

IT341 Introduction to System Administration

Project II – Configuring and Using a LAN

Before we start, please make sure that you have a **snapshot** of your finished VM/base installation from Project 1, so that you can easily restore it – if you should somehow break your system during this project's process.

In the following we set up our local area network. This involves configuring...

- the server **it20** (**which has been – or will be – done**)
- and the clients, *itvm21-1a*, *itvm21-1b*, ... *itvm27-4b*, *itvm28-4a*, *itvm28-4b* (**which you do**).

It is important that you understand the differences between the two sections in this and the following project documents.

SPRING 2019-Present NOTE:

Up until Ubuntu 17, network interfaces on an Ubuntu machine were configured with a package called **ifupdown**, which contains various commands -- such as **ifup** and **ifdown**, which activate and deactivate (respectively) specific network interfaces -- and makes use of a special config file (**/etc/network/interfaces**). Since version 17, **ifupdown** has been replaced by a newer package called **netplan** – which is installed by *default*, when you put Ubuntu 18.04 on your VMs. However, this project will have you revert back to **ifupdown** for managing your network connections. We are making this change for two reasons:

1. So that your virtual machines will be using the same network package as the IT Lab LAN's central server, **it20**.
2. Because you are likely to frequently encounter slightly older versions of Ubuntu that still use **ifupdown**, not having yet transitioned to **netplan**.

That said, we will still discuss **netplan**, and it is to your advantage to begin understanding this newer approach to network connection management. Moreover, much of my alterations to the project steps (in light of Ubuntu 18) are a bit "experimental", so it will be even more important that you are diligent about taking snapshots to preserve your successful progress, as we move forwards.

Our server (**it20**) – which presently continues to run Ubuntu Server LTS 14.04 – has two NICs (network interface cards) named **eth0** and **eth1**. **eth0** is attached to the **it.cs.umb.edu** network (inside). Each of the physical machines – and soon your VMs – can be attached (through a switch) to the server's **eth0** by connecting the physical machine to the right-hand side (RHS) network drop. Make sure your network connection is through the **RHS RJ45** network drop so you are connected to the **it20** server and the **it.cs.umb.edu** domain. As usual, you will need to be connected to the RHS.

In both the server and the clients, we must configure two files:

- the **interfaces** file, which configures the NIC
- and the **hosts** file, which associates IP addresses with each of the local hosts (both your virtual machines and real physical computers).

You DO NOT do the sections labeled On the Server. Admins for **it20** do those. You just need to understand them, as they relate to your own Project

2 tasks, for addressing in your lab report. This will apply for **all** projects that have an "**On the Server**" section.

On the server, previous admins have configured the IP tables to forward packets **out from** and **into** our LAN, and to use NAT to implement masquerading so that the outside sees just one host, our server **it20** (more about this later). Finally, on the server (**it20**), we implement DHCP (only on **it20**) for allocating IP addresses. *This will come up in the discussion questions at the end of your lab report...*

For looking at config files on **it20** – and getting some information for your own VM's config files, you can log onto it using **ssh**.

SSH into your Linux account first, then:

Host: 10.200.6.58

Username: it341

Password: Intro2sys

it341 is a special user account that can only read files on **it20** and execute some commands – and nothing else.

On the Server

(This is what previous admins have already done, along with some config changes I will make to help your VMs join the local network. You **DO NOT** do this but you need to read it, look at it on **it20**, understand it and *address it in your lab report entries.*)

1. First, the *interfaces* file **/etc/network/interfaces** had to be edited, configuring the NICs on the server **it20**:

```
# This file describes the network interfaces available on your system
# and how to activate them. For more information, see interfaces(5).

# The loopback network interface
auto lo
iface lo inet loopback

# eth1 (outside)
# This interface connects it20 to the cs.umb.edu domain.
# This is the "primary" interface; one might think that eth0
# would be the primary interface, but not on this box!
# We rely on it20's dhcp to allocate an appropriate ip address.

auto eth1
iface eth1 inet dhcp

# eth0 (inside)
# This interface connects it20 to its subdomain clients: it21,..it28
# and all the VM's itvm2x-yz, for x=1..8, y=2/3, z=a/b
auto eth0
iface eth0 inet static
network 10.0.0.0
address 10.0.0.1
broadcast 10.0.0.255
netmask 255.255.255.0
```

2. Edit the *hosts* file, **/etc/hosts**, associating IP addresses with the hosts. *Look at this file on the server **it20**. What does the **hosts** file do? This will come up in the discussion questions for*

your lab report. Here is a small part of it, on the following page:

```
# /etc/hosts

# Modified by Al Bird Jan 2016 to change the IP scopes of IT20

# Associate ip addresses with names.

# Myself (loop back)
127.0.0.1    localhost
#127.0.1.1  it20.it.cs.umb.edu it20
10.0.0.1    it20.it.cs.umb.edu it20

# Inside vm's

10.0.0.128  itvm21-1a.it.cs.umb.edu itvm21-1a
10.0.0.129  itvm22-1a.it.cs.umb.edu itvm22-1a
10.0.0.130  itvm23-1a.it.cs.umb.edu itvm23-1a
10.0.0.131  itvm24-1a.it.cs.umb.edu itvm24-1a
10.0.0.132  itvm25-1a.it.cs.umb.edu itvm25-1a
```

...

3. Download and install the package for DHCP (This is only for **it20** –not on your VM!)

```
sudo apt-get install dhcp3-server
```

4. Backup the default DHCP configuration:

```
sudo cp /etc/dhcp/dhcpd.conf /etc/dhcp/dhcpd.conf.bak
```

5. And define a new configuration in the file **/etc/dhcp/dhcpd.conf** :

In this file, two important administrative things are done:

- a. In the past, we have restricted the computers to which **it20** assigns IPs, using their MAC addresses. See the file **/etc/dhcp/dhcpd.conf** on the server **it20** for a complete listing.
- b. Also, I need to modify this file to network your virtual machine, and I need to enter the MAC of your virtual machine so the DHCP server sends your VM the correct IP address.)

6. Start the DHCP server by typing:

If running Ubuntu Server 14.04 LTS, as we are on **it20**, then we use the upstart command:

```
sudo service isc-dhcp-server start
```

Now, our DHCP server should be operational.

On the Clients

(You should do these on your virtual machine, *itvm2x-yz*)

Preliminaries

Before we can do the configuration steps on your VM, we must revert from **netplan** to **ifupdown**. To do this, execute the following steps on your VM, as **sysadmin**:

- a) Enter the command: **sudo nano /etc/default/grub**

- b) Go to the following line:

```
GRUB_CMDLINE_LINUX=""
```

- c) Change the line as follows:
`GRUB_CMDLINE_LINUX="netcfg/do_not_use_netplan=true"`
- d) Press **Control+O** and then press **Enter** to save. Press **Control+X** to exit **nano**
- e) Enter the command: `sudo update-grub`
- f) Enter the command: `sudo apt-get install ifupdown`
- g) Enter the command: `ip addr show`
Confirm that you have `ens33` and `lo`. If you do not have `ens33`, then please let me know.
- h) Enter the command: `sudo nano /etc/network/interfaces`
- i) At this point, the file will either be completely empty **OR** only contain comments, which you may ignore. Add the following lines to the file, in the manner specified below:

```
auto lo
iface lo inet loopback

auto ens33
iface ens33 inet dhcp
```

- j) Press **Control+O** and then press **Enter** to save. Press **Control+X** to exit **nano**
- k) Enter the command: `sudo reboot`

After the reboot, log-in as normal and continue...

Primary Part 1 Tasks

1. First, look at the *interfaces* file, configuring the NICs. (Take a look at the file to confirm that the non-comment parts look like the example below.)

Command: `cat /etc/network/interfaces`

```
# ifupdown has been replaced by netplan(5) on this system.  See
# /etc/netplan for current configuration.
# To re-enable ifupdown on this system, you can run:
#   sudo apt install ifupdown

auto lo
iface lo inet loopback

auto ens33
iface ens33 inet dhcp
```

2. Second, edit the *hosts* file. Here, you can start by making a backup of it, using the following command: `sudo cp /etc/hosts /etc/hosts.ORIG`

TIP: Check the `/etc` directory to confirm that `hosts.ORIG` exists

NOTE: Since you are going to be doing a lot of command line editing this semester, it pays to learn tricks such as the following...

- Pasting text into your VM's command line using VMWare's menu

- Copying text from – and pasting it into – command line utilities like PuTTY, Windows Command Prompt, Mac Terminal, etc.
- Preserving the content of a login session, for use in lab reports, etc.

3. Open `/etc/hosts` for editing

```
sysadmin@itvm28-8a:~$ sudo nano /etc/hosts
[sudo] password for sysadmin:
```

a. First, edit the *second* line, as indicated here:

```
127.0.0.1 localhost
127.0.1.1 itvm28-8a
```

→

```
127.0.0.1 localhost
127.0.1.1 itvm28-8a.it.cs.umb.edu itvm28-8a
```

b. After the second line, add the two lines indicated below, exactly:

```
127.0.0.1 localhost
127.0.1.1 itvm28-8a.it.cs.umb.edu itvm28-8a

10.0.0.1    it20.it.cs.umb.edu it20
10.0.0.91  cktest.it.cs.umb.edu cktest
```

The following lines are desirable for IPv6 cap

c. After those two lines, add a block of **64** lines, taken from `it20's /etc/hosts` file, corresponding to IP addresses `10.0.0.128` through `10.0.0.191`:

```
10.0.0.91  cktest.it.cs.umb.edu cktest

# Inside virtual clients for your section

10.0.0.128 itvm21-1a.it.cs.umb.edu itvm21-1a
10.0.0.129 itvm22-1a.it.cs.umb.edu itvm22-1a
10.0.0.130 itvm23-1a.it.cs.umb.edu itvm23-1a
...
10.0.0.190 itvm27-4b.it.cs.umb.edu itvm27-4b
10.0.0.191 itvm28-4b.it.cs.umb.edu itvm28-4b
```

The following lines are desirable for IPv6 cap

(The `...` are just a *placeholder!*)

NOTE: You *need not* type in all those lines. In class, I can demonstrate how to accomplish this *more easily!*

d. After that, add the block of 8 lines for the physical hosts, `it21` through `it28`

```
10.0.0.190 itvm27-4b.it.cs.umb.edu itvm27-4b
10.0.0.191 itvm28-4b.it.cs.umb.edu itvm28-4b
```

```
# Inside real clients
10.0.0.240 it21.it.cs.umb.edu it21
10.0.0.241 it22.it.cs.umb.edu it22
...
10.0.0.247 it28.it.cs.umb.edu it28
```

The following lines are desirable for IPv6 ca

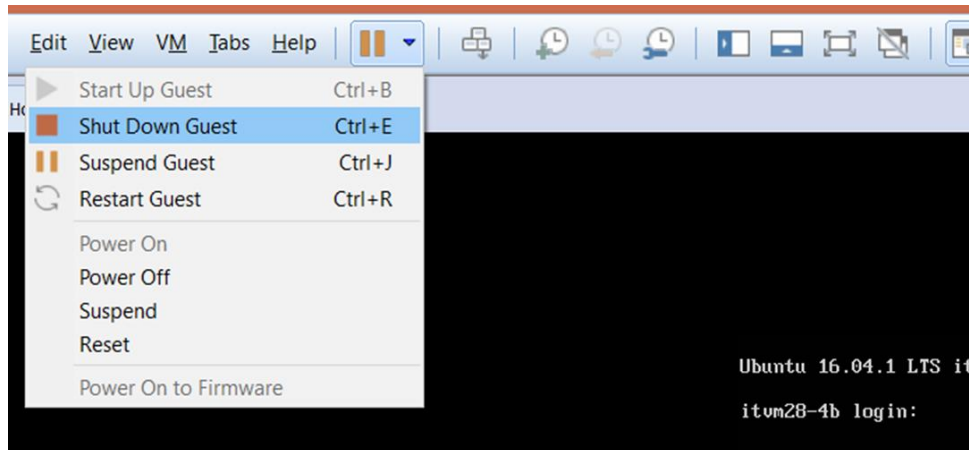
(The `...` are just a *placeholder!*)

- e. Save your work and exit the **nano** utility. When you are back at the command line, restart the networking service:

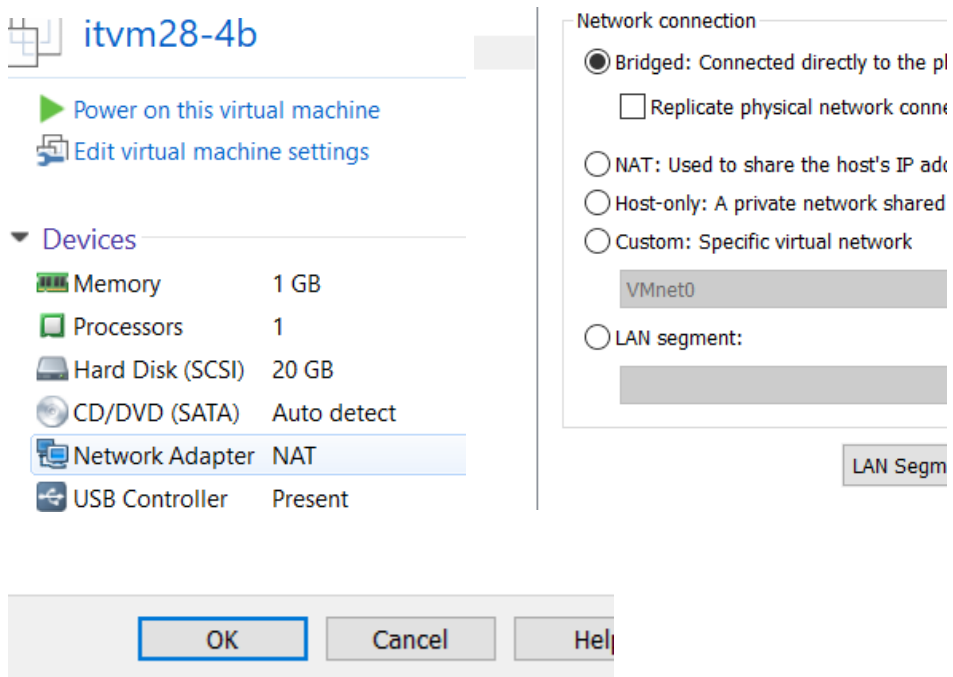
```
sysadmin@itvm28-8a:~$ sudo /etc/init.d/networking restart
```

4. When finished, log out of your VM and shut it down. Once you are sure you have these files correct, *log out* and *shut down* your virtual machine.

```
sysadmin@itvm28-4b:~$ logout
```



5. Change the network settings so, *instead of using NAT for networking, use **Bridged***. Look these (NAT and Bridged) up because you will answer questions at the end of your lab report.



- Using that same VMware dialog, set your MAC Address to: `00:0c:29:8f:XX:YY` where **XX** and **YY** are the **21-28** and **1a, 1b, 2a**, etc. parts of your team name. For example, for the VM of team **itvm24-3b** the MAC should be: `00:0c:29:8f:24:3b`
- Then power up your VM again and log back in as **sysadmin**. Then, you should do an **ip addr show** and confirm that you have the correct IPv4 address. Where will you find the correct IP address for your virtual machine?
- We should now test the networking. See if you can...

```
ping www.yahoo.com
```

```
ping cs.umb.edu
```

```
ping it20
```

`ping it2x` for an x other than your own.

`ssh sysadmin@itvm2x-yz` for a VM other than your own.

Keep in mind that *at least one* of the tests I listed here *will not work*. Think about **why** that might be the case. Come up with some more tests, and write about them (as well as the tests above) in your lab report.

NOTE: The next few steps will involve **temporarily** changing the **interfaces** file – just to see how an interface can be configured statically – and testing the alternative configuration. It is important that you remember to:

- I. Make a *backup* copy of the **interfaces** file before changing it
- II. *Revert* the **interfaces** file to the original after you are done examining/testing the static configuration. *We want our VMs to use DHCP as the default!*

- Now, let's try setting up a **static** interface, rather than relying on **it20**'s DHCP to give us an IP address. First, **save** the **current interfaces** file,

```
sudo cp /etc/network/interfaces /etc/network/interfaces.saved
```

Then, we will edit part of the **/etc/network/interfaces** file; see **/etc/hosts** for the appropriate value for **XXX** below:

```
auto ens33
iface ens33 inet static
network 10.0.0.0
address 10.0.0.xxx
gateway 10.0.0.1
broadcast 10.0.0.255
netmask 255.255.255.0
```

- One need not do a complete restart of the VM to take advantage of this new file; we can simply **restart** networking:

```
sudo /etc/init.d/networking restart
```

- We should now test this. See if you can,

```
ping www.yahoo.com
```

```
ping cs.umb.edu
```

```
ping it20
```

Try the command `ping it2x` for an **x** other than your own.

`ssh sysadmin@itvm2x-yz` for a VM other than your own.

Come up with some more tests, run them, and write about them (as well as the ones indicated above) in your lab report. Again, some tests given here *will not work*. The entire point is for you to *notice* the ones that are not working and *speculate* as to why that is the case.

12. **Restore** the previous (saved) **interfaces** file and restart networking (again)!

```
sudo cp /etc/network/interfaces.saved /etc/network/interfaces
```

```
sudo /etc/init.d/networking restart
```

13. Again, test. When you are confident that you have completed the *these* tasks successfully, be sure to take a **snapshot** of your virtual machine for safety's sake.

14. Do not forget to **shutdown** your virtual machine and **log off** of your desktop -- **UNLESS** you choose to leave it running for remote access (see following...)

Remote Access to Your Virtual Machine:

Presumably, some teams will want to be able to access their VMs remotely, which should be possible for you, as of this current point, following the steps given below:

1. Be sure that the Ethernet cord for your physical machine is plugged into the RHS.
2. Log out of **sysadmin** but **DO NOT** shutdown your VM.
3. Rather than logging off your Windows account on the physical machine...you should **lock** it, instead. *If you don't know what "locking" is – ask!*
4. Then, to access the VM, **ssh** into your Linux account, then into **10.200.6.58** as user **it341** with the (case-sensitive) password **Intro2sys**
5. From there, **ssh** into your VM as **sysadmin**, using your usual password for sysadmin. For example: `ssh sysadmin@itvm24-3b`
6. At that point, you should be able to do your usual command-line work, just as you would sitting in front of the physical machine.
7. Log out of your VM **and** out of **it20**, when finished.

Project 2, Second Part: Implementing NIS

The Network Information Service (NIS), formerly known as Yellow Pages or YP, provides a flat (unstructured) database for keeping track of users and hosts on a local area network. It works best for hundreds of users and/or hosts; Enterprises that have thousands of users and/or hosts might be better off with a more hierarchical database such as DNS or LDAP.

In this exercise, we will set up NIS on our little network and use it to keep track of users and (initially) hosts.

As stated earlier, you may wish to make a snapshot of your VM – in its state following successful completion of the first half of Project 2 – before continuing with these instructions.

See below...

NOTE ALSO that I may have to change these instructions some, if differences between Ubuntu 20 (versus 16/18) are particularly significant here; as such you will want to check for possible updates or alterations!

On the Server, it20

(This is what previous admins have already done, along with some config changes I will make to set up your personal accounts on the local network. You **DO NOT** do this but you need to read it, look at it on **it20**, understand it and *address it in your lab report entries.*)

1. Download and install NIS¹. When asked, say that the *NIS domain* is **it.cs.umb.edu**. The domain name can be anything, so long as it is the same on the *server* and *all clients* on the network.

```
sudo apt-get update
sudo apt-get install nis
sudo apt-get install sysv-rc-conf (Needed later on)
```

2. The installation of NIS includes attempting to start it up. It may not be successful.

We need to look in **/etc/default/nis**. Here, admins made a small change (again, in **boldface**) to say, yes we are the NIS master. Admins did not make any other changes to the file at that time.

```
sysadmin@it20:/etc$ cat /etc/default/nis
#
# /etc/default/nis Configuration settings for the NIS
# daemons.
#
# Are we a NIS server and if so what kind (values: false, slave,
# master)?
NISSERVER=master
# Are we a NIS client?
NISCLIENT=true
# Location of the master NIS password file (for yppasswdd).
# If you change this make sure it matches with /var/yp/Makefile.
YPPWDDIR=/etc
# Do we allow the user to use ypchsh and/or ypchfn? The
# YPCCHANGEOK fields are passed with -e to yppasswdd,
# see it's manpage.
# Possible values: "chsh", "chfn", "chsh,chfn"
YPCCHANGEOK=chsh
# NIS master server. If this is configured on a slave server
# then ypinit will be run each time NIS is started.
```

¹A good NIS reference is http://www.server-world.info/en/note?os=Ubuntu_16.04&p=nis

```
NISMASTER=master
```

```
# Additional options to be given to ypserv when it is started.  
YPSEVRARGS=
```

3. Look at `/etc/nsswitch.conf`. Nothing needs to change for now.

```
# /etc/nsswitch.conf  
#  
# Example configuration of GNU Name Service Switch functionality.  
# If you have the `glibc-doc-reference' and `info' packages  
# installed, try:  
# `info libc "Name Service Switch"  
# for information about this file.
```

```
passwd: compat  
group: compat  
shadow: compat
```

```
hosts: files dns  
networks: files
```

```
protocols: db files  
services: db files  
ethers: db files  
rpc: db files
```

```
netgroup: nis
```

4. Modify `/etc/yp.conf` to identify ourselves (`it20`) as the NIS server.

```
#  
# yp.conf Configuration file for the ypbind process. You can  
# define NIS servers manually here if they can't be found by  
# broadcasting on the local net (which is the default).  
#  
# See the manual page of ypbind for the syntax of this file.  
#  
# IMPORTANT: For the "ypserver", use IP addresses, or make sure  
# that the host is in /etc/hosts. This file is only interpreted  
# once, and if DNS isn't reachable yet the ypserver cannot  
# be resolved and ypbind won't ever bind to the server.
```

```
# ypserver it20.it.cs.umb.edu  
ypserver 10.0.0.1
```

5. Check to make sure our default domain is `it.cs.umb.edu`:

```
sysadmin@it20:/etc$ cat defaultdomain  
it.cs.umb.edu
```

6. Now we need to set up the NIS database.

```
sudo /usr/lib/yp/ypinit -m
```

There are no slave servers so enter **Ctrl-D**, then if **it20.it.cs.umb.edu** (the NIS master server) is in the list the answer **y**

7. Now we need to start the service

```
sudo service ypserv start
```

8. Now we need to start the the ypserv run-level configuration for sysV like init script links

```
sudo sysv-rc-conf ypserv on
```

9. Now we must build the NIS database, and restart NIS. To build the database, we go to directory **/var/yp**, which contains the **make** file, **Makefile**, for building the database and execute it:

```
cd /var/yp
sudo make
```

make is a Unix program that builds applications according to instructions in a "make file"; if you are interested in what a **make file** looks like you can look at **Makefile** in this directory.

The Clients²

1. Download and install NIS. To start with, we need to make sure you have the right **apt** repositories. Thus, on your VM, as sysadmin, enter the following command

```
sudo scp it341@10.0.0.1:/home/ckelly/it341_files/sources.list /etc/apt/sources.list
```

Next, we use **apt-get** to obtain NIS. When asked, say that the *NIS domain* is **it.cs.umb.edu**

```
sudo apt-get update
sudo apt-get install nis
```

When asked for a default domain, supply **it.cs.umb.edu**. It is just a name; any name will do; but, the **nis** server and all clients must have the same name.

2. The file, **/etc/default/nis** should not be changed. However, you should still open the file and look at it. *Command:* **less /etc/default/nis** Type **q** to get out of less utility

As mentioned on the project webpage, there will be something about this file on your VM that differs from the example here. Make a note of it, but it is okay to leave it as is.

```
# /etc/defaults/nis Configuration settings for the NIS daemons.

# Are we a NIS server and if so what kind (values: false, slave,
# master)?
NISSERVER=false

# Are we a NIS client?
NISCLIENT=true
# Location of the master NIS password file (for yppasswdd).
# If you change this make sure it matches with /var/yp/makefile.
YPPWDDIR=/etc

# Do we allow the user to use ypchsh and/or ypchfn ? The
```

²A good NIS reference is http://www.server-world.info/en/note?os=Ubuntu_16.04&p=nis&f=2

```

# YPCHANGEOK fields are passed with -e to yppasswdd, see it's
# manpage.
# Possible values: "chsh", "chfn", "chsh,chfn"
YPCHANGEOK=chsh

# NIS master server. If this is configured on a slave server then ypinit
# will be run each time NIS is started.
NISMASTER=

# Additional options to be given to ypserv when it is started.
YPSERVARGS=

# Additional options to be given to ypbind when it is started.
YPBINDARGS=

# Additional options to be given to yppasswdd when it is started. Note
# that if -p is set then the YPPWDDIR above should be empty.
YPPASSWDDARGS=

# Additional options to be given to ypxfrd when it is started.
YPXFRDARGS=

```

3. The file, `/etc/nsswitch.conf` should look like this (add `nis` to the four indicated lines). Notice that, on the `hosts` line, you will also need to change the order of some elements
(NOTE: You should not change the permissions for this file. Instead, use this command:
`sudo nano /etc/nsswitch.conf` .)

```

# /etc/nsswitch.conf
#
# Example configuration of GNU Name Service Switch functionality.
# If you have the `glibc-doc-reference' and `info' packages
# installed, try:
# `info libc "Name Service Switch"' for information about this file.

passwd: files systemd nis
group: files systemd nis
shadow: files nis
gshadow: files

hosts: dns files nis
networks: files

protocols: db files
services: db files
ethers: db files
rpc: db files

netgroup: nis

```

4. Modify `/etc/yp.conf` to identify `it20` as the NIS server. As the comment suggests, this should not be necessary as broadcasting ought to work, but NIS appears to work more smoothly when we explicitly identify the server...

Command: `sudo nano /etc/yp.conf`

```
#
```

Note that the lines beginning with the # symbol are just *comments*, with no actual effect on system functionality.

```
# yp.conf Configuration file for the ypbind process. You can
# define NIS servers manually here if they can't be found by
# broadcasting on the local net (which is the default).
#
# See the manual page of ypbind for the syntax of this file.
#
# IMPORTANT: For the "ypserver", use IP addresses, or make sure
# that the host is in /etc/hosts. This file is only interpreted
# once, and if DNS isn't reachable yet the ypserver cannot
# be resolved and ypbind won't ever bind to the server.

# ypserver it20.it.cs.umb.edu
ypserver 10.0.0.1
```

5. Check to make sure our default domain is **it.cs.umb.edu**

```
sysadmin@itvm2x-yz:~$ cat /etc/defaultdomain
it.cs.umb.edu
```

6. On your virtual machine, add a home directory for yourself, using your **cs.umb.edu** login name; If you don't have a **cs.umb.edu** user account, GET ONE! For example, if my username were **bj**a, the command would read:

```
sysadmin@itvm2x-yz:~$ sudo mkdir /home/bja
(Both team members need this!)
```

bja is replaced with your CS Linux username, which will also be the name of your home directory on your virtual machine.

7. Do a restart of NIS on your client side

```
sudo systemctl restart ypbind.service
sudo systemctl enable nis.service
```

8. See me for this step. I will do it on **it20** as soon as you are ready!

Then, on the server **it20**, I will add you with the same name you used for your home directory (e.g. I used **bj**a), that is, your **cs.umb.edu** user name. Then, define your new password.

```
sysadmin@it20:~$ sudo adduser --no-create-home bja
```

We *do not* want a home directory created on **it20**, since yours will be hosted on your virtual machines. Of course, it exists *only* on the VM, at present, but that is fine for now – that is where we are working. There is an alternative to **adduser** – the command **useradd** – but the one we use allows us to suppress the creation of a home directory and prompts for a user password.

9. Again, on the server **it20**, as student individual user accounts are added, I will rebuild the NIS database and restart NIS (**I will do this**):

```
cd /var/yp
sudo make
sudo sysv-rc-conf ypbind on
```

10. Then, back on your client, login using your own login name and password that you added to **it20** in step 8. Of course, you will want to use your own login name in place of **bj**a.

```
itvm2x-yz login: bja
Password: *****
```

NOTE: Your VM *should* be accepting log-ins from anyone with an account on `it20`, by way of NIS. If you are **unable** to log into your VM using your own login name, at this point, then we will need to do some troubleshooting:

- Log into your VM as `sysadmin`
- From the command line, see if you can at least log into `it20` itself: `ssh [your login name]@10.0.0.1`
- If this is successful, then logout of `it20` with the `exit` command. This should drop you back in your session on your VM as `sysadmin`.
- This is likely due to a bug where the `rpcbind` utility failing to start for `ypbind`. More details here: <https://bugs.launchpad.net/ubuntu/+source/rpcbind/+bug/1558196>
- Execute the command `ps -ef | grep bind`
- We are looking for entries for both `ypbind` and `rpcbind`, but they will probably not be there this time!
- To fix this, execute these commands

```
sudo apt-get install nfs-common
sudo apt-get install nfs-kernel-server
```
- Log off and reboot your VM.
- Log in as `sysadmin`.
- Once again, execute the command `ps -ef | grep bind`
- You should now see entries for both `ypbind` and `rpcbind`. If you do not, please let me know.
- Log off as `sysadmin` and attempt to log into your VM now as yourself. It should be working now...

This means you *will not* need to do this in the following project – because you will have *already* done it!

You may get a message, which goes like:

```
Could not chdir to home directory /home/bja:
No such file or directory
```

Whether you get this message or not, it simply means that you have no home directory on your own host. This is because you do not yet *own* it. (When you created it in step 6, it was owned by user `root`.) We can fix these things, but you will need to be (the `sudo`-er) `sysadmin` in order to do so. You actually do not need to log out of your own session to become `sysadmin`! Instead, you can start a "nested" session, using the `su` command. *What does "su" stand for? Look up this command, and be prepared to address it in your discussion questions.*

```
bja@itvm2x-yz:/$ su sysadmin
Password:
sysadmin@itvm2x-yz:/$ ls -l /home
total 8
drwxr-xr-x 3 root root 4096 2011-02-07 11:04 bja
drwxr-xr-x 4 sysadmin sysadmin 4096 2011-01-26 11:57 sysadmin
```

This command is unnecessary, if you are *already* logged in as `sysadmin`!

First, we can change your (**and** your partner's) home directory's *owner* using the `chown` command.

```
sysadmin@itvm2x-yz:/$ sudo chown bja /home/bja
```

Next, we can also change your (**and** your partner's) home directory's *group* using the **chgrp** command

```
sysadmin@itvm2x-yz:/$ sudo chgrp bja /home/bja
```

If you used **su sysadmin**, then you can exit from **sysadmin**'s session, and you will land back in your session with your *individual* account. Use the **ls** command as indicated below, and you will see the user *owns* it, and it is also in their *group*:

```
sysadmin@itvm2x-yz:/$ exit
bj@itvm2x-yz:/$ ls -l /home
total 8
drwxr-xr-x 3 bja bja 4096 2011-02-07 11:04 bja
drwxr-xr-x 4 sysadmin sysadmin 4096 2011-01-26 11:57 sysadmin
```

11. Now, if you log out, and log in again (as *yourself*), you should end up in *your* home directory. You should be able to log into any Linux host on the **it.cs.umb.edu** network –*assuming that host has also implemented NIS correctly!* – but for now, you will have a home directory on your own client.

It would be nice if **all** of our home directories were available on **all of the hosts in our network**. We will use the **Network File System (NFS)** to accomplish that in the next project.

Discussion Questions:

1. What is the IP address of the gateway server...
 - For your virtual machine? ← Who is **itvm28-4b**'s gateway server?
 - For your physical host? ← Who is **it28**'s gateway server?
2. What is the IP address...
 - Of your VM? ← What is **itvm28-4b**'s IP address?
 - Of your physical host? ← What is **it28**'s IP address?
3. Explain how you found out all of these things, for *both* the VM *and* the physical host. (**NOTE:** The **/etc/hosts** file is not a valid source for answering **either** of the preceding questions!)
4. What is **DHCP**, and why do we use DHCP? Please explain.
5. What is **NAT** (Network Address Translation), and why do we use it? Please explain.
6. Use Google to search the Internet to find how you can make Ubuntu Server do this NAT and implement the required DHCP.
 - Choose the clearest explanation and make a citation of the webpage. Include as much citation information as you can:
 - Author's name
 - Article/page name
 - Date of publication
 - Name of the overall website
 - URL for the specific page

For example, if you are **itvm28-4b**, then ...

- I should be able to enter the URL and land directly on the page
- At the very *least*, you should be able to provide a title and URL
- **The citation itself goes *here*, as your main answer to question #6.**
- **Copy the *text* of the article/page/explanation, but you will not paste it until the very end of your lab report, *AFTER* your answers to all the other discussion questions.**

7. What is the **hosts** file, and what does it do?
8. Please define and explain the networking settings of "NAT" and "bridged" *in VMWare*. In a previous question, you discussed NAT more generally, but now you need to discuss it in the context of VMs in VMWare, specifically.
9. Now is a good time to read about – and experiment with – these tools and utilities for network troubleshooting:
 - a. **ping** (again)
 - b. **ifconfig**
 - c. **route**
 - d. **traceroute**
 - e. **netstat**

Write about these, based upon

- ❖ Your reading
 - ◆ The Petersen textbook
 - ◆ The **man** pages for each utility
 - ◆ Sources found by searching Google
- ❖ Your tests/experiments during Project 2.

10. On your VM, what aspect of the **/etc/default/nis** file differs from that which was given in the **On The Clients** section of the second half of Project 2? What does that difference mean, and/or is it significant for our purposes? (**NOTE:** *This question has nothing to do with the **/etc/default/nis** file on **it20**, which is another issue entirely.*)
11. The **su** command: What does "**su**" stand for, and what is the command's purpose?
12. Why do we often need to **restart** the VM – or at least restart some specific service – after installing some software or making some change to config files?

Remember to paste into your lab report the **text from the page you found for Question #6...**