标主机。
来自 192.168.1.1 的回复: 无法访问目标主机。
来自 192.168.1.1 的回复: 无法访问目标主机。
来自 192.168.1.2 的回复: 字节=32 时间=1003ms TTL=64
来自 192.168.1.2 的回复: 字节=32 时间=3ms TTL=64
来自 192.168.1.2 的回复: 字节=32 时间=3ms TTL=64
来自 192.168.1.2 的回复: 字节=32 时间=3ms TTL=64
来自 192.168.1.2 的回复: 字节=32 时间=3ms TTL=64
来自 192.168.1.2 的回复: 字节=32 时间=4ms TTL=64
来自 192.168.1.2 的回复: 字节=32 时间=3ms TTL=64
来自 192.168.1.2 的回复: 字节=32 时间=2ms TTL=64
来自 192.168.1.2 的回复: 字节=32 时间=3ms TTL=64
来自 192.168.1.2 的回复: 字节=32 时间=3ms TTL=64
来自 192.168.1.2 的回复: 字节=32 时间=2ms TTL=64
来自 192.168.1.2 的回复: 字节=32 时间=3ms TTL=64
来自 192.168.1.2 的回复: 字节=32 时间=3ms TTL=64
来自 192.168.1.2 的回复: 字节=32 时间=3ms TTL=64
来自 192.168.1.2 的回复: 字节=32 时间=3ms TTL=64
来自 192.168.1.2 的回复: 字节=32 时间=2ms TTL=64
来自 192.168.1.2 的回复: 字节=32 时间=2ms TTL=64
```

## 五.实验验收

验收要求：

- 1、正确启动软件路由。
- 2、连接的三台主机之间可以两两 ping 通，并且在管理节点主机的 l2switch 程序运行界面上能够看到交换表；之后尝试交换主机连接端口、断开连接、重新接入 netmagic08 等方式观测交换表的变化。
- 3、在实验报告中记录学习交换表的代码、老化的代码。

## 六.实验思考

在实验报告中回答本实验环境的 ubuntu 系统内核修改了什么部分，为什么要做修改？

ubuntu 系统内核配置了以太网接口的 IP 地址和子网掩码。因为只有把主机和虚拟机的 IP 地址设置为同一个子网掩码和同一个 IP 地址，才能使得主机和虚拟机属于同一个局域网，这样才能最终 ping 通。

## 七.实验心得

通体来说，这次实验比较困难，是第一次在 Ubuntu 的 Linux 系统上配置相应的 IP 地址进行实验。实验本身难度并不高，难点是环境的配置以及相应步骤的实现，特别是对实验的连接，即代码充当了交换机的功能。需要注意的是：一定要把主机和从机的 IP 地址设置为同一子网掩码以及同一 IP 地址，即 IP 地址的前三位需要一致，否则实验运行步骤会有错误。通过这次实验我了解了路由交换表，和计算机网络课程中所学的关于路由器的知识结合起来，对路由转发和路由表路由器等相关概念更加清晰，自己也可以明显区分网络层在数据平面和控制平面的区别和统一。