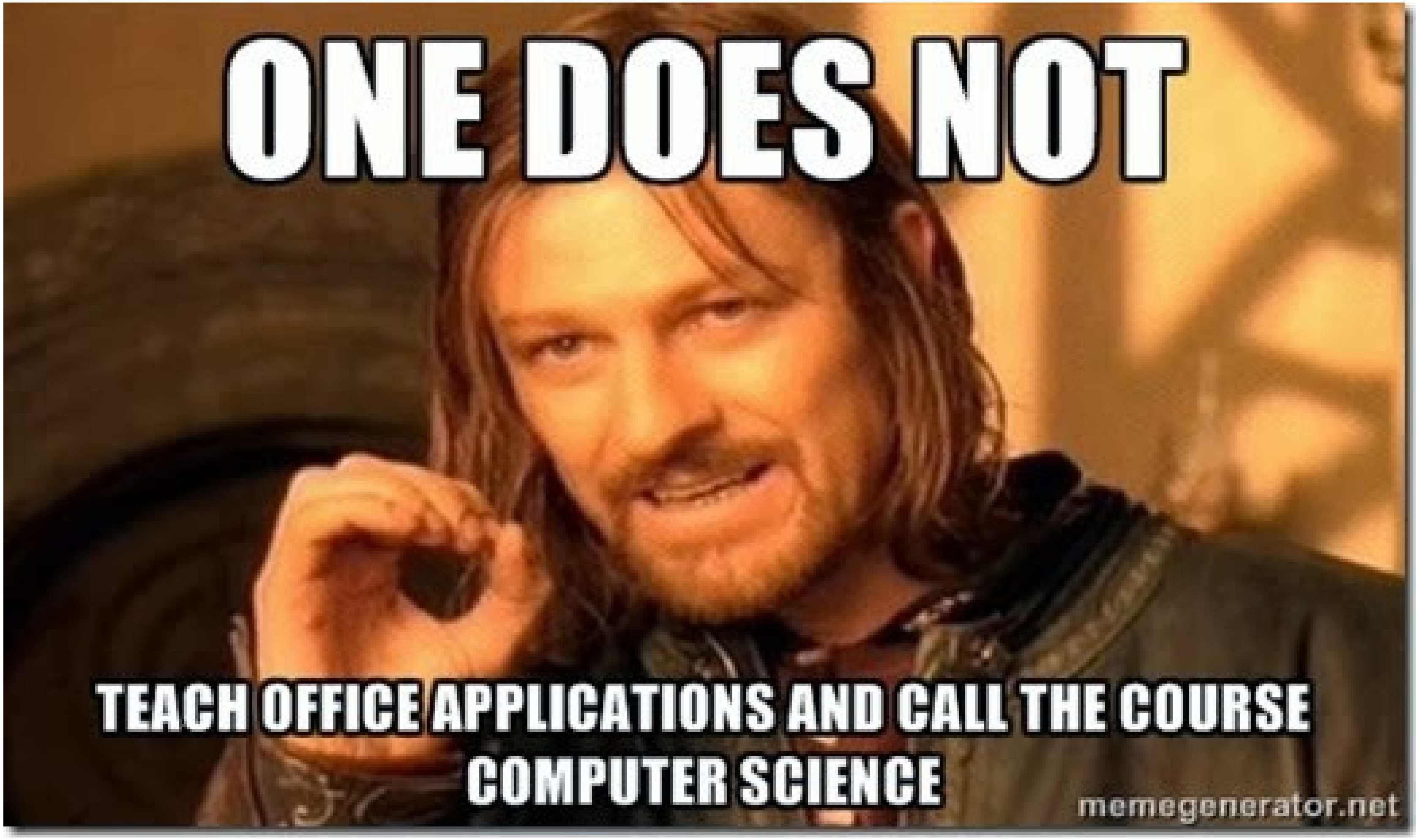

ONE DOES NOT
TEACH OFFICE APPLICATIONS AND CALL THE COURSE COMPUTER SCIENCE
memegenerator.net

# Development in Linux

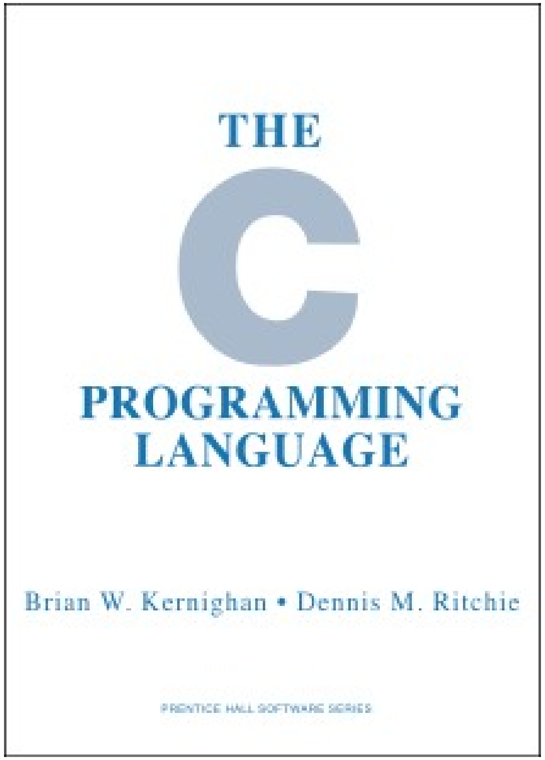
CS2308
Gentry Atkinson

# History of Unix

- In the beginning, every computer manufacturer made their own OS.
- **Multics** was an experimental time-sharing mainframe OS. It was very influential but unwieldy
- **Unix** was developed at **Bell Labs** to correct issues with Multics and was introduced in 1973.
- Bell could not sell Unix due to an anti-trust case, so they gave it away.

# C and Unix

- The **C** language was developed for Unix, and Unix was developed with C.
- Two developers Kernighan  and Ritchie, along with many others, developed Unix and C simultaneously.
- Many commercial and open-source version of Unix exist.


THE
C
PROGRAMMING LANGUAGE
Brian W. Kernighan • Dennis M. Ritchie
PRENTICE HALL SOFTWARE SERIES

# Linux

- In 1991 a Finnish developer named Linus Torvald created his own version of the MINIX operating system, which cam from Unix.
- Many distributions of Linux exist, most are free.
- Linux is highly customizable, and offers a similar graphic interface to Windows and MacOS.
- Very popular with developers.

# The Linux Shell

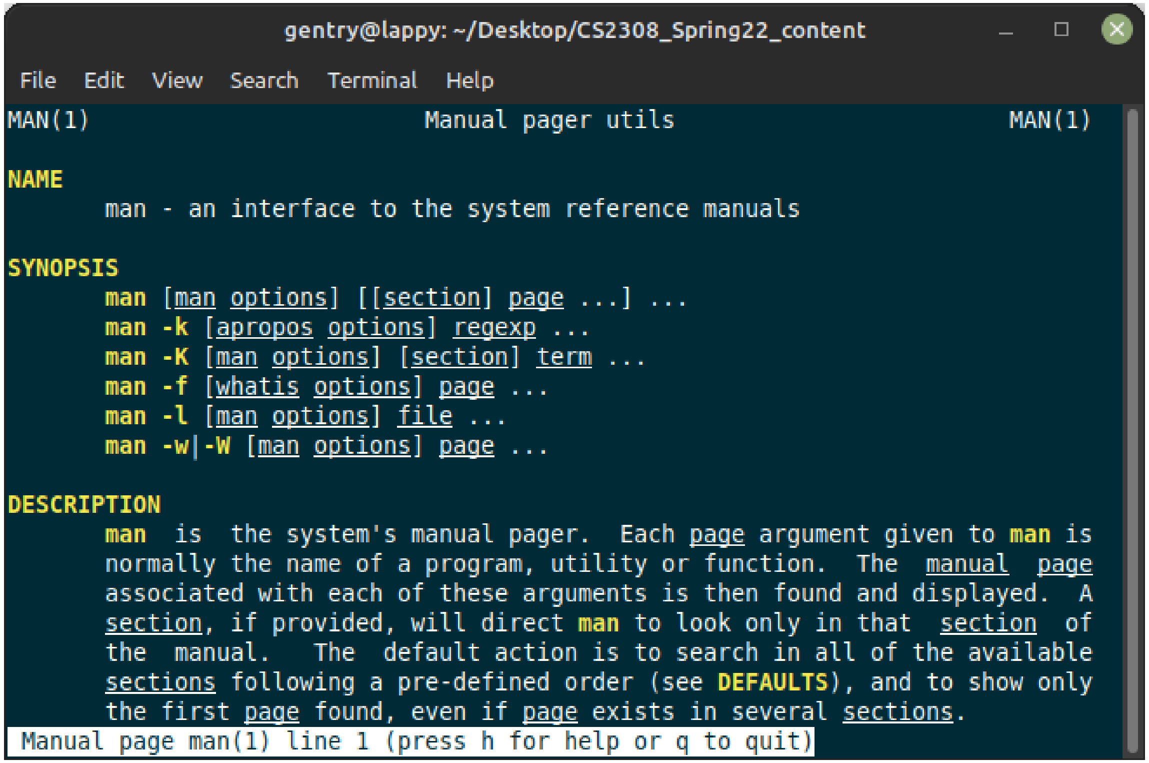
- The human interface of an operating system is called a “shell”
  - GUI: graphic user interface
  - CLI: command line interface
- CLIs have their own language and syntax:
  - Bourne Again SHell (most common)
  - C Shell (C-like syntax)
  - Friendly Interactive Shell (friendly)

# Using the CLI

- The command is always the first word in an instruction to the shell.
- Some commands take “arguments” (like functions do), usually the name of a file or directory.
- Switches are options for changing the behavior of a command, always preceded by a dash -
- Argument files can be relative to the system root or relative to the directory you are “in”

# man Pages

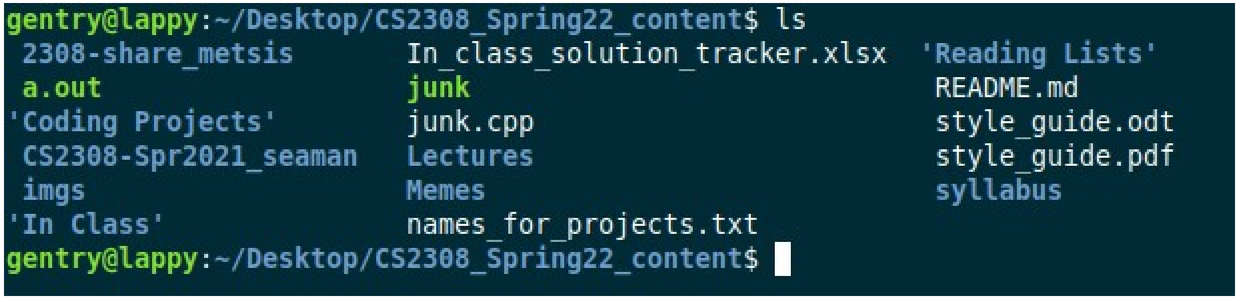
- “man” (short for manual) is a command-line tool for retrieving information about other tools.
- Type **man** followed by another word



```
MAN(1)                        Manual pager utils                       MAN(1)

NAME
       man - an interface to the system reference manuals

SYNOPSIS
       man [man options] [[section] page ...] ...
       man -k [apropos options] regexp ...
       man -K [man options] [section] term ...
       man -f [whatis options] page ...
       man -l [man options] file ...
       man -w|-W [man options] page ...

DESCRIPTION
       man  is  the system's manual pager.  Each page argument given to man is
       normally the name of a program, utility or function.  The  manual  page
       associated with each of these arguments is then found and displayed.  A
       section, if provided, will direct man to look only in that  section  of
       the  manual.   The  default action is to search in all of the available
       sections following a pre-defined order (see DEFAULTS), and to show only
       the first page found, even if page exists in several sections.
 Manual page man(1) line 1 (press h for help or q to quit)
```

# ls

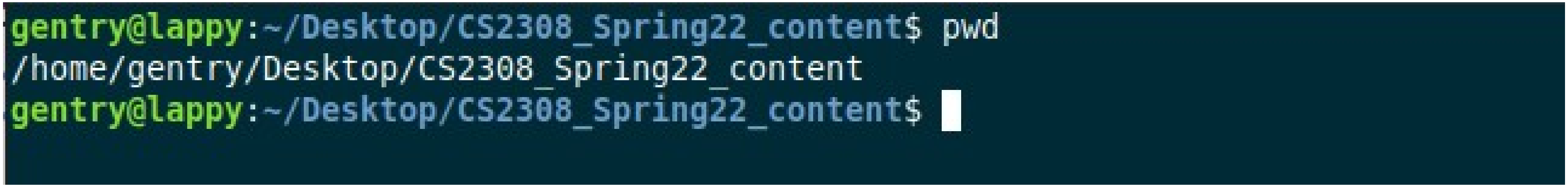
- The ls command is used to list the files in a directory.
- Possible arguments: the name of a directory.
- Possible switches:
  - **-a** show hidden files too
  - **-s** show the size of each file
  - **-f** unsorted



```
gentry@lappy:~/Desktop/CS2308_Spring22_content$ ls
 2308-share_metsis          In_class_solution_tracker.xlsx  'Reading Lists'
 a.out                      junk                             README.md
'Coding Projects'           junk.cpp                         style_guide.odt
 CS2308-Spr2021_seaman      Lectures                         style_guide.pdf
 imgs                       Memes                            syllabus
'In Class'                  names_for_projects.txt
gentry@lappy:~/Desktop/CS2308_Spring22_content$
```

# pwd

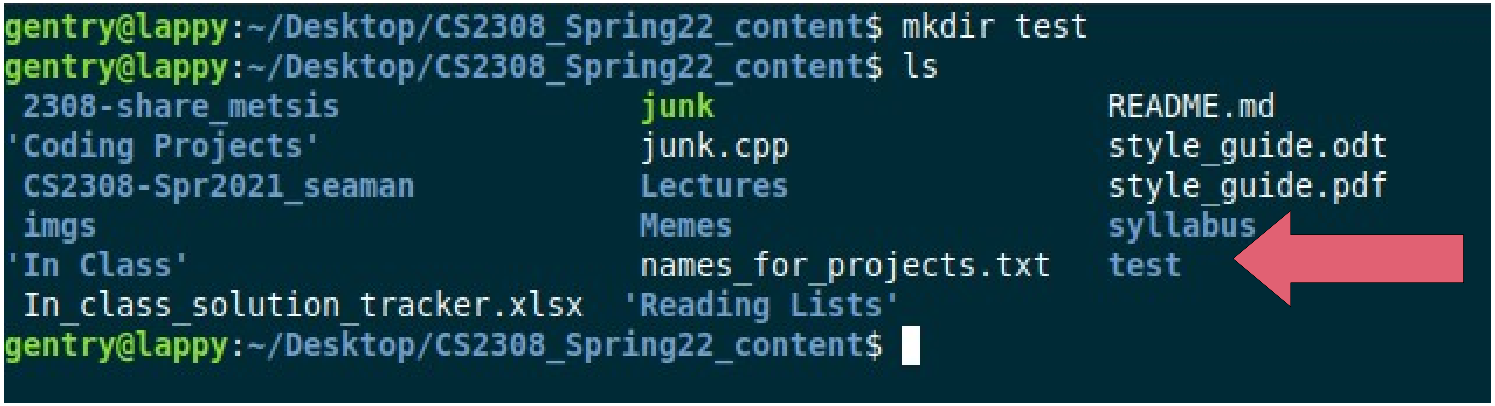
- Used to print the working directory. Shows the directory you are currently in.
- Takes no arguments.
- Can take switches, but none that are easy to explain.



```
gentry@lappy:~/Desktop/CS2308_Spring22_content$ pwd
/home/gentry/Desktop/CS2308_Spring22_content
gentry@lappy:~/Desktop/CS2308_Spring22_content$ 
```

# mkdir

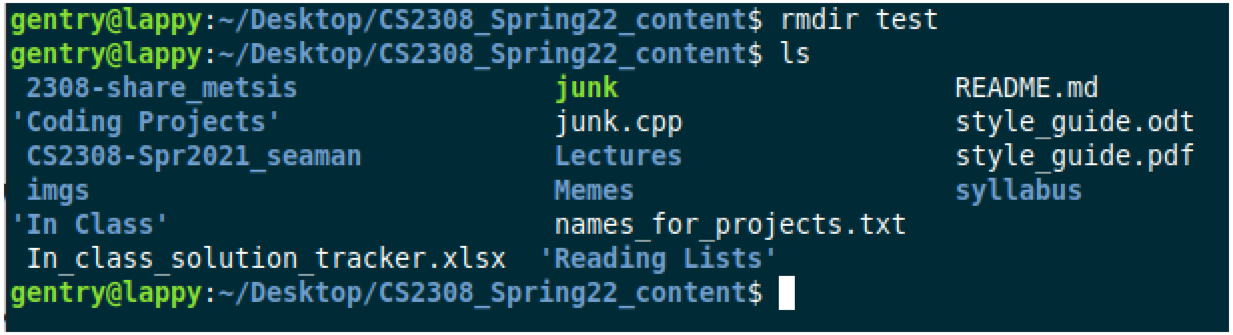
- Make a new directory.
- Mandatory argument, the name of the new file.
- Optional switches:
  - **-m** set access permission (mode) of the file.



```
gentry@lappy:~/Desktop/CS2308_Spring22_content$ mkdir test
gentry@lappy:~/Desktop/CS2308_Spring22_content$ ls
 2308-share_metsis                 junk                     README.md
'Coding Projects'                  junk.cpp                 style_guide.odt
 CS2308-Spr2021_seaman             Lectures                 style_guide.pdf
 imgs                              Memes                    syllabus
'In Class'                         names_for_projects.txt   test
 In_class_solution_tracker.xlsx   'Reading Lists'
gentry@lappy:~/Desktop/CS2308_Spring22_content$ 
```

# rm and rmdir

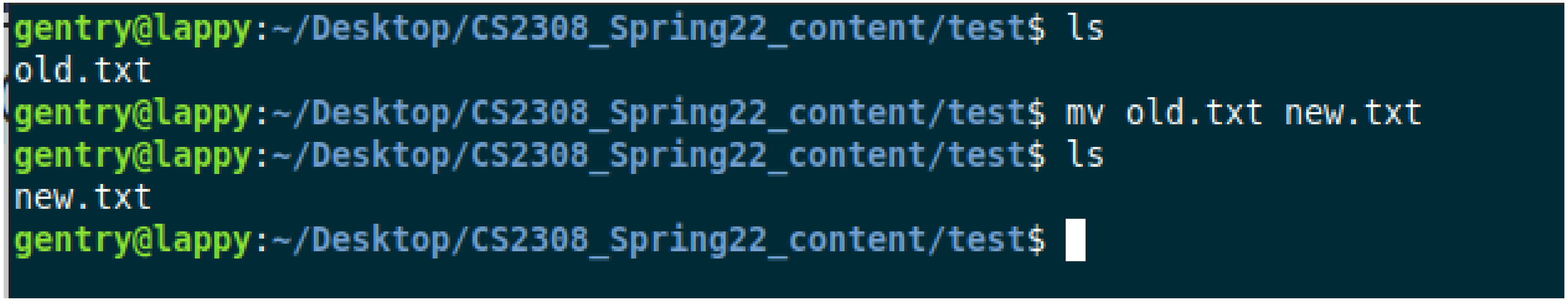
- Remove (delete) a file or directory.
- Mandatory argument, the name of a file or directory.
  - * can be used for “everything”
- Switches:
  - **-r** recursive, delete files inside the deleted directory.
  - **-f** force, make it happen



```
gentry@lappy:~/Desktop/CS2308_Spring22_content$ rmdir test
gentry@lappy:~/Desktop/CS2308_Spring22_content$ ls
 2308-share_metsis                junk                     README.md
'Coding Projects'                 junk.cpp                 style_guide.odt
 CS2308-Spr2021_seaman            Lectures                 style_guide.pdf
 imgs                             Memes                    syllabus
'In Class'                        names_for_projects.txt
 In_class_solution_tracker.xlsx  'Reading Lists'
gentry@lappy:~/Desktop/CS2308_Spring22_content$ 
```

# cp and mv

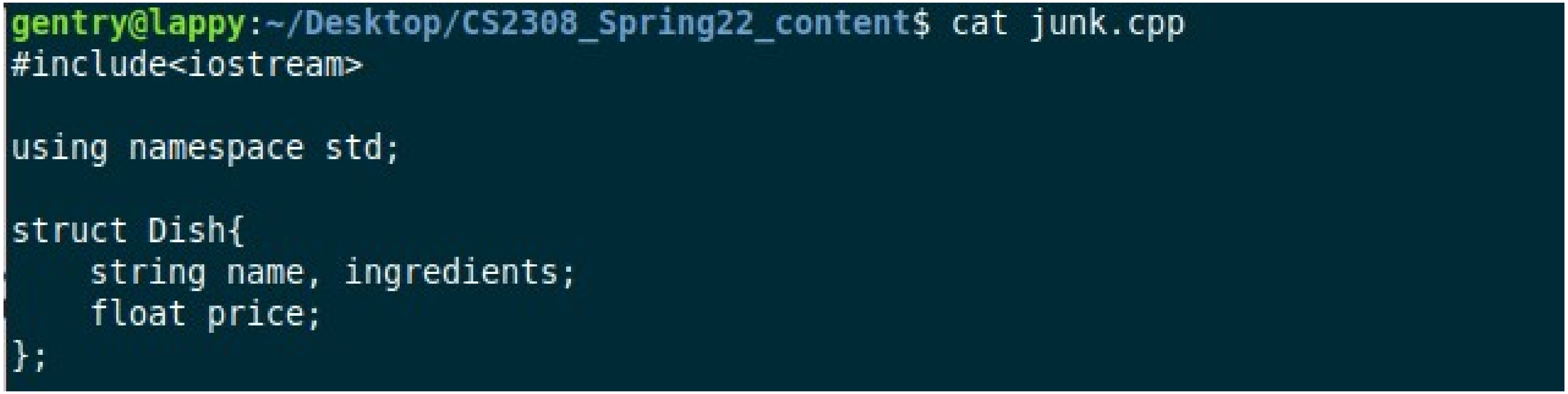
- Copy or move a file. Similar to copy-paste and cut-paste.
- Mandatory arguments: old file name and then the new file name. In that order.
- Switches:
  - **-f** force the action



```
gentry@lappy:~/Desktop/CS2308_Spring22_content/test$ ls
old.txt
gentry@lappy:~/Desktop/CS2308_Spring22_content/test$ mv old.txt new.txt
gentry@lappy:~/Desktop/CS2308_Spring22_content/test$ ls
new.txt
gentry@lappy:~/Desktop/CS2308_Spring22_content/test$ 
```

# cat

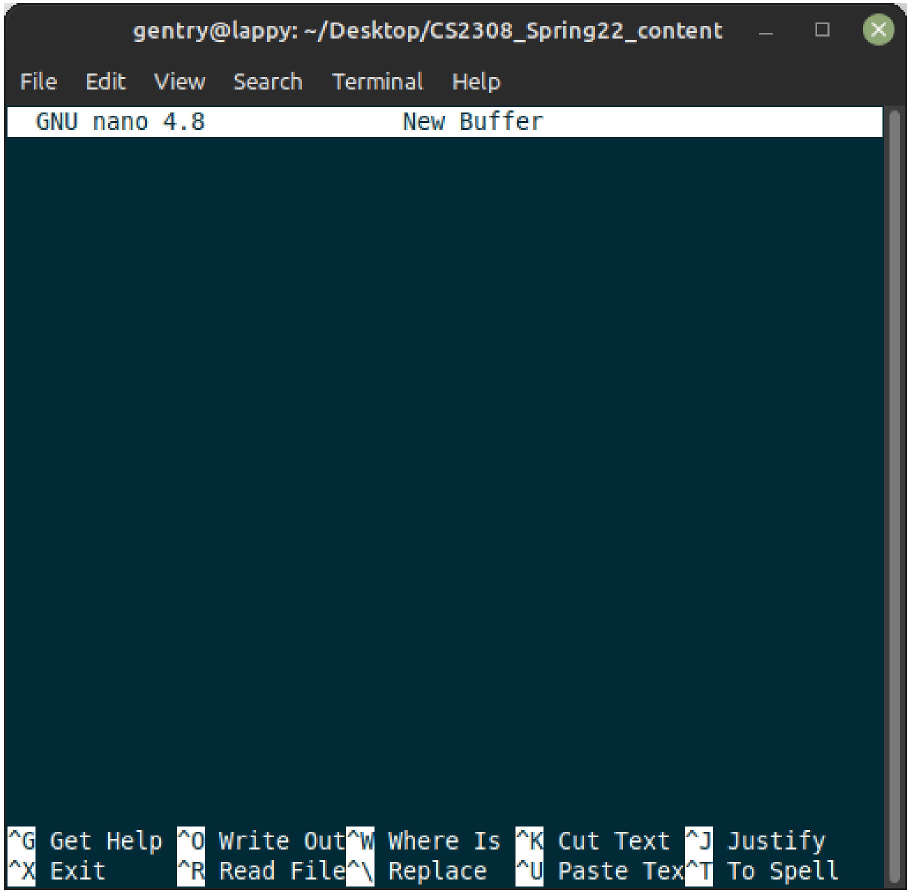
- Prints the contents of a file.
- Mandatory argument, the name of a file.
- Switches:
  - **-n** numbers output lines



```
gentry@lappy:~/Desktop/CS2308_Spring22_content$ cat junk.cpp
#include<iostream>

using namespace std;

struct Dish{
    string name, ingredients;
    float price;
};
```

# nano

- Launch the nano text editor.
- Optional argument, the name of file to edit.
- Switches:
  - **-B** backup old versions
  - **-D** bold text
- Other tools are listed in the editor.

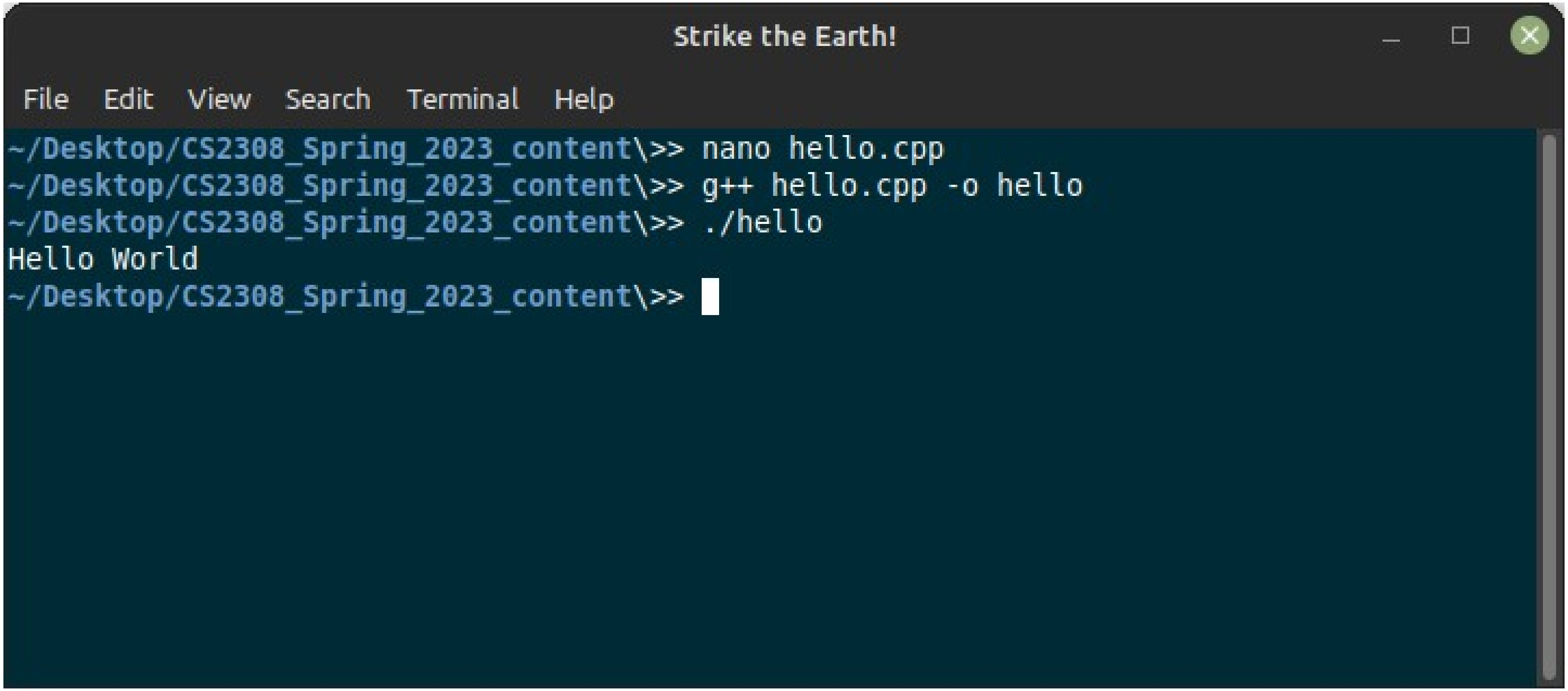


```
  GNU nano 4.8                 New Buffer

^G Get Help  ^O Write Out^W Where Is  ^K Cut Text  ^J Justify
^X Exit      ^R Read File^\ Replace   ^U Paste Tex^T To Spell
```

# g++

- Runs the GNU C Compiler (gcc) on a C++ file.
- Mandatory argument, a file to compile.
- Switches:
  - **-o** names the output (executable) file
  - many, many more.
- Outputs a program called **a.out** if the -o switch is not used.

# g++



```
~/Desktop/CS2308_Spring_2023_content\>> nano hello.cpp
~/Desktop/CS2308_Spring_2023_content\>> g++ hello.cpp -o hello
~/Desktop/CS2308_Spring_2023_content\>> ./hello
Hello World
~/Desktop/CS2308_Spring_2023_content\>>
```

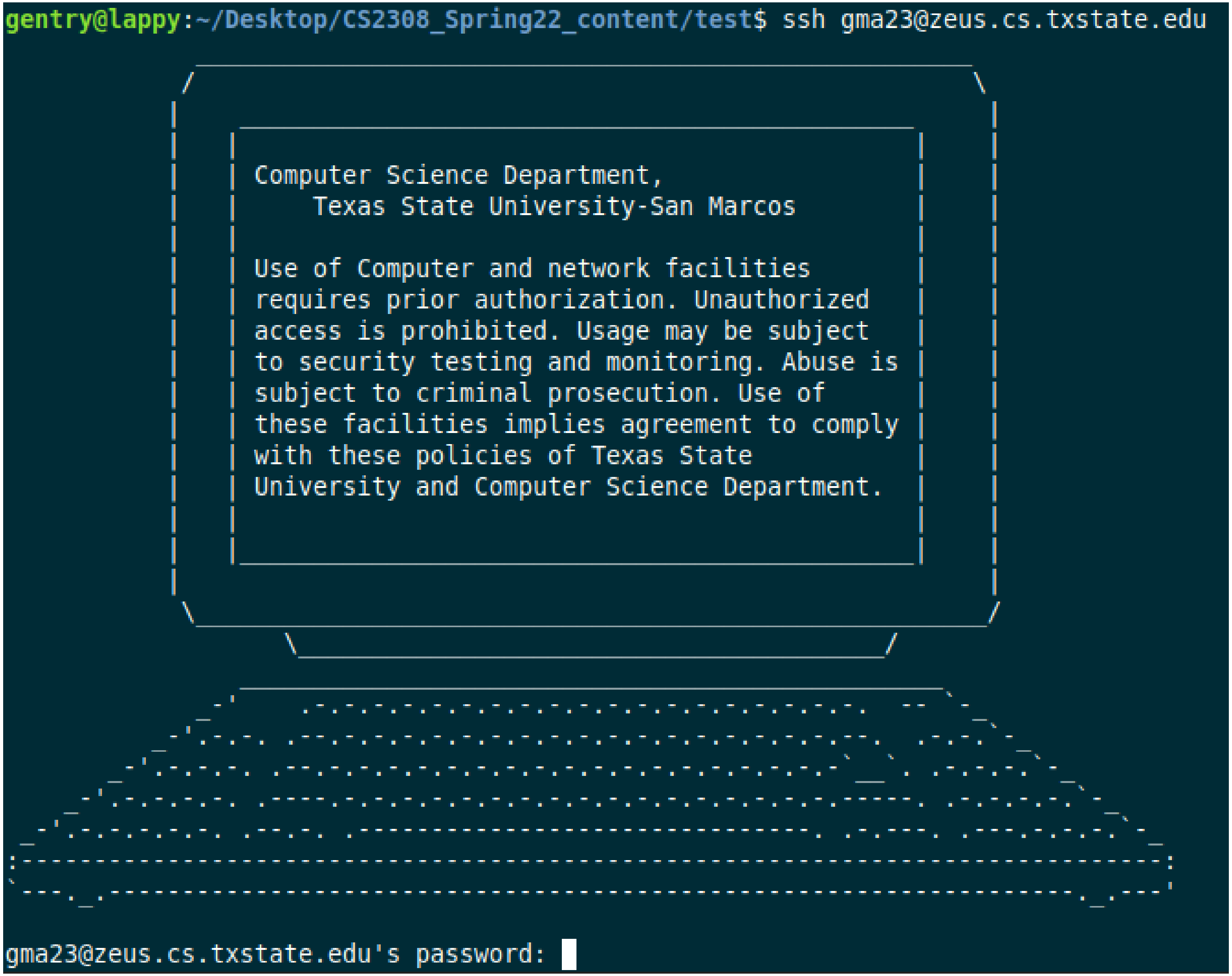
# Linux File System

- The file system is a tree that branches upwards from the root: /
- Each user has a **home** which is referenced using a ~
- . is the current directory
- .. is the parent of the current directory
- ./program runs a program in the current directory.

# Our Linux servers

- Use **putty** from Windows or the **ssh** command line tool on Mac or Linux.
- Connect to:
  - netID@zeus.cs.txstate.edu OR
  - netID@eros.cs.txstate.edu
- Type your password when prompted. You will not see the input (not even *******).
- https://cs.txstate.edu/resources/labs/accounts/linux/

# Successful Connection



```
gentry@lappy:~/Desktop/CS2308_Spring22_content/test$ ssh gma23@zeus.cs.txstate.edu
             _____________________________________________________
            /                                                     \
           |    ______________________________________________     |
           |   |                                              |    |
           |   | Computer Science Department,                 |    |
           |   |     Texas State University-San Marcos        |    |
           |   |                                              |    |
           |   | Use of Computer and network facilities       |    |
           |   | requires prior authorization. Unauthorized   |    |
           |   | access is prohibited. Usage may be subject   |    |
           |   | to security testing and monitoring. Abuse is |    |
           |   | subject to criminal prosecution. Use of      |    |
           |   | these facilities implies agreement to comply |    |
           |   | with these policies of Texas State           |    |
           |   | University and Computer Science Department.  |    |
           |   |                                              |    |
           |   |______________________________________________|    |
           |                                                       |
            \______________________________________________________/
                   \________________________________________/
                ________________________________________________
             _-'    .-.-.-.-.-.-.-.-.-.-.-.-.-.-.-.-.-.-.-.  -- `-_
          _-'.-.-. .--.-.-.-.-.-.-.-.-.-.-.-.-.-.-.-.-.-.--.  .-.-.`-_
       _-'.-.-.-. .--.-.-.-.-.-.-.-.-.-.-.-.-.-.-.-.-.-.-`__`. .-.-.-.`-_
    _-'.-.-.-.-. .----.-.-.-.-.-.-.-.-.-.-.-.-.-.-.-.-.-----. .-.-.-.-.`-_
 _-'.-.-.-.-.-. .--.-. .-------------------------------. .-.---. .---.-.-.-.`-_
:------------------------------------------------------------------------------:
`---._.------------------------------------------------------------------._.---'

gma23@zeus.cs.txstate.edu's password: █
```

# Questions or Comments?