# Comp 280 Notes:

To access the virtual machine on the Witt computers:

Wittenberg student
Password: #Learn2Linux

# Processes:

Basically, anything that is running is a process.

Fork - a process that starts a new process by forking. Forking is creating a copy of the thing that is currently running. The original is the parent, and the new one created is the child. When a process forks, a clone is made. The child process will be assigned a new PID number.

Exec - a way for a process to execute another program. Often a parent will fork and a child will exec another program.

init - the init daemon - the first process run by the Linux system. It has the PID (process identification number) of 1. The init daemon is responsible for starting all processes on the Linux system. The init is the parent or ancestor of everything in the system. NOW, Linux is replacing init with systemd.

PID - every time a process forks, the child will be given a new PID number.

ps -e shows every process
pstree -p shows parent-child relationships (shows the tree)

Fork and sleep - the we run a command in the shell the shell will fork a new shell that runs the command, and the original shell sleeps (does nothing) until the new shell running the command is finished,

Can use the '&' to run commands in the background, then the shell will not sleep.

**Processing a command:**
First, we perfor, History expansion. Hurn history command into executable command.

Ex:
$ !!
!! - run the previous command

Second, the shell performs Alias Substitution. If an alias is the first word in a simple command, the shell will replace it with its aliased value.

Ex:
If la is aliased to ls -a
$ la
5 la gets changed to
$ ls -a

Next is parsing the command.
The shell will scan the line and parse it (separate) into tokens. After parsing into tokens, the shell performs Command Line Expansion.

Command Line Expansion is broken into 9 parts, executed in the following order:
1. Brace Expansion
2. Tilde Expansion replaces - with your home folder
3. Parameter and Variable Expansion
4. Arithmetic Expansion
5. Command Substitution
6. Word Splitting
7. Pathname Expansion
8. Process Substitution
9. Quote Removal

Order of expansion affects the outcome of a command:
Brace Expansion - Expands braces that contain a comma-separated list of tokens. It can be used to match filenames or to generate arbitrary strings.

The syntax for braces: {t1,t2, t3, ..., tn} - Tokens separated by commas, and no whitespace.
touch {a,b,c}.txt -makes a.txt, b.txt, and c.txt - good to make things faster

Tilde Expansion - replaces ~ with your home folder. If the ~ is followed by a valid username, replace it with that user's home folder. The tilde must be the first character in the token.

Parameter and Variable Expansion - replaces any variables or parameters referenced with their value. We use a dollar sign $ without any parentheses to reference a variable.

$ $var
$ $PATH

Arithmetic Expansion - The shell will try to evaluate an Arithmetic expression and replace it with the result. The arithmetic expression should appear as the following syntax:

$((expression)) - any variable in the expression doesn't need any '$'

Command Substitution - Replaces a command with the output of the command.

The syntax: $(command)

Word Splitting - This will take results from previous expansions and split the words from them.

Ex: $(1s a*.txt) expanded to al.txt a2.txt a3.txt a4.txt a5.txt
These are one token without word splitting.

Word splitting will split this into individual tokens. How we split is defined in the IFS variable. You can modify this variable to split based on additional characters, such as commas, periods, and colons. If the variable is empty, then no splitting happens.

Pathname Expansion: Also called filename generation or globbing, interprets ambiguous file references and substitutes appropriate lists of filenames. Tries to match *, ?, []  In bash, if it doesn't match the wild card characters, then it interprets the token as is.

Quotations - Double quotes suppress all expansion except parameter and variable Single quotes suppress all expansion