

```
1. bash
Shell bash
thunter@thuntermba:~/Sites/home-notebook-node (master) $ glg
* 6de4879 - (HEAD, origin/master, master) removing unused config section (6 minutes ago) <Thomas Hunter>
* a245be6 - merge (9 minutes ago) <Thomas Hunter>
| \
| * 36b0000 - linked up landing pages (11 minutes ago) <Joshua T Kalis>
* | 0ffdb3c - merged maybe (12 minutes ago) <Thomas Hunter>
| \ \
| | /
| * d519133 - forms to prototypes... (13 minutes ago) <Joshua T Kalis>
| * 5f54ff5 - fixing blunder (15 minutes ago) <Joshua T Kalis>
| * e40bb2b - blah (20 minutes ago) <Joshua T Kalis>
| * 872a706 - Merge branch 'master' of qlgithub:thunter/home-notebook-node (26 minutes ago) <Joshua T Kalis>
| | \
| | * | cd3fe1c - removing landing pages from ql3.0 (44 minutes ago) <Joshua T Kalis>
* | | cc4b03e - Auto-Reconnecting to MySQL if connection is lost (14 minutes ago) <Thomas Hunter>
| | /
| / |
* | 88fdca3 - Verbose messages for MySQL connection (28 minutes ago) <Thomas Hunter>
* | 9f2d728 - starting server and loading modules after connecting to db, running a query when visit '/' (30 minutes ago) <Thomas Hunter>
* | c1ad7f0 - passing arguments into controller as one main object (34 minutes ago) <Thomas Hunter>
* | aaaef65 - Successful MySQL connection (42 minutes ago) <Thomas Hunter>
* | 01776ed - mysql (55 minutes ago) <Thomas Hunter>
| /
* f9ef42b - ignoring node_modules (64 minutes ago) <Joshua T Kalis>
* 01aee34 - moved from whiskers templating to handlebars (69 minutes ago) <Joshua T Kalis>
* 0486088 - beginning to ignore npm packages (69 minutes ago) <Joshua T Kalis>
* f5286a2 - removing no longer used dependencies (2 hours ago) <Thomas Hunter>
* 013626d - Init (2 hours ago) <Thomas Hunter>
thunter@thuntermba:~/Sites/home-notebook-node (master) $
```

ATMS 305 WEEK 1: INTRODUCTION TO UNIX/LINUX

Lecture 1: linux/unix basics

LINUX IS THE WORKHORSE THAT BRINGS YOU THE INTERNET AND COMPUTATION

Invented in the 1960's, UNIX, and then LINUX, is the operating system that powers the majority of servers today. It is an operating system that allows for executing computer code, storing files, and accessing the internet. It also has an optional graphical user interface (GUI).

Why?

- Mostly free and “open source”
- Works on almost all PCs
- Mac OS is a flavor of linux
- Windows 10 (post 2016 Anniversary Update) has a linux kernel
- Huge user community that is generally helpful
- Most computing software/internet servers are easiest to use on linux

LINUX HAS MANY “FLAVORS”

Types of “kernels” – a.k.a. the “brand of linux”

- Fedora
- Ubuntu
- Linux Mint
- CentOS → what we will be using
- Debian/OS X
- OpenSUSE

Types of “shells” (available on any kernel) – terminal syntax for issuing commands

- csh
- tcsh
- sh
- bash → what we will be using
- ksh

ACCESSING THE LINUX/BASH TERMINAL

Local machine:

PC: Bash for Windows 10 (need Anniversary Edition, still in beta), otherwise there are some bash interpreters for Windows (not recommended)

Usually we use a 3rd party ssh application.

For its simplicity, I recommend putty.exe. Get it here if you don't have it:

<http://go.illinois.edu/getputty> After download, just drag it to your desktop.

OS X: Terminal.app (it is in Applications/Utilities – or just use spotlight to search)

linux desktop: Terminal is available, accessed through the menus

ACCESSING THE LINUX TERMINAL IN THIS COURSE

We will use the Atmospheric Sciences cluster `keeling.earth.illinois.edu` in this course.

This host is behind the campus firewall, so if you are off campus, you need to use the CITES VPN software available from the Software Webstore to enable access. The hostname `vpn.cites.illinois.edu`. Use the ‘Split Tunnel’ profile.

Connect using the secure shell or `ssh`, log in with your NetID and password:

PC: Use one of the terminal emulators on the previous slide.

OS X/linux: From a terminal window, type

```
ssh netID@keeling.earth.illinois.edu
```

```
λ Cmder
KEELING

Documentation:
https://wiki.illinois.edu/wiki/display/manabeccluster/keeling+Home
Please contact help@earth.illinois.edu for additional support.

REMINDER:
This is the login node for keeling, and it is shared among multiple
users. Do not run parallel programs requiring more than four compute
threads on this machine; please use the batch system for such programs
instead.

12/6/2016
Per-group CPU core limits have been updated in the SLURM batch system.

Checking quota on home file system...
Disk quotas for user snesbitt (uid 393079):
  Filesystem blocks quota limit grace files quota limit grace
san-a1-ib:/data/keeling/a
          46512M 51200M 52224M      473k      0      0
snesbitt@keeling ~$ |
```

ssh.exe Search

WOO HOO! NOW WHAT?

NOW YOU'RE READY TO ROLL, BUT ONE SEC

A few one time setup items:

Change your shell to bash

Type `chsh`

Enter netID password

Enter `/bin/bash` when prompted.

Now you're running with the bash shell.

A WORD ABOUT KEELING

Keeling is named after Charles Keeling, the UIUC grad that discovered that CO₂ was rising due to anthropogenic sources, established Mauna Loa Observatory in Hawaii

Keeling (the computer) is a linux cluster operated by the Department of Atmospheric Sciences supported by SESE Computer Services.

We have been given computing accounts for class purposes only on this machine. Please respect our privileges on this machine and do not abuse the resources.

We have a class data directory in the folder `/data/atms305/a/netID` which has about 5 GB of space per user.

We also have the ability to use the batch queueing system to run jobs on the cluster. Interactive jobs are run with the command `qlogin`. Try to run `qlogin` when possible.

BASH ON LINUX: NOW WHAT?

Linux allows you to run commands, write scripts that can perform tasks automatically or repetitively, interact with the file system, and access the internet, among other things you would expect a computer to do. Using the terminal, we can run commands that are at a very low level in the operating system.

However, the terminal requires knowledge of basic linux commands.

Also, to generate and edit scripts and files, you need to be able use a text editor.

To use the internet from the terminal, you need other commands.

Let's go through some useful commands!

EDITING FILES

There are several text editors available on linux, including pico, vi, and emacs.

pico is probably the easiest, but least powerful.

It has a keyboard menu system that helps guide you through the process of saving, opening, finding and replacing text, etc.

Example: `pico test.txt`

This will open test.txt for editing (if you have permissions, we will talk about that later) 😊

TOP 15 LINUX COMMANDS

1: cd

The "Change Directory" command enables you to navigate to another directory.

The cd command is THE most important command there is in linux i think. As the command suggests, it enables the user to change / jump to a directory.

Example: cd Downloads

Example: cd /etc/

Example: cd .. (Directory up!)

note: After you have typed cd and entering the first or two letters of the directory you can press the TAB key to autocomplete the directory! so... cd Do (press tabkey) will autocomplete it to Downloads.

TOP 15 LINUX COMMANDS

#2: `man`

The `man` command shows the users the "manual" of the command. In some situation you might need to get more information about the command you are using. The `man` command shows you this information about the command.

Example: `man cp`

This will open up the "`cp`" manual document for us in the shell. The manual shows us the parameters available for the commands.

note: To close the manual simply press "`Q`".

TOP 15 LINUX COMMANDS

#3: ls

In the absolute top 15 there's no way the ls command is missing. On the third place, the ls command. The ls command is used to list the files/directories within a directory.

Example: `ls`

It shows us the directories and files available.

Another example: `ls -a`

This shows the directories and “details” about the files.

TOP 15 LINUX COMMANDS

#4: cp

The cp command is available for us to "Copy" things. This might be usefull for duplicating files f.e.

Example: `cp file file2`

note: file is on this case the file the user wants to be copied... file2 is the name of the copied file. It's that simple.

Another example: `cp -R /data/directory/test /data/directory/test2`

This will copy the directory test into the directory test2

TOP 15 LINUX COMMANDS

#5 mv

The mv command is used for "Move" operations. The mv command enables the users to move a file/directory to a specified location.

Example: `mv /home/bas/Desktop/bla /home/bas/Desktop/blahblah`

note: the first part of the command is the file that has to be moved. The second part (after the whitespace) is the target directory. Make sure you type in the path you intend to use with this command!

Another example: `mv -fi /data/keeling/a/snesbitt/testdirectory /data/keeling/a/ahuang11`

This will move the directory and everything in it called testdirectory to the specified path.

TOP 15 LINUX COMMANDS

#6 mkdir

This command is used to "make" directories, NOT Files.. (this is possible with another command which I will bring up later in this tutorial)

Example: `mkdir testdirectory`

note: The name of the directory is case sensitive which means that Testdirectory is a complete different directory as testdirectory.

TOP 15 LINUX COMMANDS

#7 rmdir

When you are able to make directories, you also want to know how to remove them. Removing directories is done by the rmdir command and belongs to the absolute basic commands in shell scripting.

Example: `rmdir testdirectory`

note: When the directory is not empty the command will prompt an error message:

```
rmdir: failed to remove `testdirectory': Directory not empty
```

So make sure it's completely empty before removing it.

TOP 15 LINUX COMMANDS

#8 touch

Now we know how to make directories and deleting them, i now want to use the touch command. The touch command is used to make files.

Example: `touch vogel`

This will create the file vogel for us in the directory.

TOP 15 LINUX COMMANDS

#9 rm

The rm command stand for remove. The rm command is used in order the delete files instead of directories.

Example: `rm vogel`

note: To remove files you must have the right permission bits set on the specific file.

Another example: `rm -rf Downloads`

Remove everything forcibly and recursively, i.e., everything below and including that file.

Be careful with this command, it can ruin your day if not used with care!

`rm -rf /Users/snesbitt` ☹️

TOP 15 LINUX COMMANDS

#10 cat

The cat command gives a listing of a file.

Example: `cat test.txt`

This will print the file to the terminal.

See also more, less, head, and tail.

TOP 15 LINUX COMMANDS

#11 **grep**

The command `grep` searches for matching strings in files.

Example: `grep test file.txt`

This will list the matching lines in the files.

TOP 15 LINUX COMMANDS

#12 history

The history command will list the commands you have entered, so you can find what you have done already.

Example: `history`

A number is displayed next to the command. You can either use the up arrow to go back to the previous command, or type an exclamation point followed by the job number in the history.

Example: `!45`

TOP 15 LINUX COMMANDS

#13 top

The top command tells you what the most intensive processes are currently running on the system.

Example: `top`

A table of the top cpu using commands is listed, as well as other relevant information, including the process identification number (or PID).

`kill [PID]` will kill an executing program (which you have permission to halt).

TOP 15 LINUX COMMANDS

#14 pwd

What is my current directory? pwd will tell you!

Example: pwd

It just tells you the path in the file system at the current location.

Useful if you are lost!

TOP 15 LINUX COMMANDS

#15 chmod

In linux, files have 3 levels of users (yourself, a group, and everyone), and 3 levels of access (read, write, and execute).

You can set the permissions of the files you own using the chmod command. The simplest approach is to assign numbers to each capability (4 for read, 2 for write, 1 for execute) and then to use addition when combining numbers. Thus, $6 = 4+2$ (read plus write permission), and $7 = 4+2+1$ (all three permissions added together).

Within this framework, you set permissions for a file by specifying the desired capabilities for the user, group, and world (in that order). Thus, when she set up her directory for CSC 105 above, Ms. Coahran might have issued the command

```
chmod 755 csc105
```

Here, the user (Ms. Coahran) has full permissions ($7 = \text{read} + \text{write} + \text{execute}$); while the others can read and execute, but not write (5).

Use `ls -a` to inspect permissions.