

LINUX Overview

What is Linux?

Linux is an **operating system**. You have probably used a flavor of the Microsoft Windows operating system (e.g., MS DOS, Windows NT, Windows XP, Windows 8). An operating system manages program execution and users, allocates storage, manages file systems, and handles interfaces to peripheral hardware (e.g., printers, keyboards, mouse).

Linux has two main parts:

- | | |
|------------------------|---|
| kernel | Allocates machine resources including memory, disk space, and CPU cycles. |
| system programs | Includes device drivers, libraries, utility programs, shells (command interpreters), and configurations |

Linux is a flavor of the UNIX operating system.

1971 UNIX was released on 11/03/1971. It provided a command line interface.

1972 Ritchie rewrote B and called it C.

1983 AT&T released UNIX System V. It included most of the command line capabilities including pipes.

1987 X11 released as the foundation for X Windows.

1991 Linux is introduced by Linus Torvalds, a student in Finland

2004 Ubuntu Linux released.

Why is Linux so popular?

- Not proprietary.
- Portable to most hardware. Linux was written in C.
- Inexpensive since it is open source.
- Supports multiple simultaneous users.
- Easy integration with system programs and libraries via command shell languages

X Windows

Like MS Windows, Linux provides a GUI which frees users to navigate the file system. Full GUI capability is only available if you have started **X Windows**.

If you login directly to a Linux machine (like one in our lab in [NPB 2.118](#)), you will automatically have X Windows.

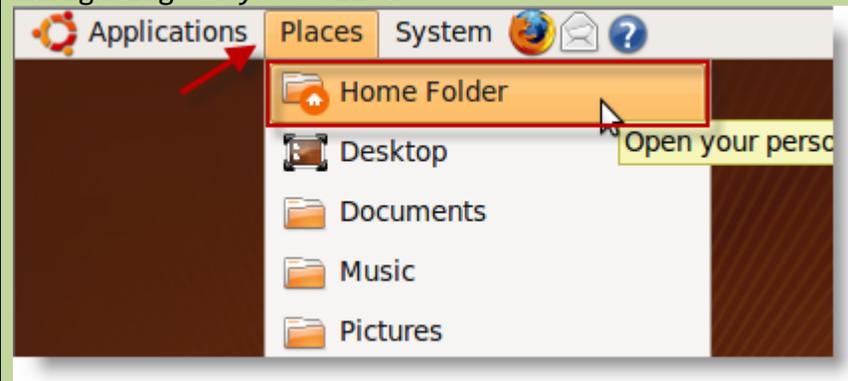
Depending on the desktop, there are multiple different ways to get to the File Browser:

Approach #1 (Linux 9.04)

- On the menu, select **Places**

Sample UI for approach #1 (Linux 9.04)

Navigating to your files

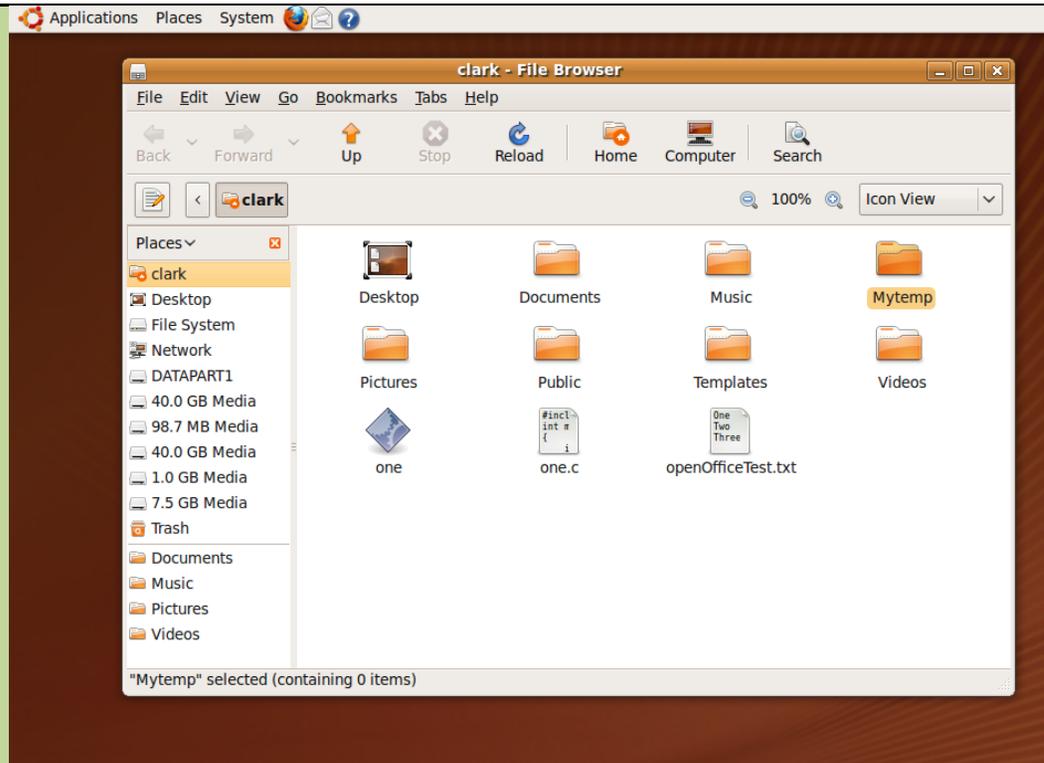


Linux will show a window for navigating your files.

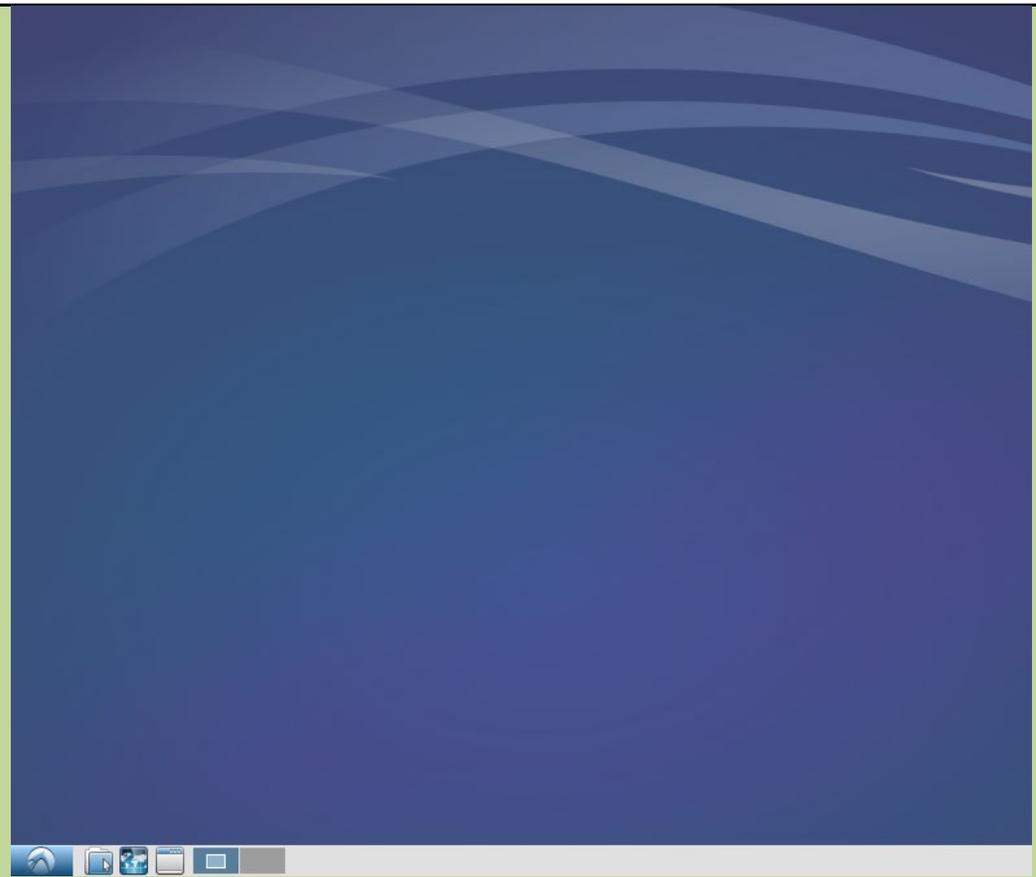
- Select **Home Folder**.

Approach #2 (Linux 14.04, see below):

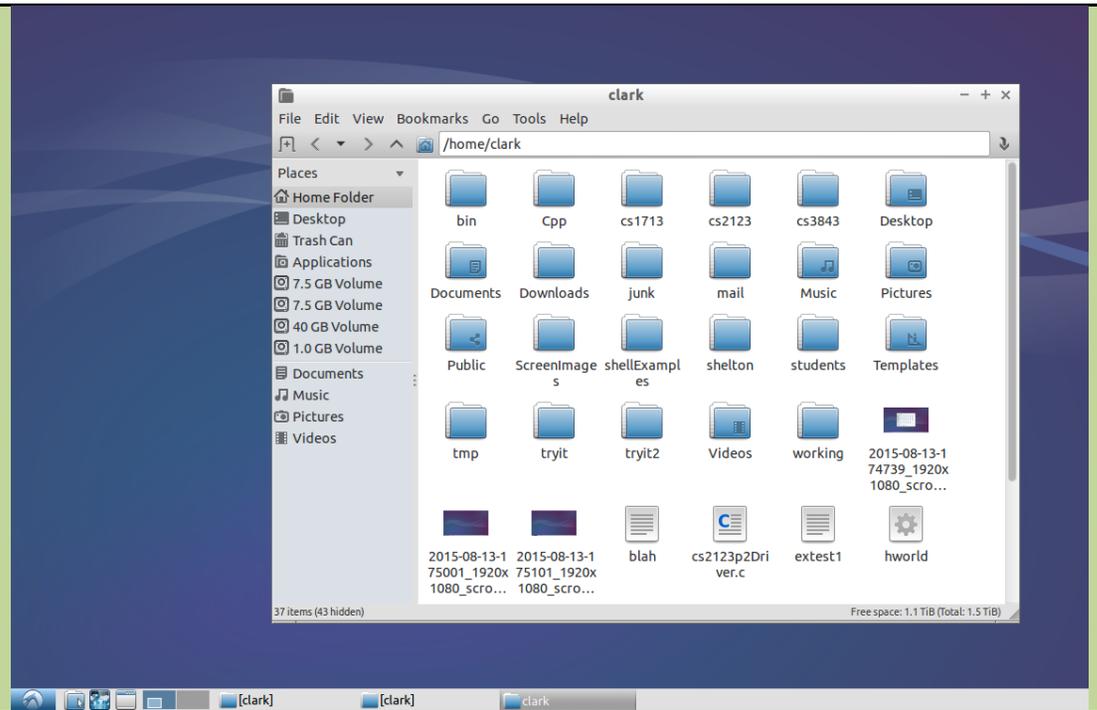
- On the icon button list at the bottom of the screen, click the Folder icon Button (second from left).



Sample UI for Approach #2 (Linux 14.04 in the CS Lab)



Linux will show a window for navigating your files



Remote Access

When you remote into UTSA's Linux servers from MS Windows, you must first use a product that establishes an X Window Server if you want GUI capabilities.

Some products which provide an X Window Server:

- Xming
- Cygwin/X

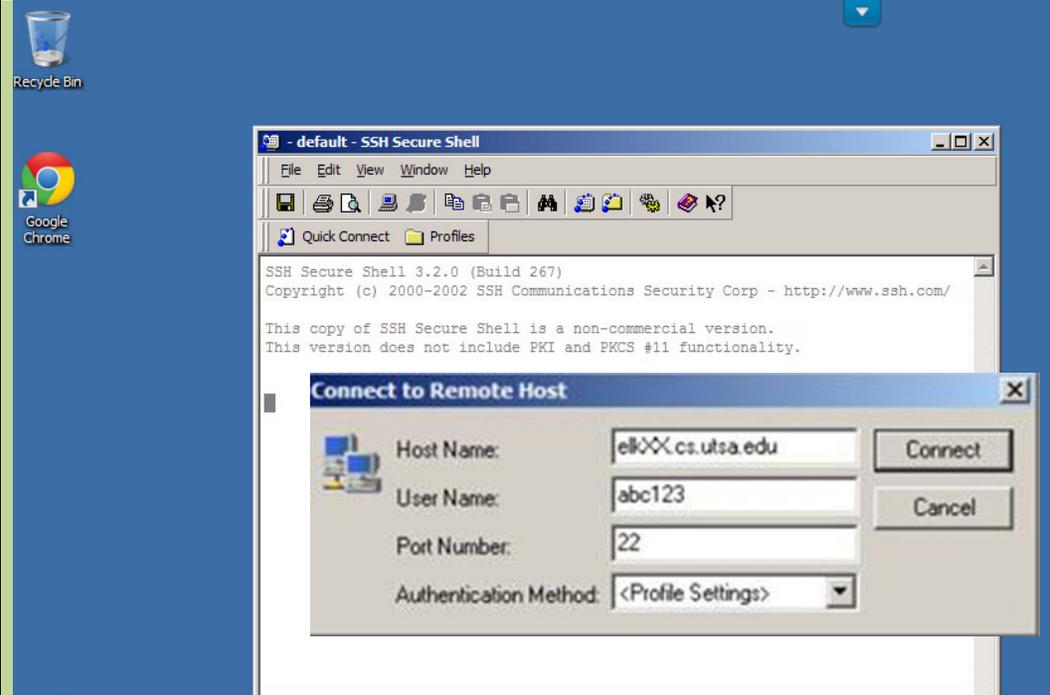
Note that our Virtual Desktop Infrastructure (VDI) Windows clients have Xming. (See my setup instructions for more information.)

You can then use **ssh** to either

- send files between your windows workstation and UTSA's Linux servers
- start a terminal session

Using ssh to access a UTSA Linux server

See my **setup instructions** for more information.



Linux Command Shell

It is a command interpreter which acts as an interface between you and the operating system. When you enter a command on the screen, the Linux command shell interprets the command and invokes the specified program.

The power of the Linux command shell is that it can be used interactively and as a programming language. You can do just about anything from the Linux command shell.

This type of interface between you and the operating system is called a **command line** interface. Everything you do through the Linux Command Shell is done by executing lines of commands.

Note that MS Windows also provides a command shell, cmd.exe.

```
# Comments in Shell begin with a #
# In the examples below, assume $ is the command line prompt
# To compile we type:
$ gcc -g -o p0 cs1713p0.c

# That creates an executable program named p0 from the C language
# source file cs1713p0.c.

# List contents of a directory (aka, folder)
$ ls -al

# Changing to another directory
$ cd cs1713
```

Terminal Session

When you use **ssh** to remotely access one of the UTSA Linux servers, you will automatically see a **Terminal** window.

If you login directly to a Linux machine with Linux 9.04 (approach #1), you can show a terminal session by doing the following:

- On the menu, select **Applications**
- Select the **Accessories** submenu
- Select the **Terminal** application

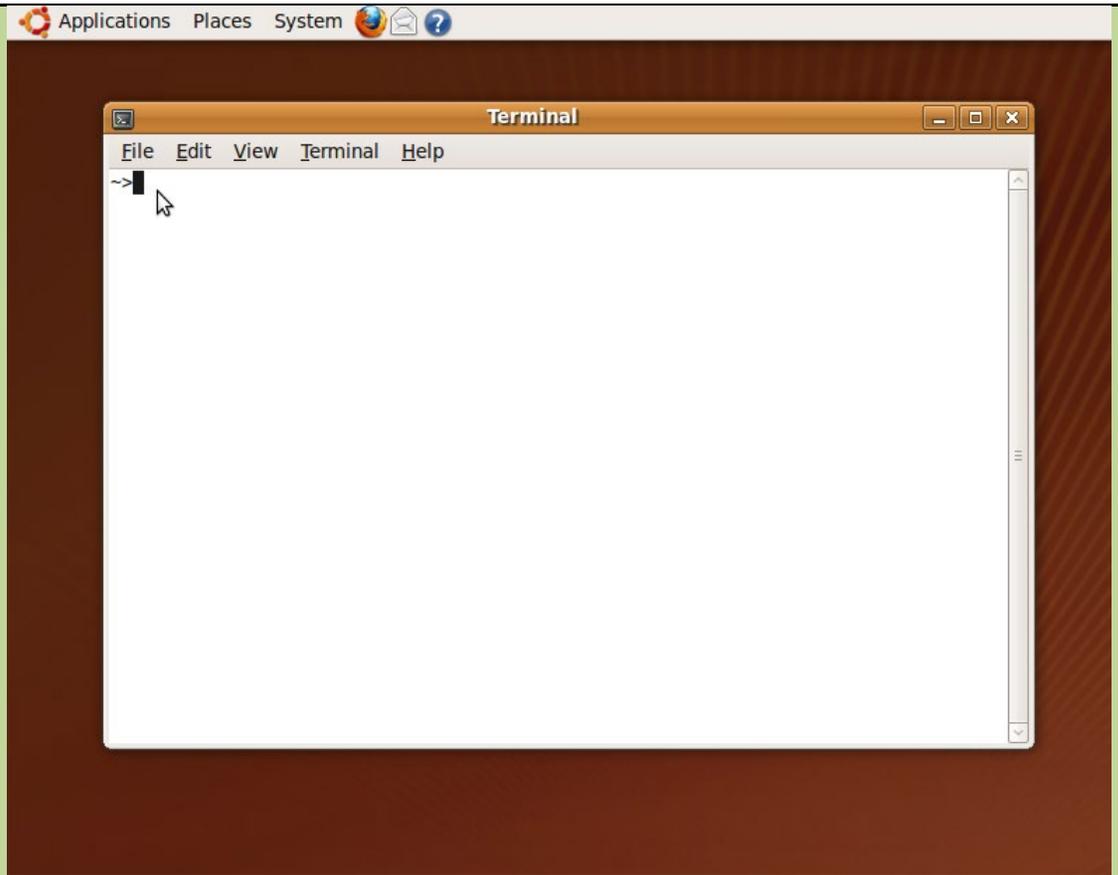
That will bring up a terminal window which allows entry of command lines.

If you use one of our Linux machines in NPB 2.118 (approach #2), see below.

Approach #1: Linux 9.04



The terminal window (where you can type command lines) in Linux Ubuntu 9.04

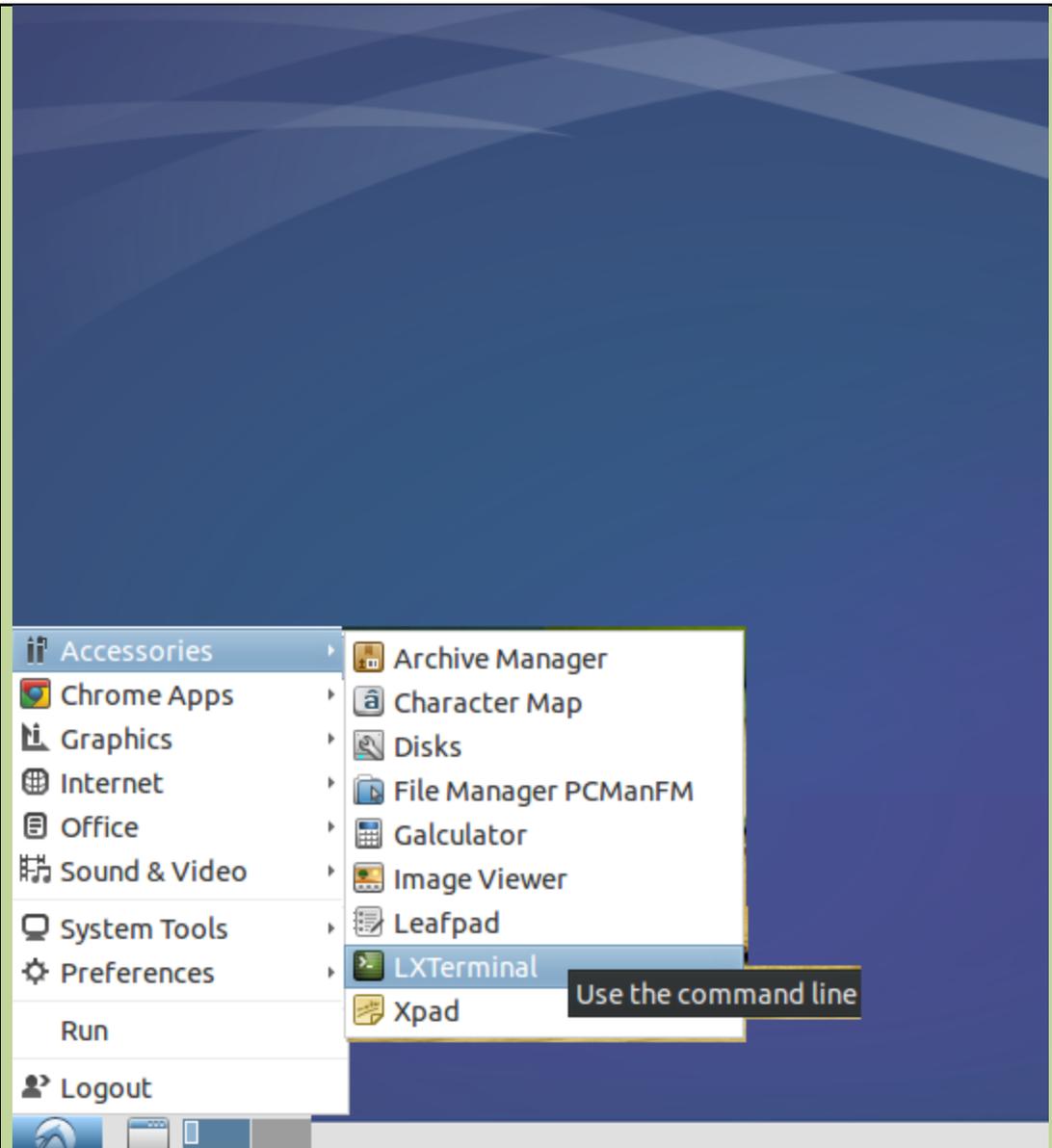


This will bring up the terminal window for the Linux 14.04 workstations in our CS Lab.

With Linux 14.04 in our CS Lab, you can show a terminal session by doing the following:

- Click the start button (bottom left). A menu will appear as shown to the right.
- Select **Accessories**
- Select **LXTerminal**

Approach #2 (Linux 14.04 in the lab):



Some popular Command Line Commands

ls - list files in a directory

cd - change directory (navigate to a different directory)

cat - show the contents of a file or create a new file (also can be

In the examples below, assume the command line prompt is \$
The shaded text is the response from Linux.

change to my home directory. ~ is defined to be your home directory
(i.e., highest directory)
\$ cd ~

used to concatenate files)

pwd - print working directory (i.e., show the current directory)

mkdir - create a new directory

rm - remove files

cp - copy files

mv - move or rename files

For more information, see programming assignment #0 and the Unix Cheat Sheet.

```
# get the current directory
$ pwd
/home/clark

# make a directory named Mytemp
$ mkdir Mytemp

# change to the Mytemp directory and get the current directory
$ cd Mytemp
$ pwd
/home/clark/Mytemp

# create a file named animal using the cat command, enter some text, and use
# the CTRL and D keys to exit input.
$ cat >animal
dog
cat
dolphin
CTRL-D

# show the contents of animal using the cat command
$ cat animal
dog
cat
dolphin

# copy animal to mammals
$ cp animal mammals

# list all files in the current directory with details
$ ls -al
drwx----- 2 clark faculty 4096 Jul 18 14:06 .
drwx--x--x 50 clark faculty 4096 Jul 18 14:04 ..
-rw----- 1 clark faculty 16 Jul 18 14:05 animal
-rw----- 1 clark faculty 16 Jul 18 14:06 mammal

# change to the directoy above the current directory
$ cd ..
$ pwd
/home/clark

# remove Mytemp and recursively its contents
$ rm -r Mytemp

# attempt to change to the Mytemp directory
$ cd Mytemp
Mytemp: No such file or directory
```

Compiling and Linking

```
# Compile and link first.c, producing an
# executable named "first".
```

You will use a Linux Command Shell to compile and link.

To compile in Unix, we use **gcc** to generate compiled object and executables:

```
gcc -g -o executableName cSourceFileName.c
```

where `-g` tells the compiler to generate information suitable for the debugger.

This generated an executable with the name specified after the output switch (`-o`).

Until we have more complex assignments, we will use that form of the `gcc`.

To compile a `c` source file, producing only a compiled object:

```
gcc -c -g -o cSourceFileName.o cSourceFileName.c
```

where specifying `-c` will cause it to only compile, producing a compiled object named `cSourceFileName.o`

To link (combines compiled objects and libraries, resolving external references) a compiled object file, producing an executable:

```
gcc -g -o executableName cSourceFileName.o  
anyOther.o ...
```

```
$ gcc -g -o first first.c
```

```
# Compile first.c producing the compiled object  
# named "first.o".
```

```
$ gcc -c -g -o first.o first.c
```

```
# Link first.o with standard libraries, producing  
# an executable named "first".
```

```
$ gcc -g -o first first.o
```

```
# Link two compile object files with standard  
# libraries, producing an executable named "p6"
```

```
$ gcc -g -o p6 cs1713p6Driver.o cs1713p6.o
```

Executing

You will use a Linux Command Shell to execute linked executables. Linux uses a `PATH` environment variable to tell it where to look for commands. Your executable code is just another command to Linux. Syntax for invoking a command that isn't on the `PATH`:

```
path/executableName commandArguments
```

```
# Invoke the p6 executable which is in the  
# current directory. Using "./" tells Linux  
# to look in the current directory for the command  
# This example uses the file "p6Input.txt"  
# as the standard input.  
$ ./p6 <p6Input.txt
```

Logging Off

If you used `ssh` to remote access our Linux servers, enter the **logout** command.

If you have a Linux desktop (which uses Gnome), select the power icon on the upper right side and then select the logout option.

If you are using Linux 14.04 in our CS Lab, click the bottom left button and then select logout.

```
# To logout when you remotely accessed Linux, use logout  
$ logout
```

To logout of a Linux Desktop:

Guest session
Lock screen

Log Out...

Suspend

Hibernate

Restart...

Shut Down...