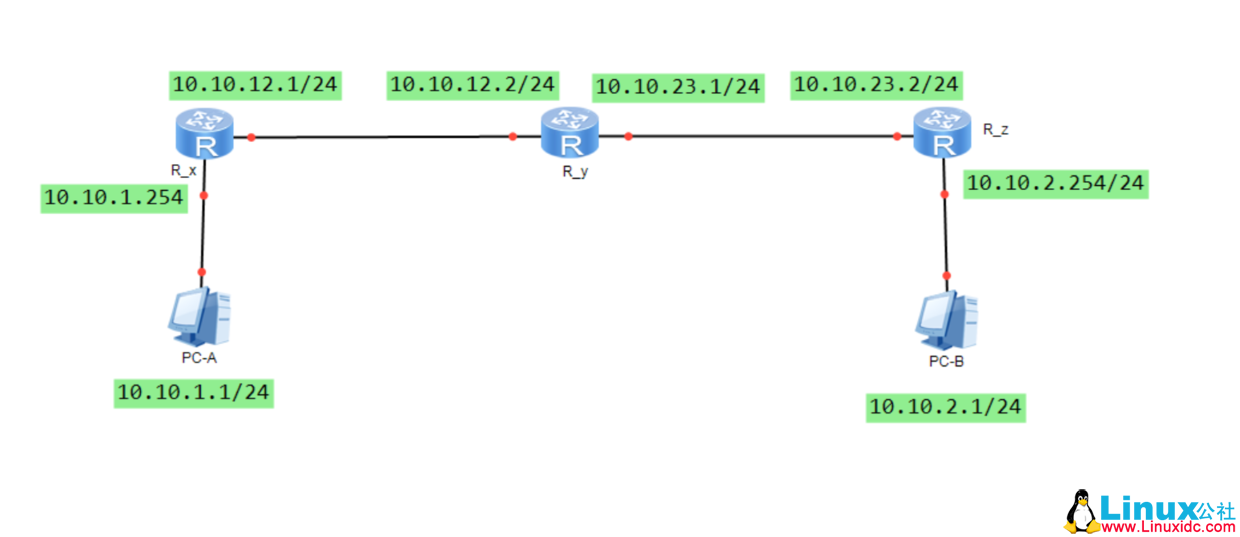
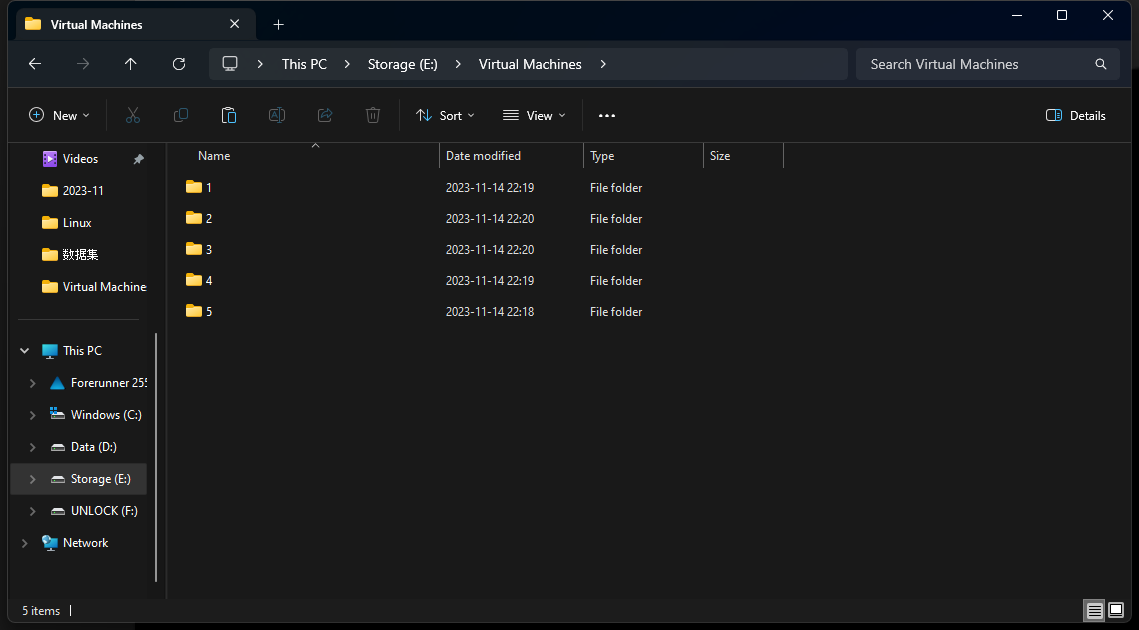
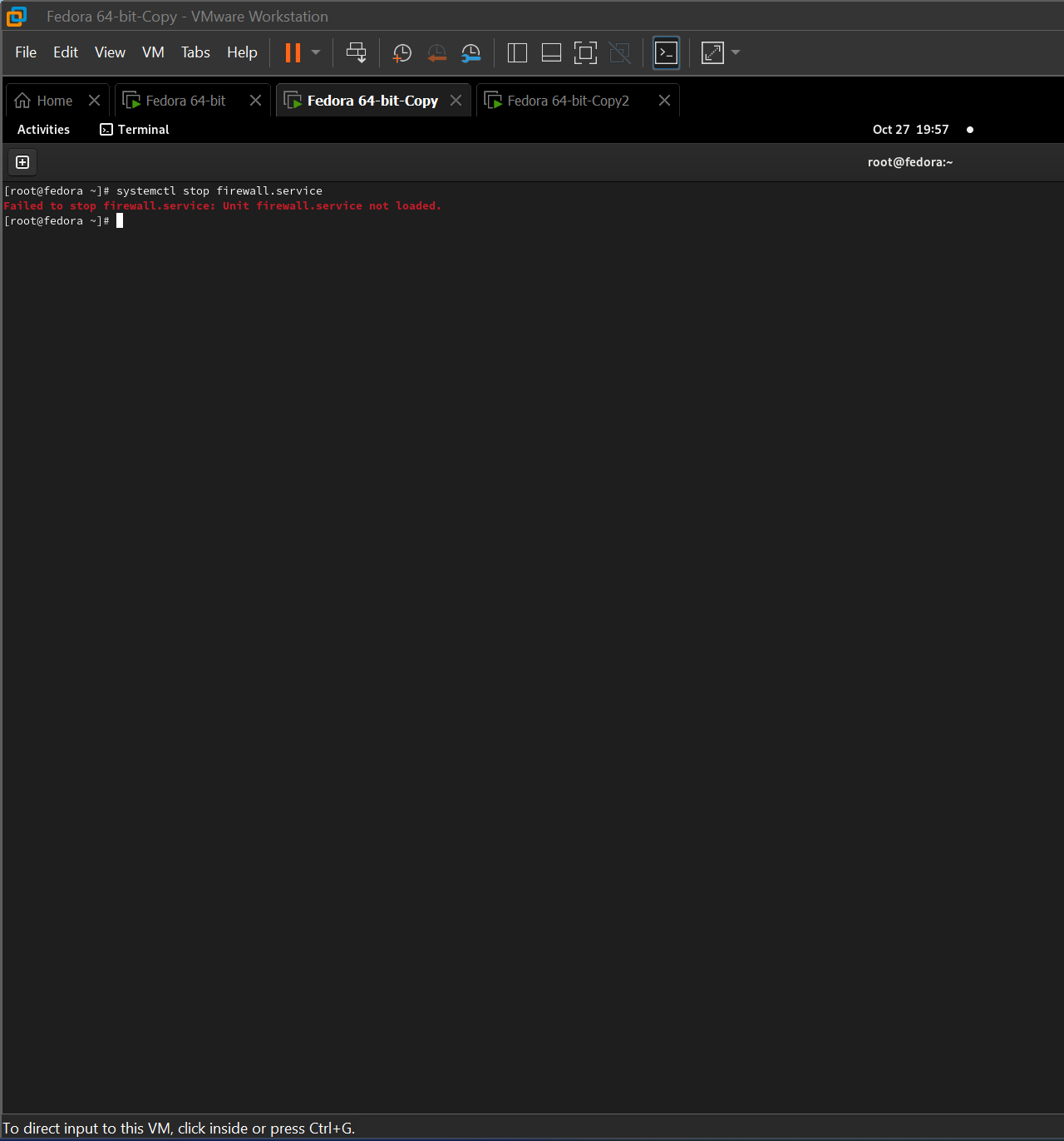
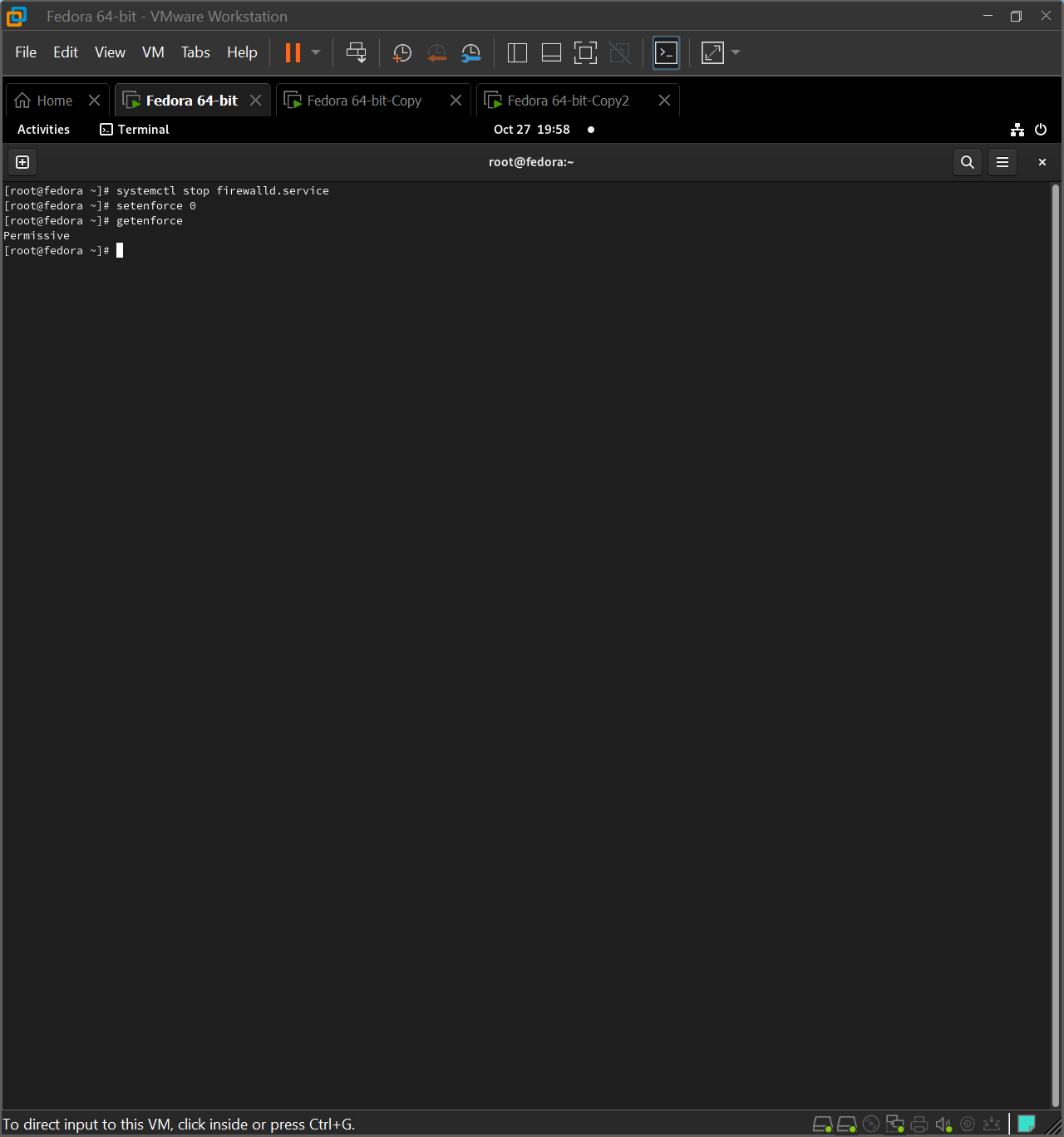
**Linux模拟路由器配置实验**

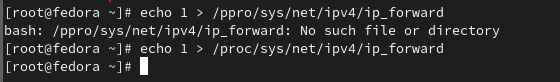
实验目标：  
将3台主机设置成路由器。  
实现两台终端能够相互通信。  
网络拓扑结构如下：



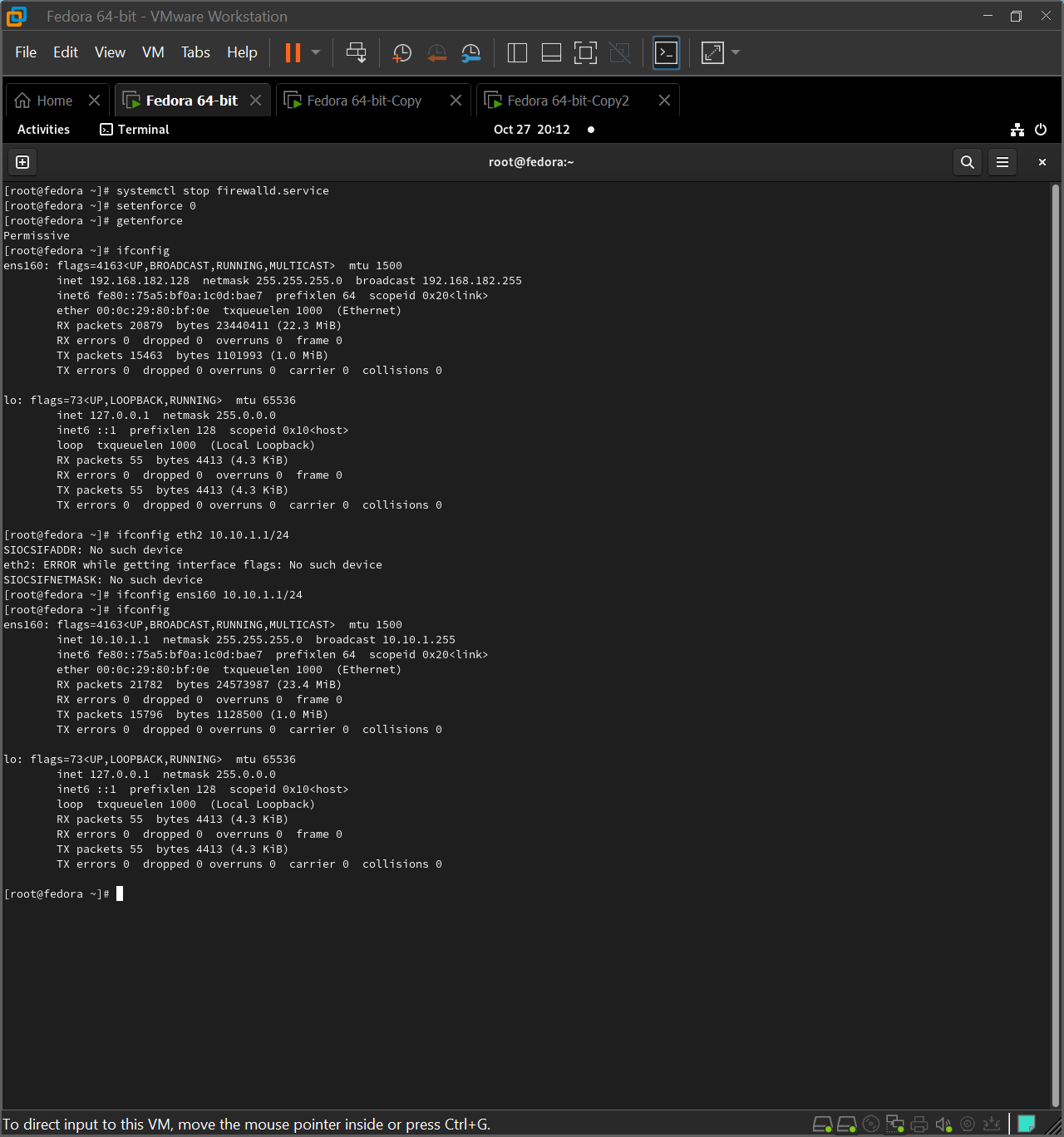
准备工作：  
1、批量复制虚拟机文件，创建多个系统，提前修改mac地址。或者使用多台计算机进行实验。  
2、关闭防火墙  
systemctl stop firewalld.service

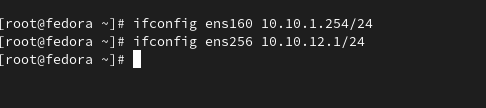
  
3、关闭selinux  
setenforce 0  //关闭  
getenforce  //查看

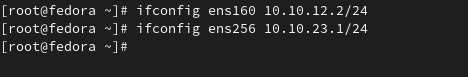
  
4、在三台“路由器”上启用路由转发功能  
echo 1 > /proc/sys/net/ipv4/ip\_forward   
echo 1 > /proc/sys/net/ipv4/ip\_forward

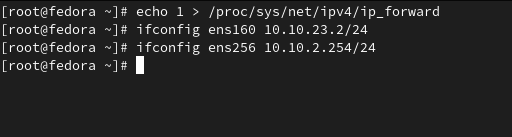


操作过程：  
第1步：配置网卡地址，测试直连  
配置ip地址  
    ifconfig 接口 ip地址/掩码   
    或者  
    ip addr add ip地址/掩码  dev 接口  
检查ip配置  
    ip add  
    或者  
    ifconfig   
12345678910111213 PC-A:   
ifconfig eth2 10.10.1.1/24

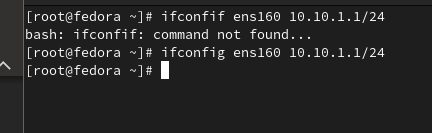
  
R\_x:   
ifconfig eth1 10.10.1.254/24  //PC-A的网关   
ifconfig eth2 10.10.12.1/24    //与R\_y直连

  
R\_y:   
ifconfig eth0 10.10.12.2/24    //与R\_x直连   
ifconfig eth2 10.10.23.1/24    //与R\_z直连

  
R\_z:   
ifconfig eth0 10.10.23.2/24    //与R\_y直连   
ifconfig eth1 10.10.2.254/24  //PC-B的网关

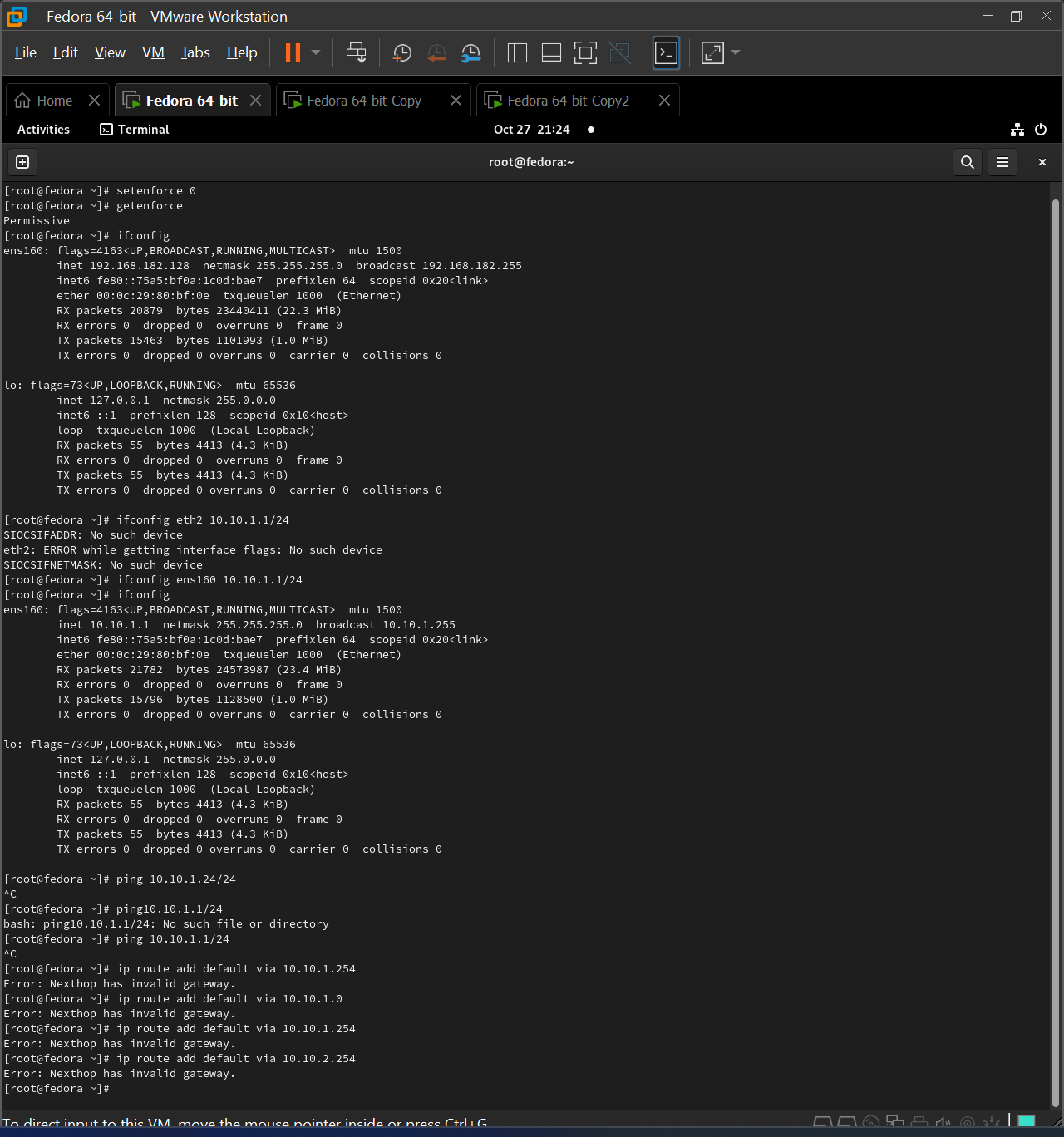


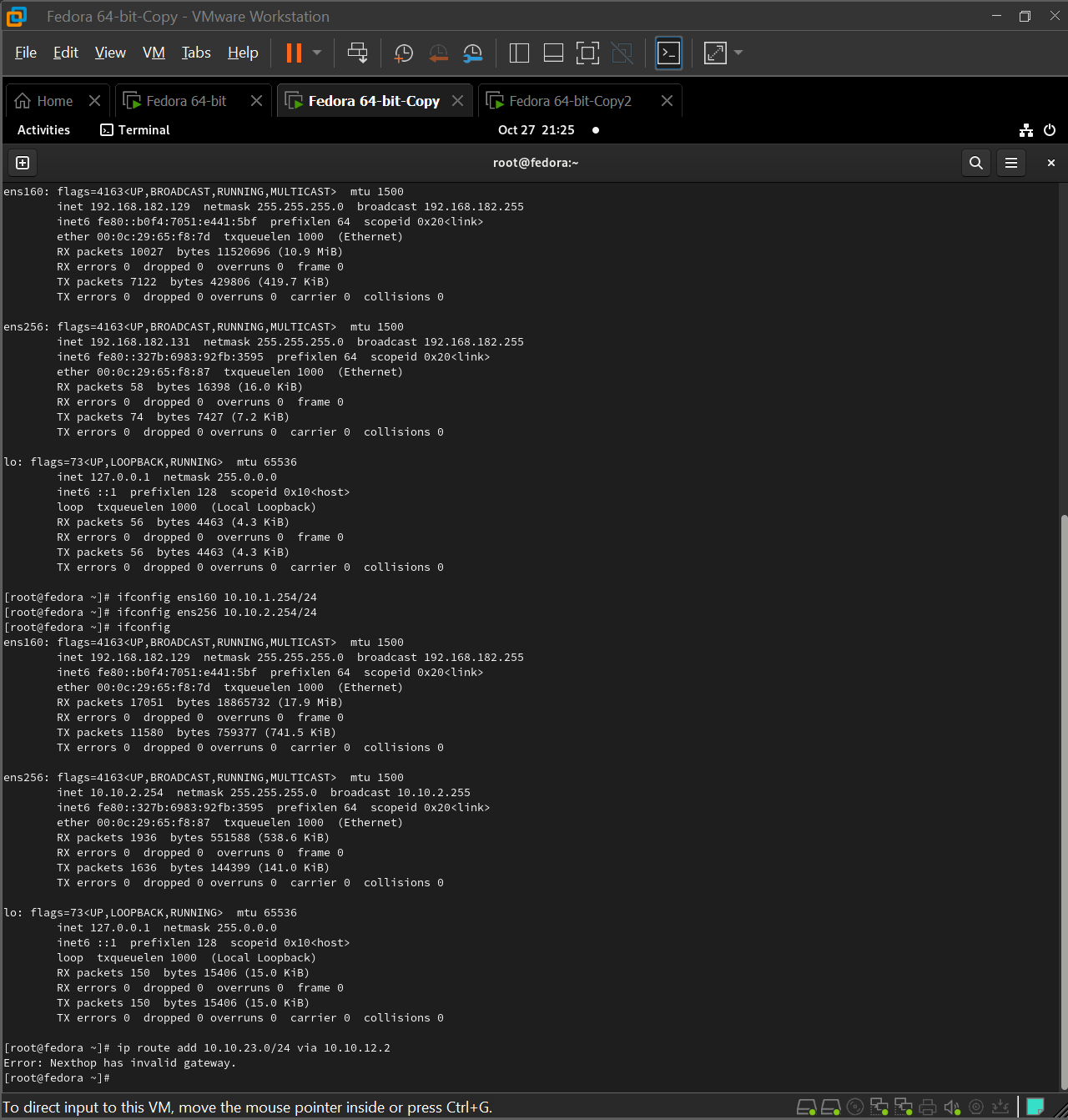
PC-B:   
ifconfig eth1 10.10.1.1/24



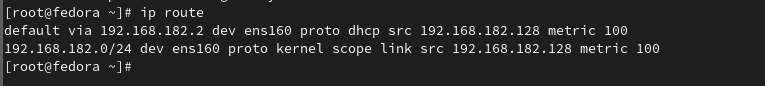
测试直连地址使用工具：ping  
如果直连不通，后续就无法操作，一定要保证ip配置正确  
网卡的配置文件存放在下述目录中：  
/etc/sysconfig/network-scripts/  
如果想要永久生效可以修改目录中对应的网卡文件

第2步：添加路由，测试  
**配置路由**  
    A) ip route add 目的网段/掩码 via  下一跳ip地址  
    或者  
    B) route add -net 目的网段/掩码 dev 本机出接口  
    意思就是要想到达某一网络，就要从本机指定接口出去，或者指定的邻接路由器接口ip地址，而这就是下一跳  
查看路由信息  
    route -n  
    或者  
    ip route  
PC-A：   
ip route add default via 10.10.1.254    //主机A不管去哪里的网络，都会把数据包交给网关10.10.1.254也就是R\_x ++

  
R\_x：   
ip route add 10.10.2.0/24 via 10.10.12.2      
ip route add 10.10.23.0/24 via 10.10.12.2    //要想到达目标网络，本段必须有去的路由，但是并不关心回的路由，因为那是由对端设备负责的   
R\_y：   
ip route add 10.10.1.0/24 via 10.10.12.1   
ip route add 10.10.2.0/24 via 10.10.23.2   
R\_z：   
ip route add 10.10.1.0/24 via 10.10.23.1   
ip route add 10.10.12.0/24 via 10.10.23.1   
PC-B：   
ip route default via 10.10.2.254

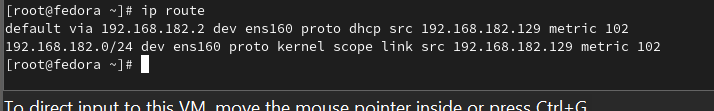
  
//每条路由是只负责去的路径，由对方负责回来的路径。对本段而言回的路由就是对端去的路由

查看各个设备的路由表  
[root@PC\_A ~]#ip route    
192.168.1.0/24 dev eth1  proto kernel  scope link  src 192.168.1.100  metric 1    
10.10.1.0/24 dev eth2  proto kernel  scope link  src 10.10.1.1    
default via 10.10.1.254 dev eth2        //主机A有一条默认路由，去往目标的数据包全部交给eth2接口



[root@R\_x ~]#ip route    
10.10.23.0/24 via 10.10.12.2 dev eth2    
10.10.2.0/24 via 10.10.12.2 dev eth2        //路由器R\_x到达目的网络的路径是10.10.12.2，也就是R\_y   
10.10.1.0/24 dev eth1  proto kernel  scope link  src 10.10.1.254    
10.10.12.0/24 dev eth2  proto kernel  scope link  src 10.10.12.1  //以上两条路由是直连网络

[root@R\_y ~]#ip route    
10.10.23.0/24 dev eth2  proto kernel  scope link  src 10.10.23.1    
10.10.2.0/24 via 10.10.23.2 dev eth2    
10.10.1.0/24 via 10.10.12.1 dev eth0      //由于R\_y是3台路由器的中间一台，所以需要配置到两端的路由，因此下一跳也不一样   
10.10.12.0/24 dev eth0  proto kernel  scope link  src 10.10.12.2



[root@R\_z ~]#ip route    
10.10.23.0/24 dev eth0  proto kernel  scope link  src 10.10.23.2    
10.10.2.0/24 dev eth1  proto kernel  scope link  src 10.10.2.254    
10.10.1.0/24 via 10.10.23.1 dev eth0    
10.10.12.0/24 via 10.10.23.1 dev eth0    //至此经过的路由器有三个了，到达目的网络网关

[root@PC-B ~]#ip route   
default via 10.10.2.254 dev eth1      //到达目的主机B   
10.10.2.0/24 dev eth1  proto kernel  scope link  src 10.10.2.1    
192.168.2.0/24 dev eth0  proto kernel  scope link  src 192.168.2.100    
192.168.122.0/24 dev virbr0  proto kernel  scope link  src 192.168.122.1



出现的192.168.x.x网络可以忽略

测试：  
配置没问题的话，到这一步PC-A就能够ping通PC-B  
[root@PC\_A ~]#ping -c 5 10.10.2.1   
PING 10.10.2.1 (10.10.2.1) 56(84) bytes of data.   
64 bytes from 10.10.2.1: icmp\_seq=1 ttl=61 time=0.752 ms    //ttl之所以是61，是经过3个路由器R\_x、R\_y、R\_z，ttl值原是64，经过1个路由器减一   
64 bytes from 10.10.2.1: icmp\_seq=2 ttl=61 time=0.983 ms   
64 bytes from 10.10.2.1: icmp\_seq=3 ttl=61 time=1.74 ms   
64 bytes from 10.10.2.1: icmp\_seq=4 ttl=61 time=2.12 ms   
64 bytes from 10.10.2.1: icmp\_seq=5 ttl=61 time=1.34 ms   
--- 10.10.2.1 ping statistics ---   
5 packets transmitted, 5 received, 0% packet loss, time 4007ms   
rtt min/avg/max/mdev = 0.752/1.389/2.123/0.498 ms

下面是在PC-A上traceroute到PC-B的路径   
[root@PC\_A ~]#traceroute 10.10.2.1   
traceroute to 10.10.2.1 (10.10.2.1), 30 hops max, 60 byte packets   
 1  10.10.1.254 (10.10.1.254)  0.195 ms  0.066 ms  0.051 ms  //先到网关R\_x   
 2  10.10.12.2 (10.10.12.2)  0.816 ms  0.795 ms  0.764 ms  //再到R\_y   
 3  10.10.23.2 (10.10.23.2)  0.730 ms  0.701 ms  0.670 ms  //然后到达R\_z   
 4  10.10.2.1 (10.10.2.1)  0.920 ms  0.893 ms  0.862 ms  //最后到达PC-B   
//能到PC-B说明已经通了，如果在PC-B上traceroute得到的结果正好与此相反

**linux下静态路由修改命令**

方法一：

添加路由

route add -net 192.168.0.0/24 gw 192.168.0.1

route add -host 192.168.1.1 dev 192.168.0.1

删除路由

route del -net 192.168.0.0/24 gw 192.168.0.1

add 增加路由

del 删除路由

-net 设置到某个网段的路由

-host 设置到某台主机的路由

gw 出口网关 IP地址

dev 出口网关 物理设备名

增加默认路由

route add default gw 192.168.0.1

默认路由一条就够了

route -n 查看路由表

方法二：

添加路由

ip route add 192.168.0.0/24 via 192.168.0.1

ip route add 192.168.1.1 dev 192.168.0.1

删除路由

ip route del 192.168.0.0/24 via 192.168.0.1

add 增加路由

del 删除路由

via 网关出口 IP地址

dev 网关出口 物理设备名

增加默认路由

ip route add default via 192.168.0.1 dev eth0

via 192.168.0.1 是我的默认路由器

查看路由信息

ip route

保存路由设置，使其在网络重启后任然有效

在/etc/sysconfig/network-script/目录下创建名为route- eth0的文件

vi /etc/sysconfig/network-script/route-eth0

在此文件添加如下格式的内容

192.168.1.0/24 via 192.168.0.1

重启网络验证

/etc/rc.d/init.d/network中有这么几行：

# Add non interface-specific static-routes.

if [ -f /etc/sysconfig/static-routes ]; then

grep "^any" /etc/sysconfig/static-routes | while read ignore args ; do

/sbin/route add -$args

done

fi

也就是说，将静态路由加到/etc/sysconfig/static-routes 文件中就行了。

如加入：

route add -net 11.1.1.0 netmask 255.255.255.0 gw 11.1.1.1

则static-routes的格式为

any net 11.1.1.0 netmask 255.255.255.0 gw 11.1.1.1