

IT 341: Introduction to System Administration

Project I (REMOTE-Summer 2020-Present):

Initial Setup and Using a LAN

Due to the unique situation with the Summer 2020 session -- instruction being completely remote -- the first project for this course will be structured very differently than it would be during a standard session where class is conducted in-person. Normally, students would:

1. Create their own virtual machines (VMs) on the Windows workstations in the IT Lab, using the program **VMWare Workstation Pro 15**.
2. Install **Ubuntu LTS Server 18.04** on those VMs. (*Recently, 20.04!*)
3. Prepare VMs for local network connection, including the installation of **ifupdown** and alteration of virtual hardware configuration.

However, because instruction is remote, this usual progression is not possible. Accordingly, I have created VMs for everyone, with those VMs "developed" and "configured" to the point that each student can access theirs remotely. As such, you will completely the remainder of the preliminary setup using...

- The login credentials that I have e-mailed to you
 - Administrative account username: **sysadmin**
This is the same for every student's VM
 - The name of your VM, of the following form:
This is different for each student (see my e-mail)
- The steps that follow in the sections below

The following section constitutes **Part A** of **Project I**. In order to expedite the process of creating VMs, I created one template -- and then copied that template to the various physical workstations, with minor modification specific to each individual VM. You will carry out the remaining steps to set up your own VM's unique configuration.

Project I, Part A

In the steps that follow below, I am using the non-existent "placeholder" name **itvm28-8a** in the examples. Please replace said placeholder with the name of your own VM, as provided in my recent e-mail to you.

1. Log into your Linux account for the *command line*, using the utility of your choice -- PuTTY, PowerShell, Command Prompt, Mac OS X Terminal, etc.

```
$ ssh cs110ck@users.cs.umb.edu
cs110ck@users.cs.umb.edu's password: _
```

You will, of course, replace **cs110ck** with your own Linux username!

- From there, log into it20 using the indicated command

```
cs110ck@itserver6:~$ ssh it341@10.200.6.76
it341@10.200.6.76's password:
```

The (*case-sensitive*) password is: **Intro2sys**

Your resulting prompt will look something like this...

```
Last login: Mon Jun 15 18:58:10 2020 from itserver6.cs.umd.edu

Could not chdir to home directory /home/it341: No such file or directory

it341@it20:/$
```

- SSH into your VM as **sysadmin**. Replace **itvm28-8a** with your VM name. For your first login as **sysadmin**, right now, you will use this password: **template**

```
it341@it20:/$ ssh sysadmin@itvm28-8a

Could not create directory '/home/it341/.ssh'.

The authenticity of host 'itvm28-8a (10.0.0.91)' can't be established.
ECDSA key fingerprint is 0d:7c:a0:58:de:e8:fd:e9:55:ae:87:76:c1:a0:58:de.
Are you sure you want to continue connecting (yes/no)? yes
Failed to add the host to the list of known hosts (/home/it341/.ssh/known_hosts).

sysadmin@itvm28-8a's password:
```

You will *change* the **sysadmin** password *shortly*!

- When you first login, you will see that your VM's *internal* hostname is currently **template**, which is the case for *all* student VMs, in the beginning.

```
0 updates are security updates.

Last login: Tue Jun 9 19:50:42 2020

sysadmin@template:~$
```

This, too, will be changed very soon! This is part of the initial setup for all students in the class. Your job, as you will soon see, will include changing your VM's internal hostname.

For now, run the command `mkdir sessions` to create a directory inside `sysadmin`'s home, followed by the `ls` command, to confirm directory creation

```
sysadmin@template:~$ mkdir sessions
sysadmin@template:~$ ls
sessions
```

5. Run the `script` command, with the `--flush` option, to preserve your command-line interaction for later use.

```
sysadmin@template:~$ script --flush $HOME/sessions/$(date +%Y%m%d_%H%M%S").txt
Script started, file is /home/sysadmin/sessions/20200616_020552.txt
```

Here it is again, with selectable text in this PDF:

```
script --flush $HOME/sessions/$(date +%Y%m%d_%H%M%S").txt
```

Notice how this creates an output file with a *unique* filename! Later on, we will see how you can upload these files *from* your VM *to* your Linux account.

6. First, we want to change `sysadmin`'s password from `template` to the unique VM name. In the example, the hypothetical VM name is `itvm28-8a`, but in *your* case, it will be the name I provided to you. Use the appropriate passwords, as indicated below the image....

```
sysadmin@template:~$ sudo passwd sysadmin
[sudo] password for sysadmin:
Enter new UNIX password:
Retype new UNIX password:
passwd: password updated successfully
```

```
[sudo] password for sysadmin: template
Enter new UNIX password: itvm28-8a
Retype new UNIX password: itvm28-8a
```

Henceforth, `sysadmin`'s password on your VM will be the VM name...

7. Use the `hostnamectl` command to confirm your VM's *current* hostname, which will still be `template`, at this point.

```
sysadmin@template:~$ hostnamectl
Static hostname: template
Icon name: computer-vm
```

- Now, change your VM's hostname to the appropriate name. In the example, the VM name is `itvm28-8a`, but in *your* case, it will be the name I provided to you.

```
sysadmin@template:~$ sudo hostnamectl set-hostname itvm28-8a
```

- Use the `hostnamectl` command *again* to confirm your VM's *new* hostname, which should now be the one specified in Step #8, at this point.

```
sysadmin@template:~$ hostnamectl
Static hostname: itvm28-8a
Icon name: computer-vm
Chassis: vm
Machine ID: 3ae743eb48164b2aac7948558e0f7bed
```

- Look at your VM's `/etc/hosts` file, as it currently exists. Notice that `template` is still listed as the host name here, which will be changed in the next step.

```
sysadmin@template:~$ cat /etc/hosts
127.0.0.1 localhost
127.0.1.1 template

# The following lines are desirable for IPv6 capable hosts
::1 ip6-localhost ip6-loopback
fe00::0 ip6-localnet
ff00::0 ip6-mcastprefix
ff02::1 ip6-allnodes
ff02::2 ip6-allrouters
```

- Use the following command to make this small edit in `/etc/hosts`

```
sysadmin@template:~$ sudo sed -i 's/template/itvm28-8a/' /etc/hosts
```

Be sure you are typing the command correctly, substituting your own VM name in place of `itvm28-8a`

- Look at `/etc/hosts` again, to confirm the change took place:

```
sysadmin@template:~$ cat /etc/hosts
127.0.0.1 localhost
127.0.1.1 itvm28-8a

# The following lines are desirable for IPv6 ca
::1 ip6-localhost ip6-loopback
```

Eventually, you will need to make some more extensive edits to this file...

13. Exit from `script`

```
sysadmin@template:~$ exit
exit
Script done, file is /home/sysadmin/sessions/20200616_0
```

14. Reboot your VM

```
sysadmin@template:~$ sudo reboot
[sudo] password for sysadmin:
Connection to itvm28-8a closed by remote host.

Connection to itvm28-8a closed.
```

At this point, you will be automatically dropped back at your `it341@it20` prompt

15. After a couple minutes, try to ping your VM to confirm it has finished rebooting

```
it341@it20:/$ ping -c 5 itvm28-8a
PING itvm28-8a.it.cs.umb.edu (10.0.0.91) 56(84)
64 bytes from itvm28-8a.it.cs.umb.edu (10.0.0.91): icmp_seq=1 ttl=64 time=0.740 ms
64 bytes from itvm28-8a.it.cs.umb.edu (10.0.0.91): icmp_seq=2 ttl=64 time=1.477 ms
64 bytes from itvm28-8a.it.cs.umb.edu (10.0.0.91): icmp_seq=3 ttl=64 time=1.812 ms
64 bytes from itvm28-8a.it.cs.umb.edu (10.0.0.91): icmp_seq=4 ttl=64 time=0.417 ms
64 bytes from itvm28-8a.it.cs.umb.edu (10.0.0.91): icmp_seq=5 ttl=64 time=0.740 ms

--- itvm28-8a.it.cs.umb.edu ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 2500ms
rtt min/avg/max/mdev = 0.740/1.477/1.812/0.417 ms
```

Replace `itvm28-8a` with your *own* VM name, naturally.

16. Sign into your VM as `sysadmin` again, this time using your VM name as the password

```
it341@it20:/$ ssh sysadmin@itvm28-8a
Could not create directory '/home/it341/.ssh'.

The authenticity of host 'itvm28-8a (10.0.0.91)' can't be established.
ECDSA key fingerprint is 0d:7c:a0:58:de:e8:fd:e9:55:ae:87:76:
Are you sure you want to continue connecting (yes/no)? yes
Failed to add the host to the list of known hosts (/home/it341/.ssh/known_hosts)

sysadmin@itvm28-8a's password:
Welcome to Ubuntu 18.04.3 LTS (GNU/Linux 4.15.0-101-generic x86_64)
```

17. On login, you will probably notice your prompt now reflects your VM's unique name, as in the image below. Run `ls` to see the `sessions` directory within `sysadmin`'s home

```
sysadmin@itvm28-8a:~$ ls
sessions
```

18. Run the `script` command again with `--flush` to preserve your CLI session interaction

```
sysadmin@itvm28-8a:~$ script --flush $HOME/sessions/$(date +"%Y%m%d_%H%M%S").txt
Script started, file is /home/sysadmin/sessions/20200616_021118.txt
```

```
script --flush $HOME/sessions/$(date +"%Y%m%d_%H%M%S").txt
```

19. Run `hostnamectl` again to see your VM's current hostname

```
sysadmin@itvm28-8a:~$ hostnamectl
Static hostname: itvm28-8a
Icon name: computer-vm
Chassis: vm
Machine ID: 3ae743eb48164b2aac79485586
```

20. View the contents of `/etc/network/interfaces`

```
sysadmin@itvm28-8a:~$ cat /etc/network/interfaces

# ifupdown has been replaced by netplan(5) on this system
# /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
```

21. Run `ip addr show` to see your current network configuration, especially `ens33`

```
sysadmin@itvm28-8a:~$ ip addr show
ens33: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 10.0.0.91 netmask 255.255.255.0
    inet6 fe80::20c:29ff:fe8f:ffff pref
    ether 00:0c:29:8f:ff:ff txqueuelen 1000
    RX packets 1071 bytes 87517 (87.5 KB)
    RX errors 0 dropped 0 overruns 0
    TX packets 117 bytes 16735 (16.7 KB)
    TX errors 0 dropped 0 overruns 0 carrier 0
lo: flags=73<UP,LOOPBACK,RUNNING> mtu 65536
```

22. Use the following **ping** commands, one after the other:

```
ping -c 5 www.google.com
ping -c 5 www.cs.umb.edu
ping -c 5 10.0.0.1
ping -c 5 10.0.0.91
```

```
sysadmin@itvm28-8a:~$ ping -c 5 www.google.com
PING www.google.com (172.217.11.4) 56(84) bytes of data:
64 bytes from 172.217.11.4: icmp_seq=1 ttl=64 time=0.123 ms
```

```
sysadmin@itvm28-8a:~$ ping -c 5 www.cs.umb.edu
PING vm54.cs.umb.edu (158.121.106.224) 56(84) bytes of data:
```

```
sysadmin@itvm28-8a:~$ ping -c 5 10.0.0.1
PING 10.0.0.1 (10.0.0.1) 56(84) bytes of data:
```

```
sysadmin@itvm28-8a:~$ ping -c 5 10.0.0.91
PING 10.0.0.91 (10.0.0.91) 56(84) bytes of data:
```

We would expect *successful* results for all four.

23. View contents of **/etc/hosts** again:

```
sysadmin@itvm28-8a:~$ cat /etc/hosts
127.0.0.1 localhost
127.0.1.1 itvm28-8a

# The following lines are desirable for IPv6 capable hosts
::1 ip6-localhost ip6-loopback
```

24. Exit from **script**


```
sysadmin@itvm28-8a:~$ exit
exit
Script done, file is /home/sysadmin/sessions/202006
```

25. 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
```

- f. When finished, you may log out of your VM (which will remain running), bringing you back to **it20** :

```
sysadmin@itvm28-8a:~$ exit
logout
Connection to itvm28-8a closed.

it341@it20:/$
```

- g. Log out of **it20**, bringing you back to your Linux account:

```
it341@it20:/$ exit
logout
Connection to 10.200.6.76 closed.

cs110ck@itserver6:~$
```


Project I, Part B

(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.

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.
```

¹A good NIS reference is http://www.server-world.info/en/note?os=Ubuntu_20.04&p=nis

```
YPPWDDIR=/etc
# Do we allow the user to use ypchsh and/or ypchfn? The
# 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 ypinitt will be run each time NIS is started.
NISMASTER=master

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

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

```
# 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
```

- 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**

- Now we need to start the service

```
sudo service ypserv start
```

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

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

- 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

- Log into your VM as **sysadmin**. As a reminder, that means...
From your Linux account, log into **it20** as **it341** **OR** as yourself:

```
cs110ck@itserver6:~$ ssh cs110ck@10.200.6.76
cs110ck@10.200.6.76's password:
```

From **it20**, log into your VM as **sysadmin**:

```
cs110ck@it20:/$ ssh sysadmin@itvm28-8a
sysadmin@itvm28-8a's password:
```

```
sysadmin@itvm28-8a:~$
```

- 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, though you can use your own username instead of **it341**:

```
sysadmin@itvm28-8a:~$ sudo scp it341@10.0.0.1:/home/ckelly/it341_files/sources.list /etc/apt/sources.list
[sudo] password for sysadmin:

it341@10.0.0.1's password:
Could not chdir to home directory /home/it341: No such file or directory
sources.list                                100% 2739    1.3MB/s   00:00
```

²A good NIS reference is http://www.server-world.info/en/note?os=Ubuntu_20.04&p=nis&f=2

Command text below, though you can replace `it341` with your own username:

```
sudo scp it341@it20:/home/ckelly/it341_files/sources.list /etc/apt/sources.list
```

(Notice that we can also use `it20` in place of `10.0.0.1`, since the hostname resolves!)

Next, we use `apt-get` to perform a local database update – and then to obtain `NIS`.

```
sysadmin@itvm28-8a:~$ sudo apt-get update
```

```
sysadmin@itvm28-8a:~$ sudo apt-get install nis
```

When asked for a default domain, supply `it.cs.umb.edu`. It is just a name; any name would actually suffice; 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, just for informational purposes.

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.

```
sysadmin@itvm28-8a:~$ cat /etc/default/nis
```

```
# /etc/defaults/nis Configuration settings for the NIS daem
# Are we a NIS server and if so what kind (values: false, s
# master)?
NISSERVER=false
```

3. Edit the file `/etc/nsswitch.conf`

```
sysadmin@itvm28-8a:~$ sudo nano /etc/nsswitch.conf
```

Add `nis` to the four indicated lines. (Notice that, on the `hosts` line, you will also need to change the order of some elements!)

```
# /etc/nsswitch.conf
#
# Example configuration of GNU Name
# If you have the `glibc-doc-referen
# `info libc "Name Service Switch"
#

passwd:          compat systemd
group:           compat systemd
shadow:          compat
gshadow:         files

hosts:           files dns
networks:        files
```

```
# /etc/nsswitch.conf
#
# Example configuration of GNU Name Se
# If you have the `glibc-doc-referenc
# `info libc "Name Service Switch" fo
#

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

hosts:           dns files nis
networks:        files
```

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..

```
sysadmin@itvm28-8a:~$ 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
```

You may notice that we identify `it20` by its local IP address (`10.0.0.1`) rather than its hostname..

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

```
sysadmin@itvm28-8a:~$ 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; for example, if my username were `cs110ck`, the command would read:

```
sysadmin@itvm28-8a:~$ sudo mkdir /home/cs110ck
```

(Both team members need this, if you're working in a team scenario!)

7. Do a restart of NIS on your client side

```
sysadmin@itvm28-8a:~$ sudo systemctl restart ypbind.service
```

```
sysadmin@itvm28-8a:~$ sudo systemctl enable nis.service
```

8. During a normal semester – conducted in the IT Lab – I would do this on `it20` for individual students, on an ongoing basis, as soon as they were ready.

On the server `it20`, I would add you with the same name you used for your home directory (e.g. I used `cs110ck`), that is, your `cs.umb.edu` user name. Then, you would define your new password.

```
ckelly@it20:~$ sudo adduser --no-create-home cs110ck
```

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. During a remote semester, I have different process that I use to create individual student accounts on `it20`...

9. Again, on the server `it20`, as student individual user accounts are added, I would rebuild the NIS database and restart NIS, on an as-needed basis:

```
ckelly@it20:~$ cd /var/yp
ckelly@it20:~$ sudo make
ckelly@it20:~$ sudo sysv-rc-conf ypbind on
```

10. Then, back on your client, following the completion of Step #7 above...

Logout of `sysadmin`, bringing you back to your `it20` prompt:

```
sysadmin@itvm28-8a:~$ exit
logout
Connection to itvm28-8a closed.

cs110ck@it20:/$
```

Log back in, using *your own* login name and password that we set up in a previous class session. Of course, you will want to use *your own* login name in place of `cs110ck`.

```
cs110ck@it20:/$ ssh cs110ck@itvm28-8a
cs110ck@itvm28-8a's password:
```

You may get a normal prompt...

```
cs110ck@itvm28-8a:~$
```

...or you may get something like *this*:

```
Could not chdir to home directory /home/cs110ck: No such file or directory
cs110ck@itvm28-8a:/$
```

Whether you get one or the other, it would simply mean that you have no home directory on your own host. This is because you do not yet *own* it. (When you created the directory some steps ago, it was owned by user `root`.) We will fix this, but you will need to be (the `sudo`-er) `sysadmin` in order to do so.

However, 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.*) Please note that the password requested is that of `sysadmin` and *not* your own!

```
cs110ck@itvm28-8a:~$ su sysadmin
Password:
```

Perform a long listing on /home to see current ownership, group, and permissions on contents:

```
sysadmin@itvm28-8a:~$ ls -l /home
total 8
drwxr-xr-x 3 root root 4096 2011-02-07 11:04 cs110ck
drwxr-xr-x 4 sysadmin sysadmin 4096 2011-01-26 11:57 sysadmin
```

First, we can change your (**and**, if you're in a team, your partner's) home directory's *owner* using the **chown** command:

```
sysadmin@itvm28-8a:~$ sudo chown cs110ck /home/cs110ck
```

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

```
sysadmin@itvm28-8a:~$ sudo chgrp cs110ck /home/cs110ck
```

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@itvm28-8a:~$ exit
cs110ck@itvm28-8a:~$ ls -l /home
total 8
drwxr-xr-x 3 cs110ck cs110ck 4096 2011-02-07 11:04 cs110ck
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 machine 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 are
next page!**

Discussion Questions:

1. What does **LTS** *stand for*, and what does that *mean*? Please explain.
2. What is an **LVM**? (It stands for "logical volume manager".) Please explain
 - What a *logical volume* is
 - And what a *logical volume manager* does.
3. Please explain what the **sudo** command is, how it is used, and why we need it.
4. Please explain what **APT** and **apt-get** are and why we use them.
5. What is the IP address of...
 - Your virtual machine?
 - Your virtual machine's *gateway server*?

(You should be able to do the first using a command we already used in class! The second will likely require a bit of research for how to find that information...)

6. Explain how you found out both of these things for the previous question. (**NOTE:** *The **/etc/hosts** file is not a valid source for answering either part of the preceding question! The **hosts** file only tells us what IP address a hostname should resolve to; whether the host is actually at that IP is another matter entirely.*)
7. What is **DHCP**, and why do we use DHCP? Please explain.
8. What is the **/etc/hosts** file, and what does it do?
9. What is **NAT** (Network Address Translation), and why do we use it? Please explain.
10. 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.
11. Take a little time to read about – and experiment with – these tools for network troubleshooting:
 - a. **ping** (again)
 - b. **ifconfig** (and the **ip** cmd, too)
 - 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 the project.
12. The **su** command: What does "**su**" stand for, and what is the command's purpose?
13. 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?

This is *probably* the final version, but check back for updates to PDF, just in case!