CMPE-380: Applied Programming

Laboratory Exercise 01

Connect to Linux Systems, Review Linux Commands

Table of Contents

Connect to Linux Systems, Review Linux Commands

Pre-Lab – 5 pts

Bring the laptop computer you plan to use this semester to the lab, if you have one.

Interactive Exercises – 35 pts

This exercise introduces the tools and terminals used to connect to the Linux systems at RIT and reviews the basic Linux command set which will be used throughout the semester. You are expected to finish the interactive exercises in lab and remain in lab and start any assignment work if time allows.

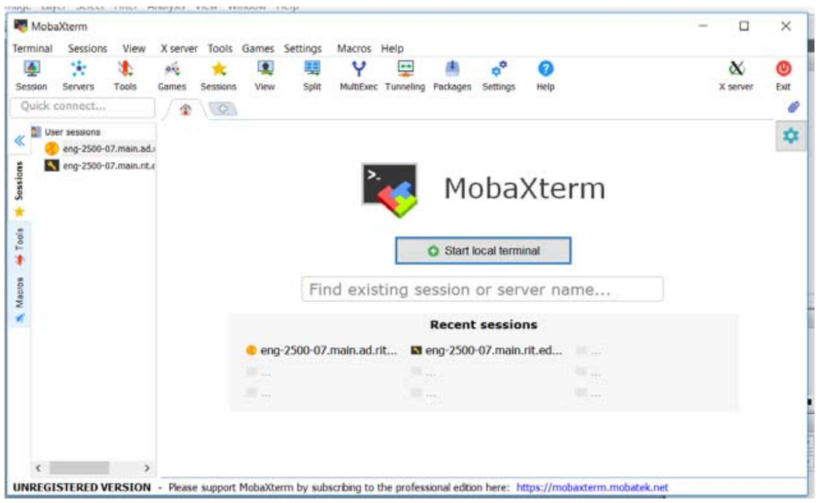
You must turn in the signed grading sheet at the end of lab to receive credit!

Connect to Linux Systems

The standard way to connect to Linux servers is via Secure Shell (SSH), Secure Copy Command (SCP) and Secure File Transfer (SFTP). SSH tools open a text window into a Linux server and allow you to enter operating system commands while SCP and SFTP provide a mechanism to transfer files. SSH, SCP and SFTP are available on all Linux systems and on Mac, which is really a Linux variant. Windows users must install 3rd party SSH, SCP and SFTP communication tools to access Linux systems. Many people will also install 3rd party SSH and SFTP tool on Mac and Linux systems to enable drag and drop features.

Windows Users

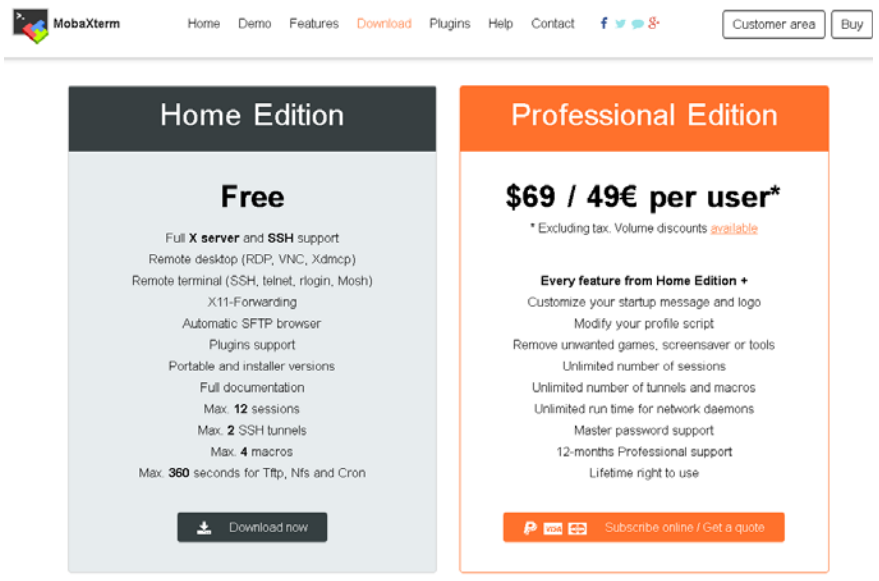
Windows users should use MobaXterm, an enhanced terminal for Windows with a X11 server, tabbed SSH client, and network tools. More information about MobaXterm can be found at http://mobaxterm.mobatek.net/download.html.


MobaXterm
Terminal Sessions View X server Tools Games Settings Macros Help
Session Servers Tools Games Sessions View Split MultiExec Tunneling Packages Settings Help X server Exit
Quick connect...
User sessions
eng-2500-07.main.ad.
eng-2500-07.main.rit.
Sessions Tools Macros
MobaXterm
Start local terminal
Find existing session or server name...
Recent sessions
eng-2500-07.main.ad.rit...
eng-2500-07.main.rit.ed...

UNREGISTERED VERSION - Please support MobaXterm by subscribing to the professional edition here: https://mobaxterm.mobatek.net

MobaXterm Installation

To download MobaXterm please visit https://mobaxterm.mobatek.net/download.html and choose the free version.



Home Edition

Free

Full **X server** and **SSH** support
Remote desktop (RDP, VNC, Xdmcp)
Remote terminal (SSH, telnet, rlogin, Mosh)
X11-Forwarding
Automatic SFTP browser
Plugins support
Portable and installer versions
Full documentation
Max. **12** sessions
Max. **2** SSH tunnels
Max. **4** macros
Max. **360** seconds for Tftp, Nfs and Cron

Download now

Professional Edition

$69 / 49€ per user*

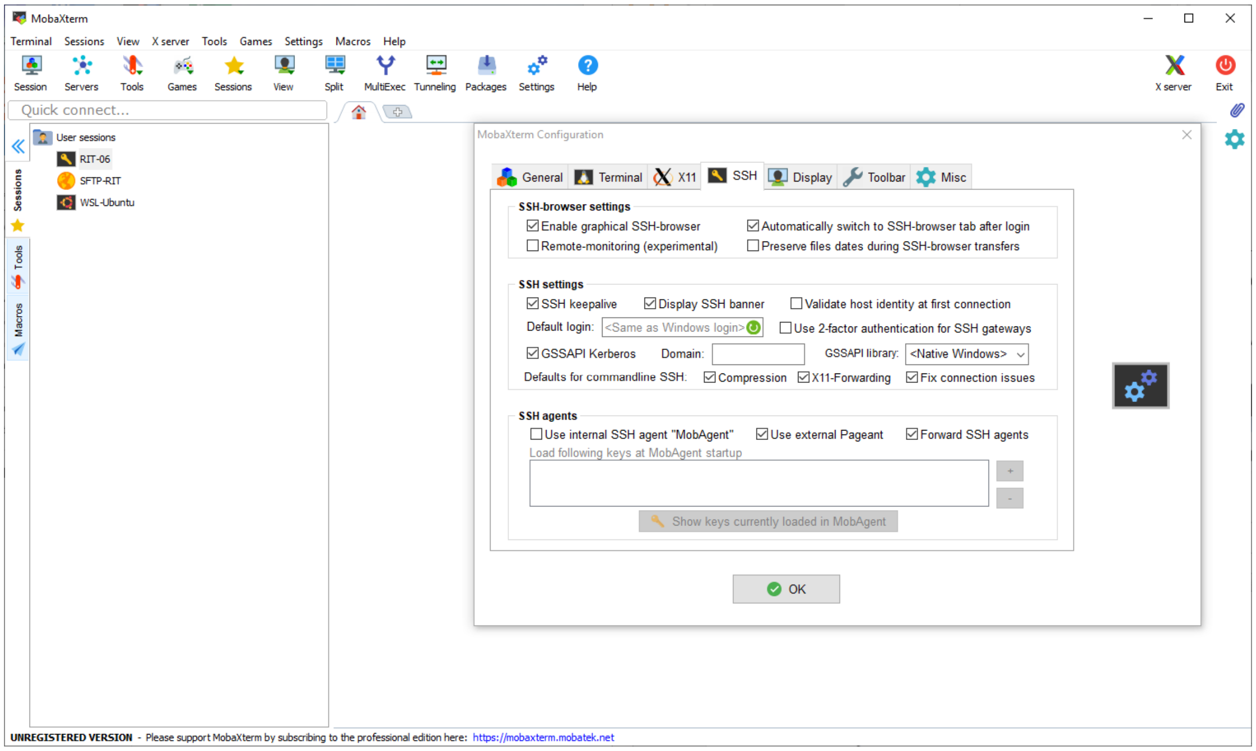
* Excluding tax. Volume discounts available

Every feature from Home Edition +
Customize your startup message and logo
Modify your profile script
Remove unwanted games, screensaver or tools
Unlimited number of sessions
Unlimited number of tunnels and macros
Unlimited run time for network daemons
Master password support
12-months Professional support
Lifetime right to use

Subscribe online / Get a quote

SSH Connection

After you download and install MobaXterm, start the program and select "Settings", then "Configuration" and then the "SSH" tab.   Check the "SSH keepalive" box, select "OK" to save your settings.


MobaXterm
Terminal Sessions View X server Tools Games Settings Macros Help
Session Servers Tools Games Sessions View Split MultiExec Tunneling Packages Settings Help
X server Exit
Quick connect...
User sessions
RIT-06
SFTP-RIT
WSL-Ubuntu
Sessions Tools Macros
MobaXterm Configuration
General Terminal X11 SSH Display Toolbar Misc

SSH-browser settings
- ☑ Enable graphical SSH-browser
- ☑ Automatically switch to SSH-browser tab after login
- ☐ Remote-monitoring (experimental)
- ☐ Preserve files dates during SSH-browser transfers

SSH settings
- ☑ SSH keepalive
- ☑ Display SSH banner
- ☐ Validate host identity at first connection
- Default login: <Same as Windows login>
- ☐ Use 2-factor authentication for SSH gateways
- ☑ GSSAPI Kerberos
- Domain:
- GSSAPI library: <Native Windows>
- Defaults for commandline SSH: ☑ Compression ☑ X11-Forwarding ☑ Fix connection issues

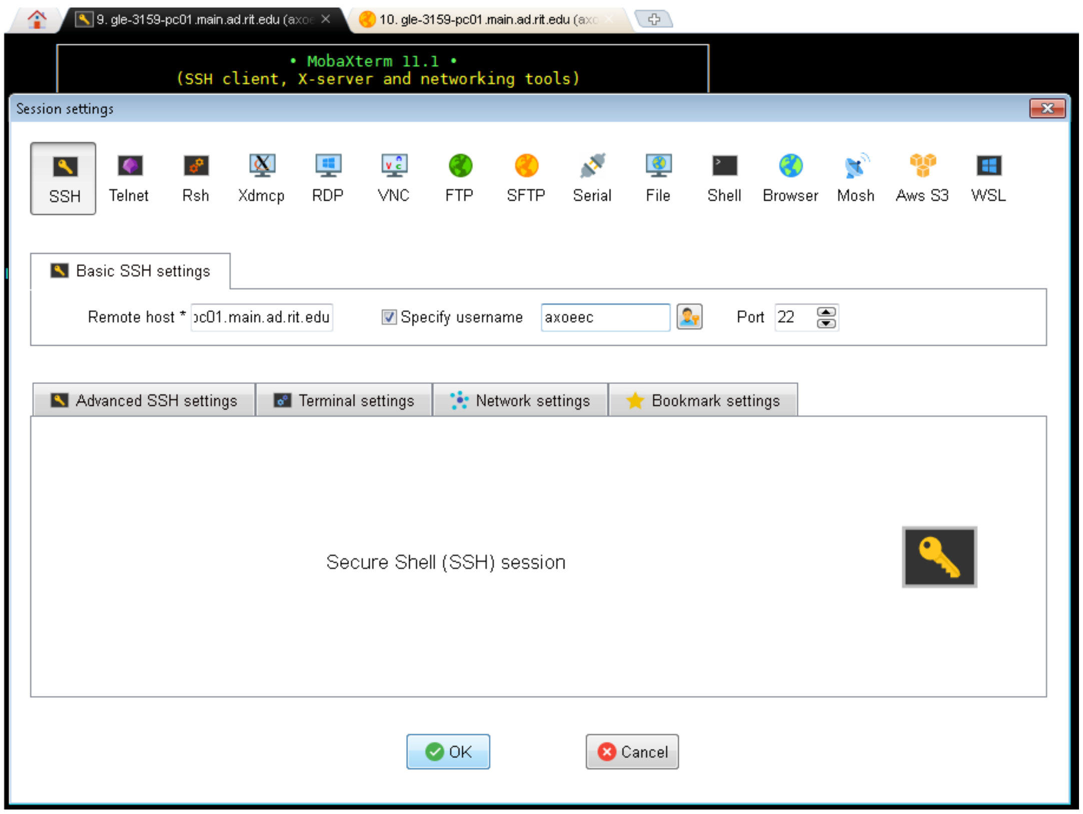
SSH agents
- ☐ Use internal SSH agent "MobAgent"
- ☑ Use external Pageant
- ☑ Forward SSH agents
- Load following keys at MobAgent startup

Show keys currently loaded in MobAgent

OK

UNREGISTERED VERSION - Please support MobaXterm by subscribing to the professional edition here: https://mobaxterm.mobatek.net

Next, click on the Session menu (at top left) and establish an SSH connection to your Linux machine in the lab. You may use any of the VLSI-lab machines (eng-2500-06.main.ad.rit.edu to eng-2500-30.main.ad.rit.edu, e.g., eng-2500-09.main.ad.rit.edu). Please check the "Specify username" check box, enter your username, and leave the default port number 22. If you are logged in with a RIT account, you should be logging in when you click on the OK button.



```
• MobaXterm 11.1 •
(SSH client, X-server and networking tools)
```

Session settings

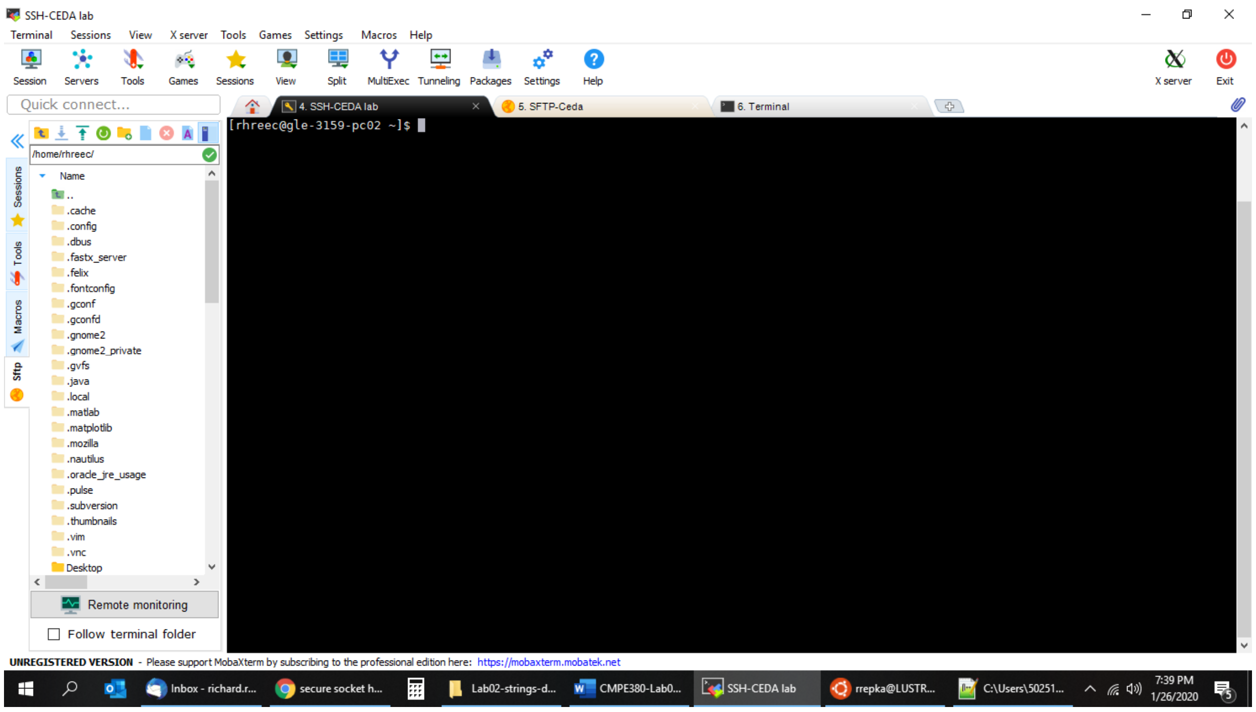
Remote host * ɔc01.main.ad.rit.edu

[✓] Specify username axoeec

Port 22

Note: If the Linux "won't take your password" verify that you entered the correct user name in the SSH settings box.

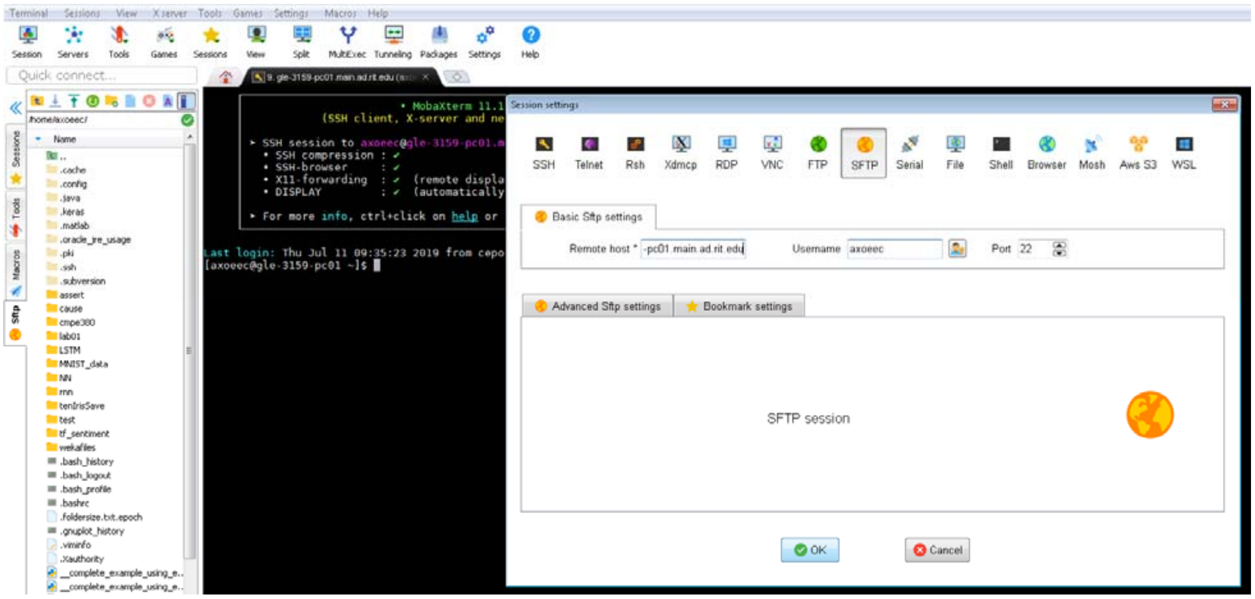
After you log in, the left side of your terminal session will contain your login id (e.g rheec) and your machine name. You can use these prompts to manage multiple sessions to multiple machines.



```
[rhreec@gle-3159-pc02 ~]$
```

SFTP Connection

Once you establish an SSH connection successfully, click on the Session menu to set up an SFTP connection as shown, in order to transfer your files between your PC and the Linux system. Please note that it does not matter which physical machine you connect to; you will see the same directory structure all the time.


Terminal Sessions View X server Tools Games Settings Macros Help
Session Servers Tools Games Sessions View Split MultiExec Tunneling Packages Settings Help
Quick connect...
9. gle-3159-pc01.main.ad.rit.edu
Session settings
SSH Telnet Rsh Xdmcp RDP VNC FTP SFTP Serial File Shell Browser Mosh Aws S3 WSL
Advanced Sftp settings
Bookmark settings
SFTP session
OK
Cancel

```
                    • MobaXterm 11.1
          (SSH client, X-server and ne

 ➤ SSH session to axoeec@gle-3159-pc01.m
   • SSH compression : ✔
   • SSH-browser     : ✔
   • X11-forwarding  : ✔  (remote displa
   • DISPLAY         : ✔  (automatically

 ➤ For more info, ctrl+click on help or

Last login: Thu Jul 11 09:35:23 2019 from cepo
[axoeec@gle-3159-pc01 ~]$ █
```

Basic Sftp settings

Remote host * -pc01.main.ad.rit.edu   Username axoeec   Port 22

/home/axoeec/

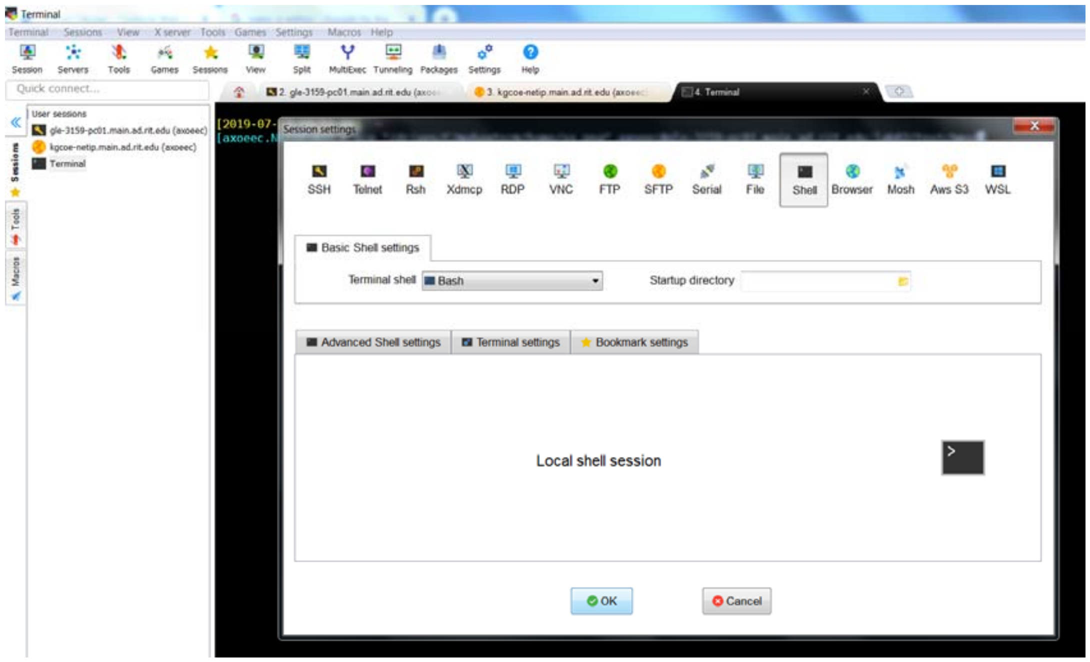
Name
..
.cache
.config
.java
.keras
.matlab
.oracle_jre_usage
.pki
.ssh
.subversion
assert
cause
cmpe380
lab01
LSTM
MNIST_data
NN
rnn
tenIrisSave
test
tf_sentiment
wekafiles
.bash_history
.bash_logout
.bash_profile
.bashrc
.foldersize.txt.epoch
.gnuplot_history
.viminfo
.Xauthority
__complete_example_using_e...
__complete_example_using_e...

To transfer files between your local computer and the Linux systems, just drag them.

Copying Files between Local and Server via SCP (Using Local Shell in MobaXterm)

Even though MobaXterm SFTP is the easiest way to drag and drop files between your local and remote system, you should experience the classical Secure Copy Command (SCP).

Open a **local** MobaXterm terminal shell as shown**Error! Reference source not found.**. Windows provides a Linux like local shell.



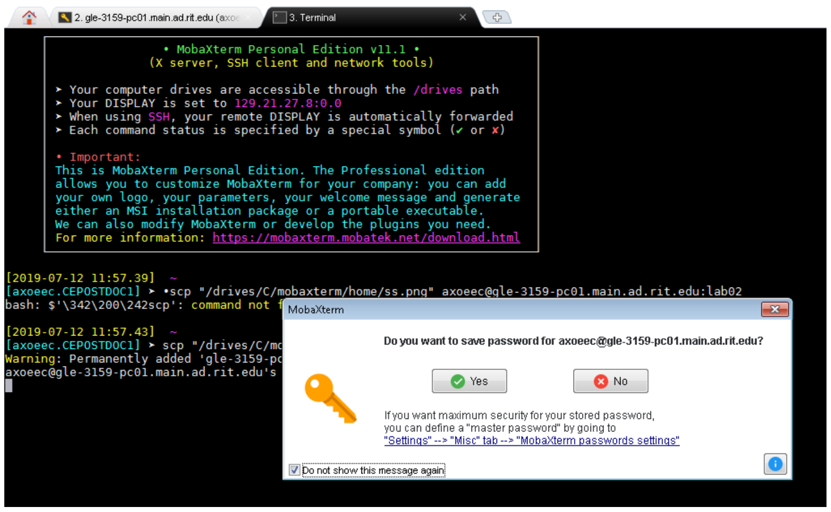
Terminal shell: Bash
Startup directory:
Local shell session

This local shell is a Bash Linux command line interface to your local PC. **If you look closely at the left-hand prompt and compare it to your SSH Linux prompt you will see the machine name is different**.

Please create a local file in the current directory by executing the following:

```
echo   "Hello World"   >   scptest.txt
```

We are going to use the standard SCP command to move individual files to/from our local PC to the Linux system. When you first execute a scp command, MobaXterm will ask you to enter your password for the remote system. Please enter your password and select either “Yes” or ”No” for the question “Do you want to save your password for…”



```
              • MobaXterm Personal Edition v11.1 •
            (X server, SSH client and network tools)

 ➤ Your computer drives are accessible through the /drives path
 ➤ Your DISPLAY is set to 129.21.27.8:0.0
 ➤ When using SSH, your remote DISPLAY is automatically forwarded
 ➤ Each command status is specified by a special symbol (✓ or ✗)

 • Important:
 This is MobaXterm Personal Edition. The Professional edition
 allows you to customize MobaXterm for your company: you can add
 your own logo, your parameters, your welcome message and generate
 either an MSI installation package or a portable executable.
 We can also modify MobaXterm or develop the plugins you need.
 For more information: https://mobaxterm.mobatek.net/download.html

[2019-07-12 11:57.39]  ~
[axoeec.CEPOSTDOC1] ➤ •scp "/drives/C/mobaxterm/home/ss.png" axoeec@gle-3159-pc01.main.ad.rit.edu:lab02
bash: $'\342\200\242scp': command not f

[2019-07-12 11:57.43]  ~
[axoeec.CEPOSTDOC1] ➤ scp "/drives/C/mo
Warning: Permanently added 'gle-3159-pc
axoeec@gle-3159-pc01.main.ad.rit.edu's
```

MobaXterm

Do you want to save password for axoeec@gle-3159-pc01.main.ad.rit.edu?

Yes   No

If you want maximum security for your stored password,
you can define a "master password" by going to
"Settings" --> "Misc" tab --> "MobaXterm passwords settings"

☑ Do not show this message again

Upload File form Local (Windows) to Remote (Linux)

Copy the local **scptest.txt** file we created to the remote Linux cluster. Execute the following command to copy scptest.txt to your remote directory as **scptest1.txt**. Note: Use your RIT userid not axoeec in all the following examples.

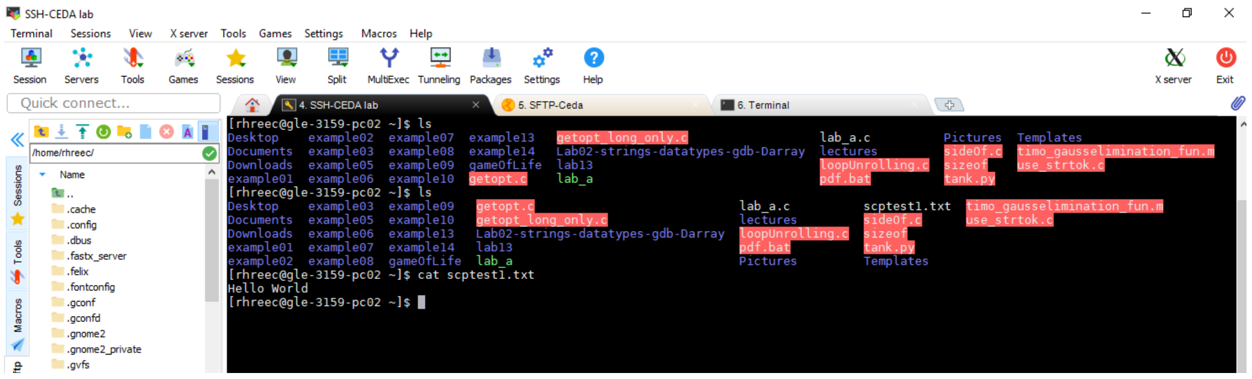
***scp "scptest.txt"* axoeec*@eng-2500-08.main.ad.rit.edu*:scptest1.txt**

Verify that the file was copied by switching to your **remote** Linux MobaXterm SSH tab and typing:

ls and then

cat scptest1.txt

You should see something like the following:



```
[rhreec@gle-3159-pc02 ~]$ ls
Desktop    example02  example07  example13  getopt_long_only.c                 lab_a.c          Pictures   Templates
Documents  example03  example08  example14  Lab02-strings-datatypes-gdb-Darray  lectures         sideOf.c   timo_gausselimination_fun.m
Downloads  example05  example09  gameOfLife lab13                               loopUnrolling.c  sizeof     use_strtok.c
example01  example06  example10  getopt.c   lab_a                               pdf.bat          tank.py
[rhreec@gle-3159-pc02 ~]$ ls
Desktop    example03  example09  getopt.c                            lab_a.c          scptest1.txt  timo_gausselimination_fun.m
Documents  example05  example10  getopt_long_only.c                  lectures         sideOf.c      use_strtok.c
Downloads  example06  example13  Lab02-strings-datatypes-gdb-Darray  loopUnrolling.c  sizeof
example01  example07  example14  lab13                               pdf.bat          tank.py
example02  example08  gameOfLife lab_a                               Pictures         Templates
[rhreec@gle-3159-pc02 ~]$ cat scptest1.txt
Hello World
[rhreec@gle-3159-pc02 ~]$
```

Download File form a Remote (Linux) to Local (Windows).

Switch back to your local terminal shell tab. Copy the scptest1.txt from the Linux system back to the Windows systems as **scptest2.txt** using the following command:

```
scp axoeec@eng-2500-07.main.ad.rit.edu:scptest1.txt  scptest2.txt
```

Verify that the file was copied and typing the following:

ls and then

cat scptest2.txt

You are now done with the local shell. Don't enter any more commands in the local shell but keep it open to show to your TA that your executed the SCP commands.

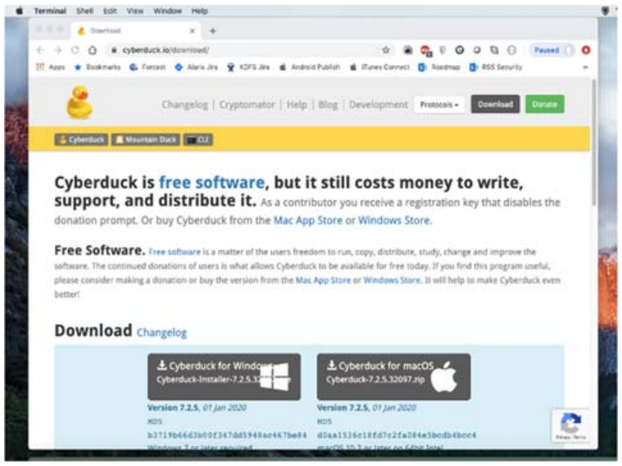
Mac Users

Mac systems are built on a Linux core and as such include a Linux terminal emulator and SCP as part of the operating system. The Mac is missing one important Linux graphic driver call "X-TERM", fortunately it is available at: https://www.xquartz.org/. Download and install "X". "X" will just run in the background of your Mac and supports displaying graphical objects.

Cyberduck

Classically, Mac users would use Secure Copy (SCP) in a Mac Linux window to move files. Many students prefer a drag & drop application so we will use Cyberduck as well.

Download and install Cyberduck from: https://cyberduck.io/download



Cyberduck is free software, but it still costs money to write, support, and distribute it. As a contributor you receive a registration key that disables the donation prompt. Or buy Cyberduck from the Mac App Store or Windows Store.

Free Software. Free software is a matter of the users freedom to run, copy, distribute, study, change and improve the software. The continued donations of users is what allows Cyberduck to be available for free today. If you find this program useful, please consider making a donation or buy the version from the Mac App Store or Windows Store. It will help to make Cyberduck even better!

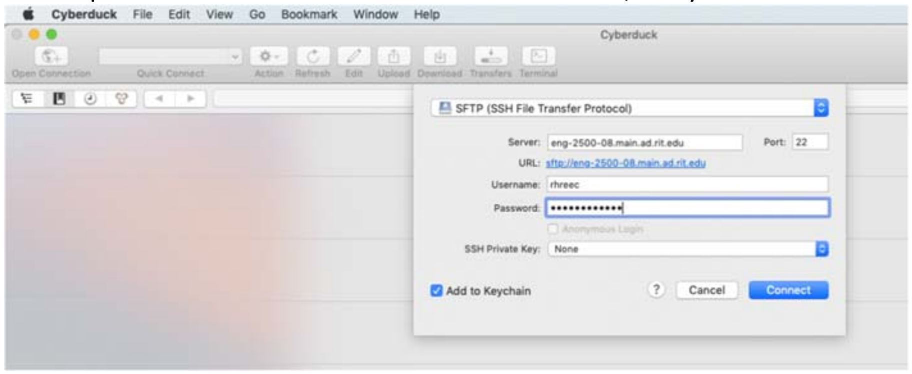
Download Changelog

Cyberduck for Windows
Cyberduck-Installer-7.2.5.3
Version 7.2.5, *01 Jan 2020*
MD5
b3719b66d3b00f347dd5948ac467be84

Cyberduck for macOS
Cyberduck-7.2.5.32097.zip
Version 7.2.5, *01 Jan 2020*
MD5
d0aa1536c18fd7c2fa084e5bodb4bcc4

Start Cyberduck, I recommend setting Cyberduck as the default FTP & SFTP tool, but this is optional.

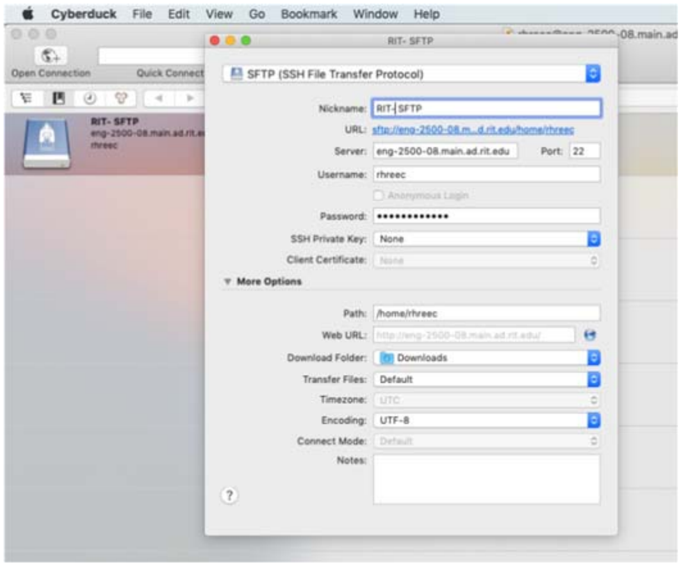
File->Open connection enter information as shown, use your username



SFTP (SSH File Transfer Protocol)

Server: eng-2500-08.main.ad.rit.edu Port: 22
URL: sftp://eng-2500-08.main.ad.rit.edu
Username: rhreec
Password: ••••••••••••
SSH Private Key: None
[✓] Add to Keychain

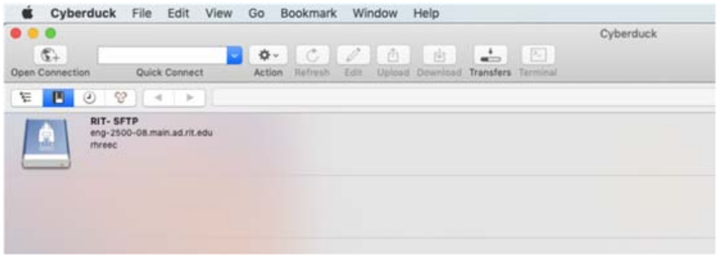
The session will open, now book mark the session for the future. Select "Bookmark" and name your bookmark as shown. Leave downloads in the Download folder for now but you will likely want to change it for future labs.



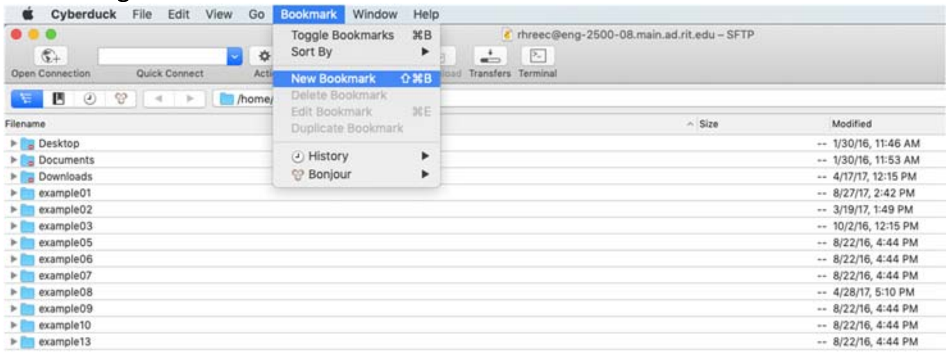
SFTP (SSH File Transfer Protocol)

Nickname:	RIT- SFTP
URL:	sftp://eng-2500-08.m...d.rit.edu/home/rhreec
Server:	eng-2500-08.main.ad.rit.edu Port: 22
Username:	rhreec
	Anonymous Login
Password:	•••••••••••••
SSH Private Key:	None
Client Certificate:	None
Path:	/home/rhreec
Web URL:	http://eng-2500-08.main.ad.rit.edu/
Download Folder:	Downloads
Transfer Files:	Default
Timezone:	UTC
Encoding:	UTF-8
Connect Mode:	Default
Notes:	

You should now have a bookmark :



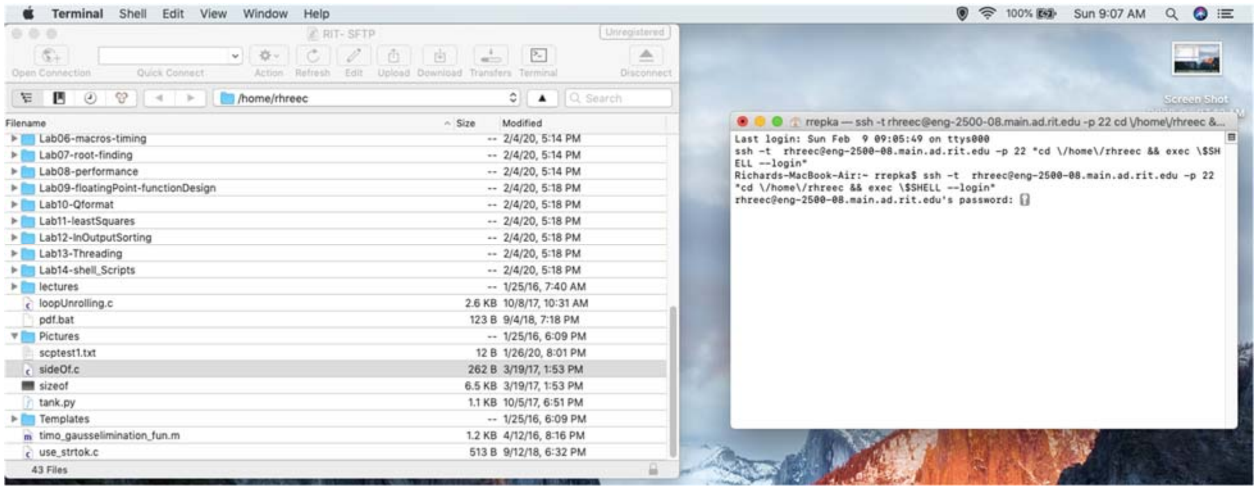
Double click on the bookmark to start an SFTP session if it is not already running. Your session will look something like:



Filename	Size	Modified
Desktop	--	1/30/16, 11:46 AM
Documents	--	1/30/16, 11:53 AM
Downloads	--	4/17/17, 12:15 PM
example01	--	8/27/17, 2:42 PM
example02	--	3/19/17, 1:49 PM
example03	--	10/2/16, 12:15 PM
example05	--	8/22/16, 4:44 PM
example06	--	8/22/16, 4:44 PM
example07	--	8/22/16, 4:44 PM
example08	--	4/28/17, 5:10 PM
example09	--	8/22/16, 4:44 PM
example10	--	8/22/16, 4:44 PM
example13	--	8/22/16, 4:44 PM

Try to drag and drop items from your Mac to the Lynx system and from the Linux system to your Mac. By default, downloads will be place in the Mac download directory.

Cyberduck also offers a Linux terminal emulator, select the "terminal" button and you should be automatically connected to your Linux machine.



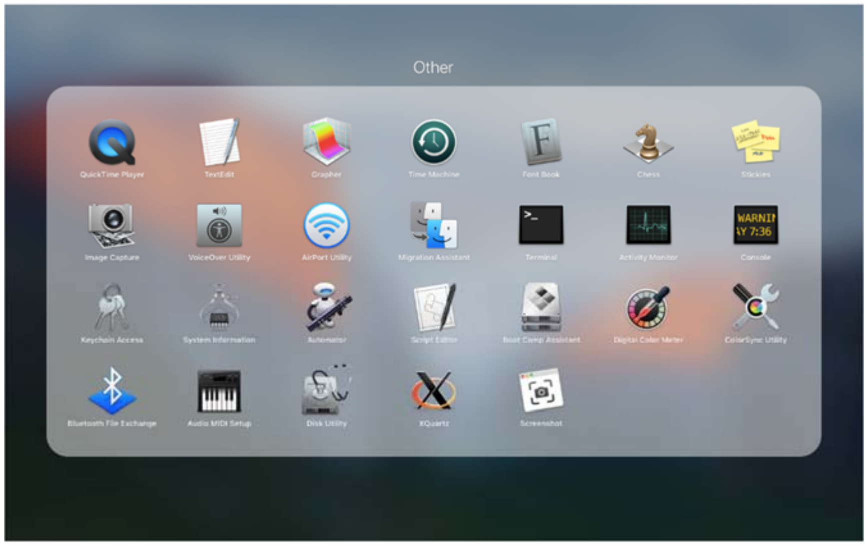
Filename	Size	Modified
Lab06-macros-timing	--	2/4/20, 5:14 PM
Lab07-root-finding	--	2/4/20, 5:14 PM
Lab08-performance	--	2/4/20, 5:14 PM
Lab09-floatingPoint-functionDesign	--	2/4/20, 5:18 PM
Lab10-Qformat	--	2/4/20, 5:18 PM
Lab11-leastSquares	--	2/4/20, 5:18 PM
Lab12-InOutputSorting	--	2/4/20, 5:18 PM
Lab13-Threading	--	2/4/20, 5:18 PM
Lab14-shell_Scripts	--	2/4/20, 5:18 PM
lectures	--	1/25/16, 7:40 AM
loopUnrolling.c	2.6 KB	10/8/17, 10:31 AM
pdf.bat	123 B	9/4/18, 7:18 PM
Pictures	--	1/25/16, 6:09 PM
scptest1.txt	12 B	1/26/20, 8:01 PM
sideOf.c	262 B	3/19/17, 1:53 PM
sizeof	6.5 KB	3/19/17, 1:53 PM
tank.py	1.1 KB	10/5/17, 6:51 PM
Templates	--	1/25/16, 6:09 PM
timo_gausselimination_fun.m	1.2 KB	4/12/16, 8:16 PM
use_strtok.c	513 B	9/12/18, 6:32 PM

43 Files

```
Last login: Sun Feb  9 09:05:49 on ttys000
ssh -t  rhreec@eng-2500-08.main.ad.rit.edu -p 22 "cd \/home\/rhreec && exec \$SH
ELL --login"
Richards-MacBook-Air:~ rrepka$ ssh -t  rhreec@eng-2500-08.main.ad.rit.edu -p 22
"cd \/home\/rhreec && exec \$SHELL --login"
rhreec@eng-2500-08.main.ad.rit.edu's password: 
```

Classic Mac Terminal session

In addition to the elegant drag and drop Cyberduck interfaces, Mac natively supports a Linux terminal session. Mac users should open the Mac terminal (Launch pad -> other -> terminal)

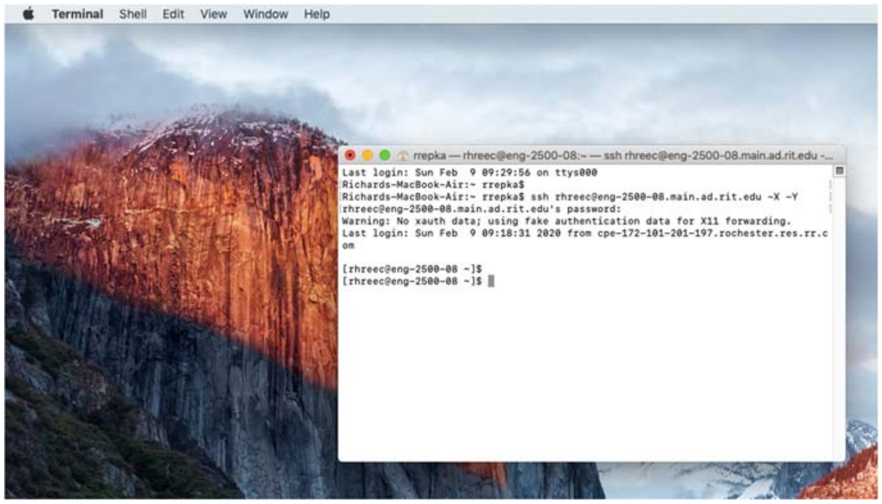


And then execute SSH commands to connect to a Lab machine as illustrated.

ssh xxxxxx@eng-2500-09.main.ad.rit.edu -X -Y

xxxxxx is your RIT computer account name.
–X to start the X server
–Y to disable connection time out

You will be connected to the remote Linux system as shown and can issue commands on the remote system from your local Mac terminal window:

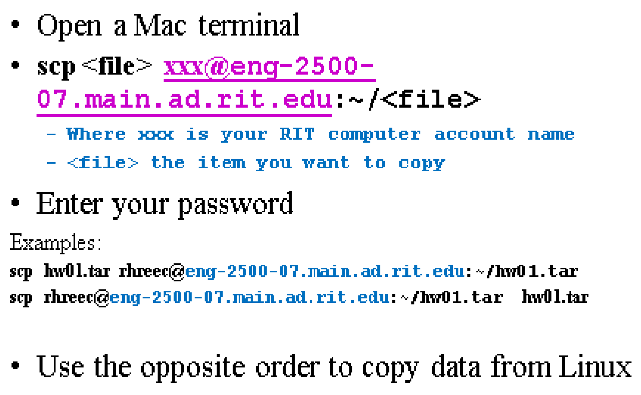

Terminal
Shell
Edit
View
Window
Help
rrepka — rhreec@eng-2500-08:~ — ssh rhreec@eng-2500-08.main.ad.rit.edu -...

```
Last login: Sun Feb  9 09:29:56 on ttys000
Richards-MacBook-Air:~ rrepka$
Richards-MacBook-Air:~ rrepka$ ssh rhreec@eng-2500-08.main.ad.rit.edu -X -Y
rhreec@eng-2500-08.main.ad.rit.edu's password:
Warning: No xauth data; using fake authentication data for X11 forwarding.
Last login: Sun Feb  9 09:18:31 2020 from cpe-172-101-201-197.rochester.res.rr.c
om

[rhreec@eng-2500-08 ~]$
[rhreec@eng-2500-08 ~]$
```

Classic Mac SCP session

The Secure Copy command (SCP) can be use in any Linux terminal to transfer files to/from other Linux machines



- Open a Mac terminal
- scp <file> xxx@eng-2500-07.main.ad.rit.edu:~/<file>
  - Where xxx is your RIT computer account name
  - <file> the item you want to copy
- Enter your password

Examples:

```
scp hw01.tar rhreec@eng-2500-07.main.ad.rit.edu:~/hw01.tar
scp rhreec@eng-2500-07.main.ad.rit.edu:~/hw01.tar  hw01.tar
```

- Use the opposite order to copy data from Linux

Open a 2nd local Mac Linux session. Please create a local file in the current directory by executing the following:

echo "Hello World" > scptest.txt

We are going to use the standard SCP command to move individual files to/from our Mac to the Linux system. When you first execute a scp command, the Mac will ask you to enter your password for the remote system. Please enter your password.

Upload File form Local (Mac) to Remote (Linux)

Copy the **scptest.txt** file we created to the Linux cluster.. Execute the following command to copy scptest.txt to your remote directory as **scptest1.txt**. Note: Use your RIT userid not axoeec in all the following examples.

***scp "scptest.txt"* axoeec@*eng-2500-08.main.ad.rit.edu*:scptest1.txt**

Verify that the file was copied by switching to your Cyberduck SSH tab and typing:

ls and then

cat scptest1.txt

Download File form a Remote (Linux) to Local (Windows).

Switch back to your local Mac terminal shell. Copy the scptest1.txt from the Linux system back to the Mac system as **scptest2.txt** using the following command:

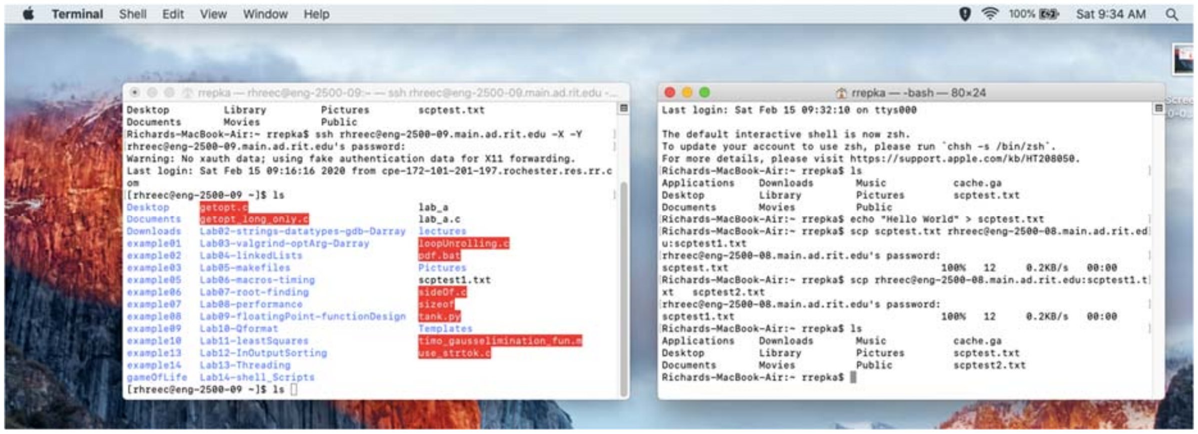
```
scp axoeec@eng-2500-07.main.ad.rit.edu:scptest1.txt  scptest2.txt
```

Issue the same Linux commands on the local system and you should see the new file.

ls and then

cat scptest2.txt

You should see something like the following:


Terminal
Shell
Edit
View
Window
Help
100%
Sat 9:34 AM
rrepka — rhreec@eng-2500-09:~ — ssh rhreec@eng-2500-09.main.ad.rit.edu -...
rrepka — -bash — 80×24

```
Desktop         Library         Pictures        scptest.txt
Documents       Movies          Public
Richards-MacBook-Air:~ rrepka$ ssh rhreec@eng-2500-09.main.ad.rit.edu -X -Y
rhreec@eng-2500-09.main.ad.rit.edu's password:
Warning: No xauth data; using fake authentication data for X11 forwarding.
Last login: Sat Feb 15 09:16:16 2020 from cpe-172-101-201-197.rochester.res.rr.c
om
[rhreec@eng-2500-09 ~]$ ls
Desktop      getopt.c                             lab_a
Documents    getopt_long_only.c                   lab_a.c
Downloads    Lab02-strings-datatypes-gdb-Darray   lectures
example01    Lab03-valgrind-optArg-Darray         loopUnrolling.c
example02    Lab04-linkedLists                    pdf.bat
example03    Lab05-makefiles                      Pictures
example05    Lab06-macros-timing                  scptest1.txt
example06    Lab07-root-finding                   sideOf.c
example07    Lab08-performance                    sizeof
example08    Lab09-floatingPoint-functionDesign   tank.py
example09    Lab10-Qformat                        Templates
example10    Lab11-leastSquares                   timo_gausselimination_fun.m
example13    Lab12-InOutputSorting                use_strtok.c
example14    Lab13-Threading
gameOfLife   Lab14-shell_Scripts
[rhreec@eng-2500-09 ~]$ ls
```

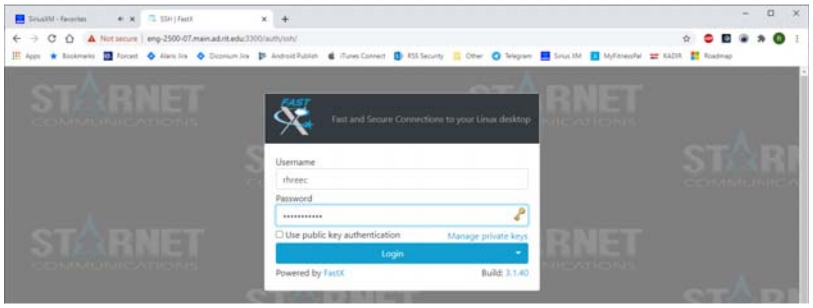
```
Last login: Sat Feb 15 09:32:10 on ttys000

The default interactive shell is now zsh.
To update your account to use zsh, please run `chsh -s /bin/zsh`.
For more details, please visit https://support.apple.com/kb/HT208050.
Richards-MacBook-Air:~ rrepka$ ls
Applications    Downloads       Music           cache.ga
Desktop         Library         Pictures        scptest.txt
Documents       Movies          Public
Richards-MacBook-Air:~ rrepka$ echo "Hello World" > scptest.txt
Richards-MacBook-Air:~ rrepka$ scp scptest.txt rhreec@eng-2500-08.main.ad.rit.ed
u:scptest1.txt
rhreec@eng-2500-08.main.ad.rit.edu's password:
scptest.txt                                   100%   12     0.2KB/s   00:00
Richards-MacBook-Air:~ rrepka$ scp rhreec@eng-2500-08.main.ad.rit.edu:scptest1.t
xt  scptest2.txt
rhreec@eng-2500-08.main.ad.rit.edu's password:
scptest1.txt                                  100%   12     0.2KB/s   00:00
Richards-MacBook-Air:~ rrepka$ ls
Applications    Downloads       Music           cache.ga
Desktop         Library         Pictures        scptest.txt
Documents       Movies          Public          scptest2.txt
Richards-MacBook-Air:~ rrepka$
```

FasxtX (optional)

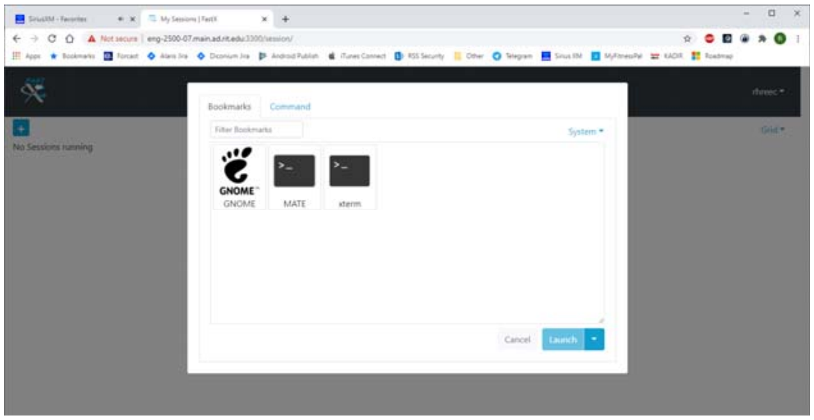
This step is optional but might come in handy if you are on a computer without a standard Xterminal client. RIT Linux servers have the FastX web server installed. This product allows for simple terminal sessions or full X terminal sessions over a browser.

1) Access FastX on any RIT Linux cluster machine https://eng-2500-07.main.ad.rit.edu:3300/

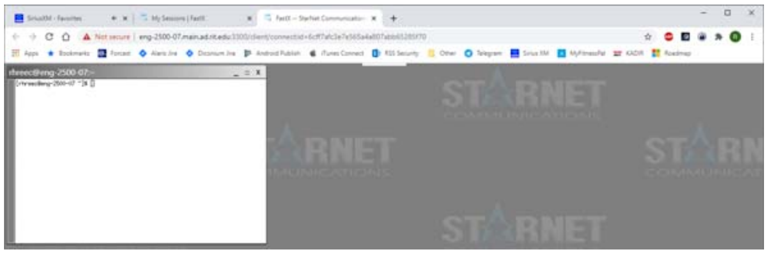

Fast and Secure Connections to your Linux desktop
Username
Password
Use public key authentication
Manage private keys
Login
Powered by FastX
Build: 3.1.40

There may be security warning.

2) Click "+" then xterm the "launch"



3) Your Session starts



Review Linux Commands

This section provides information about the most important commands that will be used frequently. Along with the individual commands, parameters are listed, and examples are provided. To learn more about a specific command, use the command *man <commandName>* where <commandName> represents the name of the command. For example, *man ls* command provides all details regarding the *ls* command.

Please note that all commands are case sensitive. Throughout this section, all commands are written in italics.

All commends should be entered in the remote Linux terminal shell NOT the local shell.

Cheat Sheets

Two cheat sheets were included in your exercise directory and are also included on MyCourses in the References section. The first is called "**unixCheatsheet.pdf**" which includes a very brief summary of popular Linux commands. We won't use most of the commands in this class but you will if you work in the Linux world. You should review the Unix cheat sheet if you are new to Linux.

The second file is called "**vi_cheat_sheet.pdf**" which includes a very brief summary of "vi" or "vim" editor commands. The "vi" or "vim" editor is available on all versions of Linux and it is automatically used by many Linux tools. We will be using vi or vim for our simple editing tasks.

If you are new to Linux and vi or vim then you should consider printing out the cheat sheets and using them.

Creating & Deleting Directories

To create a directory myDir use the command *mkdir myDir*. To delete a directory, use *rmdir myDir*.

1) Create a directory named Lab01, notice the upper case "L" (***mkdir Lab01***) and move to that directory (***cd Lab01***) to try the commands in the following sub-sections.

Using vi or vim editors

You can use the built-in editors **vi or vim** in a Linux system to create or edit files. Both editors are identical on our system and there are lots of tutorials on how to use them, such as the ones below:

- https://thomer.com/vi/vi.html
- MyCourses -> References -> vi_cheat_sheet.pdf

Creating, Editing and Deleting Files

You can create a file with the editor (e.g. *vi myFile*).

To delete a file you can use *rm <fileName>* where <fileName> is the name of the file you want to delete.

2) Create a file named myFile using the vi editor.
   - *vi myFile*
   - Type "***i***" to enter "insert" mode and then type the following text:

     Hello World

     I love Linux

   - Use the "***esc***" key and "***:wq***" to enter command mode (esc:) and then "wq" to write the file and quit.
3) Re-edit the same file and add another line.
   - *vi myFile* the use the cursor keys to move to the end of Linux word
   - Type "*a*" to enter append mode
   - Hit <enter> and type: "*I would like a F-- in class*" (yes, F--, no quotes)
   - Use the "***esc***" key and ":***wq***" to write the file and quit.
4) Re-edit the same file to fix the grade
   - *vi myFile* the use the cursor keys to move to the **F--** grade
   - Type "*R*" (capital R) to enter replace mode then type "*A++*"
   - Write the file out and exit
5) Re-edit the same file to copy the grade line 2 times and then delete 1.
   - *vi myFile* the use the cursor keys to move to the grade line
   - Type "yy" (yank) to copy the entire line then type "*p*" and "*p*" to paste the line twice.

- Type “dd” (delete) to delete the current line.
- Write the file out and exit

6) View the file with *cat myFile*
7) Delete the file named myfile
   - *rm myfile*
8) Use **vi** to create files with the indicated custom names and content.
   - me.txt: Enter your full name
   - myShortBio.txt: Write your short bio (at least 3 sentences).
   - myFirstCFile.c: Leave empty for now.

Listing Files and Their Attributes

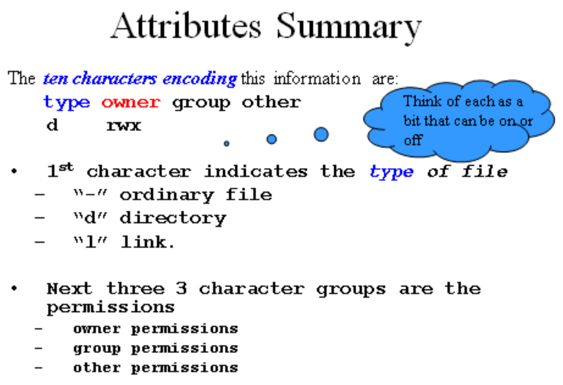
Use the **ls** command to list files (directories) and their attributes. Try each of the following examples and check your output (from within the Lab01 directory).

Examples

9) List file and directory names within the current directory
    a. *ls*
10) List file and directory names (with all information) within the current directory
    b. *ls –l*

The follow two graphics summarize the attributes and the permission fields within attributes when the *ls -l* command is executed. Please note that the attributes are seen when the ls command is executed with –l option (i.e., *ls –l)*.

11) List file and directory names (all, including hidden "dot files") within the current directory
    c. *ls –a*
12) List file and directory names (all files with all information) within the current directory
    d. *ls -la*
13) List file and directory names and their sizes(in blocks) within the current directory
    e. *ls –s*



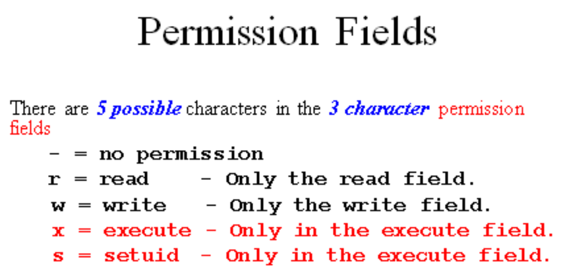
Attributes Summary

The *ten characters encoding* this information are:

```
type owner group other
d    rwx
```

Think of each as a bit that can be on or off

- 1st character indicates the *type of file*
  - "-" ordinary file
  - "d" directory
  - "l" link.
- Next three 3 character groups are the permissions
  - owner permissions
  - group permissions
  - other permissions



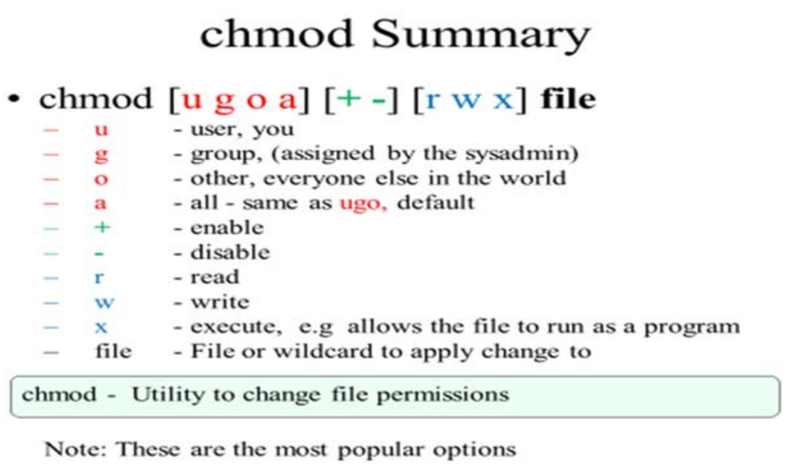
Permission Fields

There are *5 possible* characters in the *3 character* permission fields

```
 - = no permission
 r = read     - Only the read field.
 w = write    - Only the write field.
 x = execute  - Only in the execute field.
 s = setuid   - Only in the execute field.
```

Changing File Attributes

The **chmod** command sets the permissions of files or directories. The command **chmod** stands for "change mode" and determines the way a file (or directory) is accessed.



chmod Summary

- chmod [u g o a] [+ -] [r w x] **file**
  - u - user, you
  - g - group, (assigned by the sysadmin)
  - o - other, everyone else in the world
  - a - all - same as ugo, default
  - + - enable
  - - - disable
  - r - read
  - w - write
  - x - execute, e.g allows the file to run as a program
  - file - File or wildcard to apply change to

chmod - Utility to change file permissions

Note: These are the most popular options

Handy hint: To protect your files: **chmod go-rwx ***

Examples

1. Assume you created a file named “filename”, and you want to set its permissions so that the user can read, write, and execute it, the members of the user group can read and execute it, and any other user may only read it.
   a. Execute commands: *chmod u=rwx filename, chmod g=rx filename, chmod o=r filename*,
   b. You can do all at once using the octal permission notation: *chmod 754 filename*
2. Make “filename” executable (a designates all, e.g., ugo)
   a. chmod a+x filename
3. Set permission of “filename” to read and write for all:

   a. *chmod a+rw filename*
   b. *chmod +rw filename*   (a is default)
4. Make all your files private. Deny all permissions for group and other.
   a. *chmod go-rwx **
5. Check the permission for "filename"
   a. *ls –l filename*

Archiving and Listing tarballs

A tar file is called a "tarball". The *tar* command is used to create .tar.gz or .tgz archive files, also called "tarballs." The tar command has a large number of options, but you just need to remember a few letters to quickly create archives with tar. You may also use the tar command to extract the resulting archives. To archive files, we will use **tar** (**ta**pe a**r**chive)

*tar cvf myfile.tar *.c*	(**c**reate)
tar tvf myfile.tar	(lis**t** or **t**est)
tar xvf myfile.tar	(e**x**tract)
tar rvf myfile.tar	(**r**eplace or add)

Important Remarks:

- Please make sure not to gzip your tar files! Do not use the tar *tar zcvf* option or gzip!
- In the first command (with cvf option), if you forget the "myfile.tar", tar will destroy the first file in the *.c! Why? Because, the first .c file will be the name of your .tar file now!

Examples

1. Access MyCourses using your PC/Mac and download the **Lab01.tar** file.
2. Upload the tar file to your Linux system using either SFTP in MobaXTerm of SCP (for Mac users) to your **Lab01** directory.
3. Return to your Linux terminal session and extract the contents of the Lab01.tar file using:

tar -xvf Lab01.tar

4. Execute a **ls** command, you should see a new directory **Lab01** with two new subdirectores directories called: **prelab** and **exercise** under it. In the future the results of your prelab work will be placed in the prelab directory and in lab work in the exercise directory.
5. Change to the **exercise** (**cd Lab01/exercise**). Create the following files within exercise.
6. Issue a **ls** command and verify there is a **QuadraticSolver.c** file
7. Create **myinfo.txt** using a Linux editor, write your full name.
8. Take your screenshot of your session on your PC/Mac and save it as screen.png and then transfer it to your Linux system, in the **Lab01/Lab01/exercise** directory using SFTP or scp.
9. Move up one directory, back to the **Lab01 level** using **cd ..**
10. Create an **analysis.txt** file and enter "Hello World" You will document your at home work here in the future.
11. Move up one more directory, above the **Lab01 level** using **cd ..**
12. Create a tar file named name_hw01.tar (name is your last name, no spaces).tar using the files in Lab01 directory. This is the normal way you will submit your assignments to MyCourses.
    - *tar cvf name_lab01.tar Lab01*
13. List the contents of a tar file with details.
    - tar tvf name_lab01.tar

I/O Redirection

Return to the Lab01 directory level using **cd Lab01**

1. Any program taking input from the standard input (*e.g.* keyboard) can be redirected to take input from any input file.

   ```
   myprogram < input_file
   ```

2. Any program writing to standard output (*e.g.,* screen) can be redirected to write to any output file

   ```
   myprogram > output_file
   ```

3. We can also combine input and output

   ```
   myprogram < input_file > output_file
   ```

4. Create or append options

   ```
   out_file > in_file (created)
   out_file >> in_file (appended)
   ```

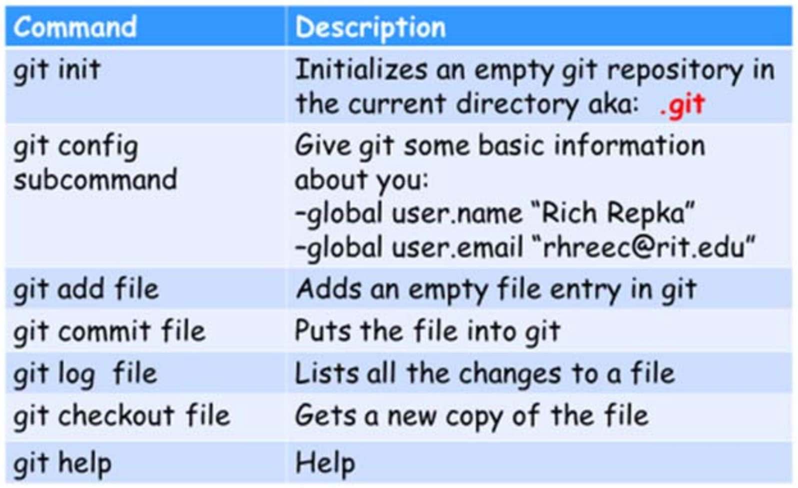
List current directory contents and redirect the command output to a file named myFile.txt. using:

```
ls –la >myFile.txt
```

Git Command Set

Git is a version control system for tracking changes in computer files created by Linus Torvalds in 2005. By default Git adds control to the current directory.

Software development in an enterprise is impossible without using a Version Control System (Git, Subversion, CVS etc.). Version Control System is used to track and manage changes in a software project. It helps to ensure the integrity of a software project



Command	Description
git init	Initializes an empty git repository in the current directory aka: .git
git config subcommand	Give git some basic information about you: -global user.name "Rich Repka" -global user.email "rhreec@rit.edu"
git add file	Adds an empty file entry in git
git commit file	Puts the file into git
git log file	Lists all the changes to a file
git checkout file	Gets a new copy of the file
git help	Help

Examples

1. Move to the Lab01 directory, if you are not already there, and create an empty Git repository there
   a. *git init*
2. Set up basic information
   a. git config --global user.name "<your user name>"
   b. git config --global user.email <your email>

3. Please note that a new directory named .git is created.
    a. *ls –la*
    b. See its contents by *ls -la .git/*
4. Create an empty C file an add it to Git.
    a. *vi test.c then <esc>:wq to write and quit*
    b. *git add test.c*
    c. *git commit test.c* A vi editor will pop up; press i and then enter a comment explaining the change made on test.c file. DO NOT ENTER "#" in your comment, as it would be considered a git comment, not a code change commet.
    d. Exit vi - <esc>:wq to save and quit. Your comments will be associated with the test.c file in the Git repository. Other developers will see your comments to help understand the purpose or the scope of your change when you push your commits.
5. Delete file test.c
    a. *rm test.c*
    b. Enter *ls –la* and see that test.c is deleted
6. Restore back from git
    a. *git checkout test.c*
    b. Type *ls –la* and verify that test.c is back
7. Enter vi test.c and make some changes on test.c
    a. *vi test.c then <esc>:wq to write and quit*
8. Verify that the changes on test.c are tracked by Git.
    a. *git status* See the "modified: test.c" in the output.
9. To see the change log enter
    a. *git log test.c*

Other Commands: Display, Cat, nano

The Linux command *display* is used to display images and graphs. Test *display Lab01/screen.jpg*

The Linux command *cat* is used to print the contents of text files to the console.

cat Lab01/myinfo.txt

While “vi” or “vim” are available on all Linux installations and are the default editor for many commands, many Linux distributions also support a graphical editor called “nano”.

- https://www.tutorialspoint.com/articles/how-to-use-nano-text-editor

All the important nano commands are shown at the bottom of the nano window.

Try: nano myInfo.txt

Assignment – 60 pts

Objective

To familiarize the student with the basic aspects of all homework assignments in this course such as, connecting to the system using a secure shell, editing, Xwindows, compiling with gcc, basic Linux operations, using git, creating make files, packing code into a tar file and submitting a tar file.

Part A

Login to any CE machines (eng-2500-XX.main.ad.rit.edu to eng-2500-30.main.ad.rit.edu) using a secure shell client (e.g., MobiXterm). Make a directory called hw01 (mkdir hw01). Then, go to that directory (cd hw01) and using your editor of choice – e.g., vi, vi, – create a text file, lastName_myinfo.txt, (lastName is your last name) with the following information (one item per line):

- Full Name
- Major and anticipated graduation date
- List of each programming language that you know and how well you know it.

Part B

You must have Xwindows (MobiXterm) installed on your PC, (Xquartz) Mac or other device. Read the exercise section of this document to find out how to enable Xwindows or MobiXterm.

- Run the Linux command: module load matlab

- Run Matlab by typing `matlab` and take a screen capture, or picture, of the Matlab logo splash screen and include it in your tar file as “screen.xxx” (xxx is the file type of your screen shot).

Note: Matlab off campus can be very, very slow.

Part C

List all the hidden files in your root directory (e.g. **cd ~ ls -la**) and redirect the output using UNIX redirection ">" to a file called "hidden.txt". Edit the hidden ".bashrc" file using your editor of choice and add the following to the bottom of the file:

```
export GDFONTPATH=/usr/share/fonts/dejavu
export GNUPLOT_DEFAULT_DDFONT="DejaVuSans.ttf"
```

then copy the .bashrc file into your hw01 directory (**cp .bashrc hw01/bashrc.txt**), calling it "**bashrc.txt**". Be sure to include your "**hidden.txt**" and "**bashrc.txt**" file in your hw01 directory and then your final TAR file.

Part D

Test "git".

Be sure to prove each of the following steps was accomplished by executing an "ls -la" between step and add the results to your analysis.txt file with appropriate comments. Be sure to include the contents of the change log in the analysis.txt file.

1 Create (Add) a "test.c" file with a single comment in it (your name).

2 Add (commit) the file to git, provide a change summary

3 Add another comment line with your major to your test.c file

4 Add (commit) the file to git, provide a change summary

5 Delete the test.c file

6 Recover test.c from git.

7 List the change log

Packaging

To submit your work please use the following guideline for all labs. Not all labs will have all files. All work and data for the out of lab assignment will be located at the root directory of the lab. Pre lab and in lab work and data will be stored in the corresponding subdirectory.

General lab directory format:

```
LabX                // X is 01 in this case
|-- prelab          // A subdirectory for all pre-lab work and answers
|   -- prelab.txt
|   -- [..]
|-- exercise        // A subdirectory for all in lab exercise work and answers
|   -- exercise.txt
|   -- [..]
|-- Makefile        // The make file (future) for the assignment.
|-- analysis.txt    // Your analysis for the assignment
|-- [..]            // All provided assignment files, your work files and any
                    //resulting data files.  E.g.  .c, .h. ,txt, etc. files
```

Part E

To submit your work go to your current homework directory, in this case hw01, and create a single tar file (lastName_hw01.tar) containing the required files (where lastName is your last name, e.g. Repka_hw01.tar)

Grading Criteria & Notes

- Each part of the assignment is worth 12 points.
- Always make sure that your tar files have been created correctly.
- Verify the content using tar -tvf lastName_hw01.tar
- Once you are sure that everything is correct, submit your homework lastName_hw01.tar to the Assignments section in MyCourses.

Student Name: ____

Laboratory Grading Sheet

Lab01 - Linux Commands

Component	**Point Value**	**Points Earned**	**Comments and Signatures**
Pre-Lab – laptop if possible	**5**		
In lab Exercise			
Demonstrate MobiXterm (or Mac) SSH and SFTP	**5**		
Linux commands oral quiz (mkdir, cp, cd, ls, chmod, IO redirection, etc.)	**10**		
Demo vi editor (create, edit, save, etc.)	**10**		
Demo tar creation	**5**		
Demo git check in and out	**5**		
Total	**40**		

You must turn this signed sheet in at the end of lab to receive credit!