Tips for Working Effectively

- Linux Tips
- Basics Points
- VMWare issues
- Ubuntu/CLI issues
- Networking issues
- Doing classwork

Linux Tips

- Ending tasks at the command line
  - I have seen some students using **Control-Z** to terminate a running task
    - What that key combo (**Control-Z**) does is quite different:
      - Suspends the running task (not terminate)
      - Pushes the suspended task into the background…
      - …to be resumed when you expressly foreground that task
    - This is actually very counterproductive as it can
      - Prevent you from logging out due to "stopped jobs"
      - Create issues when editing files
  - The standard terminator combo is **Control-C**

Linux Tips

- The standard terminator combo is **Control-C**, which more or less stops the current running process in its tracks!
- Sometimes, **Control-C** will not work, and you can try logging into the server on another shell session, find the relevant process id, and terminate it with the **kill** command. See Example:

  ps -ef | grep [your username]

  - (Identify id of offending process)

  kill [proc id #]

  - (If that does not work, retry with the **-9** option…)
- If there is a standard way to exit a utility cleanly, use that instead. (Press **q** to exit a pager, use **Control-X** to exit **nano**)

Linux Tips

- Pagers
  - These are utilities that allow you do view (potentially) long text files, one screen at a time.
  - Text editors like **nano** *can* be used for viewing text files, but are really not built for that purpose.
  - Learn to use a pager like less
    - **Enter** and **Control-P** → To move forward or backward, respectively, by one *line*
    - **Space** and **Control-B** → Forward or backward by one *screen*
    - **q** → To *quit*

Linux Tips

- Pagers
  - Standard navigation keys (up arrow, down arrow, Home, End, etc.) will also often work!
  - The **less** utility has *many* more options and customizations of which you can take advantage.
  - Where to look:
    - http://www.thegeekstuff.com/2010/02/unix-less-command-10-tips-for-effective-navigation/
    - At the command line…

```
man less
less --help | less
```

Linux Tips

- I/O redirection
  - Three I/O streams
    - `Standard input` (file descriptor: **0**)
    - `Standard output` (file descriptor: **1**)
    - `Standard error` (file descriptor: **2**)
  - Examples:
    - **Standard output into file (overwrite): [command] > file.txt**
    - **Standard output into file (append): [command] >> file.txt**
    - **Standard error into file (overwrite): [command] 2> file.txt**

Linux Tips

- More redirection examples:
  - **Standard output into file (overwrite), with standard error into standard output:**

    [command] > file.txt 2>&1

  - **Standard output and error into separate files:**

    [command] 1> output.txt 2>> error.log

- ***Pipes*** – let the *output* of one command be *input* to the next:

  tail -300 /var/log/auth.log | grep Invalid | less

  - This allows for better management of output.
  - It is a form of *redirection* that can be combined with the previous

Linux Tips

- Other tips…
  - Tab completion: Type in part of an identifier, press **TAB** for completion
  - Use the **▲** (up) and **▼** (down) arrows to get to commands in your recent CLI history
  - Remember key combos for command line:
    - **Ctrl+A** → *start* of line
    - **Ctrl+E** → *end* of line
    - **Ctrl+U** → delete everything *before* cursor
    - **Ctrl+K** → delete everything *after* cursor
    - **Ctrl** plus **L**/**R** arrow → move cursor one *word* at a time
  - On your VM, as **sysadmin** , do not forget **sudo**, especially when editing files. **nano** will not tell you that you lack permissions until you try to save!
  - But…also remember where **sudo** cannot be used! And by whom…

Some Basic Points

- Passwords are one issue where many students could stand to improve.
  - This is especially so if your major is Computer Science or Information Technology.
  - And even MORE so, if you are on the System Administration track of the IT program!
- What makes a password so valuable are two things:
  - You know it
  - Other people, programs, etc. DO NOT
    - (And, of course, cannot figure it out!)

Some Basic Points

- Two areas to consider improving:
  1. Recall: Work on your ability to memorize and recall passwords
     - Carrying around passwords written down (e.g., in a notebook) compromises their security.
       - What if someone else gets ahold of it?
       - Or even just sees it for a moment?
     - Furthermore, it is just a bit *unsettling* to see a CS or IT professional glancing at a piece of paper because they cannot otherwise remember their own password!
     - If you need help recalling many different passwords (that may have variations), you might considering keeping a guide to them in a secure, protected file -- that others cannot access!

Some Basic Points

2. Complexity: This should go without saying, but it should be extremely difficult for anyone -- amateur or professional -- to guess/crack your password.
   - Technology is always making it easier and faster to crack passwords, with more sophisticated algorithms and more powerful hardware.
   - Certainly you should avoid using things like...
     - Names and Places
     - Addresses and Dates
     - Words in the dictionary
     - Common default passwords
   - Also, be wary of brute-force approaches

Some Basic Points

- Some basic password tips:
  - Longer is better
  - Capital and lowercase letters
  - Digits and special characters (e.g., ! @ # $ % ^ & *)
- Try out some online "password strength checkers"
  - Instead of using your own password, though, try something that looks like a password you might use
  - Take note of the cracking time
    - How long would it take to crack your password?
    - Using simpler software and hardware?
    - Using more complex tools that a professional hacker might employ?
- If the kinds of passwords you might use can be cracked in under a few millenia, you should probably try harder!

Some Basic Points

- Of course, when possible, you may want to "graduate" from username/password authentication to key-based
  - Public-private key pairs are more secure!
  - But that's a topic for another day...

- Predictably, computer programming and information technology are going to involve lots of typing.
  - A system administrator will have plenty of typing to do.
  - Plus, it is a necessity in this course.
- As such, you should probably take an honest-self assessment of your personal typing strategy

Some Basic Points

- Ask yourself…
  - How fast can you type? How many words-per-minute?
  - How accurately can you type? Do you get it right the first time, or do you constantly have to go back and correct?
  - Do you know where the keys for different characters are on the keyboard?
    - Letters and Digits?
    - Special Characters?
    - Which characters require use of ***Shift***?
  - Do you actually have a strategy, in the first place? (In other words, something better than "hunt and peck"?)

Some Basic Points

- **Shift** vs. **Caps Lock**
  - In formal training for typing, they generally teach you to use **Shift+** `[letter]` to get a capitalized version
  - The **Caps Lock** key is for situations when one needs to type many capital letters in sequence
  - Some students have developed a habit of repeatedly turning **Caps Lock** on and off for individual capital letters

Some Basic Points

- It is not necessarily "wrong,"" but in many cases, this practice can actually cause errors.
  - For example, if typing in a password, you usually will not see the characters you are typing
  - As such, **Caps Lock** might cause you to unknowingly enter letters of the wrong case
- Generally, it is best to use the right tool (or key) for the task

VMWare - Regain ownership of VM

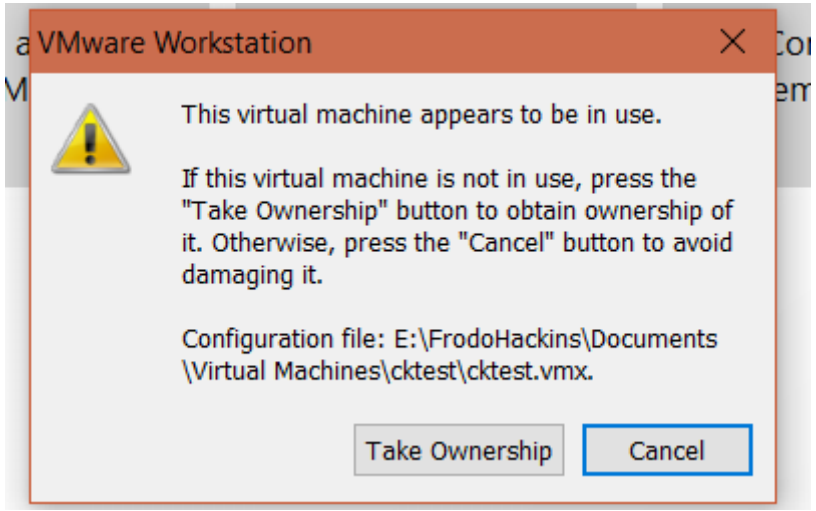
- While your VM is started up and running, it will have several temporary "lock files" amongst your other VM files inside your team's directory (e.g., `C:\IT341\section2b`, etc.)
- You can recognize them because they may look like *folders*, and they will have the extension `.lck`
- These lock files signal that the virtual machine is *in use*, in order to prevent other users from simultaneously booting your VM, writing to its virtual disk, and possibly corrupting data.

VMWare - Regain ownership of VM

- When you shut down your VM cleanly...
  - By choosing ***"Shut Down Guest"*** from VMWare's menu
  - Or by executing `sudo shutdown` from the command line
- ...VMWare will remove the lock files itself.
- In contrast, there is a such thing as a "dirty" shutdown when
  - The virtual machine crashes
  - The virtualization software itself (i.e., VMWare) fails
    - VMWare crashes
    - The physical host machine process is terminated
  - The physical host reboots or shuts down before the VM itself can receive a proper shutdown

VMWare - Regain ownership of VM

- In this event, the aforementioned "lock files" may not be deleted, so the next time you or your partner attempt to start up your VM, you will receive the following error message:



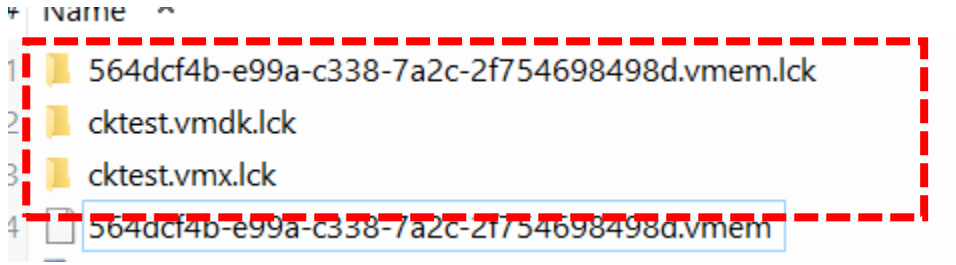
- First, click **Cancel** -- *NOT* "Take Ownership". Let's not create anymore problems, of course.

VMWare - Regain ownership of VM

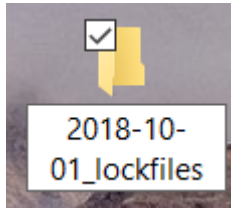
- Second, it could be that your partner is currently logged in under his/her Windows account and running your VM from there.
  - In this case, just have your partner ***"Switch User"*** into their account, which is already logged in.
  - Sometimes, this may be the case, if your team forgets who is doing what. At the very least, it's worthwhile to check and be sure.

VMWare - Regain ownership of VM

- Third, assuming that the previous does not apply, navigate to the folder containing your team's VM files (e.g., **C:\IT341\section3a**)
  - You will likely see multiple folder-like icons ending in **.lck** (Make sure you can see folders' complete names!)

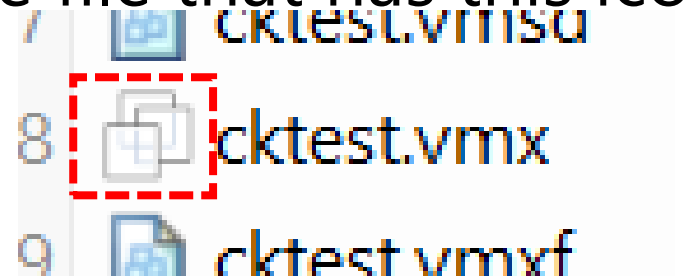


  - On your desktop, create a folder with the following name: **YEAR-MM-DD_lockfiles**
    - **YEAR** is the current *year* -- e.g., 2018
    - **MM** is the 2-digit *month* -- such as 03 or 11
    - **DD** is the 2-digit *day*



VMWare - Regain ownership of VM

- Select all of those items, ending in **.lck**, and move (not copy) them to the new folder you created on the desktop.
- From your VM files, double-click the file that has this icon and bears your team name.
  - If you have file extensions visible, then it will have the extension **.vmx**



  - When prompted, you can go ahead and choose VMWare Workstation 12 Pro as the default app for opening **.vmx** files

- At this point, if your VM was not already running, you should be able to start it up as you normally would.

VMWare - Regain ownership of VM

- You will mostly likely need to do this after the physical machine is restarted or shutdown -- whether it is an administrative update to the whole lab, an individual machine issue, a power outage, etc.
- Once you are sure your VM is running okay, you probably do not need to keep the lock files on your desktop, either!

VMWare - Fix bridge issue

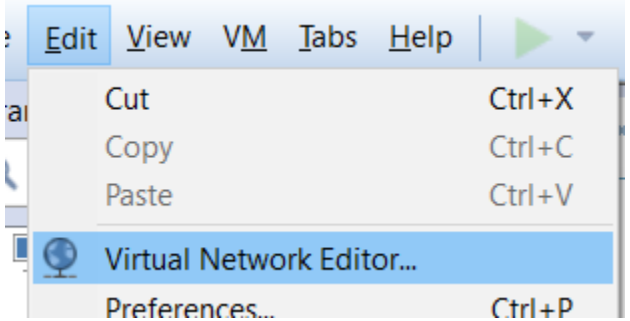
- In Project 2, you change your VM's network mode from ***NAT*** to ***Bridged***.
- Depending on how VMWare's networking settings are currently configured, when (re)booting your VM, the boot process may stall at the following point:

```
A start job is running for Raise network interfaces...
```

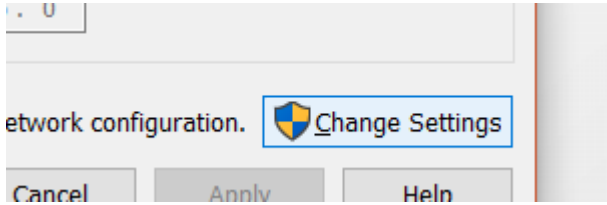
- This is usually due to VMWare's default settings for bridging an interface...

VMWare - Fix bridge issue

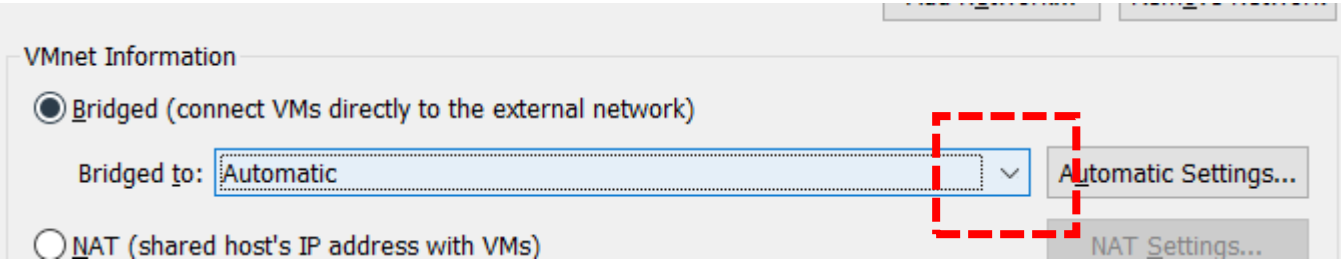
- The relevant setting needs to be changed to proceed more smoothly.
  - Fortunately, it is a VMWare-wide setting, which only needs changed once per physical host.
  - It may already be changed on your physical host, in which case you will not need to do anything about this.
- If you get the stalled boot process, one thing to do is ***"Shut Down Guest"*** (aborting the stalled boot process) and then go to VMWare Workstation's menu...
  - Go to *Edit* -> *Virtual Network Editor*



VMWare - Fix bridge issue

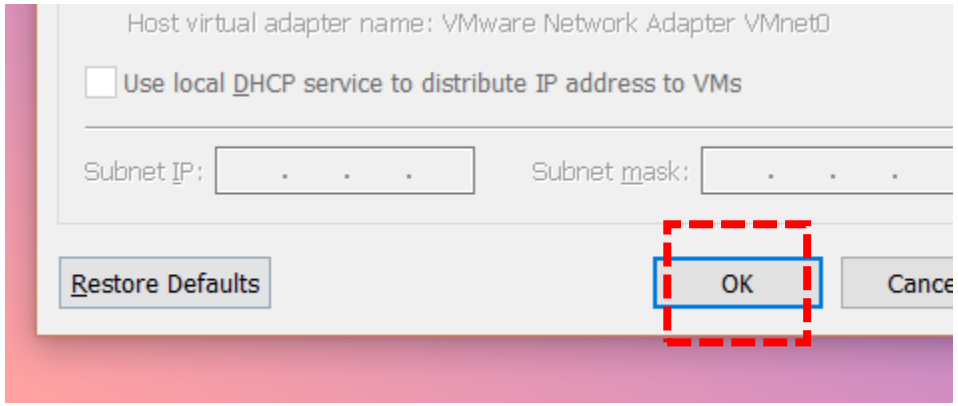


- Choose *Change Settings* (It will say that admin privileges are required, but ignore this.)
- When prompted if you want to allow the app to make changes, choose *Yes*
- Halfway down the dialog box, you will see the heading *"VMNet Information"* and the radio button for *"Bridged"* will be selected.
- Underneath, if it says *"Bridged to: **Automatic**"*, that may be your problem.
- Using the dropdown, change *"Automatic"* to the **"Intel"** interface, specifically.



VMWare - Fix bridge issue

- Click ***OK***, and then attempt to start your VM again.

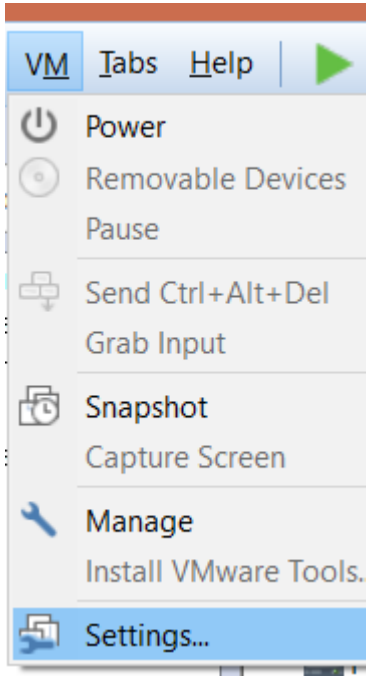


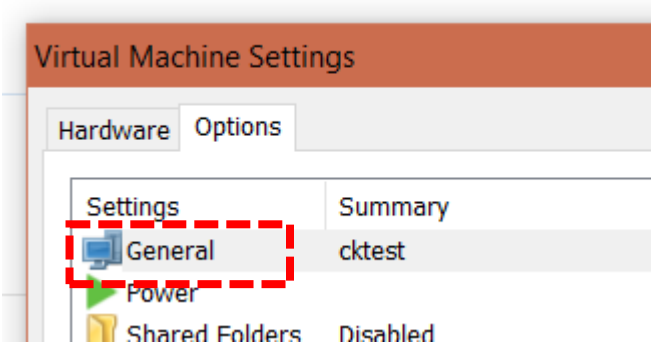
- The problem here is that your VM's virtual NIC may get bridged to the incorrect interface on the physical machine -- which we fix by explicitly telling VMWare to bridge to *the physical Intel interface*.

VMWare - Rename your VM

- If you gave your VM the wrong name while creating it in VMWare, you can change it.
- Make sure your VM is shut down
  - Log off **sysadmin** (or any other user at the command line)
  - *Shut Down Guest*
- With your VM being the active tab, go to

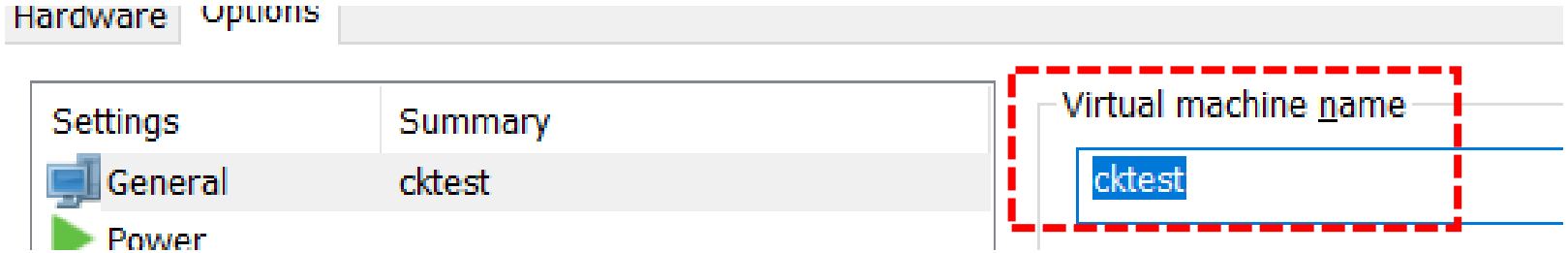
VM -> ***Settings*** -> ***Options*** -> ***General***





VMWare - Rename your VM

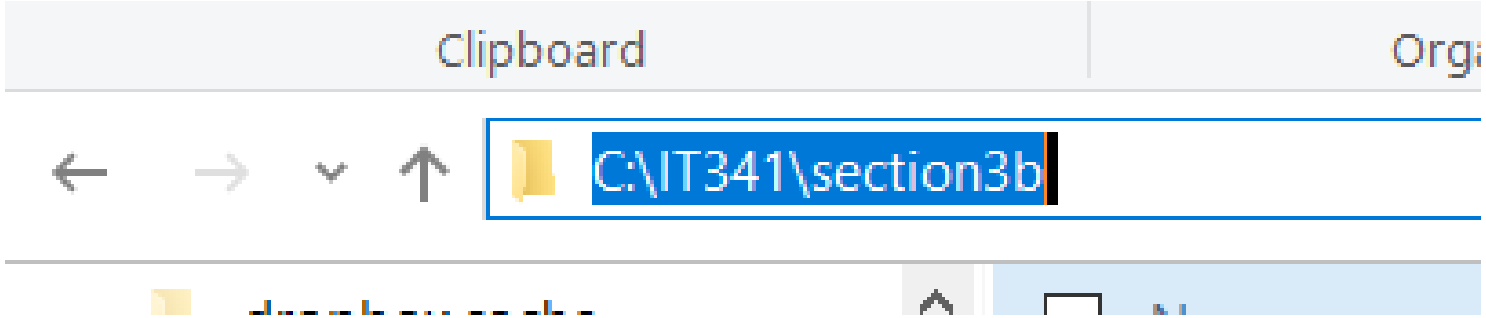
- Under *"Virtual machine name"*, put in the correct name for your VM and click ***OK***



- NOTE: This *only* changes your VM's name so far as *VMWare itself* is concerned. The hostname, as configured in the Ubuntu installation, is *another matter entirely!*

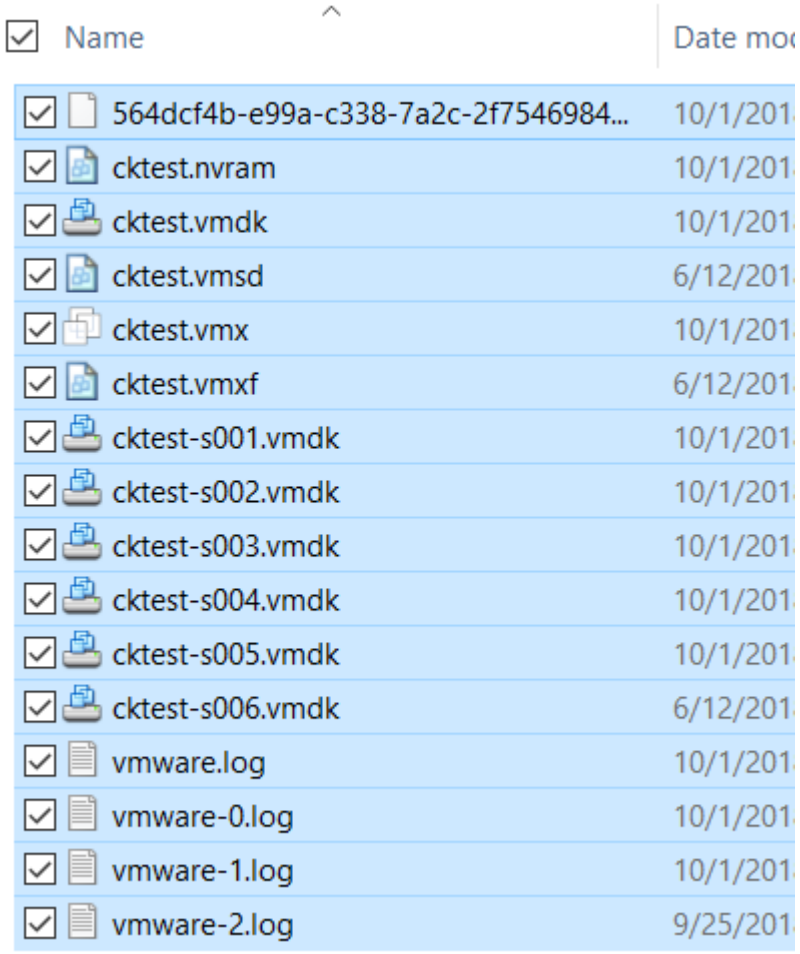
VMWare - Move/Copy your VM

- This assumes you have access to the directory where your VM files are located, for purposes of viewing and modifying
- Make sure your VM's users (**sysadmin**, etc.) are logged out, and the VM is properly and cleanly shutdown.
- Open up two windows in *File Explorer*
  - 1st window: Go to the folder where your VM files are located



VMWare - Move/Copy your VM

- 2nd window: Go to the *destination* folder, to where you wish to move or copy your VM files. This could be:
  - Another folder on the physical host
  - A folder on a USB flash drive
  - etc....
- In the ***first*** window, select *all* the files related to your VM, which may number more than you see here! Most of the files' names will include your VM name.



VMWare - Move/Copy your VM

- To copy:
  - Right click on the body of selected files in the *first* window
  - Choose ***Copy***
  - Change focus to the *second* window and right click
  - Choose ***Paste***
- To move:
  - Repeat the steps for copying, but choose ***Cut***, instead
  - OR: Click and drag the files *from the first* window *to the second* window.

VMWare - Move/Copy your VM

- You are probably better off to just copy the VM to the new location and make sure it works there. (Once you are sure it works, you can always clean up the files in the old location.)
- When you try to start your VM up again from the new location, you will be asked if you moved or copied it.
  - Choose the correct answer
  - If in doubt, just say you copied it!
- *NOTE: Check to see if your **MAC address** has changed, which affects your ability to get on the IT Lab LAN!*

VMWare - Clone your VM

- This may be a good option for you, if you installed your VM in a location where you cannot directly access your VM files.
- Make sure your VM is shut down
- With your VM in the active tab, go to ***VM*** -> ***Manage*** -> ***Clone*** and click ***Next***
- Choose to clone from the *"current state"* and click ***Next***
- Choose to *"Create a full clone"* and click ***Next***
- Choose an appropriate name and location, and ***Finish***

Change hostname on your system

- A few slides ago, we saw how you can change your VM's name within the context of *VMware* itself
  - However, you would also need to change your VM's **hostname**, in the context of *your OS installation*
  - This is how your VM identifies itself, internally and locally
- To see your VM's *current* hostname, enter the following command: `hostnamectl`
- Look for the line starting with `Static hostname:`
- You can also do this to confirm a hostname change was successful.

Change hostname on your system

- To *change* the hostname, use this command:

  `sudo hostnamectl set-hostname [newNameHere]`

  (*Without the square brackets!*)
- Edit to `/etc/hosts`: `sudo nano /etc/hosts`
  - On the line for IP `127.0.1.1`, change the previous hostname to the new one
  - Do this for both the *long* **and** *local* versions, if both are present
- ***Reboot*** your VM from the CLI (`sudo reboot`) or from VMware: Logout `sysadmin` and choose *"Restart Guest"*

Make a sysadmin account on your VM:

- After Project #1, every virtual machine should have an administrative user -- a `sudo`-er -- called `sysadmin`, whose password is the team name, per the Project #1 instructions.
- If this was not done correctly, you can correct it as follows:
- As the admin user, do the following:
  - Execute command: `sudo adduser sysadmin`

Make a sysadmin account on your VM:

- Define `sysadmin`'s password, as specified in Project #1
  - Type the password and press ***Enter***
  - Retype the password and press ***Enter***
- In response to prompts for user information, just leave *blank* and press ***Enter*** repeatedly.
- Confirm the process complete
- Next, execute this command: `sudo usermod -aG sudo sysadmin`
- Log out of your VM, and log back in as `sysadmin`

su - substitute user

- In future projects, you often need to switch between `sysadmin` and your personal account, created on `it20` in Project 2, Part II.
  - Instead of constantly logging out of one and into another, you might benefit from using the `su` command
  - This allows you to temporarily "become" another user (with that user's privileges) during a session.
- Enter this command: `su [the other username]`
- ...and enter that user's password, when prompted
- When finished type `exit` and press **Enter** . You will be back in your main session, under the original username.

su - substitute user

- Of course, only use the **su** command if you are not, in fact, that user already.
  - After all, if you are *already currently* logged in as **sysadmin** , then executing **su sysadmin** has no real benefit
  - However, if you are logged in as a normal user, but you need **sysadmin** for some quick administrative task, then **su sysadmin** would indeed make things easier and let you keep your main session.
- Naturally, you can only use the **su** command if you actually know that user's username and password

Troubleshooting VM Access via SSH

- First, your VM should have the `openssh-server` package installed, per Project #1 instructions. This is so you can have an *SSH daemon* running to take incoming log-ins
  - To check this,
    - Execute this command on your VM: `ps -ef | grep ssh`
    - One of the lines should feature an sshd process owned by `root`: `/usr/sbin/sshd -D`
    - Another check: `dpkg --list | grep openssh`
    - There should be a line for `openssh-server`

Troubleshooting VM Access via SSH

- If the `openssh-server` package is not already installed, then install as follows: `sudo apt-get install openssh-server`

- Accessing your VM from *outside* the IT Lab LAN will be a *twofold* process.
- First, you must log into `it20` itself.
  - This can be done using either the default account `it341` -- or your personal account on `it20`, created in Project 2, Part II.
  - Treat logging into `it20` itself as a ***separate*** issue from logging into your VM

Troubleshooting VM Access via SSH

- Second, from your `it20` session, you can then SSH into your VM:
  - Using `sysadmin`, to do administrative tasks on your VM
  - Using another valid log-in, when the situation calls for it.
- Example:

```
it341@it20:/$ ssh sysadmin@itvm28-4b
```

- You could also use your VM's *IP address*:

```
it341@it20:/$ ssh sysadmin@10.0.0.191
```

Troubleshooting VM Access via SSH

- If you fail to log into your VM from `it20`, then some troubleshooting will be required…
- Again, as mentioned, be sure you have the `openssh-server` package installed on your VM, in the first place!
- Assuming you do, you will attempt to diagnose and solve the issue from two sides:
  - From your `it20` session
  - From within your VM itself

Troubleshooting VM Access via SSH

- From `it20`, ping your virtual machine:
  - By its hostname: `ping itvm28-4b`
  - By its proper IP address: `ping 10.0.0.191`
  - NOTE: Its "proper IP address" is the one that we would expect to see assigned, based upon conventions defined in `it20`'s configuration.
- If your pings do not get responses, ensure your VM is
  - Booted up and running
  - Has proper *Bridged* configuration, and is bridged to the physical host's Intel NIC
  - Your MAC address is properly entered on `it20`, at the correct entry in `/etc/dhcp/dhcpd.conf`

Troubleshooting VM Access via SSH

- If pinging to your VM is successful, but you still cannot *SSH* in, make sure your VM has ***port 22*** open for incoming *SSH* connections.
  - Execute this command (from `it20`): `nmap itvm28-4b`
  - And/or this one: `nmap 10.0.0.191`
  - Replace VM name/IP address with your own, of course!
  - Under `PORT  STATE SERVICE`, you should see this line: `22/tcp open  ssh`

Troubleshooting VM Access via SSH

- If *port 22* is not open, you will need to execute the following command, as **sysadmin**, on your VM:

  ```
  sudo ufw disable
  ```

  - After doing so, try running the **nmap** command upon your VM's hostname once again, from **it20**
  - If *port 22* is not shown yet, reboot your VM and run **nmap** upon your VM's hostname, from **it20**, again
- When attempting to SSH into your VM, you may find it helpful to run the **ssh** command with the **-v** option, for verbose output.

Troubleshooting VM Access via SSH

- Example: `it341@it20:/$ ssh -v sysadmin@itvm28-4b`
- The verbose output can provide additional information, which might help you to troubleshoot a failed SSH connection attempt

- One thing you may need to do on your VM is to *regenerate its SSH host keys*
  - To view the files for the SSH host keys, execute:

```
ls -l /etc/ssh/ssh_host_*
```

  - If these files are size ***zero***, then that would be one reason to regenerate

Troubleshooting VM Access via SSH

- To regenerate the SSH host keys on your VM, have `sysadmin` execute the following on your VM:

```
sudo rm /etc/ssh/ssh_host_*
sudo dpkg-reconfigure openssh-server
```

- Run `ls -l /etc/ssh/ssh_host_*` again to confirm the files are now of ***non-zero*** size
- You probably will not need to do this, but it is good to know how, if needed.

Other Teams' VMs

- Sometimes, you will need to find another team's VM that is currently running and on the network.
- One way is to simply ask your fellow student, but that only works if you and they are both present.
- Another way involves the `nmap` command...
- From `it20`, execute this command (will take a moment to complete:

```
nmap 10.0.0.128/26 | less
```

- It will show you which machines from `10.0.0.128-191` are on the LAN, as well as what services listening on which ports.

Preserving CLI Sessions

- If you invoke a text editor on a protected system file without `sudo`, you may get finished editing the file -- only to *attempt* saving...but receive a *permission error*!
  - If it was only a little bit of typing, you might as well exit the editor without saving and just do the edits again -- this time, remembering to include `sudo`
  - If you did a lot of work, you can try saving a temporary copy of the file in the user's home directory.
    - This will at least preserve the work you did.

Preserving CLI Sessions

- Then, using `sudo cp`, you can copy that temporary file into the path of the system file you intended to edit.
- But, don't do this unless you **mean** to and **know** what you're doing!

- ***There is no way to directly copy text from your VM to the clipboard***, but you will probably still want some way of preserving your command line interaction…
  - For your personal records
  - To incorporate into your admin log for your lab reports

Preserving CLI Sessions

- This can be done IF you set things up from the moment you start working. We can do this with the `script` command.
  - If you recall, executing `script` opens up a sub-session within your main shell session. You can work at the command line as you normally would.
  - When you exit from that sub-session, you are dropped back in your *main* session, in your working directory at the time you executed `script`
  - Use `script` with the `--flush` option so work is saved as you go

Preserving CLI Sessions

- In that directory, there will be a file called `typescript`, containing your command line session.

- Normally, executing `script` again -- inside a directory with a pre-existing `typescript` file -- will overwrite the previous `typescript` file!
- However, you can execute `script` with an *argument* to specify the path and name of the output file.
- So, to save your CLI session, do these things...
  - Log into your VM as you normally would.

Preserving CLI Sessions

- Do this only once per account: `mkdir $HOME/sessions`
  - You would do this for `sysadmin`, certainly
  - You and your partner may also wish to do this on your *individual* user accounts (which you will create as part of Project 2, Part II)
- Execute this command:

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

  - What this will do is save your command line session in the user's sessions directory, with a unique filename, based on a timestamp
  - *Make a note* of that filename, for subsequent upload, later on down the line.

Preserving CLI Sessions

- Work at the command line, as you normally would -- with these caveats!
  - ***Do not do any file editing during this sub-session!***
    - Either end your sub-session, edit the file, and start a new sub-session
    - Or open a separate SSH connection into your VM, specifically for file editing
  - Do not use commands like `more`, `less`, `clear`, etc. in your sub-session.
  - I make these caveats because the aforementioned things can *drastically* affect the readability of the text file from your sub-session!
- When finished, end the sub-session by typing `exit` and pressing ***Enter***

Preserving CLI Sessions

- Work at the command line, as you normally would -- with these caveats!
  - ***Do not do any file editing during this sub-session!***
    - Either end your sub-session, edit the file, and start a new sub-session
    - Or open a separate SSH connection into your VM, specifically for file editing
  - Do not use commands like `more`, `less`, `clear`, etc. in your sub-session.
  - I make these caveats because the aforementioned things can *drastically* affect the readability of the text file from your sub-session!
- When finished, end the sub-session by typing `exit` and pressing ***Enter***

Preserving CLI Sessions

- You should now have a file in your `sessions` directory of the form `YYYYMMDD_HHMMSS.txt` -- such as this example: `20180926_162950.txt`
- To make these files more readily accessible to yourself, you can copy them to your Linux account.
  - Log into `users.cs.umb.edu` with your Linux account to create an `it341/sessions` directory

```
cd $HOME
cd it341
mkdir sessions
```

(Do this only once!)

Preserving CLI Sessions

- From your virtual machine
  - (Replace `cs110ck` with your own Linux username)

```
rsync -avzz $HOME/sessions/ cs110ck@users.cs.umb.edu:/home/cs110ck/it341/sessions/
```

  - Do this from `sysadmin`'s command line.
  - If you have any sessions from your personal user account, repeat the step from that account's command line
- Back on `users.cs.umb.edu`
  - Check your `it341/sessions` directory
  - Be sure that you have all the files you wanted
  - Be sure that you have not accidentally deleted anything
  - Make backups of your session text files to your own personal computer or data storage!

Copying and Pasting

- Sometimes, when using PuTTY, you will want to copy text from your terminal, or paste into it.
- However, the standard procedures for cutting, copying, and pasting…
  - Windows keyboard shortcuts (`Ctrl-X`, `Ctrl-C`, `Ctrl-V`, etc.)
  - Right-clicking (to cut/copy highlighted text or to paste)
- …will not work as expected in PuTTY, and if you try to use those, you will get unexpected behavior!

Copying and Pasting

- Instead, follow these guidelines:
  - To *copy text from* the PuTTY window: Highlight the text.
    - That's it. In PuTTY, highlighting the text automatically copies it to the clipboard.
    - To verify this, highlight some text, open a text editor, and then paste using normal shortcuts or mouse clicks
  - To *paste text into* the PuTTY window:
    - Make sure the desired text is on your clipboard (from a copy or cut operation)

Copying and Pasting

- Make sure your cursor (in the PuTTY window) is in the desired position, where you wish to paste the text
- Right-click in the PuTTY window.
  - Again, that's it.
  - The right-click itself performs the paste operation in PuTTY
- Sometimes, you will want to paste text into VMWare
  - At the command line
  - In your text editor

Copying and Pasting

- To do this:
  - Make sure the desired text is on your clipboard
  - In fact, you may want to paste from your clipboard into a text editor
  - This is just to confirm that your clipboard contents look like you expect it to
  - Go into your VM, and make sure your cursor is in the desired position for pasting the text

Copying and Pasting

- Go to VMWare's menu and choose ***Edit*** -> ***Paste***
  - If done correctly, you should see the text start to appear where your cursor is.
  - This may look as if someone's typing it really quickly!
- Some caveats:
  - There is a *limit* to how much text you can paste into your VM *at once*
  - When pasting in multiple lines, you may see extra lines inserted in between the actual lines.
  - Be careful when pasting into your VM's command-line. If your pasted text includes new-line or carriage-return characters, then that may be interpreted as a press of the ***Enter*** key

Creating Files to Hand In

- For starters, learning how to easily and efficiently manage remote files on an external server should be a priority for you, as an information technology student
  - Managing remote files and folders
  - Moving files and folders between local and remote
- To that end, you should figure out what methods and workflow enable you to do *your best work*
  - Command-line interface or a more traditional graphical app (an FTP client)?
  - Create files locally and upload, or work on the remote and then download?

Creating Files to Hand In

- Check out this lecture that I give my IT110 students:

  `https://www.cs.umb.edu/~ckelly/teaching/common/lecture/probsolv/02-UploadingRemote.pdf`

  - The material is *targeted* towards that class, but many *principles* still apply.
- Most of the materials you hand in will be plain text files
  - Some (such as chapter summaries) will not have too many specifications
  - Others -- most notably lab reports -- have much stricter specs regarding the file text:
    - Line endings (Unix-style, `\n` instead of `\r\n`)
    - Character encoding (`ASCII` or `ISO-8859-1`/`Latin-1`)
    - Line lengths (Generally limit to 80 characters long)

Creating Files to Hand In

- IMPORTANT: Document editors like *Microsoft Word* , *Google Documents* , etc. are a poor choice for this task!
  - Such tools are for creating documents with advanced stylistic features
  - Again, use *the correct tool for the task* -- in this case, software geared towards plain text
- An advanced text editor will be a huge help in this! Examples: *Sublime Text 3* , *Notepad++*
  - Such an editor will usually have options like:
    - Convert line endings to Windows or Unix
    - Indent with spaces versus tabs (and convert between the two)
    - Save file with a specific character encoding
    - Place a marker on the screen to indicate a particular line width

Creating Files to Hand In

- Using a sophisticated text editor on your desktop is probably easiest, but much can be done via the command line, as well...
- Get information about a file -- encoding, line-endings, etc.

```
file [the file's name]
```

- **CLI:** Converting a text file's encoding to `ISO-8859-1` (i.e., `Latin1`):
  - Making a temp file with new encoding

    ```
    iconv -c -t ISO-8859-1//TRANSLIT [orig. filename] > [temp filename]
    ```
  - Confirm temp file looks okay. (Use a pager instead of cat for longer files!)

    ```
    cat [temp file name]
    ```
  - Remove original file

    ```
    rm [original file name]
    ```
  - Rename the temp file

    ```
    mv [temp file name] [original file name]
    ```

Creating Files to Hand In

- **CLI:** Convert Windows-style line endings to Unix-style

  ```
  dos2unix [file name]
  ```

  - See also: `unix2dos`, `unix2mac`, `mac2unix`

- **CLI:** "Fold" lines to a specific length

  ```
  fold -w [length] -s [original file name] > [temp file name]
  ```

  ```
  cat [temp file name]
  ```

  ```
  rm [original file name]
  ```

  ```
  mv [temp file name] [original file name]
  ```

  - Caveat: The results will surely be messy and require further editing to neaten formatting and maintain indentation!