Connecting to UNIX/Linux at NJIT

A. V. Gerbessiotis
CS Department
NJIT

August 30, 2024

Revised 2/15/2024 (revision 1)

Revised 3/19/2024 (revision 2)

Revised 5/22/2024 (revision 3)

Revised 6/14/2024 (revision 4)

Revised 8/30/2024 (revision 5)

Previous documents dated 2021 or 2022 such as the ones listed below are superseded by this new document. Many typos have been corrected. This new version is also consistent with Windows 11 in addition to Windows 10. Note that things have changed at NJIT starting Fall 2024. Therefore previous documents describe methods or utilize machines that are not available any more.

[Ref1] **Connecting to *UNIX at NJIT, December 23, 2022.**

[Ref2] **Connecting to *NIX at NJIT, December 23, 2021.**

This 2024 document is Part A of a longer document consisting of four parts of a Brief on Unix/Linux at NJIT.

PART A

Connecting to UNIX/Linux at NJIT

Precondition: You have an NJIT-sanctioned laptop or desktop computer that satisfies NJIT computer policies as applicable to the CS Department and/or other NJIT units. As of this writing, you may find the policy at the URL below (URL: Uniform Resource Locator usually referred to as a 'link').

The policy can be summarized as follows: you have a Windows-based computer (Windows 10 or may be a later version such as Windows 11).

https://ist.njit.edu/student-computers

NJIT might start including Mac OSX laptops into the NJIT computer policy. This document does not discuss the Mac OSX option.

Moreover you have enabled Duo 2FA (two-factor authentication). Every step that requires password authentication using your myUCID credentials would also require Duo 2FA authentication.

Terminology

Credentials: The pair **(myUCIDlogin, myUCIDpassword)**. The **myUCID** login, or just **UCID**, or just **login** identifies you. The associated **myUCIDpassword** or just **password** verifies that you are the owner of myUCIDlogin. To confirm that you are the legitimate owner of myUCIDlogin rather than a person who got into the possession of the credentials, 2FA authentication is being used as well.

1. AFS machines

The term AFS machines (or previously, OSL machines) at NJIT denotes Linux hosting computers. NJIT afs machines support the AFS file system. AFS stands for Andrew File System, developed at CMU and is a distributed file system. Whether you login to one afs machine or another, your home directory (also known as the login directory) is the same and you view your files transparently. Moreover your files can even become accessible in the form of a Windows letter drive (e.g. D:\ drive) if you install client AFS Windows software on your Windows client machine that maps to a drive letter your AFS home directory files. The latter is not an easy process, thus it is not discussed here.

Effective Fall Semester 2024, you will have access to **afslogin0.njit.edu** and **afslogin1.njit.edu**. This or the other identifier is known as a **host name**. Sometimes we would use the term host name to refer to just afslogin0 or afslogin1 ignoring the **domain name** njit.edu.

In addition to those two afsmachines you may use the Linux machines available at the OSL (Open Systems Laboratory) on the 2nd floor of the GITC building. Physical access is required affective Fall 2024 to access those machines, which are also afs machines.

[Connectivity to an afs machine at the OSL Lab.] If you are physically present at the OSL Lab you may use the graphical interface to connect to the AFS machines there. We do not further discuss this step.

[Connectivity to any one of the two designated afs machines.] You may use a secure shell client to connect to any afs machine and in particular to **afslogin0** and **afslogin1**. If you are outside of NJIT, a VPN client such as the NJIT provided Cisco Anyconnect VPN MUST be used. You need to have the VPN client installed, activated, and used, prior to establishing a secure shell connection. Bear in mind that the two afs machines run VMWare on top of Linux, presenting to you a virtual environment.

2. URLs of interest

Some URLs (Uniform Resource Locator) of interest are as follows.

0. NJIT computer policies as applicable to the CS Department and other NJIT units are available, as of this writing, at
   https://ist.njit.edu/student-computers
1. [Graphical Secure Shell Client for Windows]. The URL for downloading NJIT's copy of MobaXterm that includes a Windows **secure shell client (ssh)** is shown below.
   https://ist.njit.edu/software
   One may also download a copy of MobaXterm of limited functionality directly from the manufacturer. In that case you may not even need to install MobaXterm as it can run from any directory of your client Windows computer; for more see manufacture/publisher's web site below.
   https://mobaxterm.mobatek.net/
   https:**//ist.njit.edu/how-connect-afs-mobaxterm**
2. The following URL provides also information about using ssh on Mac OSX and Linux.
   https://ist.njit.edu/accessing-afs
3. The URL for 'accessing AFS' which means connecting to a linux machine at NJIT is shown below

   https://ist.njit.edu/afs
4. The NJIT VPN URL with links to downloadable VPN clients for Windows, MacOSX and Linux with instructions is shown below.
   https://ist.njit.edu/vpn
5. An NJIT URL with info on *nix/Linux commands is shown below.
   https://ist.njit.edu/common-UNIX-commands

3. CISCO Anyconnect VPN

The discussion below uses a client computer that is a WINDOWS 11 machine. This is in accordance with current (as of this writing) YWCC guidelines. It also applies to Windows 10 machines.

A VPN connection may need to be established prior to establishing a secure shell (ssh) connection to an AFS machine. This usually happens when you are outside of NJIT.

Thus, If you plan to connect to an AFS machine from **outside of NJIT** you must

(1) Detect if a VPN client has been installed previously on your machine. If it is not installed, you should install it first, a process that usually requires a reboot. We expect computing students can figure out whether a VPN client is preinstalled or previously installed (by you) on their machine (Settings->Programs or Settings->Apps might provide some information on whether a Cisco Secure VPN client has been installed or not). If a VPN client is not installed, first download such a VPN (Virtual Private Network) client though URL 4 of Section 2, and install it. Installation is done once and might require a reboot or a restart.

(2) **Activate VPN if a VPN client is preinstalled on your machine but is currently deactivated (this is a rare case, since by default it is activated at boot time).** If you know how to deactivate it, you should know how to activate it or how to reactivate it. Read the discussion that follows.

1. On my laptop the windows taskbar is at the bottom of the screen, with a Windows icon on the bottom left corner and the time and date information on the bottom right area/corner. In the bottom right area of the taskbar you might see the icon shown below (a globe). If not, find an up-arrow in the right area of the task bar, click on it and see if the icon is depicted on the popped up window. For me, it appears as shown below in Figure 4-1 (the background color might be different from the indicated black depending on a device's configuration). This means that the VPN client is **ACTIVATED but it is not in USE ('not connected to the NJIT' network).** If the VPN client were in USE, the icon would have appeared as in Figure 4-1(a) with a yellow lock visible (it also happens that the background is white).

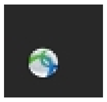


Figure 4-1.

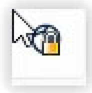
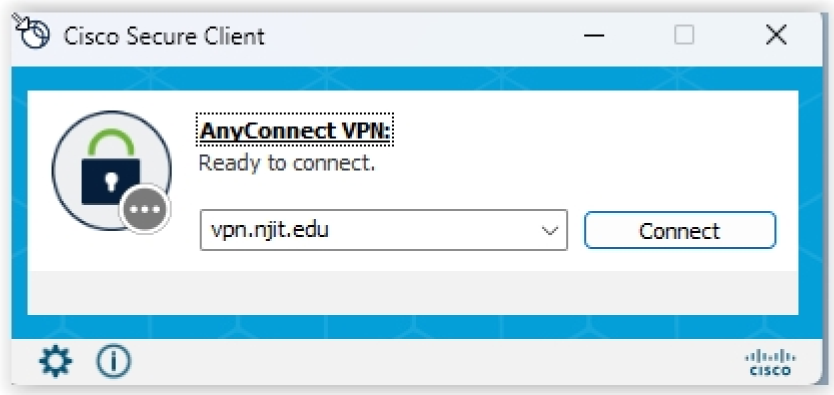


Figure 4-1(a).

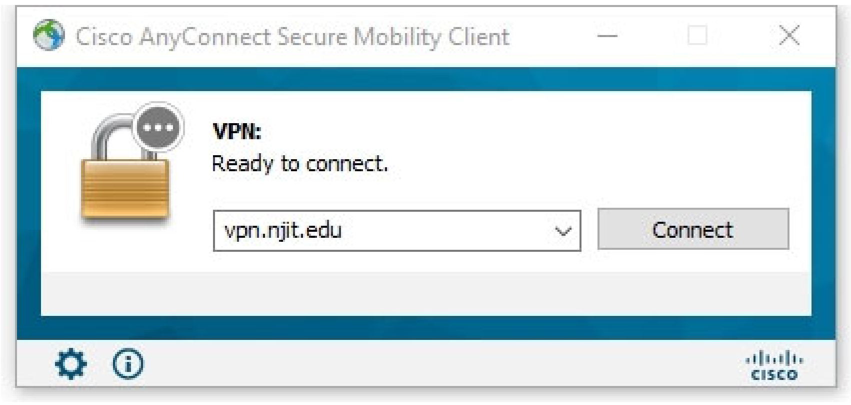
2. You may click on this icon of Figure 4-1 and the following pop-up window might appear as shown in Figure 4-2(a). In other (e.g. older) versions of the VPN client the window might look like the one in Figure 4-2(b) instead.



Cisco Secure Client

AnyConnect VPN:
Ready to connect.

Figure 4-2(a)

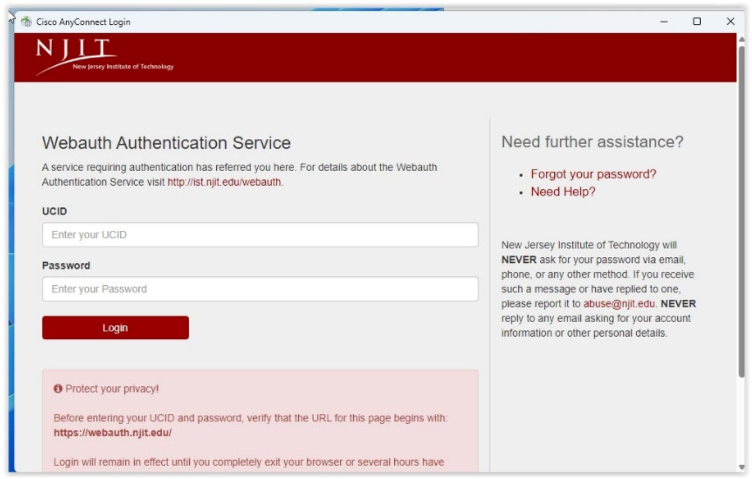


Cisco AnyConnect Secure Mobility Client

VPN:
Ready to connect.

Figure 4-2(b)

3. The Ready to connect message in Figure 4-2(a) or 4-2(b) indicates not only that your machine is disconnected (i.e. the VPN client is not in use), but also indicates that your clicking on the button labeled Connect will allow a connection to take place and thus your laptop will become part of the NJIT network. You are about ready to click the Connect button. In the next step **you need to have ready** your myUCID credentials (login , password) which must not have expired. After you click connect the pop up window of Figure 4-3(a) or the one in Figure 4-3(b) will be shown.



Cisco AnyConnect Login

NJIT
New Jersey Institute of Technology

Webauth Authentication Service

A service requiring authentication has referred you here. For details about the Webauth Authentication Service visit http://ist.njit.edu/webauth.

UCID
Enter your UCID

Password
Enter your Password

Login

Protect your privacy!

Before entering your UCID and password, verify that the URL for this page begins with: **https://webauth.njit.edu/**

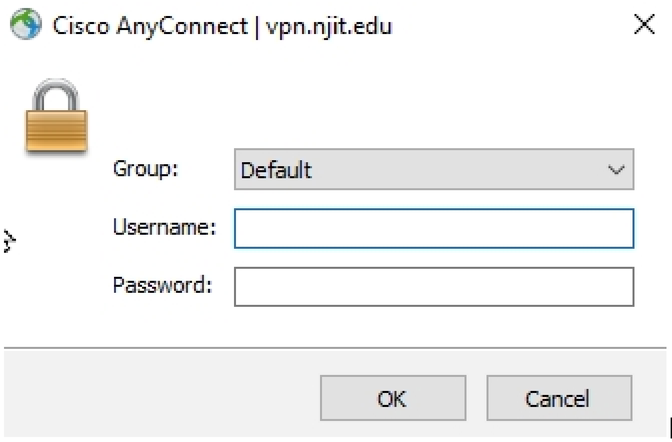
Login will remain in effect until you completely exit your browser or several hours have

Need further assistance?

- Forgot your password?
- Need Help?

New Jersey Institute of Technology will **NEVER** ask for your password via email, phone, or any other method. If you receive such a message or have replied to one, please report it to abuse@njit.edu. **NEVER** reply to any email asking for your account information or other personal details.

Figure 4-3(a)



Cisco AnyConnect | vpn.njit.edu

Group: Default

Username:

Password:

OK   Cancel

Figure 4-3(b)

4. The UCID field of Figure 4-3(a) or Username: field of Figure 4-3(b) might have been pre-populated with your myUCID. If not, type in your myUCID login in the UCID or Username: field respectively, and then type in your myUCID password in the Password: field. The use of the TAB button (or SHIFT-TAB button) of the keyboard can help you moving around quickly without a mouse. The Group: option can be left as the Default or you select an alternative according to the user instructions available during the VPN installation process or other information provided by NJIT. If you are presented with a Figure 4-3(a) prompt, a 2FA window (DUO two factor authentication) might pop up afterwards as well. Follow the DUO instructions then to authenticate.

5. If you have supplied the correct information (correct login name and correct associated password) and 2FA authenticated correctly (if needed), a connection will be established and VPN would be in use and the popup window of Figure 4-3(a) or (b) will disappear. At that point if you try to locate the VPN icon using the instructions of step 1, the icon has a lock on it as shown below in Figure 4-4.

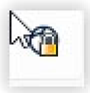
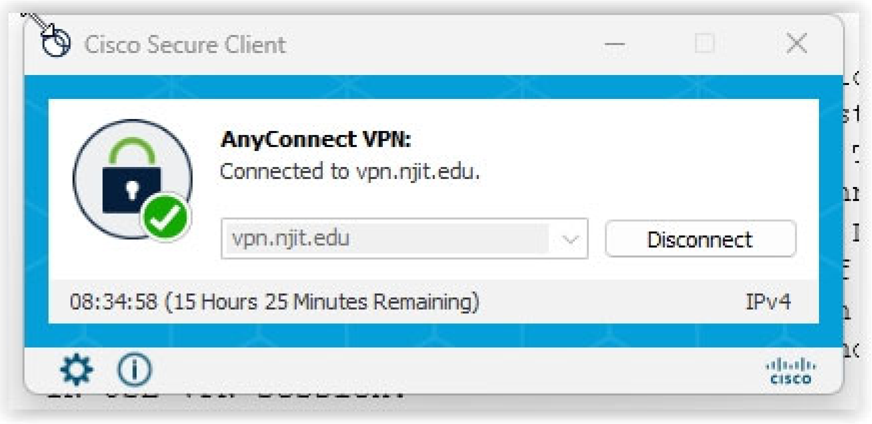


Figure 4-4.

6. If there was an active connection prior to step 2, and there will be an active connection after step 5, the window of Figure 4-2 would have looked like Figure 4-5. The button reads Disconnect now since VPN is in use vs the reading Connect in Figure 4-2 when VPN was not yet in use. You may click on Disconnect to terminate the VPN connection when it is of no need any more. An alternative is to locate the icon shown in Figure 4-4, click on it, and it will allow you to disconnect and thus terminate the IN-USE VPN session.



Cisco Secure Client

AnyConnect VPN:
Connected to vpn.njit.edu.

08:34:58 (15 Hours 25 Minutes Remaining) IPv4

Figure 4-5.

At this point even if you (in fact your device) are physically outside of NJIT , the active and in-use VPN session makes you (your device)   located within the NJIT network view.

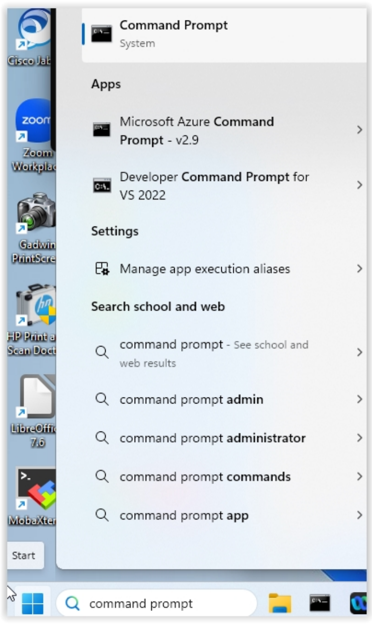
The next sections deal with the interaction with a secure shell (ssh) client and how it can be used to connect to an AFS machine using command line or a graphical interface.

4. Command line secure shell

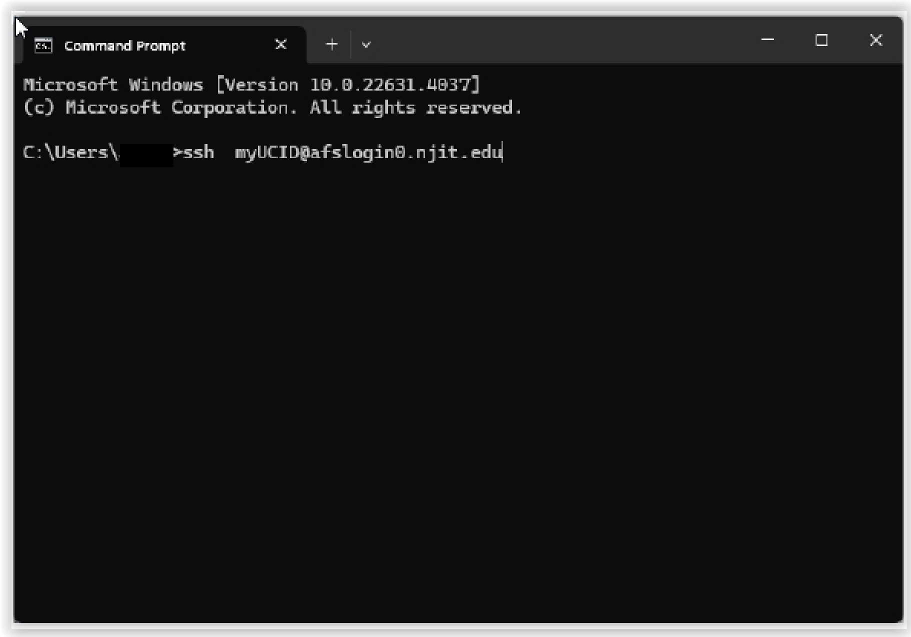
Note that Windows, Mac OSX machines and Linux machines support a non-graphical secure shell client through the command line. This means you need to open a Command prompt or a Powershell window in Windows, or a Terminal in Linux or Apple Mac OSX and invoke from the command line the secure shell by typing usually its name: ssh.

In the rest of this section we show how this works in Windows 11 though it should be applicable to Windows 10. Not much changes to do the same in Linux or Apple OSX, to our knowledge. Note that some areas have been redacted and thus have been blackened (replaced by black empty areas). These are areas that are going to be individual to you depending on your myUCID; thus you should be able to see your myUCID there or in some cases you need to supply it (e.g. in the command line).

STEP 1. Invoke the command prompt in MS Windows. See Figure 4-1. Next to the windows icon, there is the search box, and in it you type in Command prompt and when the results of the search pop up (top area of Figure 4-1) click on it.


Command Prompt
System
Apps
Microsoft Azure Command Prompt - v2.9
Developer Command Prompt for VS 2022
Settings
Manage app execution aliases
Search school and web
command prompt - See school and web results
command prompt admin
command prompt administrator
command prompt commands
command prompt app
Start
command prompt
Figure 4-1.

STEP 2. The following window shown in Figure 4-2 will pop up then. It is the Command Prompt. Depending on your prior setup or defaults it might look different. The line that starts possibly with a "C:\Users" and ends with a possibly ">" is the command line. In between you might see a user name which can be you myUCIDlogin or not.

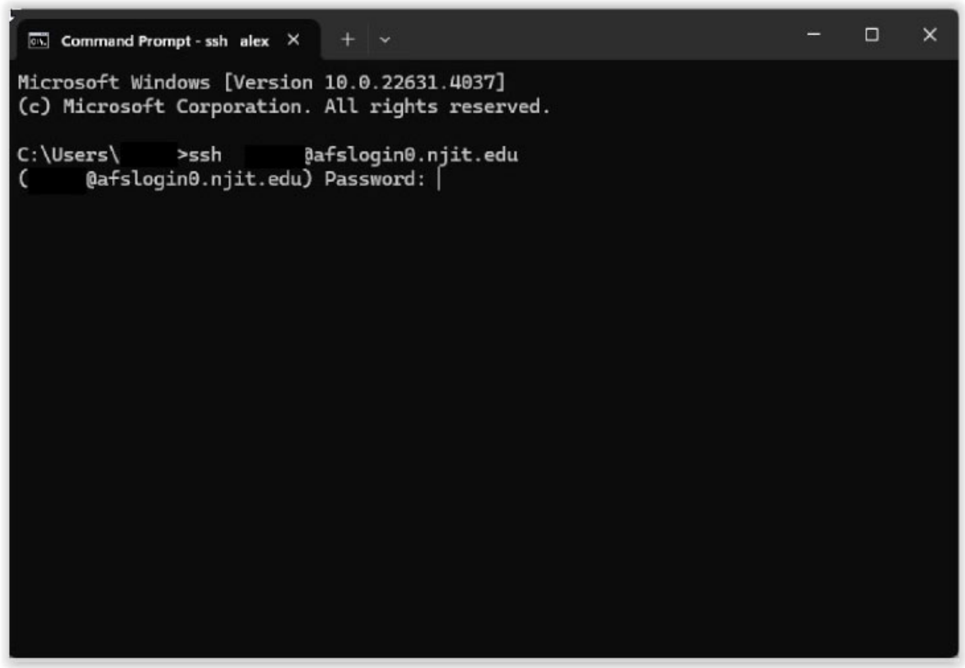


```
Microsoft Windows [Version 10.0.22631.4037]
(c) Microsoft Corporation. All rights reserved.

C:\Users\[redacted]>ssh  myUCID@afslogin0.njit.edu
```

Figure 4-2.

STEP 3. In the command line you invoke a Windows provided secure shell client which is named ssh (for secure shell) with an appropriate set of arguments **such as your myUCID login name** that I am going to assume in this example that it is a **generic `myUCID`** and the AFS host that will be used and for this example it is **`afslogin0.njit.edu`** The typed in info is shown in Figure 4-2: there is one or more spaces between ssh and the next text (the argument to ssh) and at then end of the type in line DO NOT FORGET to press ENTER (i.e. the Enter key of the keyboard). If you do not press ENTER nothing will happen other than you staring at the screen.

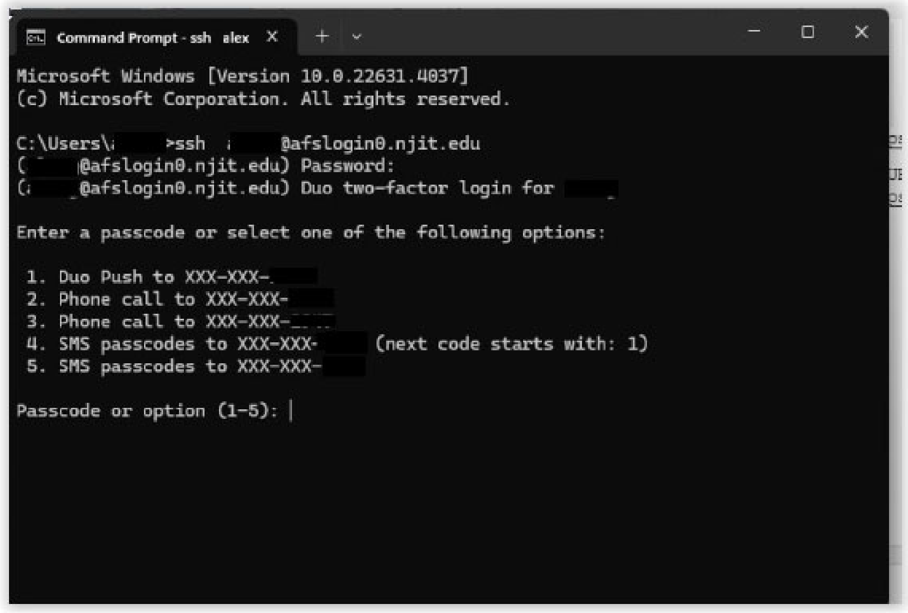
STEP 4. As shown in Figure 4-3, the local machine (Windows client's ssh) will send to the remote machine (the server afslogino.njit.edu) a request for connection for the desired generic user (in this example myUCID to be replaced by your actual UCID name). The remote machine (afslogino) responds by requesting a password authentication. If you see some blackened areas on the left of the @ symbol or the left of the > and the right of a backslash \ symbol this is where your actual userID or actual myUCID login should be displayed when you invoke ssh properly as indicated. The prompt that appears next to the word "Password : " looks like a vertical white bar. The local machine is waiting for you to supply a password that is going to be forwarded to the remote machine (in encrypted and possibly fingerprinted form).



```
Microsoft Windows [Version 10.0.22631.4037]
(c) Microsoft Corporation. All rights reserved.

C:\Users\[redacted]>ssh [redacted]@afslogin0.njit.edu
([redacted]@afslogin0.njit.edu) Password: |
```

Figure 4-3

STEP 5. In Figure 4-4, if things went well (i.e. you supplied the correct password by typing it correctly and it was accepted by the remote machine), the remote machine will request a Duo 2FA step as indicated. The list of options is dependent on your Duo 2FA setup. Pick one or if you have a hardware token type it into directly at the prompt (right of "Passcode or option" line)



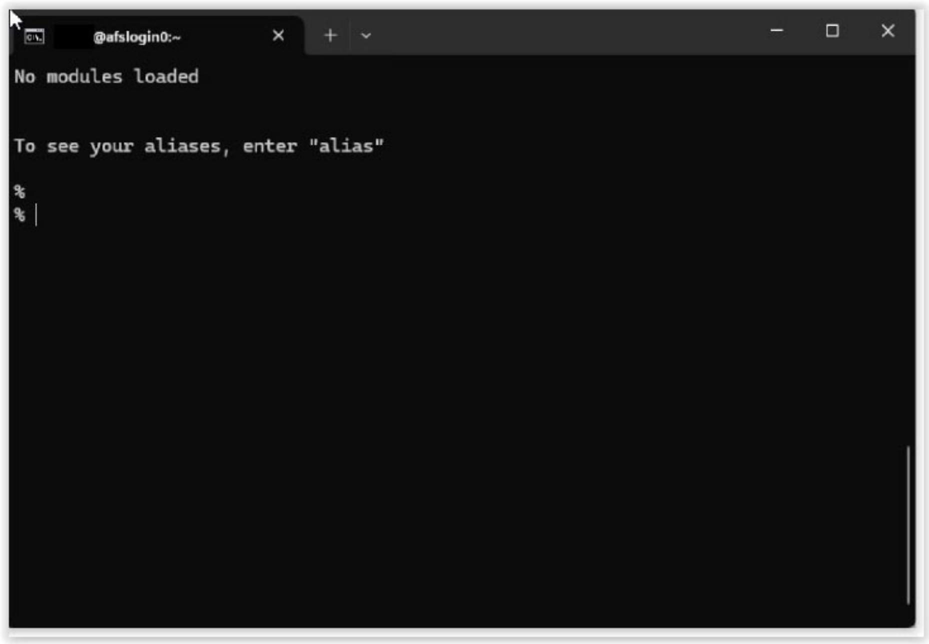
```
Microsoft Windows [Version 10.0.22631.4037]
(c) Microsoft Corporation. All rights reserved.

C:\Users\[redacted]>ssh [redacted]@afslogin0.njit.edu
([redacted]@afslogin0.njit.edu) Password:
([redacted]@afslogin0.njit.edu) Duo two-factor login for [redacted]

Enter a passcode or select one of the following options:

 1. Duo Push to XXX-XXX-[redacted]
 2. Phone call to XXX-XXX-[redacted]
 3. Phone call to XXX-XXX-[redacted]
 4. SMS passcodes to XXX-XXX-[redacted] (next code starts with: 1)
 5. SMS passcodes to XXX-XXX-[redacted]

Passcode or option (1-5): |
```

Figure 4-4

STEP 6. If things went well and you were 2FA authenticated, the window of Figure 4-5 will pop up after some interesting scrolling. You might use the scrollbar on the right to move it up or down and see what was actually displayed (or not).

At this point you are in the remote machine (afslogino). The % sign is MY (the writer's of this document) prompt for afslogino. Yours or the default one might be different and longer. Next to it after a space you see a vertical bar that is the cursor (aka cursor prompt or just prompt). The remote machine's shell is waiting for your requests. You type your requests in the form of commands at the cursor (prompt). We call it a shell because it operates as a shell to the remote operating system (Linux) thus allowing users like you to interact with the remote machine's (afslogino) operating system (OS) from your local machine (MS Windows). You may type Linux commands and after pressing the Enter key you may observe the reaction of the remote machine's OS : an output would be generated or an error will be displayed or may be an input prompt or some action taken.

Typing in for example hostname would confirm that you have been connected to afslogino.njit.edu. Typing in date prints the current date and time. There is a set of examples in the last section of this document or on a separate brief tutorial.



```
No modules loaded


To see your aliases, enter "alias"

%
% |
```

Figure 4-5.

STEP 7.   Following that step you may start a minor interaction with the remote machine as shown in Figure 4-6.

Type in pwd i.e. see below (% is displayed by the shell and you only type pwd and afterwards you press the Enter key/button)

```
%         pwd
```

This will show you your LOGIN directory (also known as home directory) on the AFS file system. The prefix you will see is going to be /afs/cad/X/Y/myUCID where X and Y you're your myUCID dependent characters: X and Y are the first and the second letters of your myUCID. At the end of the output you should recognize your myUCID login name. BTW pwd is an acronym for print working directory. Your working directory is currently your login directory because you have not moved out of it!

Type in ls (output not shown in Figure 4-6) i.e.

```
%  ls
```

This will present you with all the files created by you or by default by the system at your LOGIN directory. Confirm that there is no file of any type (a directory is a file of type directory, and we call it directory rather than folder in Linux) named mycs332. Why? Because we plan to create a directory mycs332 in the next step. By the way, command ls is an acronym for listing the contents of the current directory (i.e. home directory also known as the login directory). If you specify an argument to ls and it is a directory inside the current directory, that directory's contents (files) would only be displayed.

Assuming that there was no file named mycs332, type in mkdir mycs332 (see also Figure 4-6) i.e.

```
% mkdir mycs332
```

Command mkdir is an acronym for make directory.

Type then cd mycs332 (Figure 4-6) i.e.

```
% cd mycs332
```

Command cd is an acronym for change directory to the one indicated in its arguments. You may supply just the directory's if such directory is inside your current directory or a full path name to the directory such as

```
% cd /afs/cad/u/a/l/almn12/mycs332
```

Here we made up a fake/example myUCID: almn12. The first two letters of it are a and l.

Type then pwd (Figure 4-6) i.e.

```
% pwd
```

Its output would confirm that you have moved to the designated directory just created.

Type in ls there (not shown in Figure 4- 6) and you will confirm that there are no files in it. In fact there are two hidden files by you need to type ls with option a i.e. ls -a to see those hidden files

```
% ls -a
```

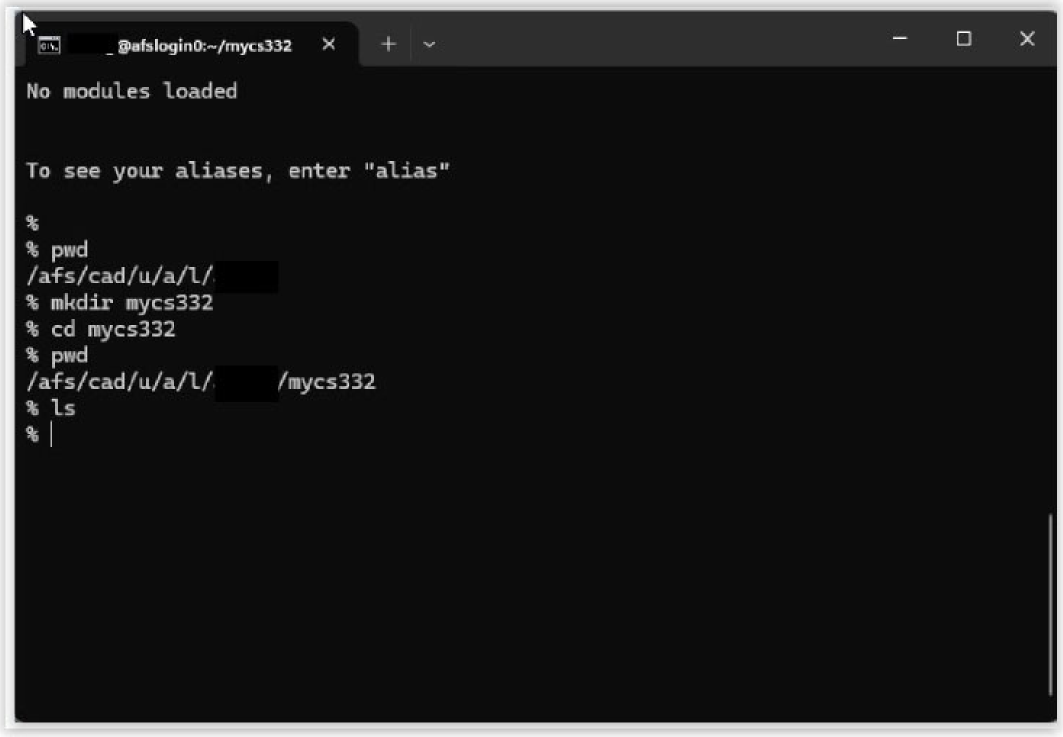
Above, there is a space between ls and -a. The dash(-) indicates the use of single letter options. The only option supplied is the lower-case letter a immediately after the dash (no space inbetween). (Do not forget to press ENTER at the end.) There is a limited Linux command overview at the end of this document and also in Section 2 link 5 there. Moreover

```
% man ls
```

or

```
% man man
```

can provide immediate information (manual) about a specific command such as ls or man itself.



```
[redacted] _@afslogin0:~/mycs332
```

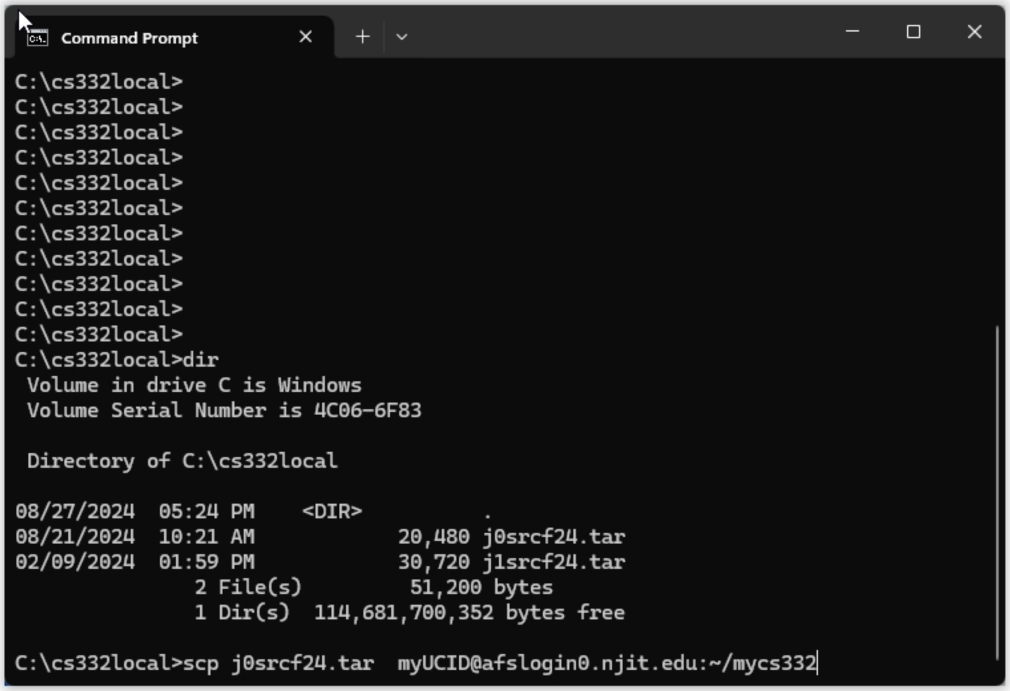
```
No modules loaded


To see your aliases, enter "alias"

%
% pwd
/afs/cad/u/a/l/[redacted]
% mkdir mycs332
% cd mycs332
% pwd
/afs/cad/u/a/l/[redacted]/mycs332
% ls
% |
```

Figure 4-6

SECURE COPY OF A FILE FROM

local machine (Windows) to remote AFS machine

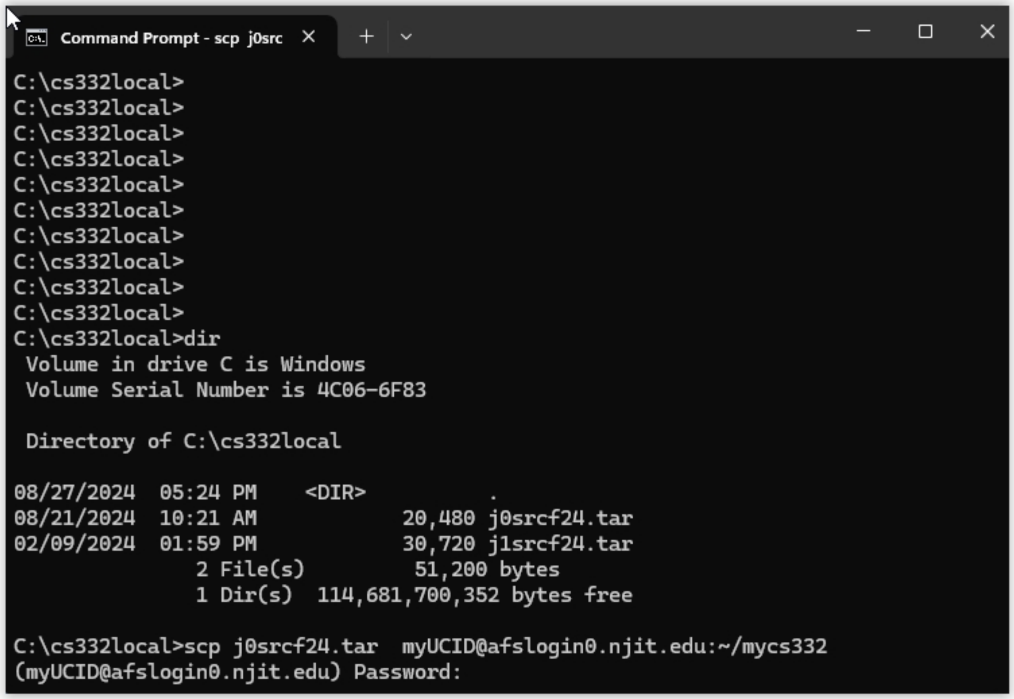
STEP 8. What if we do not want to login to the remote machine but we would like just to transfer (i.e. copy) a file to it and specifically copy a file to the remote machine's mycs332 directory? Instead of using ssh in the local (MS Windows machine) we can use instead the scp command (scp is short for secure cp i.e. secure copy). Figure4- 7 resembles Figure 4-2 and is also followed by two interactions involving password authentication and Duo 2FA authentication such as those shown in Figure4-3 and Figure 4-4 earlier. Our intent is to copy a tar file (josrcf24.tar) from the local Windows machine (the tar file is located in a chosen by the writer directory C:\cs332local of the Windows machine) to the remote (Linux) machine's directory mycs332 that was just created in previous steps. Figure 4-7 explains how to use the scp command on the Windows machine. The first argument to scp (josrcf24.tar) is a file in the current directory of the local machine (the directory is indicated as C:\cs332local). The second argument is a list that starts with the UCID of the user on the remote (Linux) machine (assumed to be a fictitious myUCID) followed by the at sign, followed by the hostname of the remote machine and after a colon : we provide the directory of the remote machine to which the file will be copied. Instead of typing /cad/afs/u/... /mycs332 we can just type instead ~/mycs332. The ~ alias expands to the long login directory of the user on the remote machine i.e. to the string /cad/afs/u/... etc. Pressing the ENTER key at the end we have the reaction of the remote machine as indicated in Figure 4-8.



```
C:\cs332local>
C:\cs332local>
C:\cs332local>
C:\cs332local>
C:\cs332local>
C:\cs332local>
C:\cs332local>
C:\cs332local>
C:\cs332local>
C:\cs332local>
C:\cs332local>
C:\cs332local>dir
 Volume in drive C is Windows
 Volume Serial Number is 4C06-6F83

 Directory of C:\cs332local

08/27/2024  05:24 PM    <DIR>          .
08/21/2024  10:21 AM            20,480 j0srcf24.tar
02/09/2024  01:59 PM            30,720 j1srcf24.tar
               2 File(s)         51,200 bytes
               1 Dir(s)  114,681,700,352 bytes free

C:\cs332local>scp j0srcf24.tar  myUCID@afslogin0.njit.edu:~/mycs332
```

Figure 4-7

STEP 9. Figure 4-8 is a request for a password authentication. This is similar to Figure 4-2. Afterwards a 2FA authentication will be requested similar to Figure 4-3. Details are not shown. After the two authentications the file will be transferred securely (securely copied) from the local to the remote machine. You may repeat some of the interactions of Figures 4-1 to 4-6 (do not try to mkdir the directory mycs332 that already exists though this second time!) to confirm that the file has been transferred and mycs332 of the remote machines contains now one additional file (or three overal if the two hidden files are included).



```
C:\cs332local>
C:\cs332local>
C:\cs332local>
C:\cs332local>
C:\cs332local>
C:\cs332local>
C:\cs332local>
C:\cs332local>
C:\cs332local>
C:\cs332local>
C:\cs332local>dir
 Volume in drive C is Windows
 Volume Serial Number is 4C06-6F83

 Directory of C:\cs332local

08/27/2024  05:24 PM    <DIR>          .
08/21/2024  10:21 AM            20,480 j0srcf24.tar
02/09/2024  01:59 PM            30,720 j1srcf24.tar
               2 File(s)         51,200 bytes
               1 Dir(s)  114,681,700,352 bytes free

C:\cs332local>scp j0srcf24.tar  myUCID@afslogin0.njit.edu:~/mycs332
(myUCID@afslogin0.njit.edu) Password:
```

Figure 4-8

5. Graphical ssh : MobaXterm

NJIT makes available to students and instructors a commercial graphical-based secure shell client known as MobaXTerm. Links on how to download it in Section 1 Link area 1.

The discussion below is for a Windows client using **Mobaxterm.**

1. Invoke Mobaxterm. An icon on your desktop might be available for clicking on it similar to the icon on the left of the word **MobaXterm** shown in Figure 5-1. A window as shown in Figure 5-1 will pop up. Depending on your settings and customizations, it might look different from the one in Figure 5-1. Click on the top left corner the icon whose label is Session (the mouse cursor in Figure 5-1 below is on top of that Session icon).

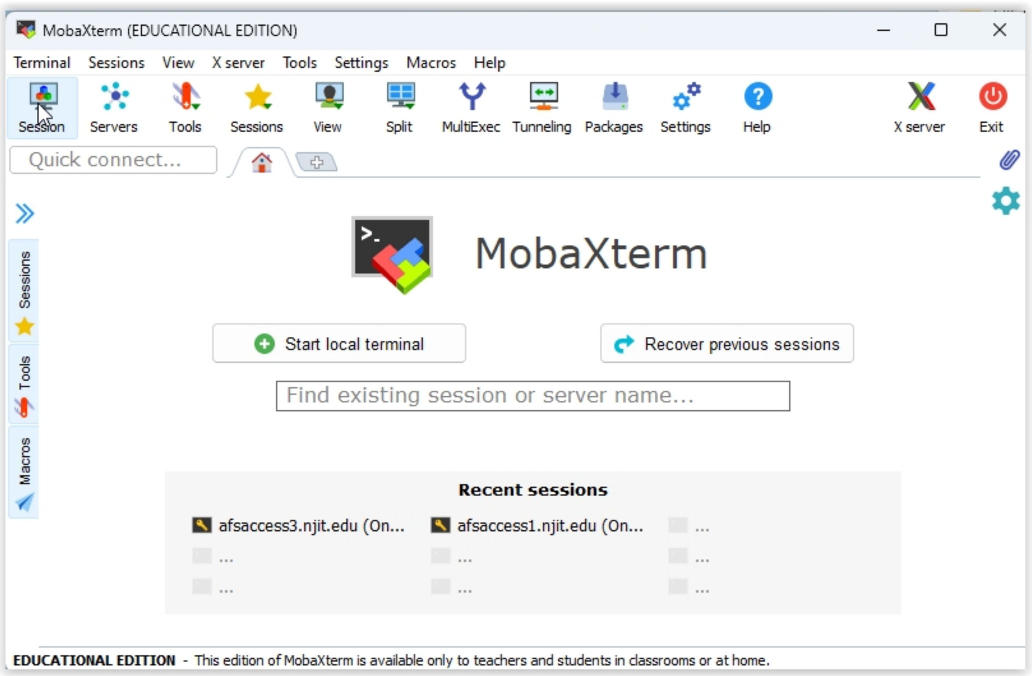

MobaXterm (EDUCATIONAL EDITION)
Terminal Sessions View X server Tools Settings Macros Help
Session Servers Tools Sessions View Split MultiExec Tunneling Packages Settings Help X server Exit
Quick connect...
Sessions
Tools
Macros
MobaXterm
Start local terminal
Recover previous sessions
Find existing session or server name...
Recent sessions
afsaccess3.njit.edu (On...
afsaccess1.njit.edu (On...
EDUCATIONAL EDITION - This edition of MobaXterm is available only to teachers and students in classrooms or at home.

Figure 5-1.

2. After clicking the button "Session", a window like the one shown below in Figure 5-2 will pop up. You choose session type SSH (top left icon, mouse cursor is indicated on top of it) and click on it.

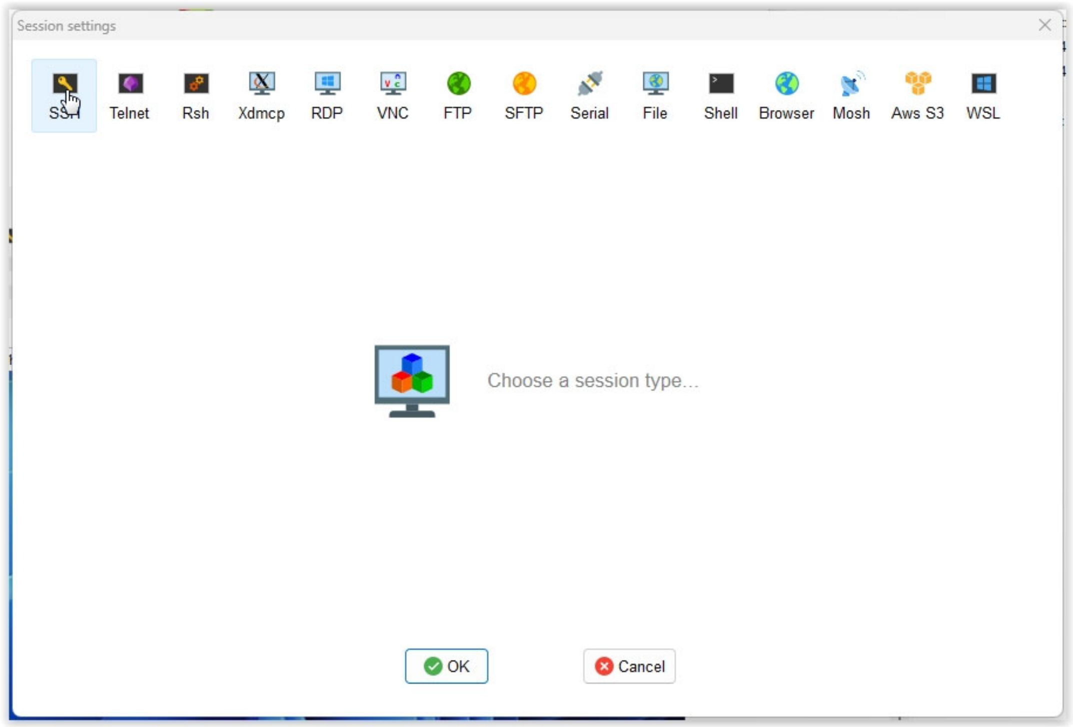


Figure 5-2.

3. You then get the window shown in Figure 5-3(a) below. In the "**Remote Host**" tab you write in the name of the osl machine you plan to connect to. You click and check on the button on the left of Specify username and on the text area on the right of Specify Username you type in your own myUCID. You might have to overwrite an indicated prefilled <default>. When you are done click on the Advanced SSH settings as shown in Figure 5-3(b) and choose for browse type SCP, and when you are done with click the **OK** button at the bottom.

Remote host is what we called before remote machine!

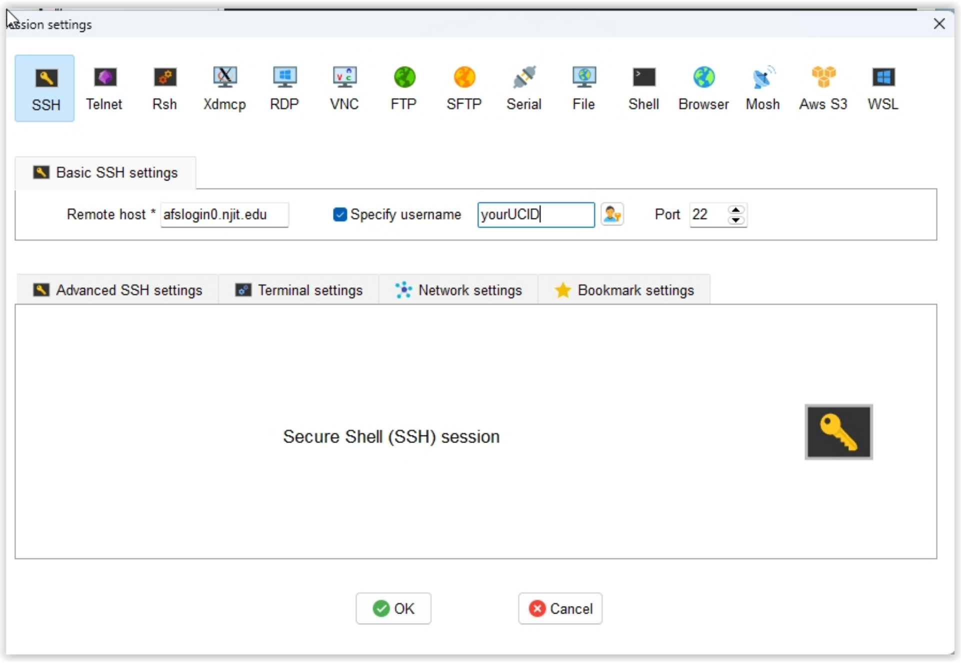
In the example below I used the following.

Example: In the **Remote host*** I typed: `afslogin0.njit.edu`

but only past of the host name is visible through the box

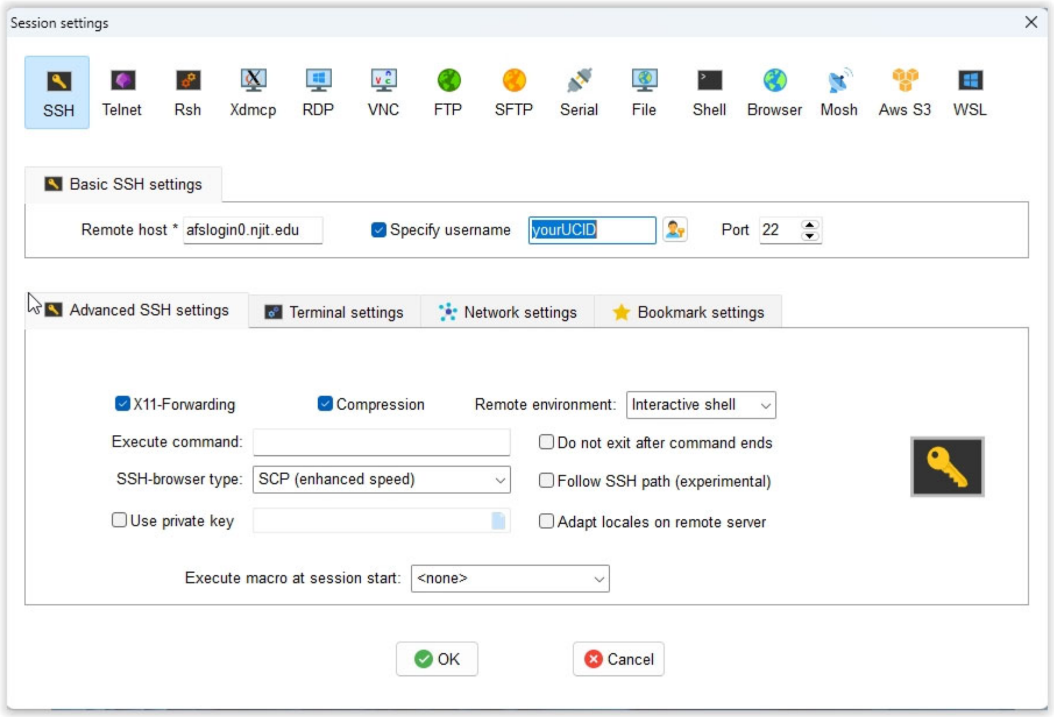
In the **Specify username** after checking the box I typed in a fictitious myUCID as an example : yourUCID

Use your own myUCID and thus replace the text myUCID as needed, and the desired host machine as needed.



Remote host * afslogin0.njit.edu
☑ Specify username yourUCID
Port 22

Figure 5-3(a)



Remote host * afslogin0.njit.edu ☑ Specify username yourUCID Port 22

☑ X11-Forwarding ☑ Compression Remote environment: Interactive shell

Execute command: ☐ Do not exit after command ends

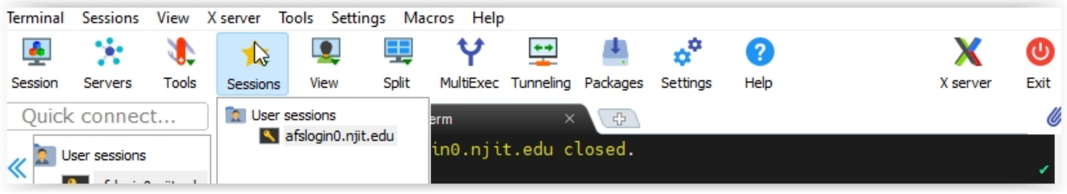
SSH-browser type: SCP (enhanced speed) ☐ Follow SSH path (experimental)

☐ Use private key ☐ Adapt locales on remote server

Execute macro at session start: <none>

Figure 5-3(b)

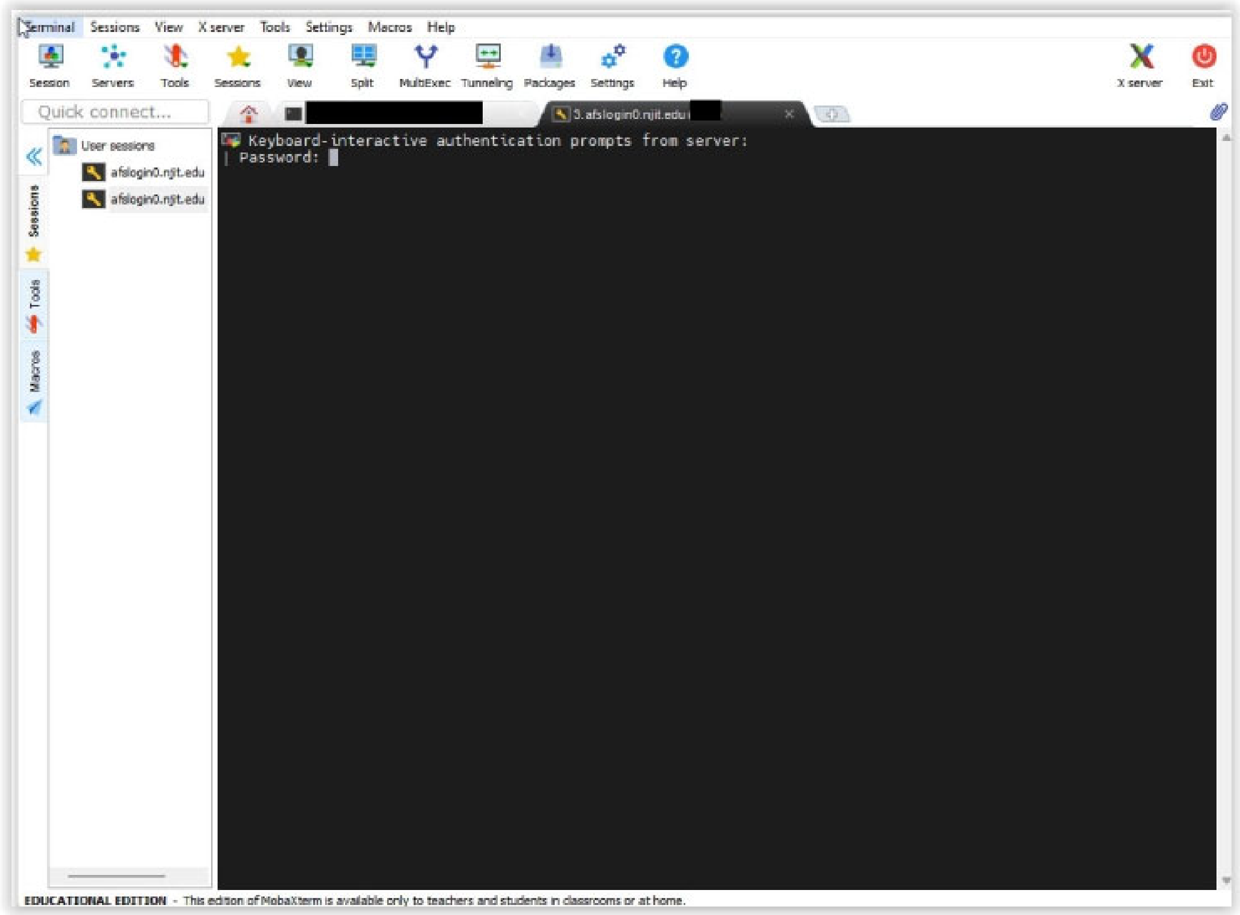
4. When you are done with step 3 you go back to the view of Figure 5-1. You navigate on Sessions as shown in Figure 5-4(a) below. The mouse button is already on top of the Sessions icons that is colored light blue and beneath it you see a User sessions tab with a yellow key in black square on the left of "afslogino.njit.edu". You click it.



```
in0.njit.edu closed.
```

Figure 5-4(a)

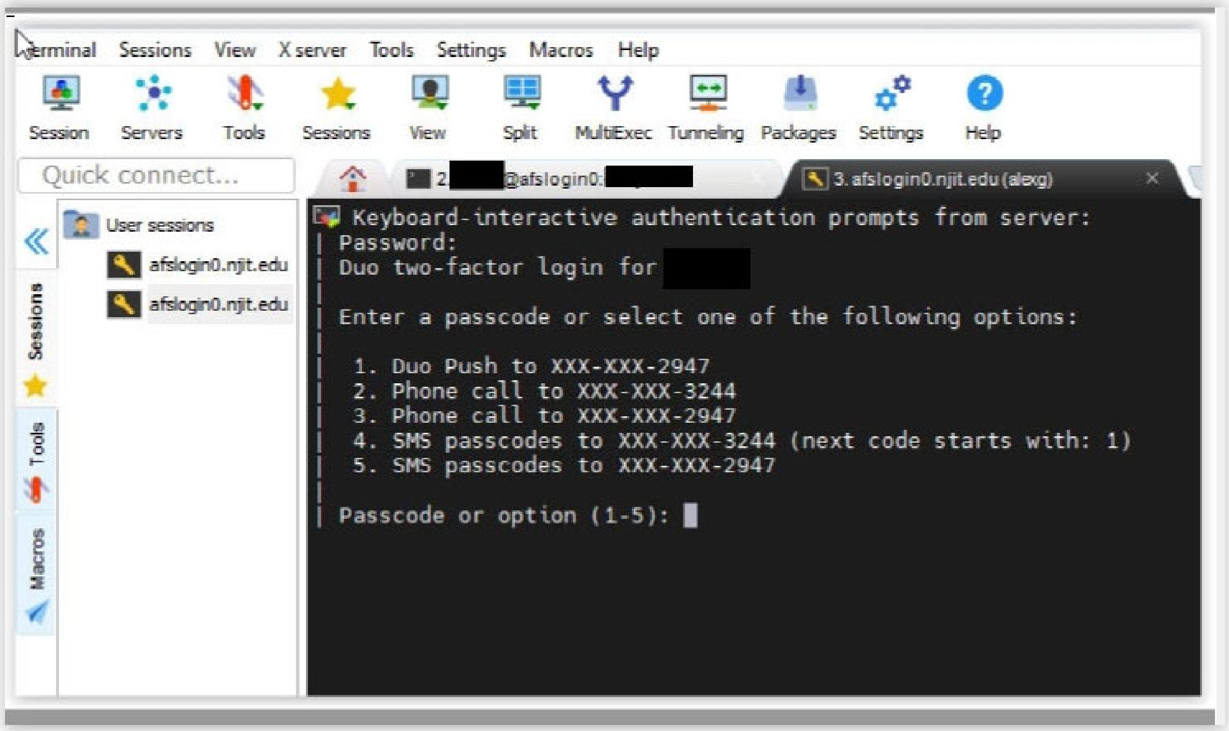
The following session appears as shown in Figure 5-4. You are prompted for your own myUCID pasword. (You have already typed in your login in the Specify username field earlier in the previous window). If you had left empty the Specify username field you would be prompted here first for the login name and then for the password. In our simpler case (no login name to specify) we type in the myUCIDpassword to the right of the Password: prompt starting with the current position of the white rectangular cursor and do not forget to press the ENTER button of the keyboard when you are done typing in your password. Note that if you mistyped a character, correcting it with a Backspace might now work. Let the system detect the error and re-prompt you with a Password:



```
Keyboard-interactive authentication prompts from server:
| Password:
```

Figure 5-4.

5. The follow-up activity after correctly typing your password is shown in Figure 5-5. If you typed it incorrectly the Password: prompt will reappear following possibly an appropriate error message. A DUO two-factor authentication request pops up. Deal with it as needed.


Terminal Sessions View X server Tools Settings Macros Help
Session Servers Tools Sessions View Split MultiExec Tunneling Packages Settings Help
Quick connect...
2.[redacted]@afslogin0:[redacted]
3. afslogin0.njit.edu (alexg)
User sessions
afslogin0.njit.edu
afslogin0.njit.edu
Sessions
Tools
Macros

```
Keyboard-interactive authentication prompts from server:
| Password:
| Duo two-factor login for [redacted]
|
| Enter a passcode or select one of the following options:
|
|  1. Duo Push to XXX-XXX-2947
|  2. Phone call to XXX-XXX-3244
|  3. Phone call to XXX-XXX-2947
|  4. SMS passcodes to XXX-XXX-3244 (next code starts with: 1)
|  5. SMS passcodes to XXX-XXX-2947
|
| Passcode or option (1-5):
```

Figure 5-5.

ALERT!

If you ignored the step indicated in Figure 5-3(b) and you did not choose scp for the browse window then it IS POSSIBLE that the default choice is sftp (transfer of files from the local to the remote machine or the other way around) . This choice can prove problematic so pay to attention to different alerts you encounter.

If sftp is to be used or desired by (because you ignored the Figure 5-3(b) setup) you then make sure you retype your myUCID password and redo 2FA on the following two windows that will pop-up then. Otherwise just ignore Fig 5-5(a) and Fig 5-5(b) (dismiss them).

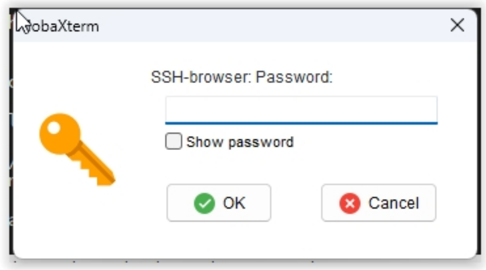


Fig 5-5(a) : one more type the myUCID password

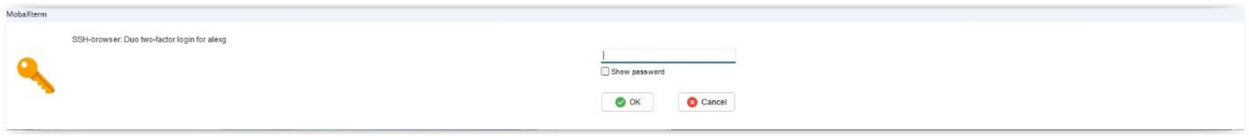
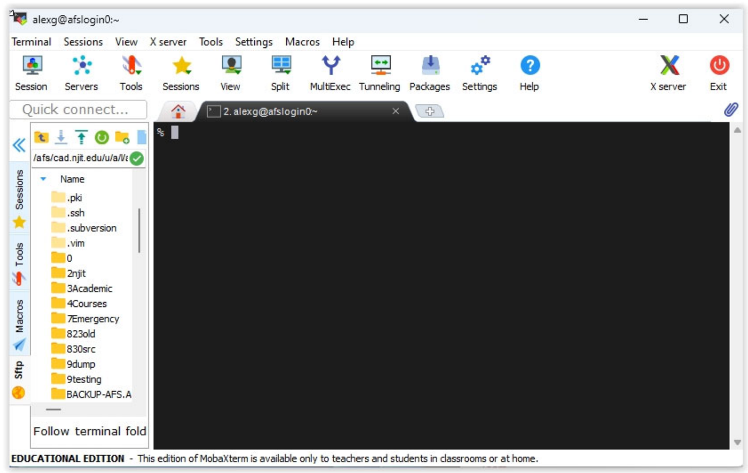


Fig 5-5(b): one more Duo 2FA

6. Note that some of this information in this step is repetition and available in section 4 step 6 - step 7. At the completion of Step 5 a secure shell connection is then established from your Windows laptop (the client local machine running Mobaxterm) to the remote Linux machine (the remote machine that will be referred to as a host, server, or remote machine) that runs a shell program named bash (for Bourne again shell) on top of the Linux operating system.


alexg@afslogin0:~
Terminal Sessions View X server Tools Settings Macros Help
Session Servers Tools Sessions View Split MultiExec Tunneling Packages Settings Help X server Exit
Quick connect...
2. alexg@afslogin0~
/afs/cad.njit.edu/u/a/
Name
.pki
.ssh
.subversion
.vim
0
2njit
3Academic
4Courses
7Emergency
823old
830src
9dump
9testing
BACKUP-AFS.A
Follow terminal fold
EDUCATIONAL EDITION - This edition of MobaXterm is available only to teachers and students in classrooms or at home.

```
%
```

Figure 5-6(a)

The layout area of Mobaxterm contains a blackened area to the right where we type commands and the shell (i.e. OS) generates output or provides error messages and the file browser white-ish left area that indicates the files located in the login (home) directory initially or other directory associated with the blackened area. In the blacked area we see a prompt for this session (user dependent) that is currently the percent sign and after a space next to it a cursor in the form of a white rectangle. This is where we type commands and we do not forget at the end to press the Enter key (button)!

The shell program is a program that runs on the remote machine (it is run on the user's session and auto activated by the user's login activity) and interacts with the OS system (Linux) of the remote machine: it accepts commands from a USER who is running the bash shell, interprets these commands, and requests the OS to execute a program related to the input command and display the output, if any, on the same window. **This is the blackened area of Figure 5-6(a).** The user is typing a command in that area next to the apparent prompt % and the rectangular white CURSOR (this is MY user defined prompt, yours or the default ONE might be different), and after the user pressing the ENTER key, the bash shell is interpreting user input and is communicating with the OS, with the OS in turn executing (running) a program (or programs) relevant to the execution of the command . The program's name is usually the command's name indicated. Do not forget every time you type in a command, complete the command by pressing the ENTER button/key. ENTER tells the bash shell that the input is over and bash should start interpreting it and acecuting it through the OS. **You will be typing commands in the CLIENT windows machine's MobaXterm but your commands will be interpreted and executed by the REMOTE AFS machine (bash and OS respectively) and the output on the REMOTE AFS machine will be displayed on the CLIENT windows machine's MobaXterm window (black area of Figure 5-5 before or 5-6 or 5-6(a) to follow).** This back and forth relay of information is realized by the ssh client of Mobaxterm and a ssh server running on the remote host. Figure 5-6 below indicates this.

The full view MobaXterm window might look like the oen in Figure 5-6(a). The layout depends on the default options of your installation or the choices you made afterwards.   You may be able to expand the left area by pointing your mouse cursor to the thick vertical bars surrounding the login area. Then the window will include not only your login area (blackened window) but some additional information as indicated in Figure 5-6(a). You might explore other options of MobaXterm. Moreover it is possible to upload files (from the client/local machine to the remote host/machine) or download files (from the remote host/machine to the local/client machine). Familiarize with mobaxterm (read its online manual/help).

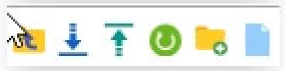


Figure 5-6(b)   :   Navigation area

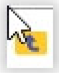


Figure 5-6(c)   : File/Filesystem Navigation

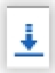


Figure 5-6(d)   : Download blue button

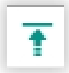


Figure 5-6(e)   : Upload green button

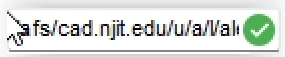
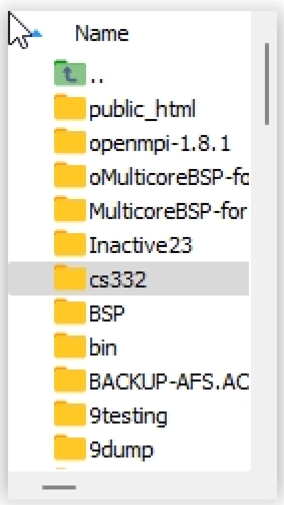


Figure 5-6(f)   : AFS directory info(name)



```
Name
..
public_html
openmpi-1.8.1
oMulticoreBSP-fc
MulticoreBSP-for
Inactive23
cs332
BSP
bin
BACKUP-AFS.AC
9testing
9dump
```

Figure 5-6(g)   : AFS home (login) directory and some of its files

The blackened area of Figure 5-6(a) is FOR MY configuration the remote machine. The secondary area on the left shows information about my file area on the remote machine's filesystem. As soon as ssh established a connection to the remote machine, the remote machine mounted my files on the file server that is supported by the remote machine. The file server supports a distributed file system known as AFS (Andew File System). My files are available at a default location in the filesystem; the location is known as the home directory or login directory. My home directory ends with my UCID.

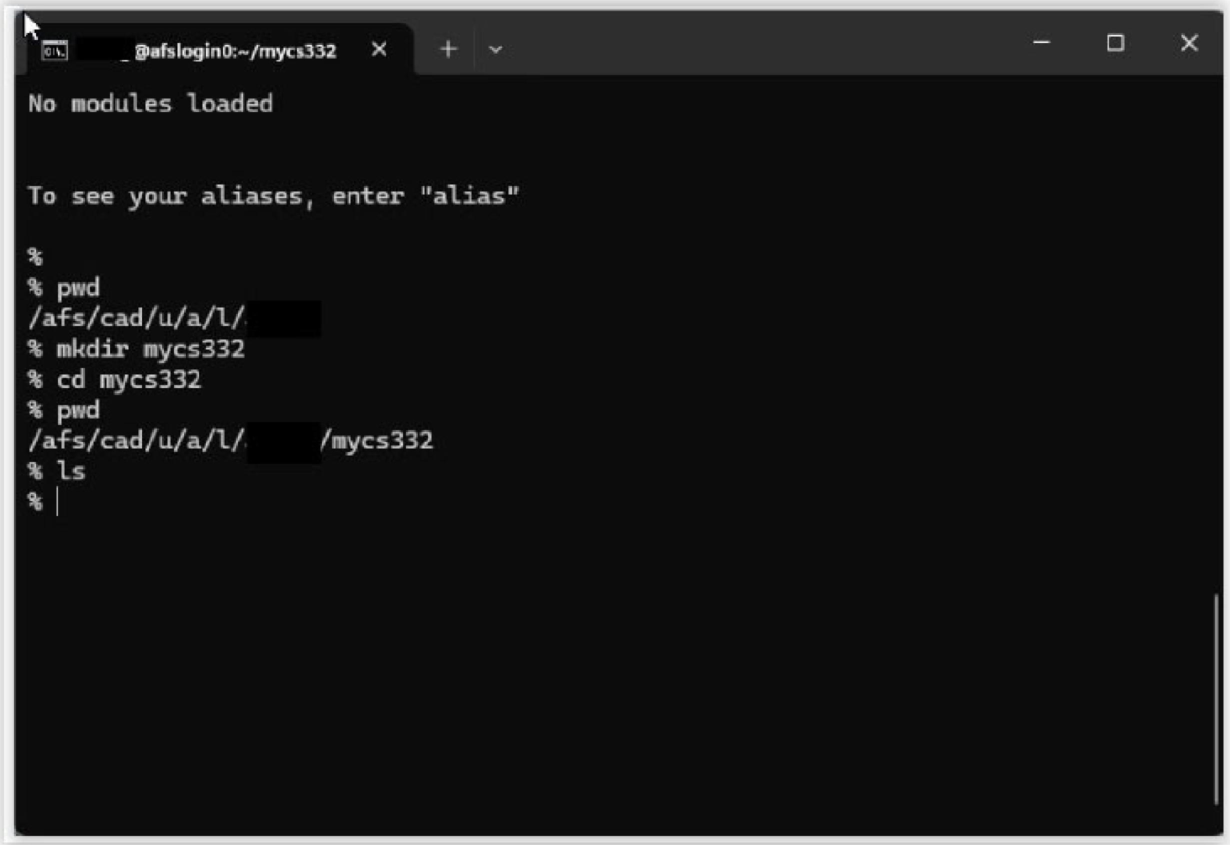
So Let us say for the sake of an example that myUCID is almn12.Beneath a sequence of icons and on the left side of a green checkmark you may observe a path name that maps to my home directory

/afs/cad.njit.edu/u/a/l/almn12

(several times it appears as /afs/cad/u/a/l/almn12)

The part of the directory /a/l/ is derived from the first two letters of myUCID. For you the part /a/l/almn12 would definitely be different.

The actions depicted in Figure 5-6 below are described for the identical window of Figure 4-6 there in Steps 6 -7 . If you have already created a directory mycs332 you cannot recreate it!



```
No modules loaded


To see your aliases, enter "alias"

%
% pwd
/afs/cad/u/a/l/[redacted]
% mkdir mycs332
% cd mycs332
% pwd
/afs/cad/u/a/l/[redacted]/mycs332
% ls
% |
```

Figure 5-6

Type in pwd i.e. see below (% is displayed you only type pwd and afterwards you press Enter)

```
%       pwd
```

This will show you your LOGIN directory on the AFS file system. The prefix you will see is going to be /afs/cad/a/l/almn12 where a and l are supposed to be the first two letter of they hypothetical almn12 UCID. At the end of the output you should recognize your myUCID login name. BTW pwd is an acronym for print working directory. Your working directory is you login directory because you have not moved out of it!

Type in ls (output not shown in Figure 5-6) i.e.

```
% ls
```

This will present you all the files created by you or by default by the system at your LOGIN directory. Confirm that there is no file of any type (a directory is a file of type directory, and we call it directory rather than folder in Linux) named mycs332. Command ls is an acronym for listing the contents of the current directory (i.e. home directory also known as the login directory). If you specify an argument to ls and it is a directory, that directory's contents would be displayed.

Type in mkdir mycs332 (see also Figure 5-6) i.e.

```
% mkdir mycs332
```

Command mkdir is an acronym for make directory.

Type then cd mycs332 (Figure 5-6) i.e.

```
% cd mycs332
```

Command cd is an acronym for change directory to the one indicated in its arguments. You may present a full (path) directory name such as mycs332 if that directory is inside your current directory or a full path such as

```
% cd /afs/cad/u/a/l/almn12/mycs332
```

if the first two letters of you myUCID are a and l and your myUCID is almn12!

Type then pwd (Figure 5-6) i.e.

```
% pwd
```

Its output would confirm that you have moved to the designated directory just created.

7. If directory mycs332 in your login (home) directory already exists you do not need to create it per actions of Figure 5-6. You just type

```
% cd mycs332
```

in order to get into (i.e. change directory from the current home/login directory to directory mycs332) it and then type ls. If you are in directory mycs332 and you type

```
% ls
```

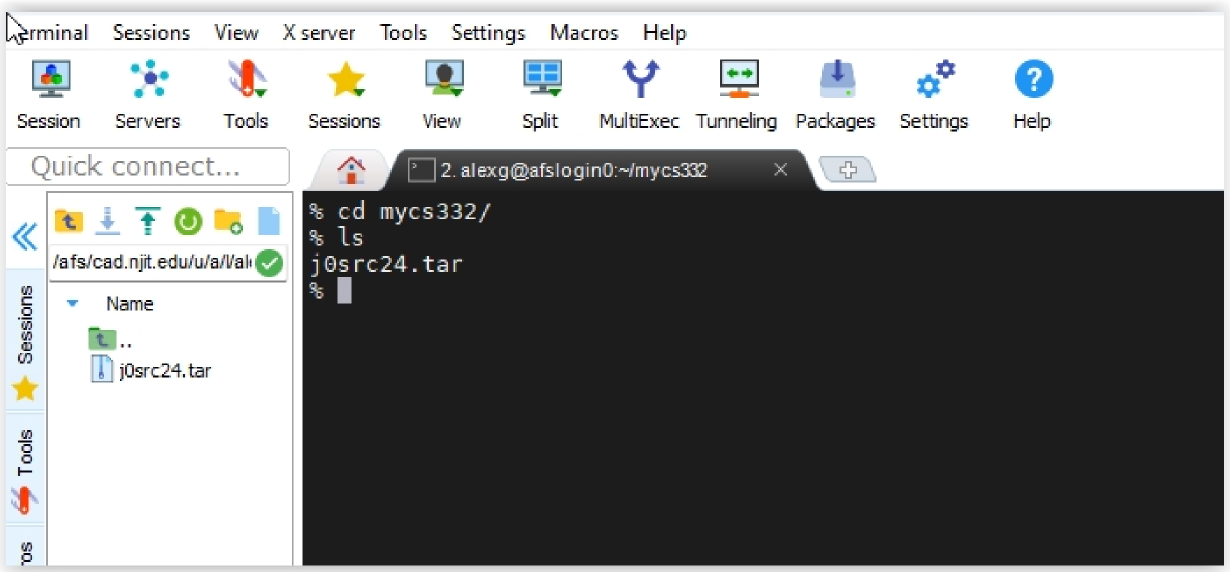
or the equivalent (. i.e. a period is a synonym to current directory)

```
% ls .
```

you will see the contents of the current directory that has become mycs332 or full path-wise

/afs/cad/u/a/l/almn12/mycs332

as indicated earlier.



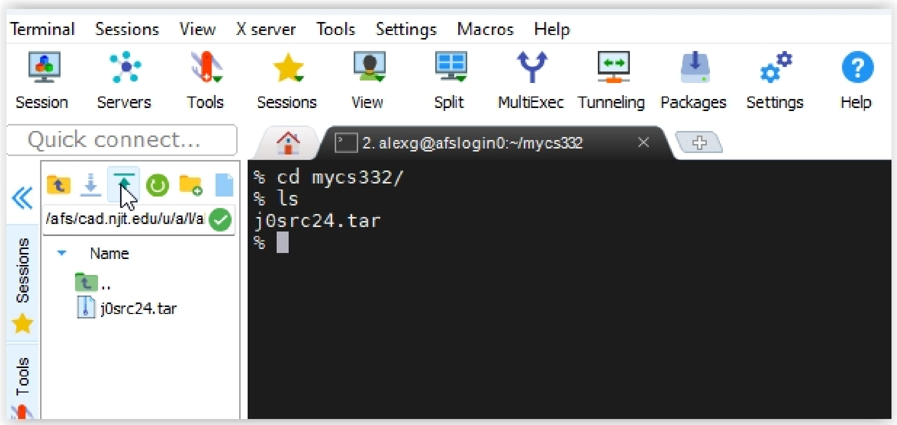
```
% cd mycs332/
% ls
j0src24.tar
%
```

Figure 5-7.

Furthermore, on the left area panel earlier identified in Figure 5-9(e) the info there reflects my current directory than my original home directory.

8. One of the icons,

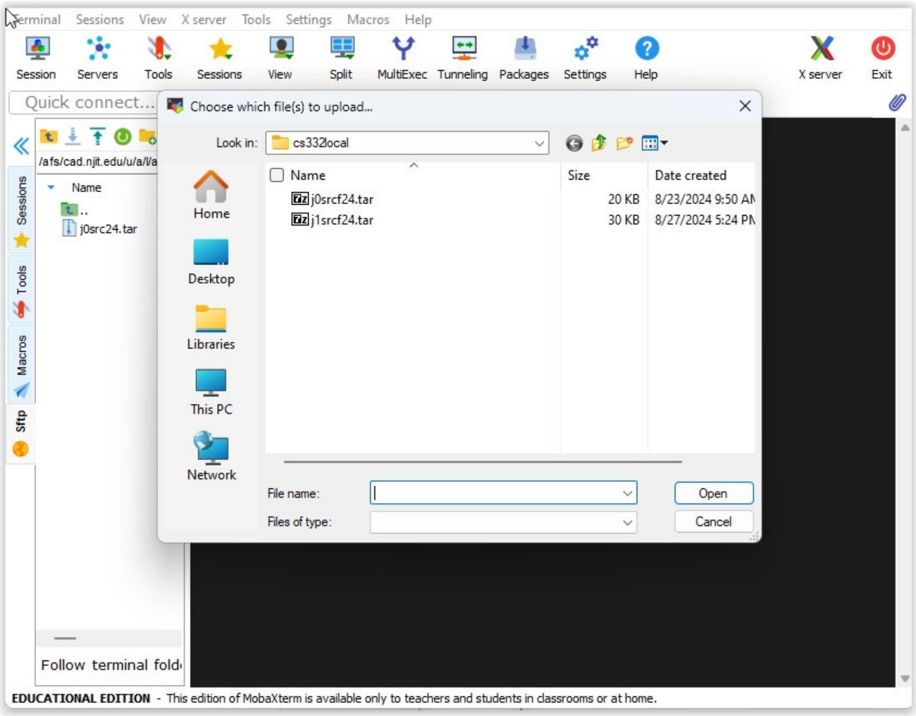
   is the upload icon to upload a file from the client (Windows) machine to the remote machine (Linux). If you navigate the mouse button on it as shown in Figure 5-8(a)


Terminal Sessions View X server Tools Settings Macros Help
Session Servers Tools Sessions View Split MultiExec Tunneling Packages Settings Help
Quick connect...
2. alexg@afslogin0:~/mycs332
/afs/cad.njit.edu/u/a/l/a
Name
..
j0src24.tar
Sessions
Tools

```
% cd mycs332/
% ls
j0src24.tar
%
```

Figure 5-8(a)

you may click on the icon with the mouse and an upload window pops up such as the one of Figure 5-8(b). The directory on the local (Windows) machine is the same as use previously in section 3 Step 6-7 when we uploaded j0srcf24.tar to the remote (Linux) machine. This time we might decide to upload the other tar file.

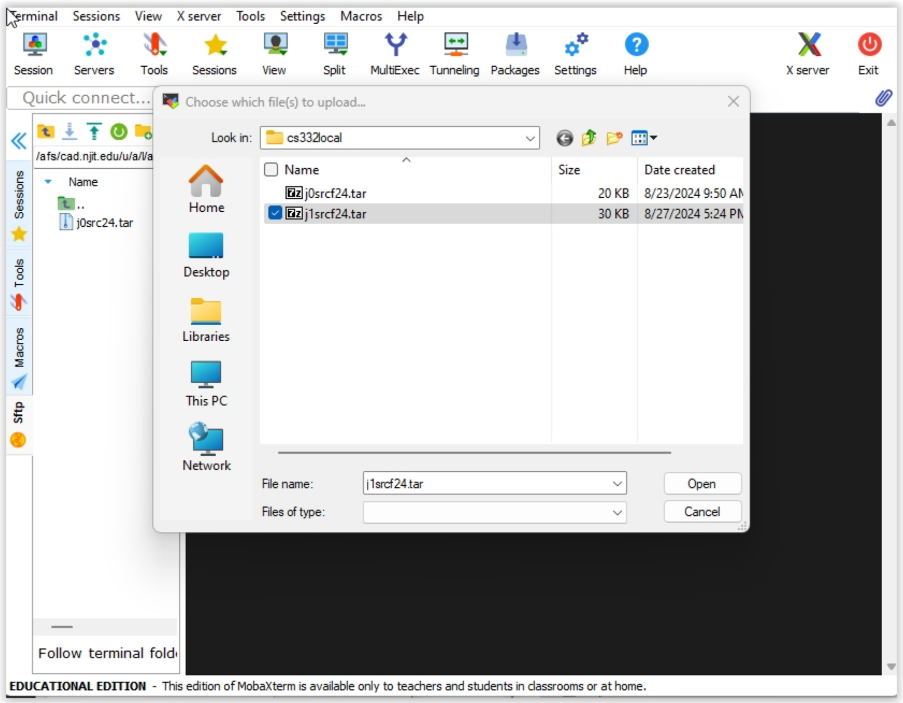

Terminal Sessions View X server Tools Settings Macros Help
Session Servers Tools Sessions View Split MultiExec Tunneling Packages Settings Help X server Exit
Quick connect...
/afs/cad.njit.edu/u/a/l/a
Name
..
j0src24.tar
Sessions
Tools
Macros
Sftp
Follow terminal fold
Choose which file(s) to upload...
Look in: cs332local
Home
Desktop
Libraries
This PC
Network
File name:
Files of type:
Open
Cancel

Name	Size	Date created
j0srcf24.tar	20 KB	8/23/2024 9:50 AM
j1srcf24.tar	30 KB	8/27/2024 5:24 PM

EDUCATIONAL EDITION - This edition of MobaXterm is available only to teachers and students in classrooms or at home.

Figure 5-8(b)

In Figure 5-8 (c) we choose the other tar file and then by clicking open the uploading process would commence


Terminal Sessions View X server Tools Settings Macros Help
Session Servers Tools Sessions View Split MultiExec Tunneling Packages Settings Help X server Exit
Quick connect...
Sessions Tools Macros Sftp
/afs/cad.njit.edu/u/a/l/a
Name
j0src24.tar
Follow terminal fold
Choose which file(s) to upload...
Look in: cs332local
Home
Desktop
Libraries
This PC
Network
Open
Cancel

Name	Size	Date created
j0srcf24.tar	20 KB	8/23/2024 9:50 AM
✓ j1srcf24.tar	30 KB	8/27/2024 5:24 PM

File name: j1srcf24.tar
Files of type:

EDUCATIONAL EDITION - This edition of MobaXterm is available only to teachers and students in classrooms or at home.

Figure 5-8(c)

ALERT

If you ignored the step higlighted in Figure 5-3(b) and you are using sftp browsing rather than scp two windows may pop-up requiring possibly authentication and definitely a Duo 2FA . These are Figure 5-5(a) and Figure 5-5(b) shown below as Figure 5-8(d) and Figure 5-8(e)

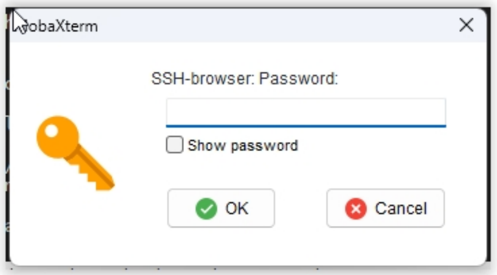


Figure 5-8(d) i.e. 5-5(a)

Figure 5-8(e) i.e. 5-5(b)

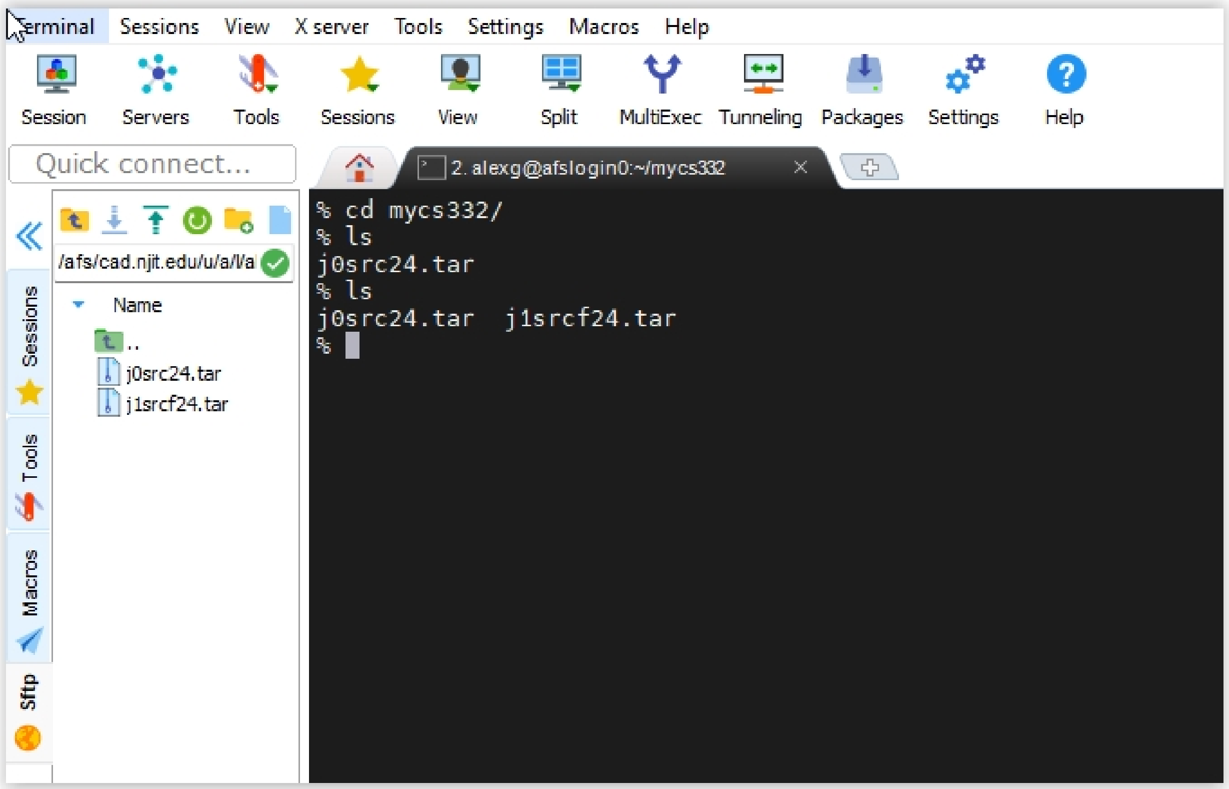
If you have enabled scp in Figure 5-3(b) those two windows (most likely as of this writing) will NOT APPEAR.

Then after a nervous pause of 10-15s, the upload would start and complete successfully.

You know things worked fine when as indicated in Figure 5-9 below both an

% ls

in the black working area confirms the coexistence of the two tar files as indicated, or you observe in the vertical white file area on the left of the working area the presence of the same two tar files in the current (working) directory mycs332!
A double verification confirms and reconfirms the successful uploading.

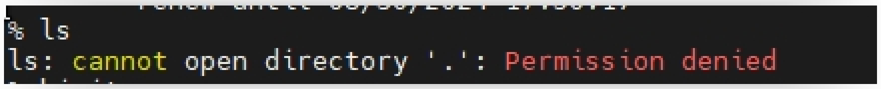


```
% cd mycs332/
% ls
j0src24.tar
% ls
j0src24.tar  j1srcf24.tar
%
```

ALERT and WARNING :

If you see the message **"sftp error #3"** you should realize the importance of enabling the action of Figure 5-3(b)

If you ever see the message "ls : cannot open directory " and "Permission denied" as in (see below)



```
% ls
ls: cannot open directory '.': Permission denied
```

do the following: type in a shell prompt in the blackened area the command

aklog i.e

% aklog

At this point you are ready to start interacting with the remote host. Type in Unix/Linux commands, and when done typing them, press the ENTER button and observe the output.
If you are not familiar with Unix or Linux there are several tutorials or summaries out there. Pick the one you are more comfortable with it.

Finally, to terminate the session type exit. (Sometimes, quit or logout might also work.) You will 'logout' from the remote host.

If you are done with MobaXterm, terminate it by clicking on its top right corner the X symbol that will 'kill' the window. Likewise locate the icon of Figure 4-4 and stop VPN. The icon of Figure 4-4 will revert to its form in Figure 4-1 without a lock on it. At that point if you rightclick on Figure 4-1 you will be able to deactivate VPN. Deactiving VPN is NOT EQUIVALENT to an uninstall.

PART B

Limited UNIX command overview

6. Limited Unix command overview

In a Unix system, a user is logged on to the system by providing a set of credentials (login name and an associated password), a process already familiar to you from the earlier sections of this document. At NJIT we call the credentials MyUCID credentials consisting of a myUCID (login name) and a myUCID password.

Immediately after the login has been completed successfully the Unix environment would become available to the logged on user and a program (process) would start executing in the user's environment after the user's login: the Unix shell process.

The UNIX shell process would allow the user to interact with the operating system and start, stop, suspend and resume the execution of services provided by the OS or create and manage user created processes. These services are programs and when run they become processes. The definition of a process is 'a program in execution'. All interaction is done through the terminal and its associated keyboard that was used by the user to gain access to the system: the user types in commands to the shell and after the shell interprets these commands, it invokes services of the OS to execute/realize those user commands as needed and as privileged. The OS might decline to execute some of these commands for safety or security reasons based on the credentials (privileges) of the user.

To keep interaction short UNIX commands to the shell are short and sometimes intuitive. For example command **ls** is short for list, **cat** for catenate (list contents), **mkdir** for make directory, **ps** for process status, **cd** for change directory, **pwd** for print working directoy, **mv** for move, and **cp** for copy.

Moreover, the UNIX shell provides command line editing, and history of interactions with the shell thus allowing for editing a previously lengthy command instead of retyping it before re-execution or repeating a frequently executed command by easily recalling it from a history of prior interactions.

Every command of the shell such as ls, ps, mv, cp, pwd, cat, cd, is an executable program residing in secondary memory (disk). It was originally written in C and compiled and assembled subsequently into the executable file named ls, ps, etc. Thus typing a command such as

```
%  ls
```

would cause the shell (an OS process) to load the program named ls from secondary memory into main memory, thus turning it into a process and then executing it as needed.

A reminder: the % is the shell prompt. You do not type it. It is output by the shell to remind you that 'I, the shell, have your full attention: please type in your request'. Moreover when you do so and type your request (ls in this case) do not forget to tell the shell that you are done when you are done typing your command. You do so by pressing the ENTER key of the keyboard at the end.

At that moment the shell interprets your input (in the example above it is an ls), the text between the prompt and the ENTER, and executes it as needed. Every execution of a process in linux (in this case ls) by default creates and interacts with three files associated and connected with three devices:

(a) standard input also known to the user as the file with fd (file descriptor) 0,

(b) standard output with fd equal to 1, and

(c) standard error with fd equal to 2.

Unless the shell is instructed otherwise by you, standard input is associated with your terminal's keyboard, and standard output and standard error are both associated with the terminal's screen.

Moreover in Linux, multiple commands can be executed one after the other in the command line. Thus

```
% ls ; date ; echo "Hi"
```

Thus in the above example after ls is executed, the current date and time is printed, and afterwards a message gets printed on the standard output, the terminal used by the user.

Moreover in Linux, multiple commands can interact with each other with a mechanism known as an unnamed pipe. A pipe is a FIFO (First In First Out) queue that accepts input from the output of one command and generates output that will become the input of another command. Thus

```
% ls | egrep filename
```

consists of the command ls that prints the contents of the current directory (this description makes sense after you read the next section if you are not familiar with any operating system's structure) and directs this output to the unnamed pipe indicated by the pipe | symbol. The unnamed pipe indicated receives as input the transmitted by ls output and then it generates its own output that is to become the input of the command egrep. The command egrep filters its input by discarding all lines that do not contain the string/word filename and thus preserving to the output that it will generate the lines that contain the string filename. The combined execution thus prints the output lines of ls that contain the string filename.

Pipes can allow multiple cascade communication such as the following one.

```
% ls -l |  egrep filename | sort  | less
```

A list of useful commands

```
% pwd          #print current working directory

% cd path      # change the current working directory to path

% cd           # if you are lost go to (i.e. cd) to home
directory

% cd /         # goto the root of the filesystem

% man pwd      # manual page for command pwd

% man cd       # manual page for cd

% whoami

% who

% date

% hostname

% cal          # calendar; use also eg. cal 5 2023

% stat filename

% ls path      # list contents of directory path

% ls filename  # confirm whether filename exists

% ls           # list contents of current working directory
% ls .          # same as ls; . alias for current directory

% ls ..         # list contents of parent directory

% ls –l         # equivalent to ls -l .

% ls -l filename

% ls -l directory        %ls -l path

% ls –a         # Hidden files . and .. included

% ls –lai       # the inode (numeric name of the file) is also
listed on the extrement left.

% ls -lR        # R for recursive ; r means reverse!

% ls file*      # list all files starting with file

% ls *file      # list all files ending with file
```

The characters after a dash are option of the command (eg. ls)

```
% rm file       # delete a file
```

or

```
% rm pathTofile
```

but

```
% rmdir directory        # remove an empty directory

% rmdir pathTodirectory  # remove an empty directory

% mkdir newdirectory     # create a new directory

% rm –rf file       # A pretty dangerous command... Avoid it.

% rm –rf directory  # A pretty dangerous command... Avoid it.

% mkdir directory   # Create a directory

% mv oldname newname  # rename file

% mv olddir  newdir   # rename a directory (newdir must not
preexist)

% cp file1 file2      # make a copy of file file1


% cat filename            #list contents of filename
% more filename          #similar to less
% less filename          # controlled flow version of cat
# head filename
# head -10 filename      # print 10 first lines of filename
% tail -10 filename      # last 10 lines
% egrep me file          # search for me in file
% echo string            # string on screen
% wc filename            # num of chars, lines, words in file
% ps                     # process status
% ps -ef |egrep myID
% sleep 10               # screen freezes for 10sec
```