Lecture Notes:

This includes the beginning of when Silvestro goes over directory commands. Stops when he hits octal notation.

fox01: ~$ **\ls -la**

- directory listing where -l gives long-form listing
- "-a" gives hidden files

a lot of single letter utilities can be combined
so instead of -l -a, we can do -la or -al
"-" is short option
"--" is long option

stdin/stdout/stderr

- file descriptor is how the OS identifies open files.
- CTRL-D sends the end of file byte to the remote host.

  Example:
  This will read from stdin (keyboard) and print to stdout (computer screen)

  ```
  fox01:~$ cat
  line one
  line one
  line two
  line two
  CTRL-D
  ```

- CTRL-D recognizes that there's no more data to read, flushes it's buffer, and exits. (Voluntary)
- CTRL-C sends a signal to terminate the process. (Forceful)

  ```
  fox01:~$ cat > mytext.txt
  #cat will read from the stdin (my keyboard) and secretly write to stdout. It parses the file before recognizing whether or not it already exists, so it'll create the file or obliterate an already existing one.
  this is a test
  this is line two
  fox01:~$ cat mytext.txt
  this is a test
  this is line two
  fox01:~$ cat > mytext.txt
  this is a NEW test
  this is a new line two
  final line.
  fox01:~$ cat mytext.txt
  this is a NEW test
  ```

this is a new line two
final line.

- Things are only displayed when stdout goes to your terminal. If it's redirected to the file, it'll go to your file.
- To avoid file obliteration and append, we use ">>"

```
fox01:~$ ./stderr
This is standard output
This is standard error
fox01:~$ ./stderr > stdout.txt
This is standard error
fox01:~$ cat stdout.txt
This is standard output
fox01:~$ ./stderr 2> stderr.txt
This is standard output
fox01:~$ cat stderr.txt
This is standard error
fox01:~$ ./stderr &> stdall.txt
fox01:~$ cat stdall.txt
This is standard error
This is standard output
```

0 stdin
1 stdout
2 stderr

```
fox01:~$ ./stderr 1> stdout.txt
This is standard error
```

This is pointless, but just to show using the file descriptor numbers.

more fileNm (pager utility)
Lets you view a long file in your scrollback buffer. Using it will show you one line at a time each time you hit ENTER, or SPACE for one page at a time.

less fileNm
Same as above, but extra features including going between beginning and end, searching, up/down arrow keys.

When using \ls we see output of all our files in columns, but using ls -l displays everything in rows with extra metadata.

ex:
-rw------ 1 namehere faculty 596 Jan 22 13:14 test.c
- The hyphen in “-rw-...” represents the type of file it is. If it's a hyphen it’s a regular file. There is “s” for sockets, “p” for pipes, etc. The next line characters after are the traditional Linux file permissions. Determines who can read, write, and execute this file.

- “1” is the hard-link account that represents the number of directory entries that point to this same file’s data.
Most regular files will have 1, Directories will have atleast 2.
“namehere” is the owner, “faculty” is the owning group, “596” is the file size, “Jan 22 13:14” is the timestamp.

```
fox01:` $ vi whoson.bash
#creates/opens a file named whoson

file: whoson.bash
#!/bin/bash
date
echo “who is on?”
who

fox01:` $ whoson.bash #this searches a bunch of different directories
-bash: ./whoson.bash: Permission denied
fox01:` $ ./whoson.bash
#says look for a file in my current directory named whoson.bash
fox01:` $ chmod u+x whoson.bash
#allows you to change permissions so this says allow “user” + “who owns it” to execute whoson.bash
fox01:` $ ls whoson.bash
-rwx---- 1 namehere faculty etc…
#now ./ or simply typing it will work too.
```