Introduction to Linux

Technical Session

Fall 2017

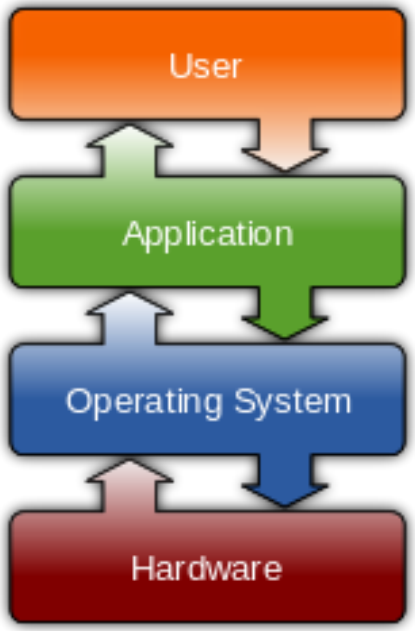
Tae-Hyuk (Ted) Ahn
Department of Computer Science
Saint Louis University

SAINT LOUIS UNIVERSITY™

EST. 1818

Linux drives the computer hardware

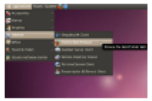
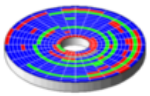
- Linux is an operating system (OS) that is system software to manage computer hardware and software resources.
- The operating system is an essential component in a computer system.



- The development of Linux is one of the most prominent examples of free and open-source software collaboration.

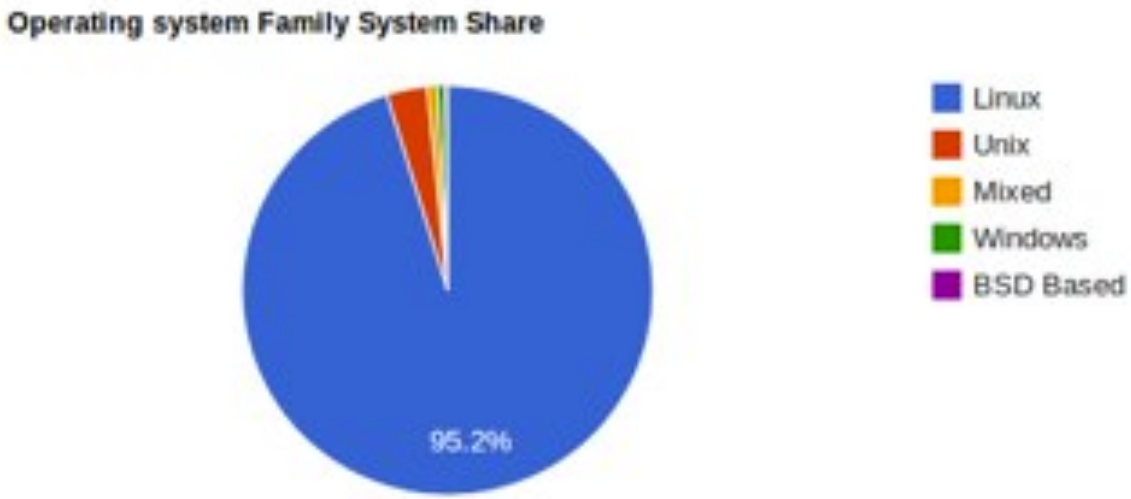
Why Linux?

- Stable, safe, and adjustable

 Forget about viruses.	 Is your system unstable?	 Linux protects your computer.	$$ Don't pay $100 for your operating system.	 No more c**pware.	 Freedom!
 When the system has installed, why would you *still* need to install stuff?	 Forget about drivers.	 Update all your software with a single click.	 Why copy software illegally if you can get it for free?	 Need new software? Don't bother searching the web, Linux gets it for you.	 Jump into the next generation of desktops.
 Does your digital life seem fragmented?	 Choose what your desktop looks like.	 Why does your Windows get slower day after day?	 Do something for the environment.	 No back doors in your software.	 Enjoy free and unlimited support.

Why Linux?

- 95 percent of the world's top 500 supercomputers run Linux



Popular Linux Distributions

- Red Hat Enterprise Linux
- Fedora
- CentOS
- Debian
- Ubuntu
- Suse Linux
- Linux Mint

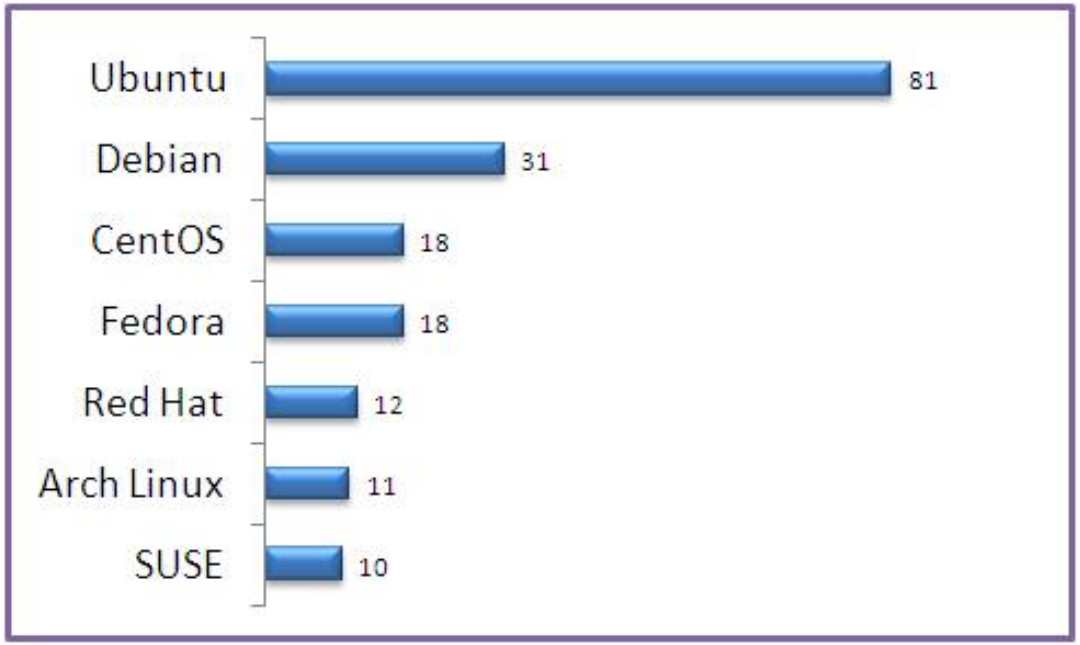
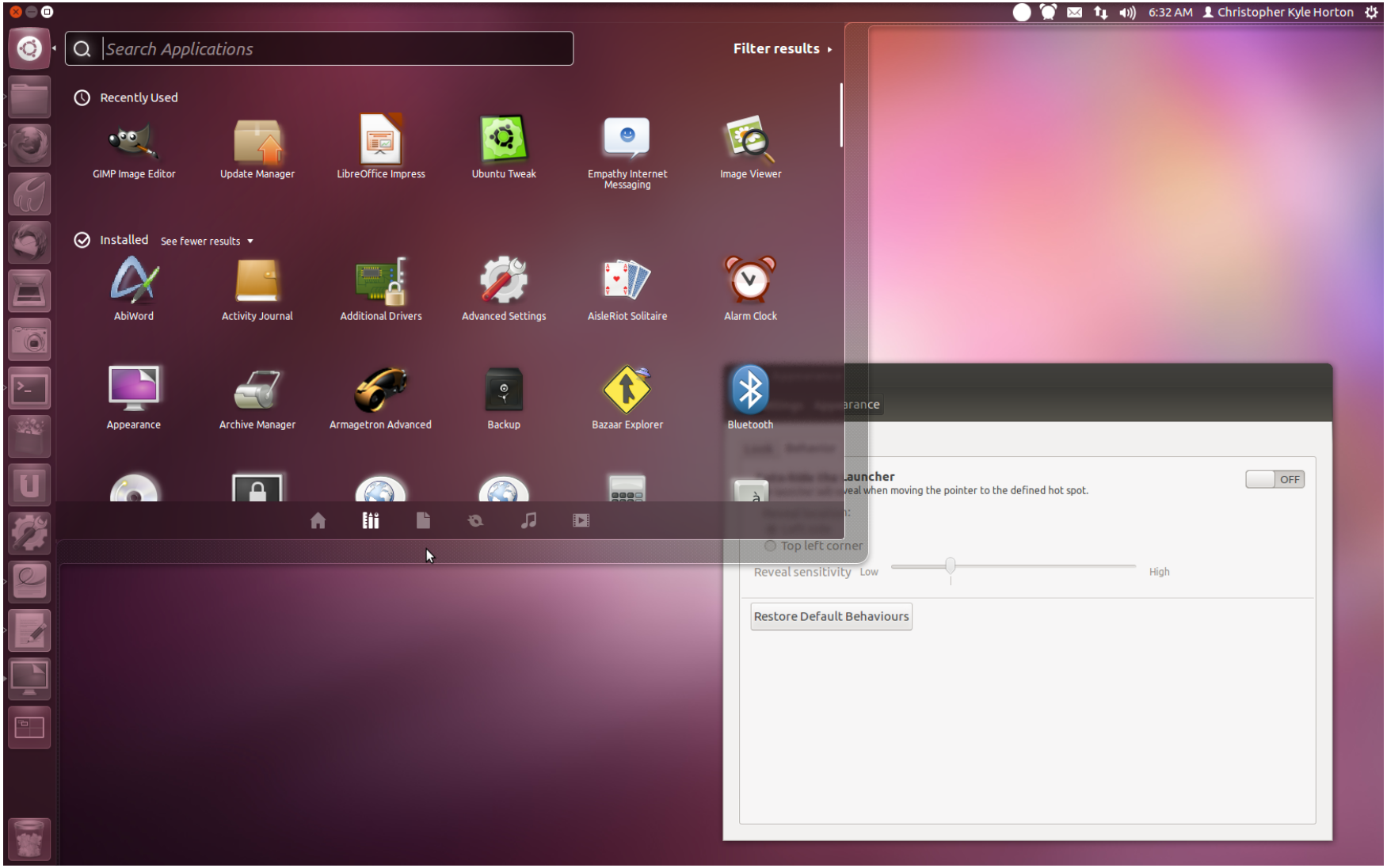


Fig: Favorite Linux Distribution Voting Results

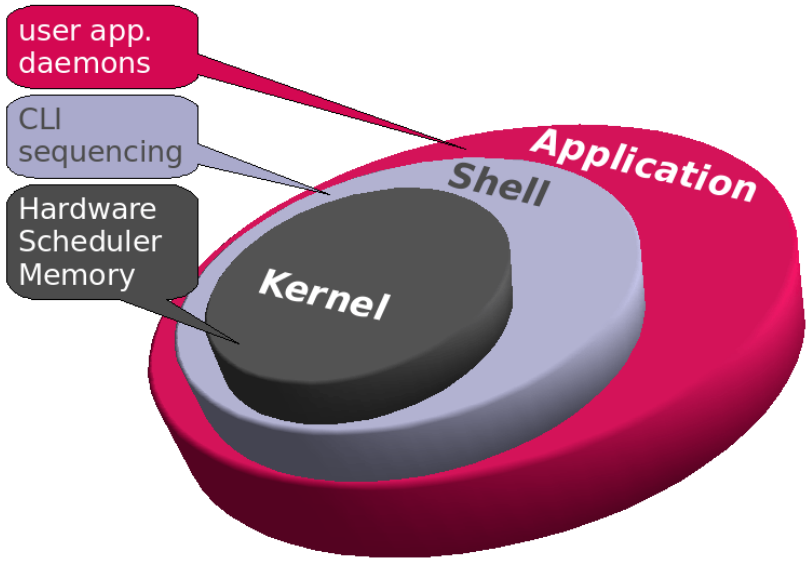
Is it Linux?


6:32 AM
Christopher Kyle Horton
Search Applications
Filter results
Recently Used
GIMP Image Editor
Update Manager
LibreOffice Impress
Ubuntu Tweak
Empathy Internet Messaging
Image Viewer
Installed
See fewer results
AbiWord
Activity Journal
Additional Drivers
Advanced Settings
AisleRiot Solitaire
Alarm Clock
Appearance
Archive Manager
Armagetron Advanced
Backup
Bazaar Explorer
Bluetooth
OFF
Top left corner
Reveal sensitivity
Low
High
Restore Default Behaviours

Terminal

```
[root@localhost ~]# ping -q fa.wikipedia.org
PING text.pmtpa.wikimedia.org (208.80.152.2) 56(84) bytes of data.
^C
--- text.pmtpa.wikimedia.org ping statistics ---
1 packets transmitted, 1 received, 0% packet loss, time 0ms
rtt min/avg/max/mdev = 540.528/540.528/540.528/0.000 ms
[root@localhost ~]# pwd
/root
[root@localhost ~]# cd /var
[root@localhost var]# ls -la
total 72
drwxr-xr-x. 18 root root 4096 Jul 30 22:43 .
drwxr-xr-x. 23 root root 4096 Sep 14 20:42 ..
drwxr-xr-x.  2 root root 4096 May 14 00:15 account
drwxr-xr-x. 11 root root 4096 Jul 31 22:26 cache
drwxr-xr-x.  3 root root 4096 May 18 16:03 db
drwxr-xr-x.  3 root root 4096 May 18 16:03 empty
drwxr-xr-x.  2 root root 4096 May 18 16:03 games
drwxrwx--T.  2 root gdm  4096 Jun  2 18:39 gdm
drwxr-xr-x. 38 root root 4096 May 18 16:03 lib
drwxr-xr-x.  2 root root 4096 May 18 16:03 local
lrwxrwxrwx.  1 root root   11 May 14 00:12 lock -> ../run/lock
drwxr-xr-x. 14 root root 4096 Sep 14 20:42 log
lrwxrwxrwx.  1 root root   10 Jul 30 22:43 mail -> spool/mail
drwxr-xr-x.  2 root root 4096 May 18 16:03 nis
drwxr-xr-x.  2 root root 4096 May 18 16:03 opt
drwxr-xr-x.  2 root root 4096 May 18 16:03 preserve
drwxr-xr-x.  2 root root 4096 Jul  1 22:11 report
lrwxrwxrwx.  1 root root    6 May 14 00:12 run -> ../run
drwxr-xr-x. 14 root root 4096 May 18 16:03 spool
drwxrwxrwt.  4 root root 4096 Sep 12 23:50 tmp
drwxr-xr-x.  2 root root 4096 May 18 16:03 yp
[root@localhost var]# yum search wiki
Loaded plugins: langpacks, presto, refresh-packagekit, remove-with-leaves
rpmfusion-free-updates                                   | 2.7 kB     00:00
rpmfusion-free-updates/primary_db                        | 206 kB     00:04
rpmfusion-nonfree-updates                                | 2.7 kB     00:00
updates/metalink                                         | 5.9 kB     00:00
updates                                                  | 4.7 kB     00:00
Updates/primary_db                    73% [=============          ]  62 kB/s | 2.6 MB     00:15 ETA
```

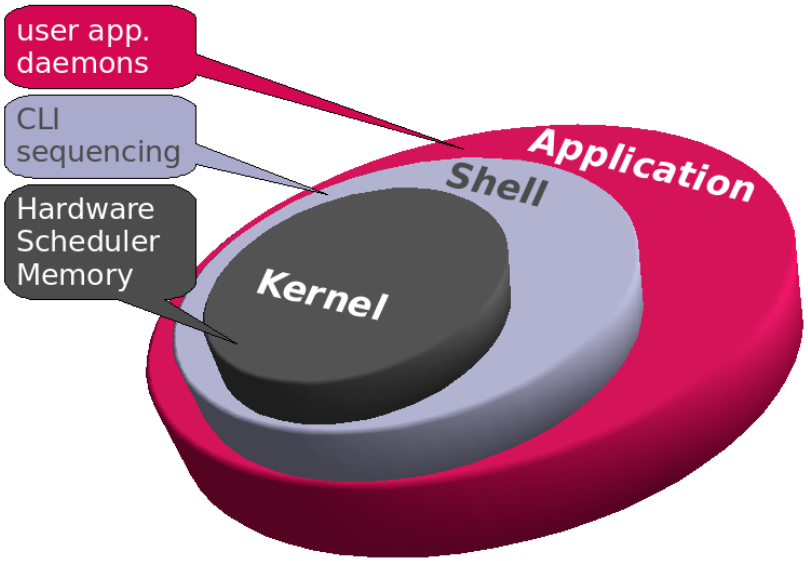
Concepts: Kernel



- Operating system "kernel" is the core software used to "talk" to computer hardware
- Is a core and modular system of drivers used to create a standardized environment for interfacing with hardware

- Kernel operates in its own memory or "kernel-space"
- Responsible for allocating memory and time to system and user processes as well as interacting with files.

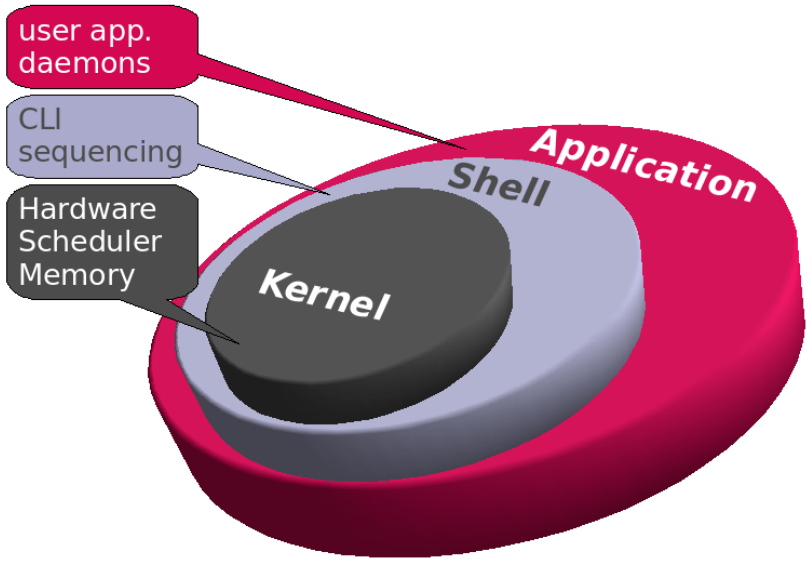
Concepts: Shell



- On log-in, the system runs a shell
- A shell is the environment within which you will interface with the kernel via commands
- It determines the syntax for complex command-line operations and shell scripting

- The shell you're using is called "bash," the successor to the venerable "Bourne Shell" called "sh"
- BASH: "Bourne Again SHell"

Concepts: Various Shell



- sh – the original UNIX shell
- bash – written as a replacement/ extension of sh
- csh – C shell based on the C programming language developed in the late 1970s
- tcsh – enhanced version of C shell
- ksh – Korn shell developed in the early 1980's, backward compatible with sh, but with some features of csh

Linux Account

- To access a Linux system, you need to have an account
- A Linux account includes the following:
  - username and password
  - uid and gid
  - a home directory, which is where you are placed by default when you log in
  - a default shell

Let's jump into Linux

- Log-in to the Linux machine in the room with your account and password

- Any problem?

Using SSH to log in remote

- SSH is the "Secure SHell"
- All traffic over SSH is encrypted
- SSH supports a file-transfer subsystem
- Need to know the hostname or IP address with your account

```
ssh username@hostname
```

```
$ ssh ahnt@hopper.slu.edu
```

- hoppoer.slu.edu: CS and BCB gateway server
- If you have a MAC, search/open a terminal application and connect to remote machine using the SSH (no install required)
- If you have a Windows, install SSH Client software (e.g., Putty or SSHSecureShellClient) and log-in using the hostname/account.
  - If you use Windows, send me email. Then, I will send you SecureShellSSH client.

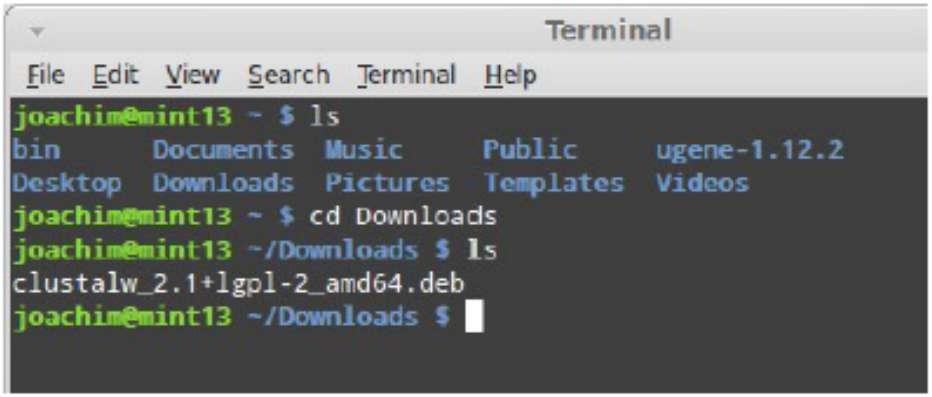
The home and root folder

- If you log-in to a Linux, then Linux leads you to your home directory.
- Type `$ pwd` and press enter. What do you see? That's your home directory.
- Type `$ ls` to list the content.
- Type `$ ls -al` to list all in detail.

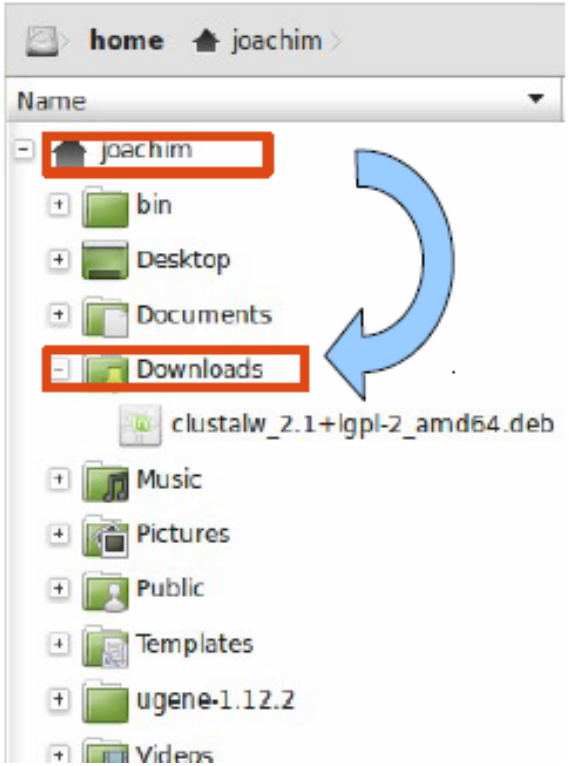
Navigating in the file system

```
$ cd Downloads
```

- cd : change directory



The result is that the prompt has changed position from:



Basic Linux (Unix) commands

Commonly Used Linux Commands	
Command	**Description**
man	display details about an instruction
ls	list contents of the current directory
ls -l	detailed listing of directory contents, shows permissions, owner, etc.
ls -a	list all files (including hidden files)
ls -la	detailed listing of all files (note that options can be combined)
cd	change directory
cd ../	backup one level from the current directory
pwd	print current working directory
touch	create an empty file
mkdir	make a new directory
rm	remove files and directories
rm -rf	recursively remove all files and directories under the specified directory
cat	list file contents
less	list file contents – one screen at a time
tail	list the end of the file (default – displays last 10 lines)
tail -n	list the last n lines of a file
cp	copy file
mv	rename a file
echo	echo values to the screen example: echo $PATH – prints the value of the PATH variable
grep	command line text search utiltity example: grep blue colors.txt – list all lines with the word blue from the colorlist.txt file
ps	list currently running processes

Common Steps to run Application on Linux

- Download a program and move it to a working directory.
- Mostly, the program is compressed format. So, extract the compressed file.
- Build/Install the program.
- Set the PATH environment to run the program in anywhere.
- Run the program with input files.

Compress / Decompress the file(s)

	Compressing files	Decompressing files
tar	Not compressing, just archiving `$ tar -cvf mydir.tar mydir/`	Dearchiving `$ tar -xvf mydir.tar`

zip (.zip)	`$ zip mydata.zip mydata.fasta`
	`=> mydata.zip`
	`$ zip -r mydir.zip mydir`
		`$ unzip mydata.zip`

gzip (.gz)	`$ gzip mydata.fasta`
	`=> mydata.fasta.gz`
	`$ tar -zcvf mydir.tar.gz mydir/`
	`=> mydir.tar.gz`
		`$ gunzip mydata.fasta.gz`
		`$ tar -zxvf mydir.tar.gz`

bzip2 (.bz2)	`$ bzip2 mydata.fasta`
	`⇒ mydata.fasta.bz2`
	`$ tar -jcvf mydir.tar.bz2 mydir/`
	`=> mydir.tar.bz2`
		`$ bunzip2 mydata.fasta.bz2`
		`$ tar -jxvf mydir.tar.bz2`

Lab 1: Install a program and run it

- Make a directory “Course” at your home directory.
- Make a directory “BCB5200” at “Course” directory.
- Make a directory “Software” at “BCB5200” directory.
- Bowtie2 program is already installed in hopper, but we will install it again at local.
- Download Bowtie2 alignment software (search on google) and install it at “Software” directory.
  - If you downloaded the “bowtie2-2.2.9-linux-x86_64.zip” file, then just unzip it.
- Go to ~/Course/BCB5200/ and make a directory “Lab1”
- Copy Bowtie-directory/example/reference/lambda_virus.fa and Bowtie-directory/example/reads/reads_1.fq files into the “Lab1” directory.
- Index the reference and align the read file into the reference index.
  - Refer the bowtie2 website
- Report overall alignment rate.

Another Task: Be familiar to Terminal Editor

- If you are familiar to one of terminal editors such as “Nano”, “VI(M)”, “Emacs”, and so on, just use it.
- If you don’t know any terminal editor, I strongly recommend you to use VIM editor.
- Search VIM editor cheat sheet.
- VIM editor example.
  - Open a file named “README.txt”
  - Add a sentence “SLU BCB program is awesome.”
  - Save and quit.

Setting Environment Variables

- Our default shell is Bash and can be used for the following purposes:
  - Configure look and feel of shell.
  - Setup terminal settings depending on which terminal you're using.
  - Set the search path such as JAVA_HOME, and ORACLE_HOME.
  - Set environment variables as needed by programs.
  - Run commands that you want to run whenever you log in or log out.
  - Setup aliases and/or shell function to automate tasks to save typing and time.
  - Changing bash prompt.
  - Setting shell options.

Setting Environment Variables

- Display current environment

```
$ echo $HOME
$ echo $PATH
```

- Set environment variable (in the terminal session)

  Use **export** command

```
$ export PATH=/faculty/ahnt/Course/BCB5200/
Software/bowtie2-2.2.9:${PATH}
```

- How do I make all settings permanent?

  `$ vi ~/.bash_profile`

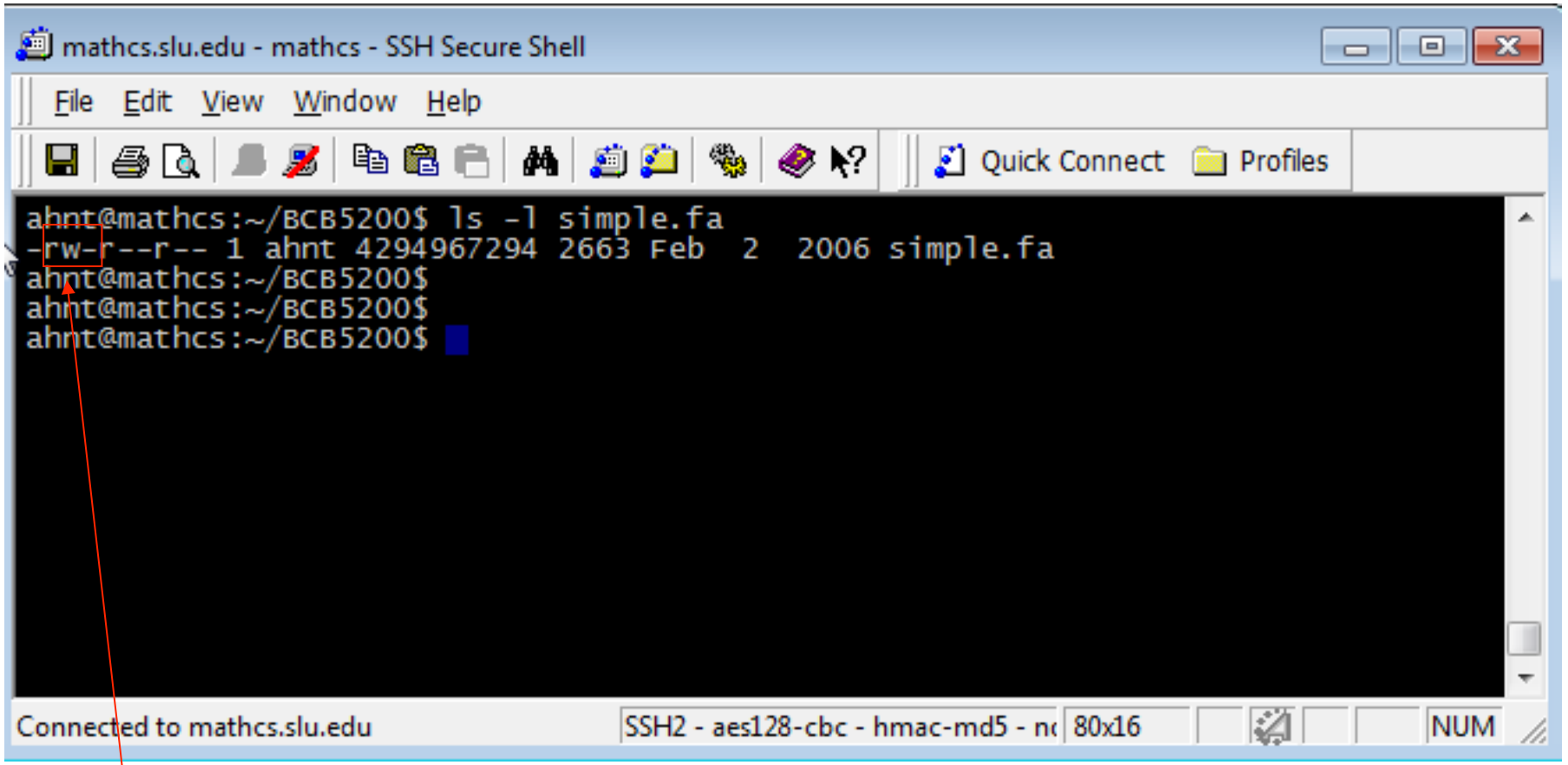
  Appending $PATH settings, save, and close.

```
export PATH=/faculty/ahnt/Course/BCB5200/
Software/bowtie2-2.2.9:${PATH}
```

File permission

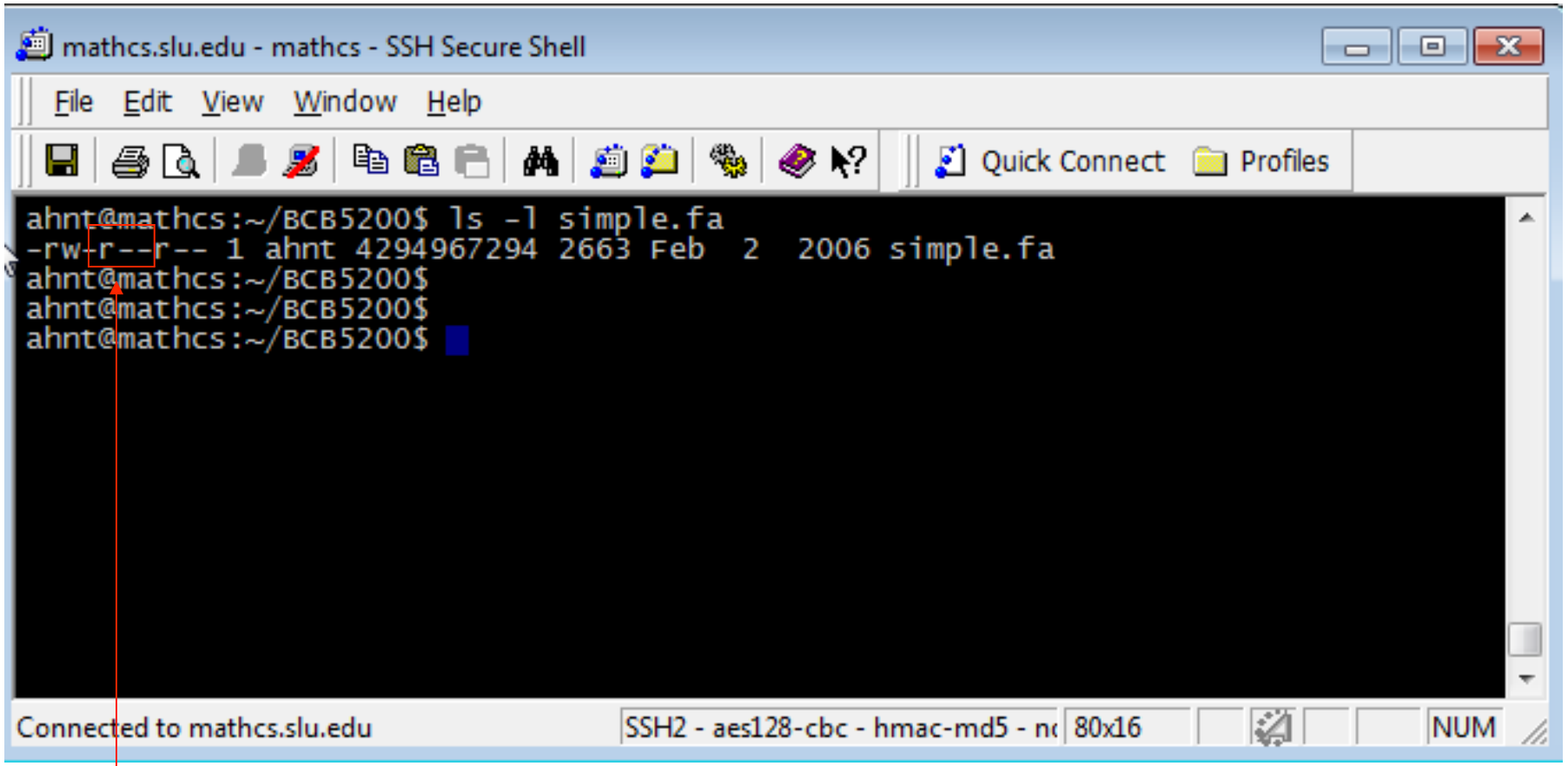
- Each file in Unix/Linux has an associated permission level
- This allows the user to prevent others from reading/writing/ executing their files or directories
- Use “ls -l filename” to check the current permission level
- “r” means “read only” permission
- “w” means “write” permission
- “x” means “execute” permission
- `$ id` can check my user id and group id

File Permissions



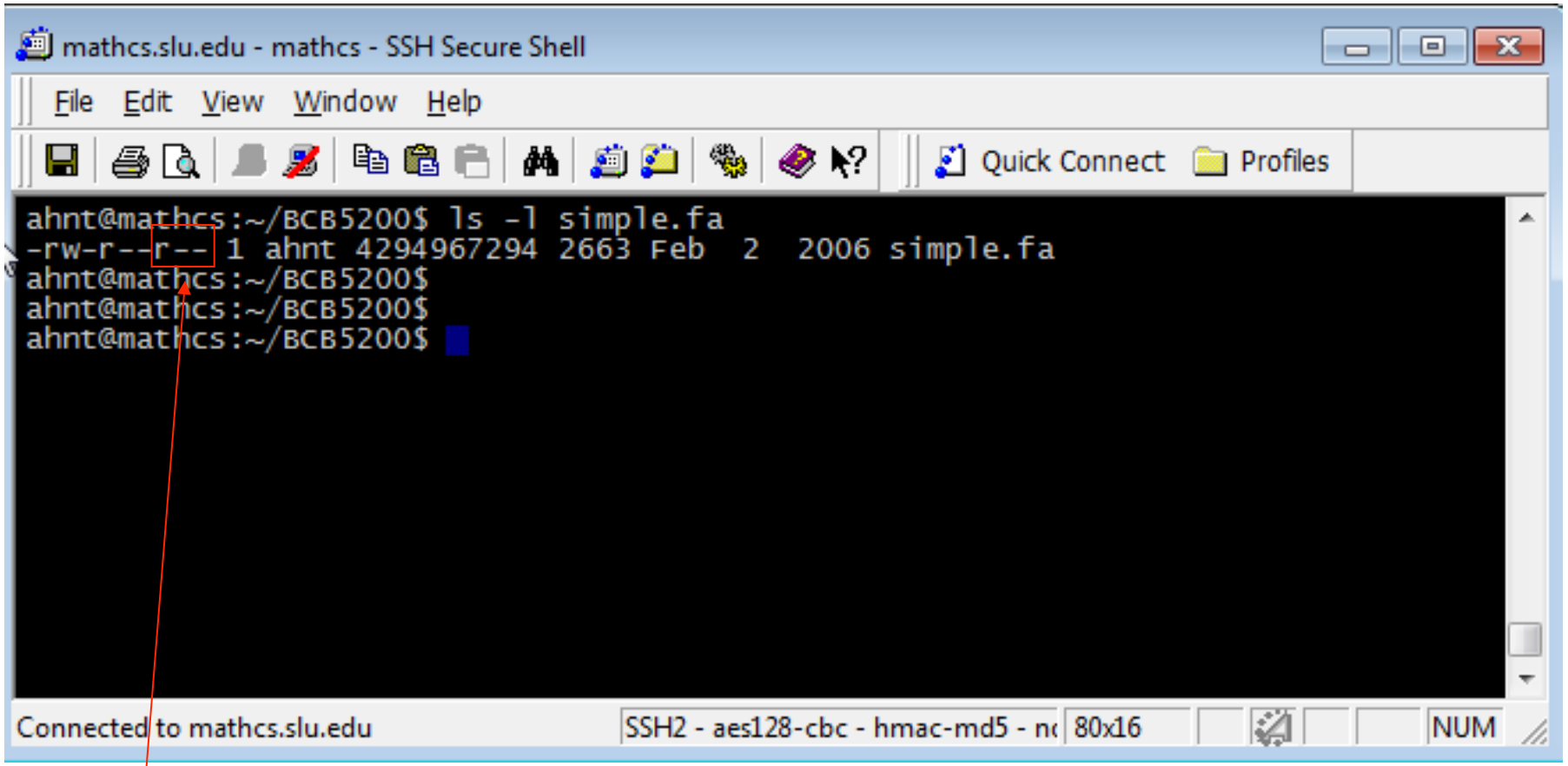
User (you)

File Permissions



Group

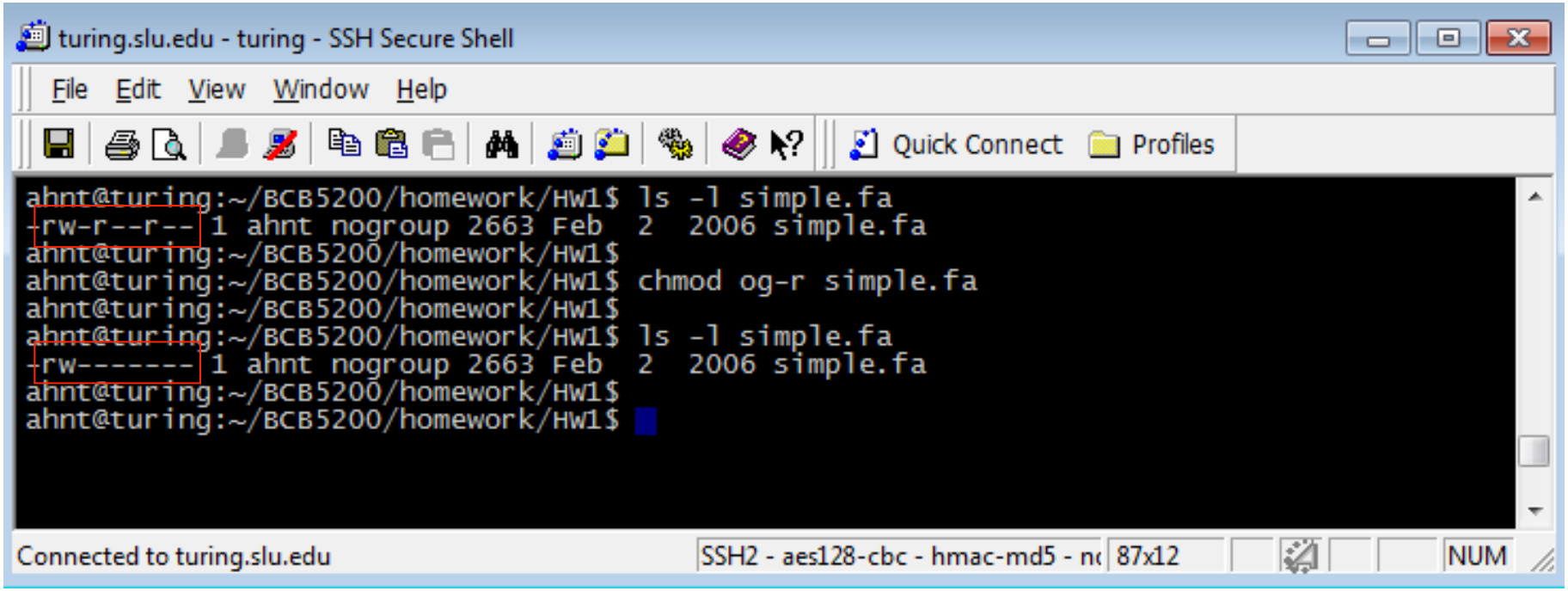
File Permissions



"Others"

Change file permission using chmod

- If you own the file, you can change it's permissions with "chmod"
  - Syntax: chmod [**u**ser/**g**roup/**o**thers/**a**ll]+[permission] [file(s)]
  - Syntax: chmod [**u**ser/**g**roup/**o**thers/**a**ll]-[permission] [file(s)]
  - Below we grant execute permission to all:



Changing Permissions

- **chmod** – change permissions on a file or directory
- **chgrp** and **chown** – change group ownership to another group (only the superuser can change the owner)
- Both options support '-R' for recursion.

What everyone else is up to

- **`top`** – show a detailed, refreshed, description of running processes on a system.
- **`ps`** – display process information on the system
- **`uptime`** – show the system load and how long the system has been up.
- To view only the processes owned by a specific user

```
$ top -U [username]
or
$ ps -u [username]
or
$ ps -ef | grep [username]
```

Killing Badly Behaving Processes

- Commands or programs on the system are identified by their filename and by a process ID which is a unique identifier. Then, kill the process
  - Check the process ID using ps or top
  - $ kill [pid] – terminates the process id
  - $ kill -9 [pid])
  - ^c (control+c) terminates the running program
  - ^d (control+d) terminates your session.
- You can only kill your process, not others.
- Only the superuser (root) has permissions to kill other's processes

Background Run

- Often we must run a command in the background with the ampersand '&' character

```
$ command –options &
```

runs command in background, prompt returns immediately

- Run a Command or Shell-Script Even after You Logout

```
$ nohup command –options &
```

Useful commands

- find – utility to find files
- cat – utility to print files to standard out (monitor)
- head – prints the first 10 lines of a file
- tail – prints the last 5 lines of a file
- history – prints your command history
- | – Using a pipe operator '|' commands can be linked together. The pipe will link the standard output from one command to the standard input of another.
- wc – word count
- grep – search pattern
- du – check the file and directory size
- scp or rsync – transfer files between systems

Lab 2

1. Make a directory ~/Course/BCB5200/Lab2

2. Copy /public/ahnt/Course/BCB5200/Lab2/sim01.fna.gz into the Lab2 directory.

3. Decompress sim01.fna.gz

4. Count the number of lines in the fasta file.

5. Count the number of reads in the fasta file (one read starts with identifier > read name followed by sequences).

6. Search a read that contains "ATCGGCAAATCAATCATTGATGCTTCATCGACCACCAGCACGTCC" sequence string and get the read ID for it.

You should save all commands that you used to solve the problems with answers.

Hint: Check Linux commands "wc" and "grep" to solve this homework.