EE 332 CAD Lab 0 Autumn 2022

Setting up Linux for a remote connection & Basic Linux Command Lines

Turn in a report for each two labs. Please submit electronically through CANVAS.

Original authors: John Uehlin and Chris Rudell

Last modified: Cindy Liu

Last Updated: October 2022

1. Introduction and Linux Remote Connection Setup

Computer Aided Design (CAD) software and many of the advanced engineering software packages run in an environment with Unix/Linux as the operating system. An example is the software that will be used for this class to analyze complicated circuits. Cadence Virtuoso® is the industry standard for integrated circuit design and includes all the features necessary to design a complete system-on-chip (SoC) from schematic capture, to layout and fabrication. We will go through the steps to make a remote connection to the servers in order to use thus software. ***For Fall 2022:*** lab sections will be hybrid, both in-person and online over Zoom. In case you need physical access to workstations for these servers use the computer room ***ECE 351***. These workstations are actual machines, so you can skip the steps regarding remote connection. The ECE 351 machines do not support Zoom, so please use your laptop for online lab sections.

Connecting to a Server

In order to log into the ECE Linux machines, you need to establish a Secure Shell (SSH) connection. You can use your preferred SSH client for remote access. While you can directly use the "ssh" command in Mac OS terminal or use **PuTTY** in Windows, these approaches do not make a remote desktop for you. Meaning that your desktop session will be restarted each time you reconnect to the serves. In order to save the remote desktop session, you can use software such as **X2Go** and **TightVNC** (both available for Mac and windows)

While this tutorial covers multiple softwares, we recommend installing the **X2Go** (for both Mac and windows computers). If for any reason you could not get the **X2Go** to work on your laptop, then use the alternative which will be **TightVNC**.

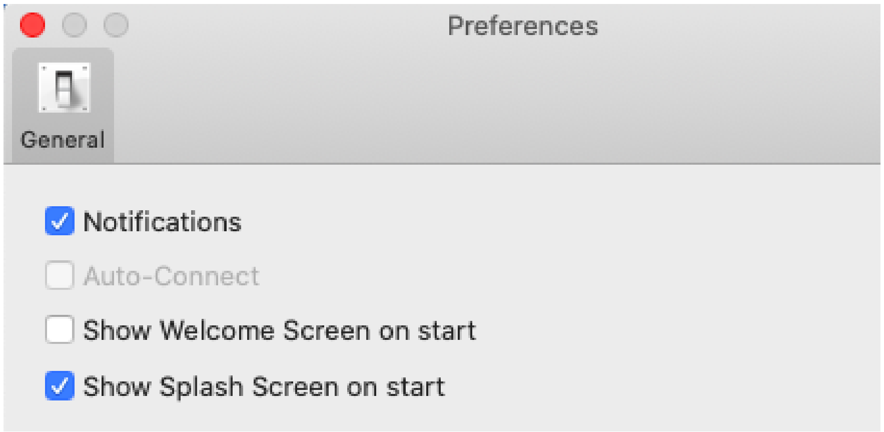
* Once you have made the successful remote desktop in this Section and can use the server machine, go to section 2 of Lab 0 ("Basic Linux Command Lines").

For instructional IT help and account issues email: help@ece.uw.edu

Connecting to the VPN (Required to access the servers!)

Husky OnNet is the secure campus VPN system required for accessing servers for this class. Please follow the instructions from: https://itconnect.uw.edu/connect/uw-networks/about-husky-onnet/use-husky-onnet/ to install the VPN software and log-in with your UW netID. Note that this website has specific instructions for Mac vs Windows, so make sure to follow the right one.

If you have multiple user accounts on your computer, you may have to turn off the VPN on all other accounts before starting it on the account you are using. If you are having trouble getting Husky OnNet to prompt you to login with your username and password, you can go to preferences and ask to show "welcome screen" or "splash screen" on start.



Preferences

General

☑ Notifications

□ Auto-Connect

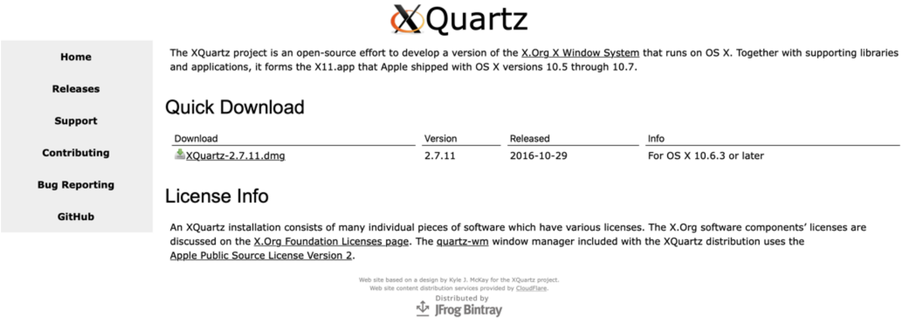
□ Show Welcome Screen on start

☑ Show Splash Screen on start

Connecting via X2Go

We strongly recommend both Windows and Mac users to use **X2Go** to connect to the servers.

On the Mac, you first need to make sure you have the latest version of **XQuartz** installed. Please go to https://www.xquartz.org and download the latest version. Install it using the dmg file on your computer.



XQuartz

Home

Releases

Support

Contributing

Bug Reporting

GitHub

The XQuartz project is an open-source effort to develop a version of the X.Org X Window System that runs on OS X. Together with supporting libraries and applications, it forms the X11.app that Apple shipped with OS X versions 10.5 through 10.7.

Quick Download

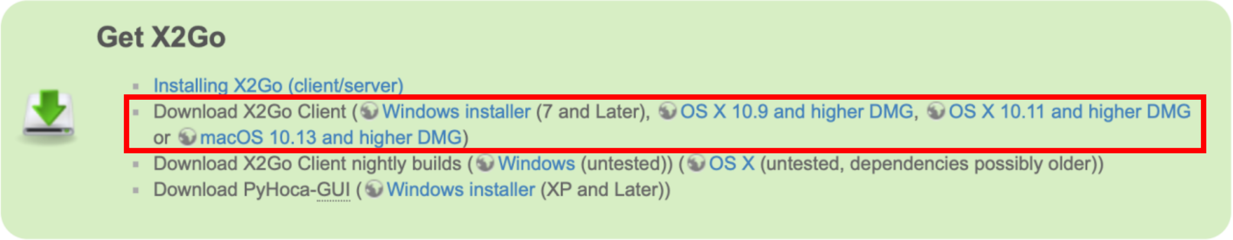
Download	Version	Released	Info
XQuartz-2.7.11.dmg	2.7.11	2016-10-29	For OS X 10.6.3 or later

License Info

An XQuartz installation consists of many individual pieces of software which have various licenses. The X.Org software components' licenses are discussed on the X.Org Foundation Licenses page. The quartz-wm window manager included with the XQuartz distribution uses the Apple Public Source License Version 2.

Web site based on a design by Kyle J. McKay for the XQuartz project.
Web site content distribution services provided by CloudFlare.
Distributed by
JFrog Bintray

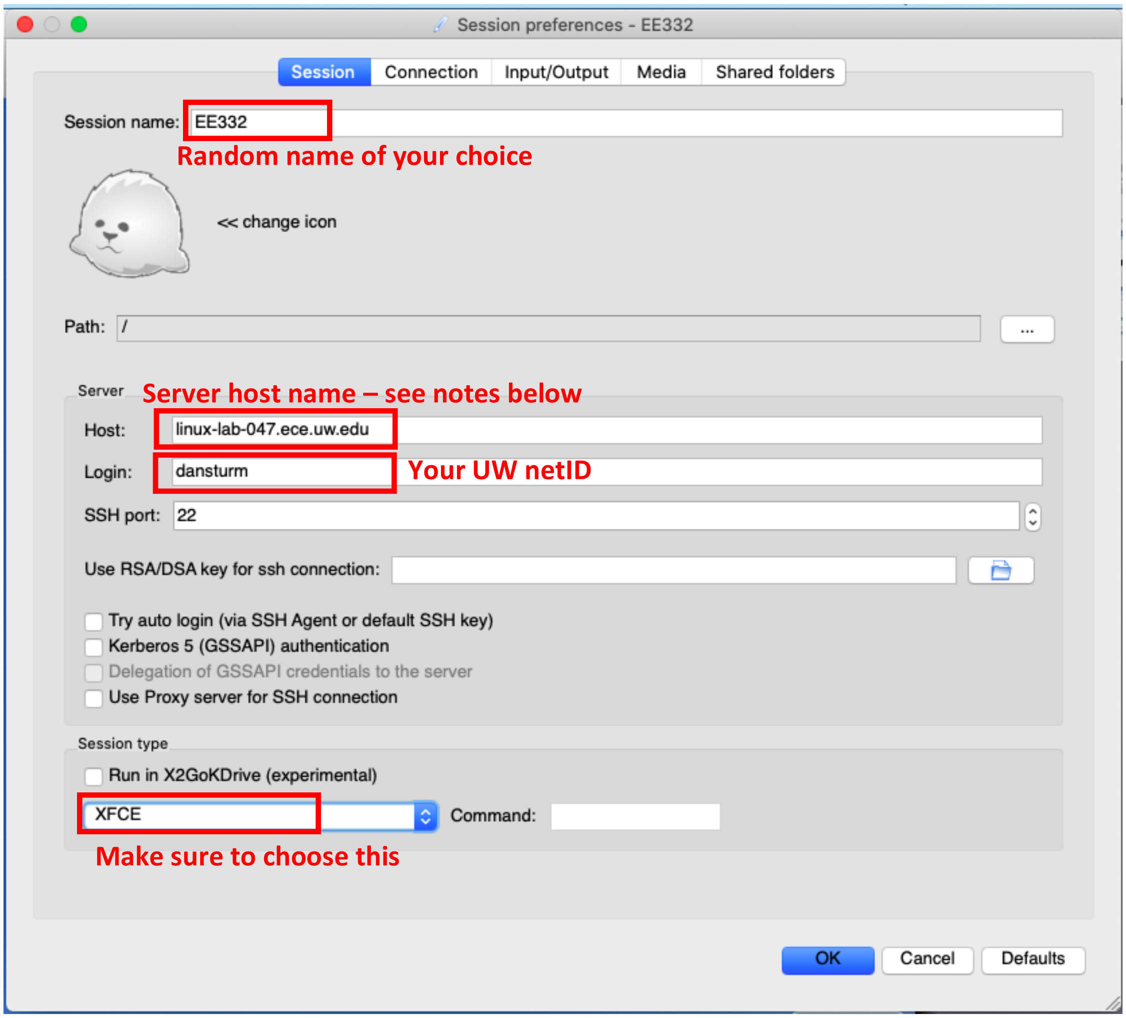
Next, you need to install the latest version of **X2Go** from https://wiki.x2go.org/doku.php



Get X2Go

- Installing X2Go (client/server)
- Download X2Go Client (Windows installer (7 and Later), OS X 10.9 and higher DMG, OS X 10.11 and higher DMG or macOS 10.13 and higher DMG)
- Download X2Go Client nightly builds (Windows (untested)) (OS X (untested, dependencies possibly older))
- Download PyHoca-GUI (Windows installer (XP and Later))

After opening the software, click on the icon on top left side (the one that looks like a sheet of paper with the corner folded) to make a new connection with configurations shown below:



Session name: EE332

Random name of your choice

Path: /

Server

Server host name – see notes below

Host: linux-lab-047.ece.uw.edu

Login: dansturm

Your UW netID

SSH port: 22

Use RSA/DSA key for ssh connection:

□ Try auto login (via SSH Agent or default SSH key)
□ Kerberos 5 (GSSAPI) authentication
□ Delegation of GSSAPI credentials to the server
□ Use Proxy server for SSH connection

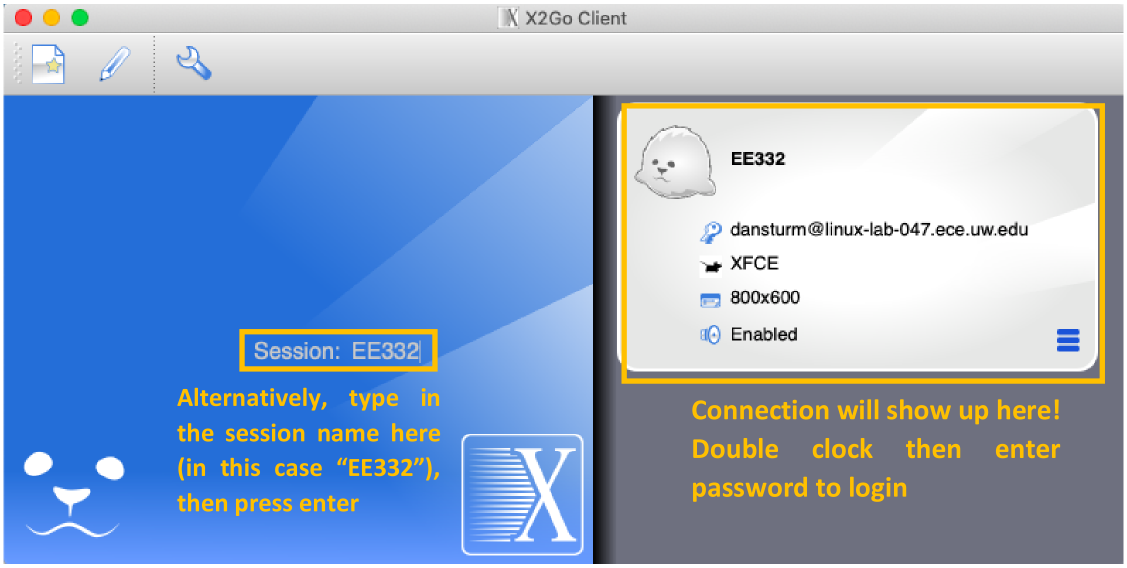
Session type

□ Run in X2GoKDrive (experimental)

XFCE Command:

Make sure to choose this

After clicking OK the connection is made and shown on the right-side panel. Simply double click on it and enter your password to log in. Sometimes this does not work, in which case you can enter the session name in the "session" section and press enter.



Session: EE332

Alternatively, type in the session name here (in this case "EE332"), then press enter

EE332
dansturm@linux-lab-047.ece.uw.edu
XFCE
800x600
Enabled

Connection will show up here! Double clock then enter password to login

Valid academic Host Names are as follows: **linux-lab-###.ece.uw.edu (### = 042-052)**

* In order to avoid overloading any of these servers, we have assigned each student a server to use. Check the server assignment file on Canvas to find the server number assigned to you (go to “files” section then “lab” folder). You may note from other classes that servers 041 – 080 are available; for this class please stick to 042 - 052.

If you connect successfully to the Linux servers, you can skip ahead to section 2 of the tutorial! If you are still struggling with getting X2Go to work on your computer, follow the rest of instructions below to use VNC:

Connect with VNC

There are three steps to connecting with VNC:

1. SSH into the host machine to start a Linux terminal (we do this using “PuTTY” on Windows or “Terminal” on Mac)
2. Start a VNC session in this host machine terminal
3. Use TightVNC to access this VNC session

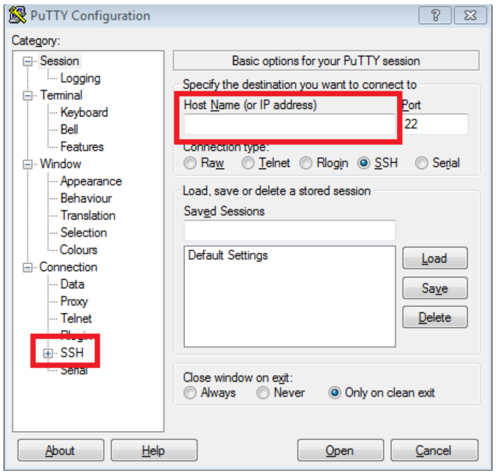
SSH via PuTTY (for Windows)

PuTTY comes pre-installed on Windows machines in ECE, you can also download it from the following site if you don’t find it in your PC:

https://www.chiark.greenend.org.uk/~sgtatham/putty/latest.html

(More software resources for UW ECE, including PuTTY, can be found here: https://vannevar.ece.uw.edu/computing/EESoftwareCatalog.html)

Open PuTTY and you should see the below window. Insert your netID and desired hostname (from the valid host names listed above) into the SSH tab of PuTTY, using the format ***netID@hostname*** (such as “dansturm@linux-lab-042.ece.uw.edu”)

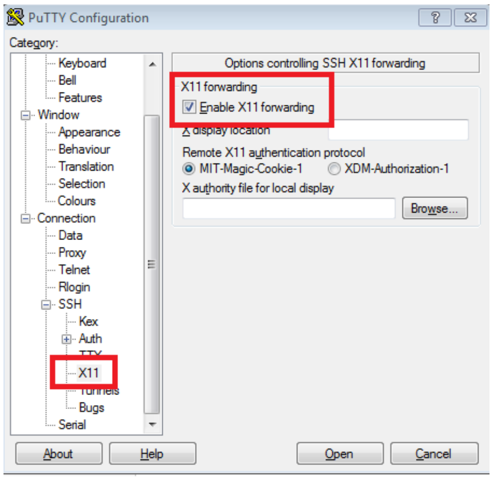


Port: 22

Connection type: ● SSH

Close window on exit: ● Only on clean exit

Make sure that X11 forwarding is enabled, as shown below. This allows the SSH host to forward active GUI windows (similar to the Windows OS) to your client desktop. If you're running SSH from a Linux machine, make sure you use to the *SSH –X11* option to enable X11 forwarding.



Options controlling SSH X11 forwarding

X11 forwarding

☑ Enable X11 forwarding

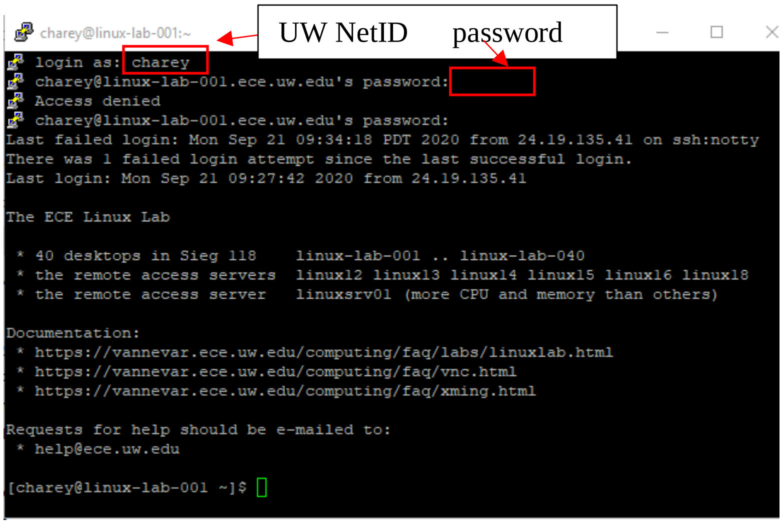
X display location

Remote X11 authentication protocol

● MIT-Magic-Cookie-1 ○ XDM-Authorization-1

X authority file for local display

Hit "Open" to start the SSH session. This will open a Linux terminal on the host machine. You will need to login using your UW NetID and password.



```
login as: charey
charey@linux-lab-001.ece.uw.edu's password:
Access denied
charey@linux-lab-001.ece.uw.edu's password:
Last failed login: Mon Sep 21 09:34:18 PDT 2020 from 24.19.135.41 on ssh:notty
There was 1 failed login attempt since the last successful login.
Last login: Mon Sep 21 09:27:42 2020 from 24.19.135.41

The ECE Linux Lab

 * 40 desktops in Sieg 118    linux-lab-001 .. linux-lab-040
 * the remote access servers  linux12 linux13 linux14 linux15 linux16 linux18
 * the remote access server   linuxsrv01 (more CPU and memory than others)

Documentation:
 * https://vannevar.ece.uw.edu/computing/faq/labs/linuxlab.html
 * https://vannevar.ece.uw.edu/computing/faq/vnc.html
 * https://vannevar.ece.uw.edu/computing/faq/xming.html

Requests for help should be e-mailed to:
 * help@ece.uw.edu

[charey@linux-lab-001 ~]$ 
```

In the screenshot above, we are using linux-lab-001 server as an example. However, note that this server is currently not available – the available servers are 042 – 052 (see X2Go section).

SSH with Terminal (Mac)

If you are on a Mac, open terminal. To SSH into a Linux machine, type "*ssh <NetID>@hostname*", where hostname is the hostname for the server you've been assigned (see in X2Go section). For example, I might type in "*ssh dansturm@linux-lab-042.ece.uw.edu*". You will be prompted to enter your password.

Configuring VNC

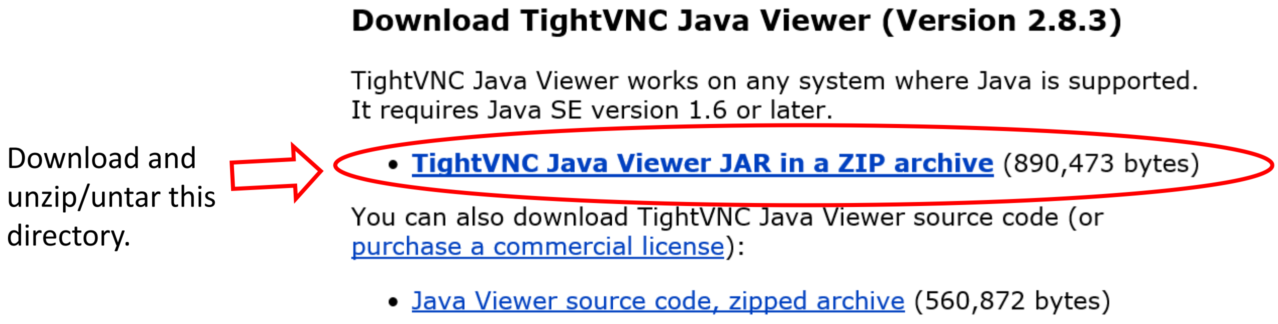
Running TightVNC will require Java, so first download the latest Java from the following website:

https://www.oracle.com/java/technologies/downloads/

Note that if you have a Mac, the correct Java to download will depend on which processor your computer has (Apple recently switched from Intel processors to in-house Apple M1 processors). If your Mac has an M1 chip, you want to download the "Arm 64 DMG Installer"; if your Mac has an Intel chip, you want to download the "x64 DMG Installer".

With Java installed on your computer, you can now run the "TightVNC" installation package. Once at the TightVNC website, scroll down until you see the following "Java Viewer" version of TightVNC. We recommend that you copy the install program to your computer, then make a copy of TightVNC to your desktop. The TightVNC download can be found here:

https://www.tightvnc.com/download.php



Download TightVNC Java Viewer (Version 2.8.3)

TightVNC Java Viewer works on any system where Java is supported. It requires Java SE version 1.6 or later.

- **TightVNC Java Viewer JAR in a ZIP archive** (890,473 bytes)

You can also download TightVNC Java Viewer source code (or purchase a commercial license):

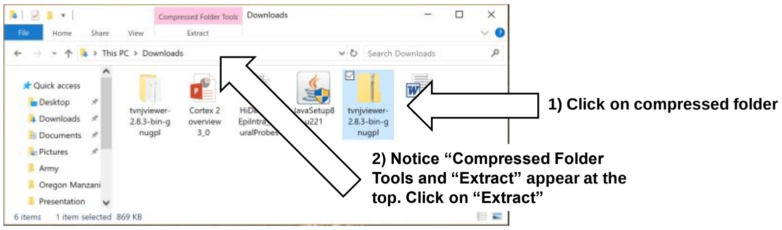
- Java Viewer source code, zipped archive (560,872 bytes)

Download and unzip/untar this directory.

Note – links to downloading Java and TightVNC can also be found on the UW ECE website:

https://vannevar.ece.uw.edu/computing/faq/vnc.html

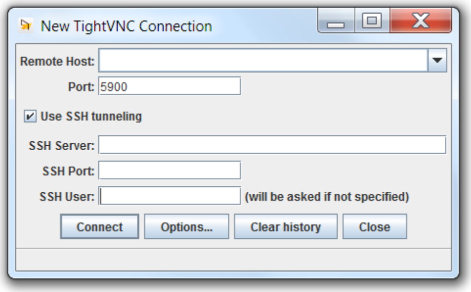
Note: Once you download this file it will be in a compressed format. You need to "unzip" or "untar" this file using software that may already be available on your laptop. If not, you will need to download the most appropriate software to untar/unzip this directory. For the latest versions of Windows, you can go to the folder that has the compressed file you just downloaded. You can extract in two steps, 1) click on the folder to uncompress and 2) go to the tool bar at the top of the folder and select "Extract", this is shown below.



1) Click on compressed folder

2) Notice "Compressed Folder Tools and "Extract" appear at the top. Click on "Extract"

Run the executable .jar file in the main folder, tightvnc-jviewer.jar. Note: definitely add a shortcut for tightvnc to your desktop, this will save time later when you return. The following window will appear:



New TightVNC Connection

Remote Host:

Port: 5900

☑ Use SSH tunneling

SSH Server:

SSH Port:

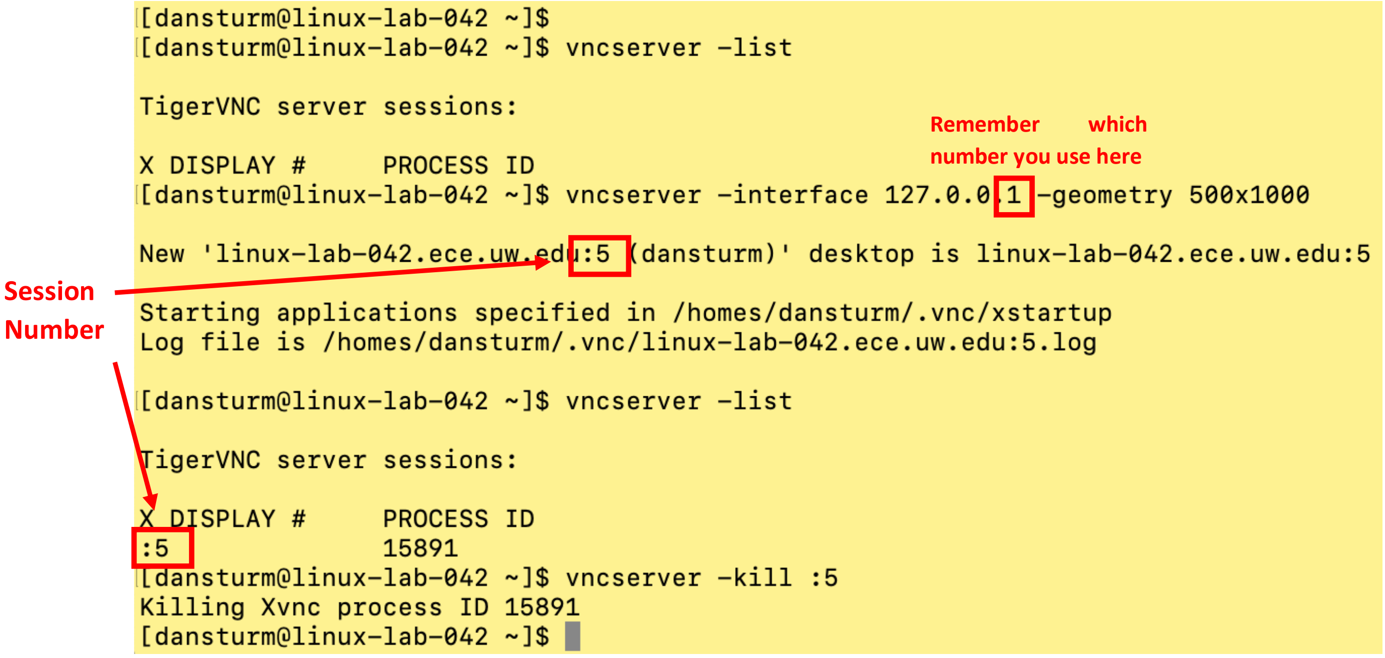
SSH User: (will be asked if not specified)

Starting a VNC Session on the Linux Terminal and Viewing it in TightVNC

Now we are ready to start the VNC server on whichever Linux machine you choose to use. This tutorial is using linux-lab-042, but this process will work on any of the servers or physical machines mentioned in the X2Go section. You will need to open a terminal on the Linux machine of your choice (using PuTTY on Windows or Terminal on Mac). In short, we use the SSH terminal simply to run the "vncserver" commands, to tell the Linux machines that you are going to open a new TightVNC session. Once we have established a new VNC session, you can now launch TightVNC from your laptop to access the session that we created. To launch a new VNC session, you will need to become familiar with some simple commands to run on the Linux server.

- To start your session, use: ***"vncserver -interface 127.0.0.X".*** In my experience, "X" can be any number from 0-255 (inclusive), but traditionally, we have used "1", so use that. You can add ***"-geometry WxH",*** where "W" and "H" are the desired height and width of the window (in pixels)
- To view open sessions, use: **"vncserver -list"**
- To kill (close) open sessions, use ***"vncserver -kill :#"*** (# is the number of the session to be killed). Note: there is a space between "-kill" and ":#"

The example terminal below shows some of the commands you will need to start and kill a VNC server session. Take note of how to find the VNC server session number.



```
[dansturm@linux-lab-042 ~]$
[dansturm@linux-lab-042 ~]$ vncserver -list

TigerVNC server sessions:

X DISPLAY #     PROCESS ID
[dansturm@linux-lab-042 ~]$ vncserver -interface 127.0.0.1 -geometry 500x1000

New 'linux-lab-042.ece.uw.edu:5 (dansturm)' desktop is linux-lab-042.ece.uw.edu:5

Starting applications specified in /homes/dansturm/.vnc/xstartup
Log file is /homes/dansturm/.vnc/linux-lab-042.ece.uw.edu:5.log

[dansturm@linux-lab-042 ~]$ vncserver -list

TigerVNC server sessions:

X DISPLAY #     PROCESS ID
:5              15891
[dansturm@linux-lab-042 ~]$ vncserver -kill :5
Killing Xvnc process ID 15891
[dansturm@linux-lab-042 ~]$ 
```

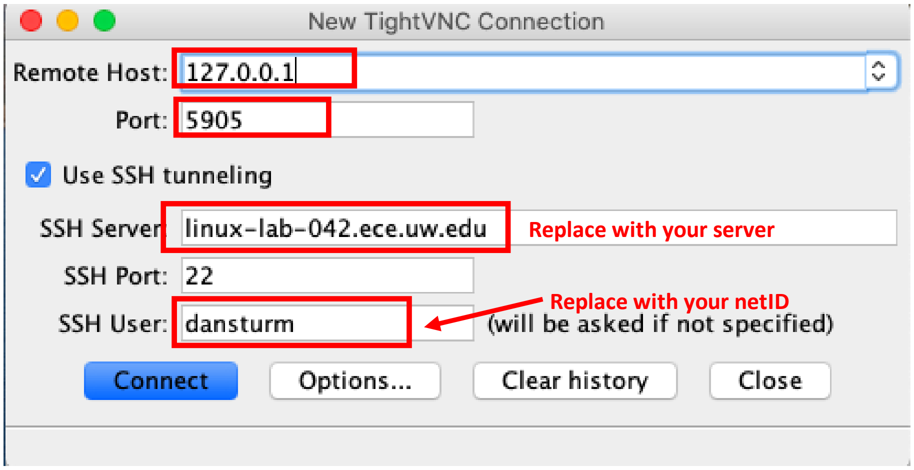
Remember which number you use here

Session Number

“vncserver” is the program name, and various dash options manipulate the program “vncserver” command to add some options to manipulate the graphics. The –interface and –geometry options specify trusted interface addresses and window size (in pixels), respectively. For maximum security, always use the –interface option. **The first time you use this command, you may be asked to create an additional password, which we will call the “host password”.** Make sure to remember this password.

After you use vncserver -interface (without killing them of course), return to your personal computer to view the VNC session opened on the Linux server. Fill out the TightVNC connection window as below.

The “Remote Host” should be the number of form 127.0.0.X you used when creating the connection in terminal. In this case, it’s 127.0.0.1 for me. Similarly, the Port field should contain “59Y”, where again, “Y” is the VNCserver session number, in this case “05”. You will be asked to fill in the password of your VNC account (the password of your UW NetID) and the host password (the password you set when you first used vncserver command). In case you don’t remember the host password, you could use this command to reset it: vncpasswd ~/.vnc/password.



Remote Host: 127.0.0.1

Port: 5905

☑ Use SSH tunneling

SSH Server: linux-lab-042.ece.uw.edu

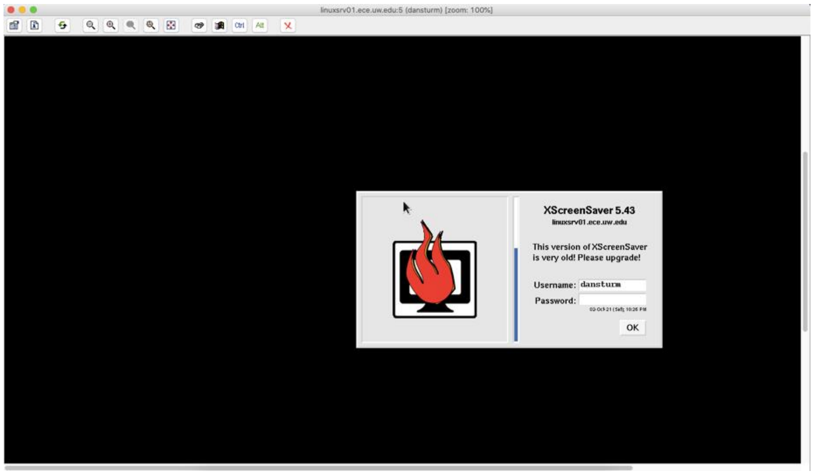
Replace with your server

SSH Port: 22

SSH User: dansturm (will be asked if not specified)

Replace with your netID

This will open a virtual desktop on the Linux machine. If the screen is blank when you first open it, scroll down and look for a popup saying “this version of XScreenSaver is very old”, and then enter your UW netID and password (see picture below). This desktop operates just as if you physically opened a Linux desktop. When you close the VNC viewer window, everything done on the desktop will remain open until the session is killed on a terminal (as described above) or the Linux computer restarts.



XScreenSaver 5.43
linuxsrv01.ece.uw.edu

This version of XScreenSaver is very old! Please upgrade!

Username: dansturm
Password:
02-Oct-21 (Sat); 10:26 PM

OK

2. Basic Linux command lines

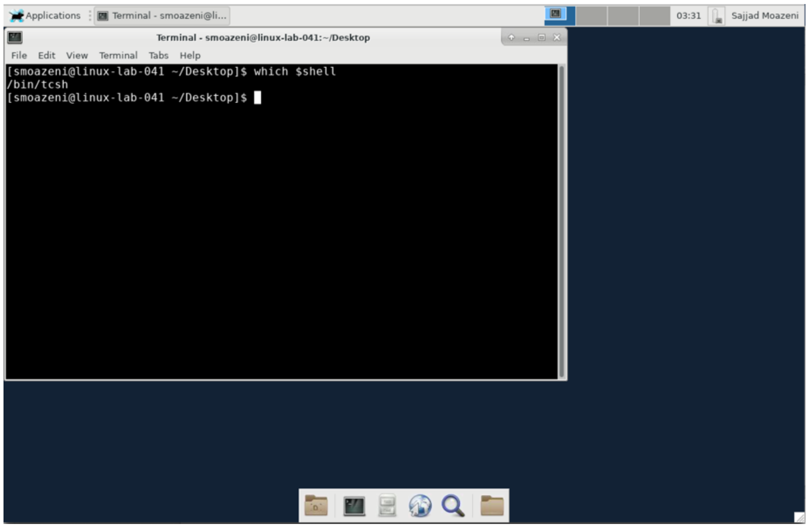
Brief Introduction to Linux

Linux is an open-source operating system kernel. It's widely used in Internet servers thanks to its advantages such as fast, secure and free compared with Windows servers.

A shell is a program that receives commands and send it to Linux to process. Linux OS has a Graphic User Interface (GUI) or Command Line Interface (CLI).

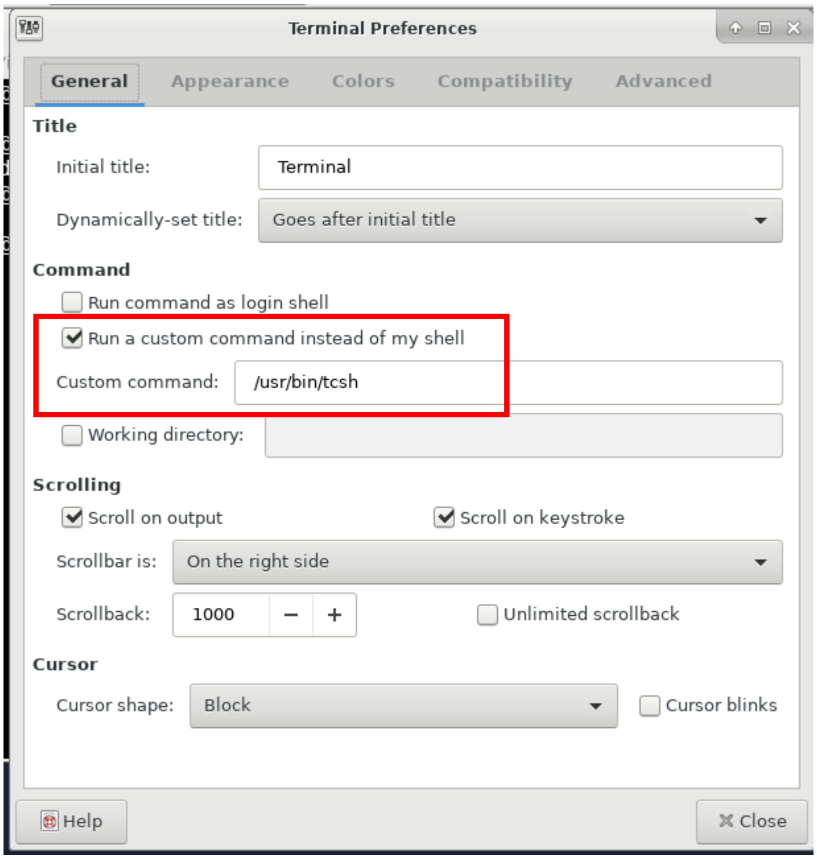
Here is a list of most useful commands you may need in this course. Please open the shell terminal and explore the Linux command line! The best way to get to know these commands is to experiment with them!

Linux Terminal Shell


Applications
Terminal - smoazeni@li...
03:31
Sajjad Moazeni
Terminal - smoazeni@linux-lab-041:~/Desktop
File
Edit
View
Terminal
Tabs
Help

```
[smoazeni@linux-lab-041 ~/Desktop]$ which $shell
/bin/tcsh
[smoazeni@linux-lab-041 ~/Desktop]$ 
```

Once you connect open the terminal (either right click and select terminal or click on the icon from the dock) and type: *“which $shell”*. This will give you whether your shell command line is “bash” or “tcsh”. The scripts that have to be loaded (in Lab1 we will explain them) to open Virtuoso are all tcsh-based. You can permanently set your command line to be tcsh by going to *edit* → *preferences* → *general*, and then clicking *“Run a custom command intead of shell”*, and entering *“/usr/bin/tcsh”* as the custom command (see photo below). You can also switch the shell to tcsh (just a one-time fix) by using the following command: *“/bin/tcsh”* or *“exec tcsh -I”*.



Title

Initial title: Terminal

Dynamically-set title: Goes after initial title

Command

☐ Run command as login shell

☑ Run a custom command instead of my shell

Custom command: /usr/bin/tcsh

☐ Working directory:

Scrolling

☑ Scroll on output ☑ Scroll on keystroke

Scrollbar is: On the right side

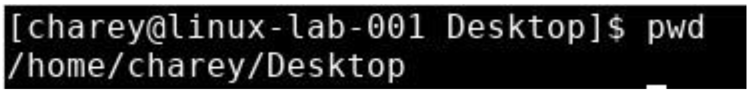
Scrollback: 1000 ☐ Unlimited scrollback

Cursor

Cursor shape: Block ☐ Cursor blinks

Basic Linux commands

1. pwd - print working/current directory



```
[charey@linux-lab-001 Desktop]$ pwd
/home/charey/Desktop
```

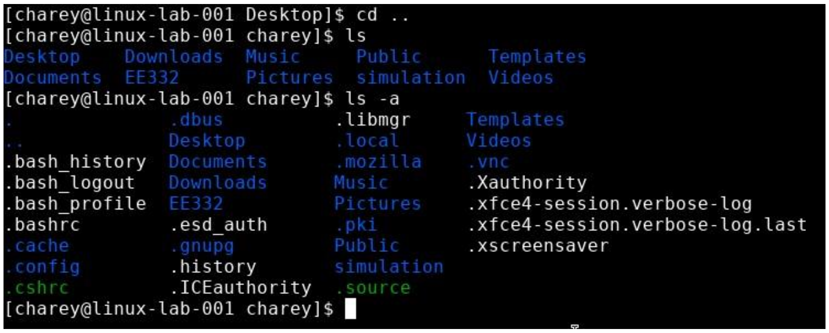
2. cd - change directory. Use it in the form of **cd filedirectory**. If you do not know the files in the current directory, you could use **ls** command as described below. You can use **Tab** key to auto-complete the next directory name.

Note: directory tips

- . means current directory
- ~ means home directory
- .. means directory before

Example: cd .. means go to the directory before

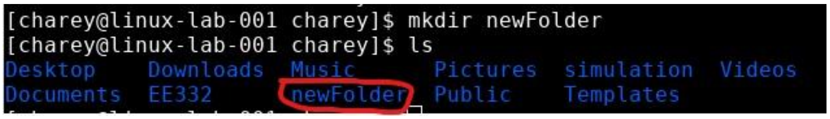
3. ls - list all file names in the current directory. Hidden files could be seen using **ls -a**



```
[charey@linux-lab-001 Desktop]$ cd ..
[charey@linux-lab-001 charey]$ ls
Desktop    Downloads  Music     Public      Templates
Documents  EE332      Pictures  simulation  Videos
[charey@linux-lab-001 charey]$ ls -a
.               .dbus          .libmgr     Templates
..              Desktop        .local      Videos
.bash_history   Documents      .mozilla    .vnc
.bash_logout    Downloads      Music       .Xauthority
.bash_profile   EE332          Pictures    .xfce4-session.verbose-log
.bashrc         .esd_auth      .pki        .xfce4-session.verbose-log.last
.cache          .gnupg         Public      .xscreensaver
.config         .history       simulation
.cshrc          .ICEauthority  .source
[charey@linux-lab-001 charey]$ 
```

ls

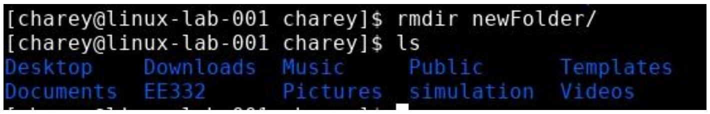
4. mkdir - create a folder. **mkdir directoryName**



```
[charey@linux-lab-001 charey]$ mkdir newFolder
[charey@linux-lab-001 charey]$ ls
Desktop    Downloads  Music      Pictures  simulation  Videos
Documents  EE332      newFolder  Public    Templates
```

mkdir

5. rmdir/rm - rmdir can remove an empty directory. rm can remove any directory. **rmdir directoryName** or **rm directoryName**



```
[charey@linux-lab-001 charey]$ rmdir newFolder/
[charey@linux-lab-001 charey]$ ls
Desktop    Downloads  Music     Public      Templates
Documents  EE332      Pictures  simulation  Videos
```

rmdir

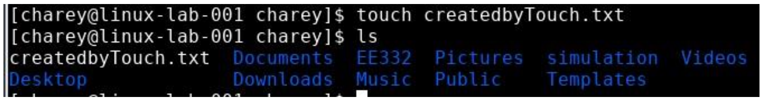
Note: after the command, you could add options as we mentioned in man, here I do a demo about how to use it. For rm, it has options -d, -f, -I and so forth.

-f Attempt to remove the files without prompting for confirmation, regardless of the file's permissions.

-r Attempt to remove the file hierarchy rooted in each file argument.

So, try **rm -r -f someDirectory** means delete all the files in the directory without asking permission.

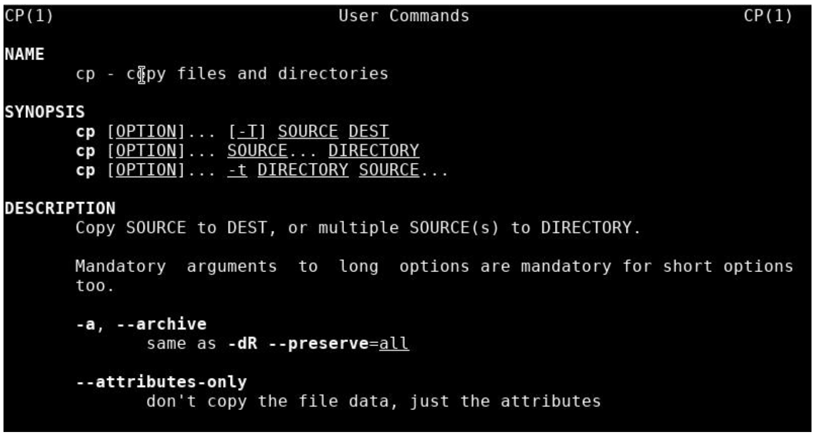
6. touch - create a file, **touch fileName**



```
[charey@linux-lab-001 charey]$ touch createdbyTouch.txt
[charey@linux-lab-001 charey]$ ls
createdbyTouch.txt  Documents  EE332  Pictures  simulation  Videos
Desktop             Downloads  Music  Public    Templates
```

touch

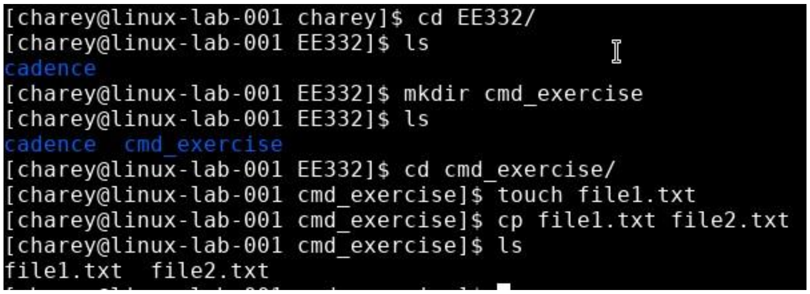
7. man - show manual page for a command **man commandName**



```
CP(1)                          User Commands                          CP(1)

NAME
       cp - copy files and directories

SYNOPSIS
       cp [OPTION]... [-T] SOURCE DEST
       cp [OPTION]... SOURCE... DIRECTORY
       cp [OPTION]... -t DIRECTORY SOURCE...

DESCRIPTION
       Copy SOURCE to DEST, or multiple SOURCE(s) to DIRECTORY.

       Mandatory  arguments  to  long  options are mandatory for short options
       too.

       -a, --archive
              same as -dR --preserve=all

       --attributes-only
              don't copy the file data, just the attributes
```

man

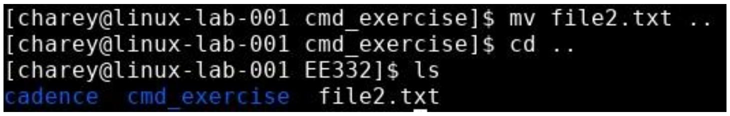
8. cp - copy files, takes two arguments, the first is the location of the file to be copied and the second is to copy to where **cp fileToBeCopied copyToWhere**



```
[charey@linux-lab-001 charey]$ cd EE332/
[charey@linux-lab-001 EE332]$ ls
cadence
[charey@linux-lab-001 EE332]$ mkdir cmd_exercise
[charey@linux-lab-001 EE332]$ ls
cadence  cmd_exercise
[charey@linux-lab-001 EE332]$ cd cmd_exercise/
[charey@linux-lab-001 cmd_exercise]$ touch file1.txt
[charey@linux-lab-001 cmd_exercise]$ cp file1.txt file2.txt
[charey@linux-lab-001 cmd_exercise]$ ls
file1.txt  file2.txt
```

Touch & cp

9. mv - move files or rename the file. Two arguments similar to cp command

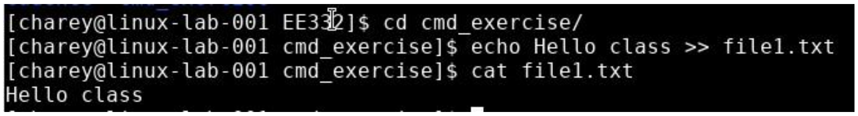


```
[charey@linux-lab-001 cmd_exercise]$ mv file2.txt ..
[charey@linux-lab-001 cmd_exercise]$ cd ..
[charey@linux-lab-001 EE332]$ ls
cadence  cmd_exercise  file2.txt
```

mv

10. echo - move data, typically text into a file **echo data >> fileName**

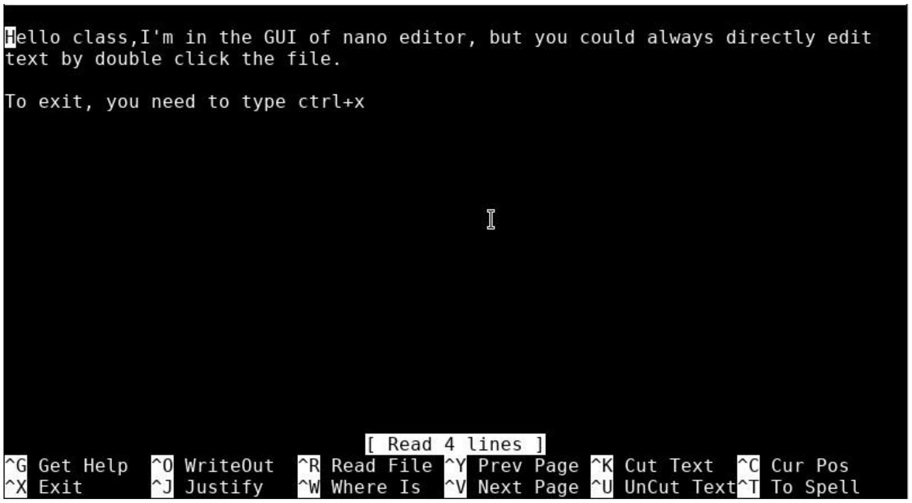
11. cat - display the contents of a file **cat fileName**



```
[charey@linux-lab-001 EE332]$ cd cmd_exercise/
[charey@linux-lab-001 cmd_exercise]$ echo Hello class >> file1.txt
[charey@linux-lab-001 cmd_exercise]$ cat file1.txt
Hello class
```

Echo & cat

12. nano/vi - bring up Linux built-in text editor in the command line. Sometimes this could be nasty, you can always edit the text file in GUI by double click the file. **nano/vi fileName**, if filename dose not exist in current directory, this command will create the file

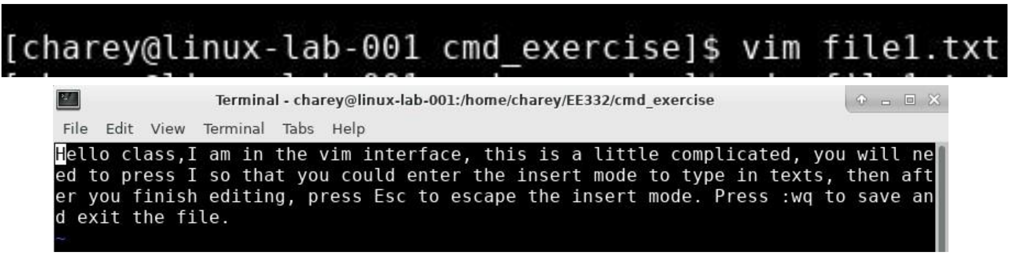


```
Hello class,I'm in the GUI of nano editor, but you could always directly edit
text by double click the file.

To exit, you need to type ctrl+x
```

```
                              [ Read 4 lines ]
^G Get Help  ^O WriteOut  ^R Read File ^Y Prev Page ^K Cut Text  ^C Cur Pos
^X Exit      ^J Justify   ^W Where Is  ^V Next Page ^U UnCut Text^T To Spell
```

nano

13. gedit/vim – the gedit/vim text editor. **gedit/vim filename** The gedit is pretty similar to what you would normally expected in windows or IOS. The vim editor is a bit complicated, after enter the vim editor interface, you need to press **I** key to enter the insert mode. In the insert mode, you can edit your text file just like normal txt editor, after editing, press **Esc** key to get out of insert mode. Type **:wq** to save and quit vim editor

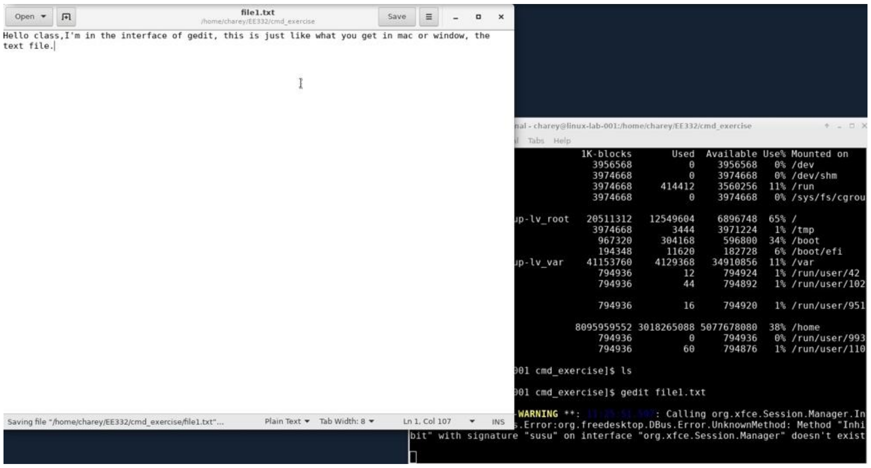


```
[charey@linux-lab-001 cmd_exercise]$ vim file1.txt
```

Terminal - charey@linux-lab-001:/home/charey/EE332/cmd_exercise

```
Hello class,I am in the vim interface, this is a little complicated, you will ne
ed to press I so that you could enter the insert mode to type in texts, then aft
er you finish editing, press Esc to escape the insert mode. Press :wq to save an
d exit the file.
~
```

Vim



file1.txt
/home/charey/EE332/cmd_exercise

```
Hello class,I'm in the interface of gedit, this is just like what you get in mac or window, the
text file.
```

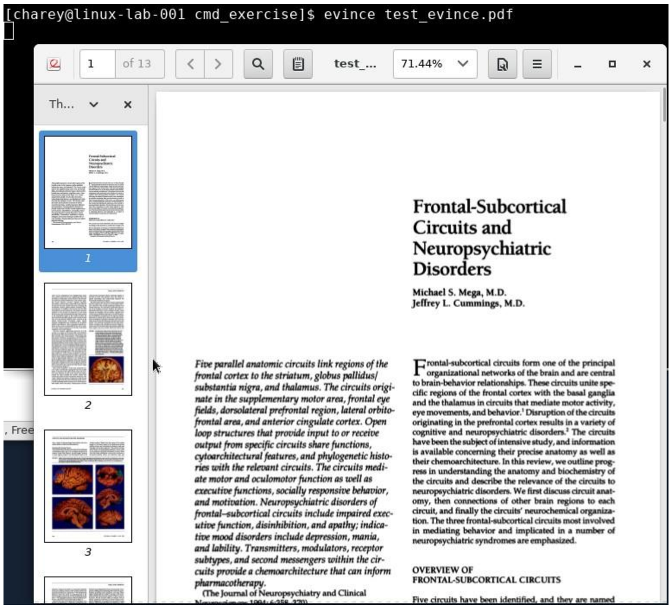
Saving file "/home/charey/EE332/cmd_exercise/file1.txt"... Plain Text Tab Width: 8 Ln 1, Col 107 INS

nal - charey@linux-lab-001:/home/charey/EE332/cmd_exercise

```
               1K-blocks       Used   Available Use% Mounted on
                 3956568          0     3956568   0% /dev
                 3974668          0     3974668   0% /dev/shm
                 3974668     414412     3560256  11% /run
                 3974668          0     3974668   0% /sys/fs/cgrou

up-lv_root      20511312   12549604     6896748  65% /
                 3974668       3444     3971224   1% /tmp
                  967320     304168      596800  34% /boot
                  194348      11620      182728   6% /boot/efi
up-lv_var       41153760    4129368    34910856  11% /var
                  794936         12      794924   1% /run/user/42
                  794936         44      794892   1% /run/user/102

                  794936         16      794920   1% /run/user/951

              8095959552 3018265088  5077678080  38% /home
                  794936          0      794936   0% /run/user/993
                  794936         60      794876   1% /run/user/110

001 cmd_exercise]$ ls

001 cmd_exercise]$ gedit file1.txt

WARNING **: [illegible]: Calling org.xfce.Session.Manager.In
s.Error:org.freedesktop.DBus.Error.UnknownMethod: Method "Inhi
bit" with signature "susu" on interface "org.xfce.Session.Manager" doesn't exist
```

gedit

14. evince – evince is a free and open source document viewer, you could use this command to open a pdf file. **evince fileName**



```
[charey@linux-lab-001 cmd_exercise]$ evince test_evince.pdf
```

Frontal-Subcortical Circuits and Neuropsychiatric Disorders

Michael S. Mega, M.D.
Jeffrey L. Cummings, M.D.

Five parallel anatomic circuits link regions of the frontal cortex to the striatum, globus pallidus/substantia nigra, and thalamus. The circuits originate in the supplementary motor area, frontal eye fields, dorsolateral prefrontal region, lateral orbitofrontal area, and anterior cingulate cortex. Open loop structures that provide input to or receive output from specific circuits share functions, cytoarchitectural features, and phylogenetic histories with the relevant circuits. The circuits mediate motor and oculomotor function as well as executive functions, socially responsive behavior, and motivation. Neuropsychiatric disorders of frontal–subcortical circuits include impaired executive function, disinhibition, and apathy; indicative mood disorders include depression, mania, and lability. Transmitters, modulators, receptor subtypes, and second messengers within the circuits provide a chemoarchitecture that can inform pharmacotherapy.

(The Journal of Neuropsychiatry and Clinical Neurosciences 1994; 6:358–370)

Frontal-subcortical circuits form one of the principal organizational networks of the brain and are central to brain-behavior relationships. These circuits unite specific regions of the frontal cortex with the basal ganglia and the thalamus in circuits that mediate motor activity, eye movements, and behavior.[1] Disruption of the circuits originating in the prefrontal cortex results in a variety of cognitive and neuropsychiatric disorders.[2] The circuits have been the subject of intensive study, and information is available concerning their precise anatomy as well as their chemoarchitecture. In this review, we outline progress in understanding the anatomy and biochemistry of the circuits and describe the relevance of the circuits to neuropsychiatric disorders. We first discuss circuit anatomy, then connections of other brain regions to each circuit, and finally the circuits' neurochemical organization. The three frontal-subcortical circuits most involved in mediating behavior and implicated in a number of neuropsychiatric syndromes are emphasized.

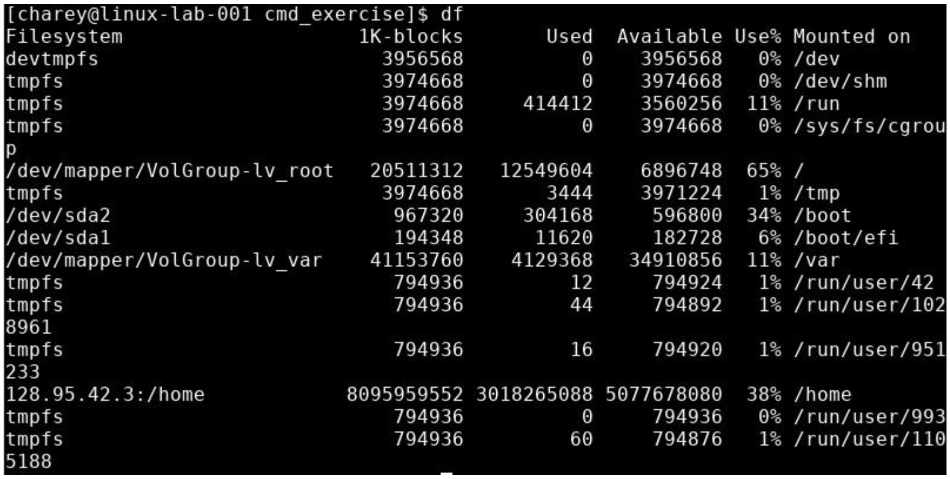
OVERVIEW OF
FRONTAL-SUBCORTICAL CIRCUITS

Five circuits have been identified, and they are named

evince

15. sudo - ``Super User Do", carry out the command with administrative/root privileges. It's widely used command, for example when you download and install a new package to your account

16. df - view available disk space



```
[charey@linux-lab-001 cmd_exercise]$ df
Filesystem                    1K-blocks       Used  Available Use% Mounted on
devtmpfs                        3956568          0    3956568   0% /dev
tmpfs                           3974668          0    3974668   0% /dev/shm
tmpfs                           3974668     414412    3560256  11% /run
tmpfs                           3974668          0    3974668   0% /sys/fs/cgrou
p
/dev/mapper/VolGroup-lv_root   20511312   12549604    6896748  65% /
tmpfs                           3974668       3444    3971224   1% /tmp
/dev/sda2                        967320     304168     596800  34% /boot
/dev/sda1                        194348      11620     182728   6% /boot/efi
/dev/mapper/VolGroup-lv_var    41153760    4129368   34910856  11% /var
tmpfs                            794936         12     794924   1% /run/user/42
tmpfs                            794936         44     794892   1% /run/user/102
8961
tmpfs                            794936         16     794920   1% /run/user/951
233
128.95.42.3:/home            8095959552 3018265088 5077678080  38% /home
tmpfs                            794936          0     794936   0% /run/user/993
tmpfs                            794936         60     794876   1% /run/user/110
5188
```

df

17. tar - compress/uncompress different types of tar archives *<u>.tar, .tar.gz, .tar.bz2</u>* and so forth. It has many different options to use, for example **tar -cvf** for creating a *.tar* archive, **tar -xvf** to uncompress the tar archive.

18. zip - create a compressed zip archive similar to tar command, but easier to use ``zip filename"/``unzip fileName"

19. apt-get - install packages, usually use company with **sudo** command

References

[1] https://maker.pro/linux/tutorial/basic-linux-commands-for-beginners

More about command line

https://www.howtogeek.com/412055/37-important-linux-commands-you-should-know

https://www.hostinger.com/tutorials/linux-commands

Lab Report Deliverable

Make sure you get yourself familiar with the commands mentioned in this instruction. Prepare yourself with the connection to server since we need that for our next labs! There is no deliverable for the report from this lab. Your first report will be from the Lab 1.