LINUX 101

Author: CJ Keist
Date: 01/03/2019

HOW TO REMOTE LOGIN TO LINUX

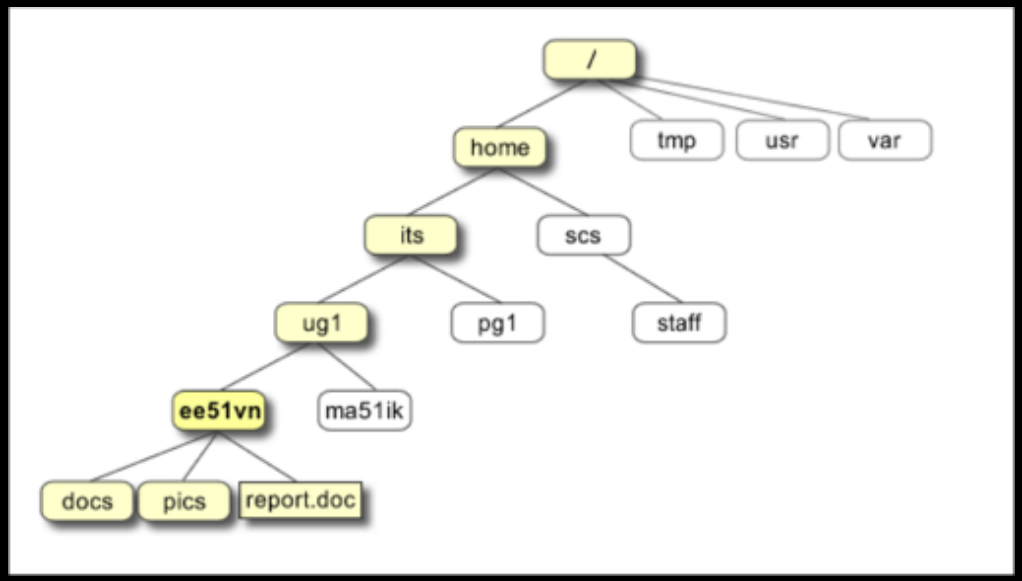
- Windows
  - MobaXterm
    - https://mobaxterm.mobatek.net/download-home-edition.html
  - Putty and Xming
    - http://www.putty.org/
    - http://www.straightrunning.com/XmingNotes/
  - X2Go
    - https://wiki.x2go.org/doku.php
- Mac OSX
  - X2Go
  - Xquartz
    - https://www.xquartz.org/
  - Term (Cannot run graphical programs)

THE SHELL

- When you remote login to a Linux server, like instruct-compute1, you will be placed in a shell.
- The Shell is an interface between you and the Linux Kernel.
- The shell is a command line interpreter (CLI). It executes commands the user types in, the commands themselves are programs.
- There are many different shell environments like csh, bash, tcsh and zsh. Each shell environment provides different features.
- The default shell for us is bash.

DIRECTORY STRUCTURE

- All files are grouped in a directory tree.
- The file system is structured hierarchically, like an upside down tree.
- The top of the hierarchy is called “root” written as a single ‘/’.



http://www.ee.surrey.ac.uk/Teaching/Unix/unix1.html

DIRECTORY STRUCTURE CONT.

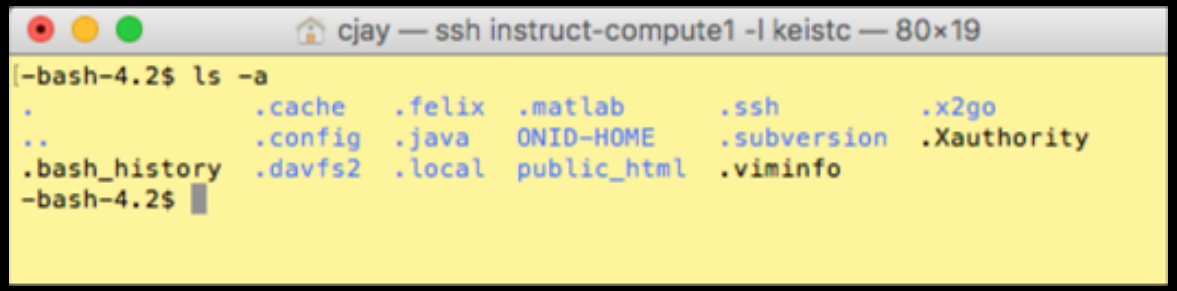
- For example, in the directory tree diagram the user ee51vn home directory would be shown as /home/its/ug1/ee51vn
- Many system programs are located in /usr/bin or /usr/sbin.
- Third party programs are usually located in /usr/local or /opt.
- User home directories are usually located under /home
- System configuration files are usually located under /etc
- For Linux, the kernel files are located under /boot

LISTING FILES AND DIRECTORIES

- When you first login to a system you will be placed in your home directory.
- To see what's in your home directory type the command "ls" (list).

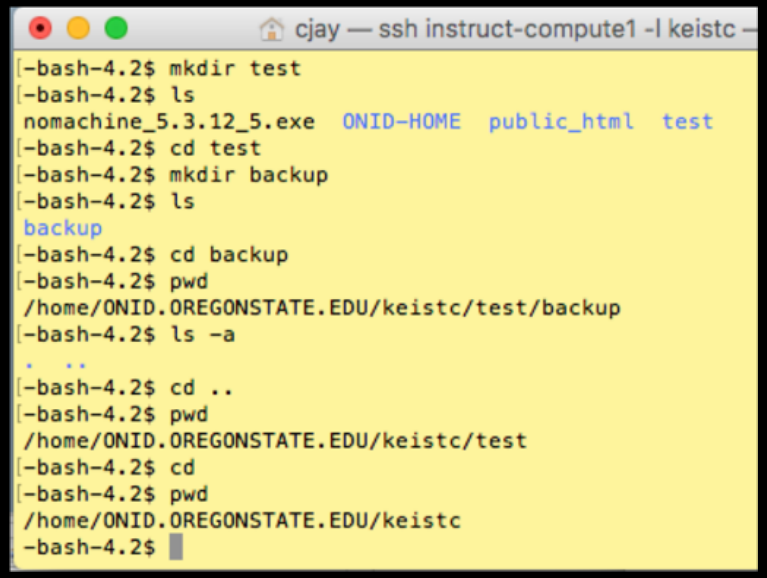
LISTING FILES AND DIRECTORIES (CONT.)

- The "ls" command doesn't show all files though. There are hidden files, file names that begin with a '.' dot. These types of files are usually for personal configuration files. You should not need to access or edit these files unless you know what you are doing.
- You can view hidden files with the '-a' flag to the "ls" command.



MAKING AND CHANGING DIRECTORIES

- The command “mkdir” is used to create new folders or directories.
- The command “mkdir test” will create a new folder in the current directory you are in.
- The command “cd” is to change directories.
- The command “pwd” prints your working directory path.
- Every directory has two special directories “.” and “..”. The “.” directory means the current directory. The “..” directory will take you to the parent directory.
- Typing “cd” by itself will always take you to your home directory.



MAKING AND CHANGING DIRECTORIES SUMMARY

Command	Description
ls	List contents of current directory
ls -a	List contents, including hidden files, of current directory
mkdir	Make a new directory
cd {directory_name}	Change to {directory_name}
cd	Change to home directory
cd ~	Change to home directory
cd ..	Change to parent directory
pwd	Print working directory path

WORKING WITH FILES

Command	Description
cp file1 file2	Copy file1 and name it file2
mv file1 file2	Move or rename file1 to file2
rm file1	Remove or delete file1
rmdir directory_name	Remove or delete directory(must be empty first)
cat file1	Display contents in file1
less file1	Display contents in file1 a page at a time
head file1	Display first few lines of file1
tail file1	Display last few lines of file1
grep ‘keyword’ file1	Search for keyword in file1 (Case sensitive)
grep –i ‘keyword file1	Search for keyword in file1 (Case insensitive)

REDIRECTING

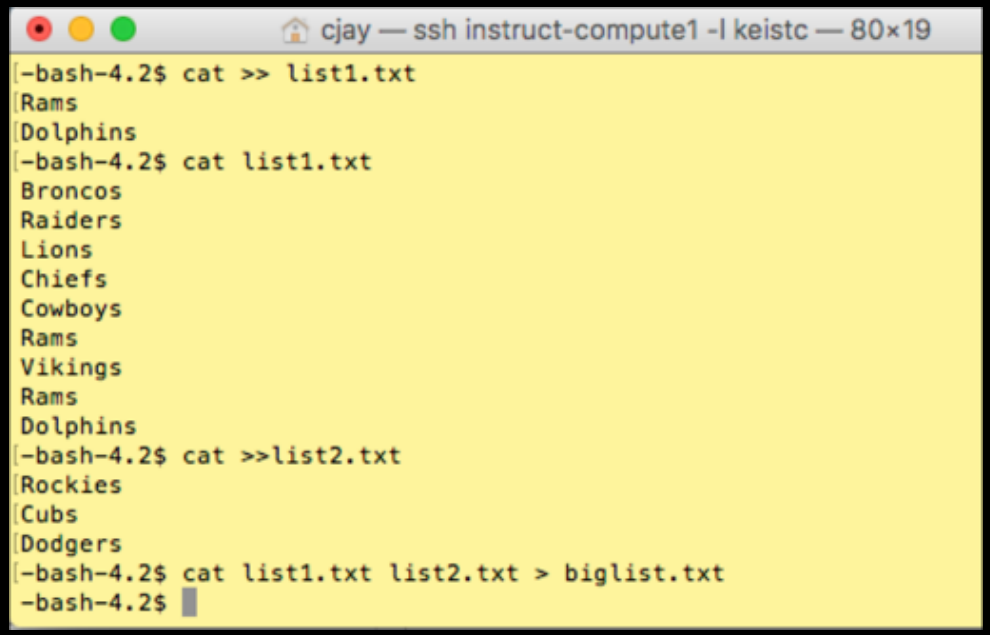
- Most commands write to standard output (print to the terminal screen) and many take input from the standard input (read from the keyboard). There is also standard error, where processes write their error messages, by default to the terminal screen.
- The “>” symbol is used to redirect the output of a command. For example try the command “cat > list1.txt” and type in some listed items. When finished press and hold down the “Ctrl” key and then press “d” on keyboard. This is end of file command.
- This will create a file named list1.txt. To view the contents type “cat list1.txt”.

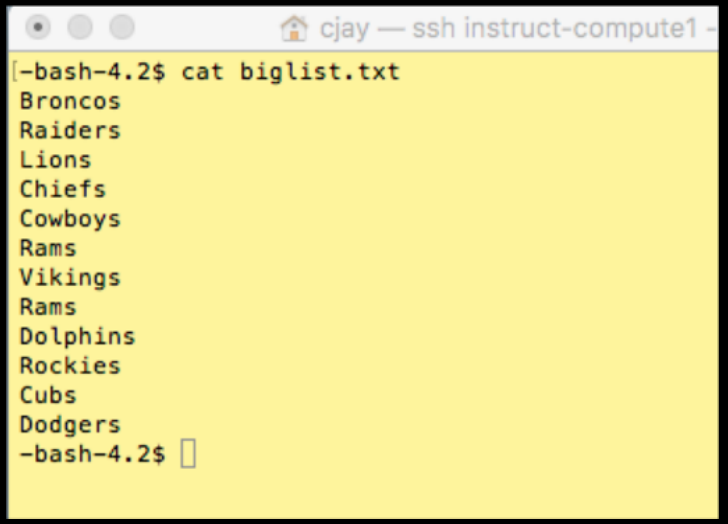
REDIRECTING CONT.

REDIRECTING CONT.

- To append to a file you use “>>”. Type “cat >> list1.txt” and type in some more items to your list. When finished type ^D (Control D to stop).
- Display contents of list1.txt file now. “cat list1.txt”
- Create a second list: “cat > list2.txt”
- Now combine both lists into a new file: “cat list1.txt list2.txt > biglist.txt”
- Display contents of biglist.txt. “cat biglist.txt”

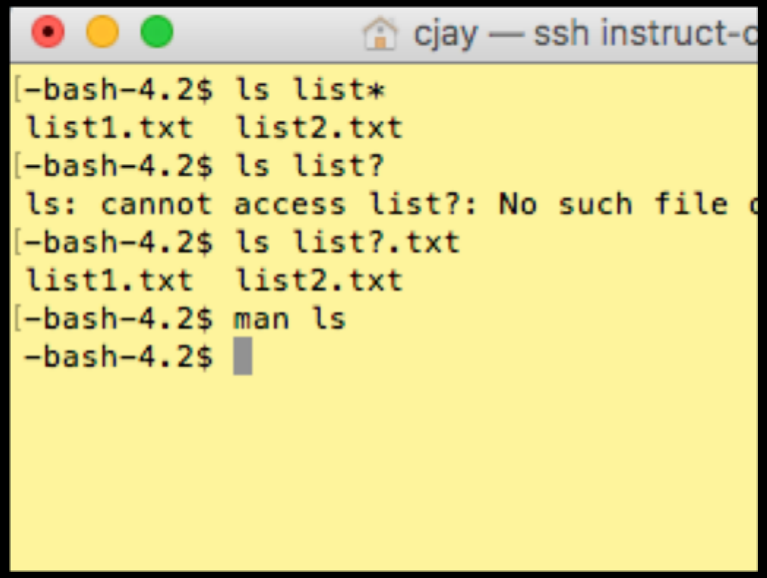
REDIRECTING CONT.





WILDCARDS

- The "*" character is called a wildcard, and will match against one or more characters in a file or directory name.
- Type "ls list*" and see what the output is.
- The "?" character will match exactly one character only.
- Type "list?" and see what output you get.
- Type "list?.txt" and see what output you get.
- If you want to know how to use a UNIX command and what command line arguments you can use, use the UNIX command "man" for manual.
- Type "man ls".



PIPES

- Pipe is a UNIX term to “pipe” the output of one command into the input of another command.
- Lets sort our biglist.txt file contents. To do this we will use the UNIX command “sort”. The pipe symbol is the “|” character. It’s usually above the “enter” key on the keyboard.
- Type the command “cat biglist.txt | sort > sorted.txt”
- Now display the contents: “cat sorted.txt”.
- Here we took the output of the “cat biglist.txt” command and piped it to the input of the “sort” command. Finally, we redirected the output into a new file”sorted.txt”.

PIPES CONT.

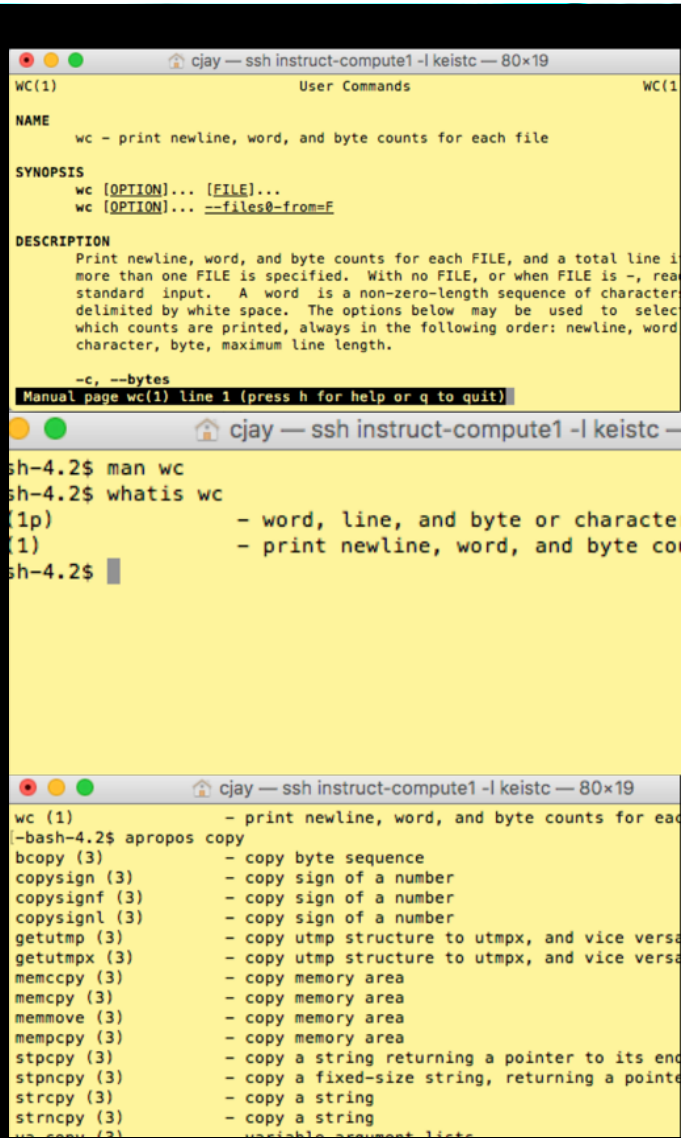

cjay — ssh instruct-compute1 -l keistc — 80×19
-bash-4.2$ cat biglist.txt | sort > sorted.txt
-bash-4.2$ ls
biglist.txt list2.txt ONID-HOME sorted.txt
list1.txt nomachine_5.3.12_5.exe public_html test
-bash-4.2$ cat sorted.txt
Broncos
Chiefs
Cowboys
Cubs
Dodgers
Dolphins
Lions
Raiders
Rams
Rams
Rockies
Vikings
-bash-4.2$

REDIRECTING SUMMARY

Command	Description
Command > file	Redirect standard output to a file
Command >> file	Append standard output to a file
Command < file	Redirect standard input from a file
Command1 | Command2	Pipe the output of command1 to the input of command2
cat file1 file2 > file3	Concatenate file1 and file2 to file3
sort	Sort data

ON-LINE MANUAL

- There are on-line manuals that gives information about most UNIX commands.
- Type “man wc”
- For short description type: “whatis wc”
- If you are not sure what command to use, type: ”apropos keyword”.
- Type “apropos copy”



FILE PERMISSIONS

- The command “ls –l” shows the long listing of contents in the current directory. The first column in the ”ls –l” output shows the security permissions on the file or directory.
    - -rw------- 1 541265 200513 85 Oct 10 14:00 biglist.txt
- They are three states files/directories can have: (r)ead, (w)rite and (x)execute.
- There are three ownership relations: 'owner', 'group' and 'all' so we need a triplet for each, resulting in nine bits. Each bit is denoted as (r)ead, (w)rite, (x)execute and (-) clear.
- For example: -rw-r----- would give read and write access to the owner, read access to anyone in the group and no access for everyone else.

FILE PERMISSIONS CONT.

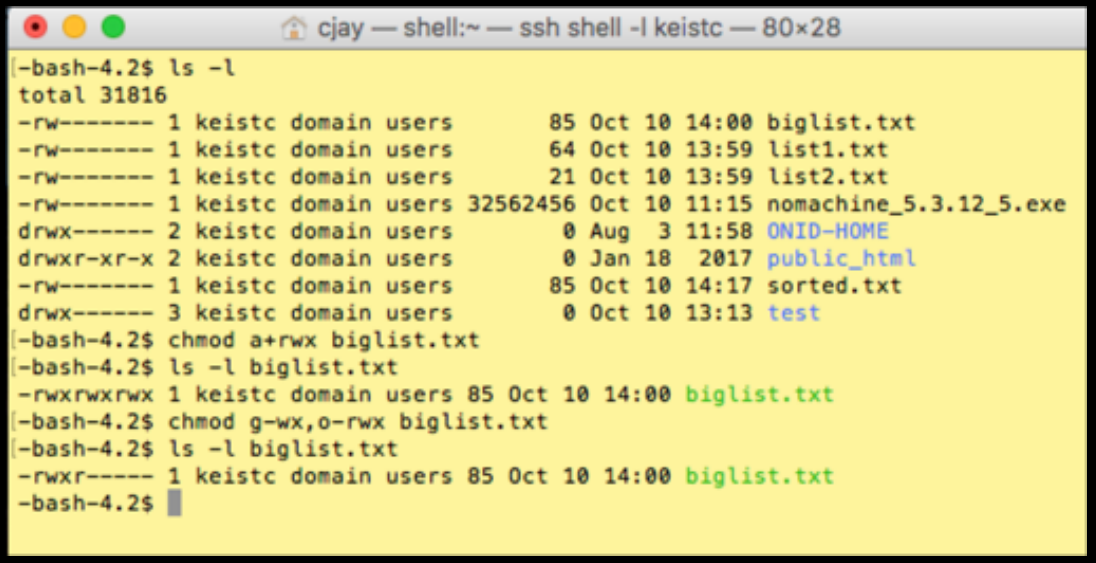
- In the previous file permission example there was a preceding (-) dash before the 9 bit permission string. That first dash represents the type of file. A dash means a regular file, if it has a "d" it states a directory.
- The permissions r,w and x have different meanings for directories. (r)ead determines if you can view the contents in a directory, i.e. will "ls" work. (w)rite determines if you can create or delete files in the directory. (x)execute determines if you can cd into the directory or not.
- To change permissions on a file/directory you need to use the "chmod" command. The syntax is "chmod {mode} file(s)". Only the owner of the file or directory can modify permissions.

FILE PERMISSIONS CONT.

- The mode specifies the new permissions to set on the file/directory. The mode is started by a letter that denotes what users will be affected by the change. These letters are:
  - u – The owner of the file/directory
  - g – The owner group
  - o – Others, everyone else
  - a – All users
- This is followed by the changes instructions which consists of a **+**(set bit) or **-** (clear bit) and the letter corresponding to the bit that should be changed.
- For example, lets change the permissions on the biglist.txt file by typing: "chmod a+rwx biglsit.txt". This will set the permissions wide open so that anyone can read and modify this file. This is not something you usually would want to do.

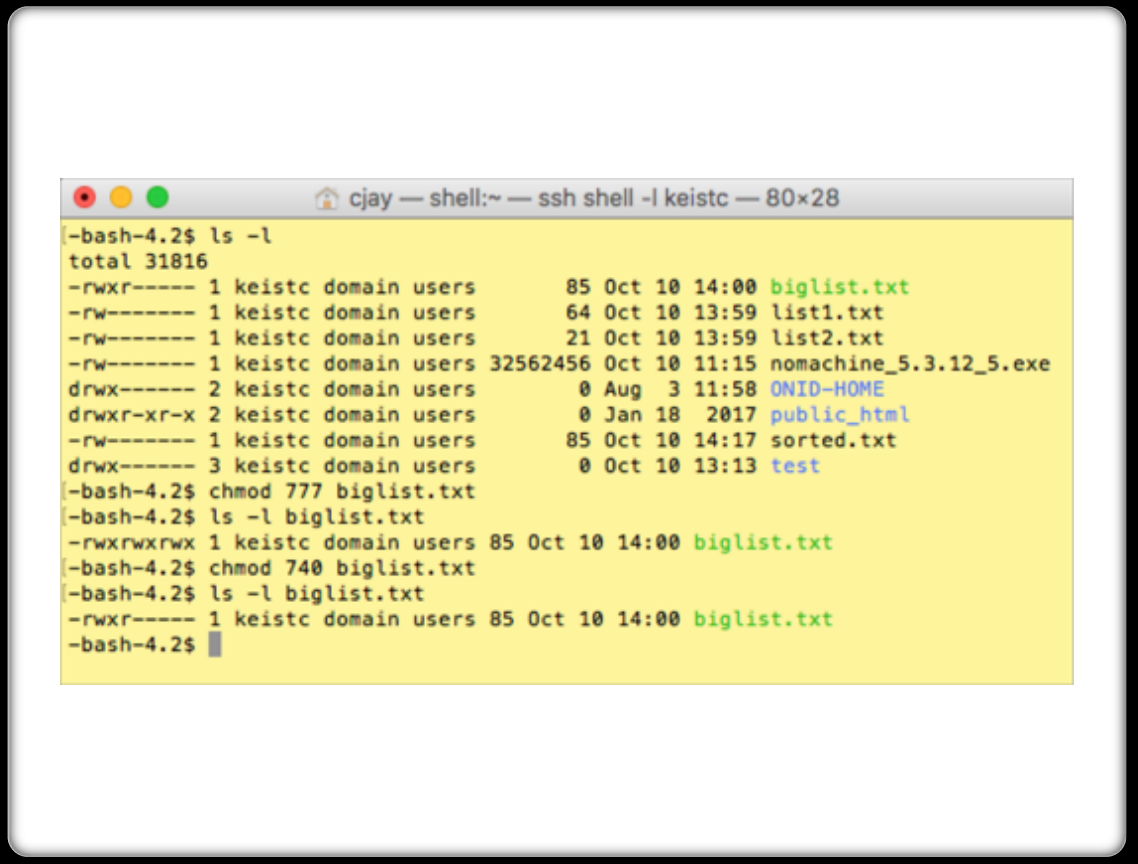
FILE PERMISSIONS CONT.

- Lets set permisson so that only members in your group can read the file now.Type: "chmod g-wx,o-rwx biglist.txt". Now do "ls –l biglist.txt" and see how the permissions look. In this command we used the "-" to clear the settings for the group and others. Notice that the group change didn't include the (r)ead bit as we wanted to make sure we kept read permission for the group on the file.



FILE PERMISSIONS CONT.

- Another way to use the chmod command is with direct values. Every permission bit in a triplet corresponds to a value: 4 for (r)ead, 2 for (w)rite, and 1 for (x)execute. By adding which bits you want set you then get the direct numeric value.
- Lets redo the security settings in the previous example using numeric values. Type: "chmod 777 biglist.txt". Then type: "chmod 740 biglist.txt".



FILE PERMISSIONS CONT.

- Table below shows all combinations and values for setting permissions on files and directories

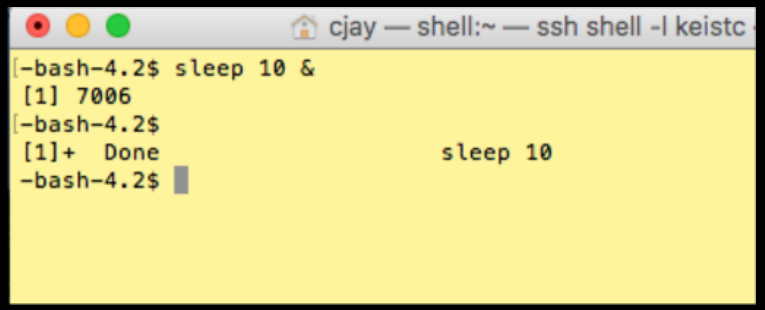
Characters	Value	Description
rwx	7	Read, write and execute
rw-	6	Read and write
r-x	5	Read and execute
r--	4	Read permission
-wx	3	Write and execute
-w-	2	Write permission
--x	1	Execute permission
---	0	No access

PROCESSES AND JOBS

- A process is an executing program. To see information about processes you own type the command "ps".
- A process can be in the foreground, background or suspended. In general a process doesn't return the shell prompt until it finishes.
- A process that will take a long time to run can tie up your terminal. To free up your terminal you can put the process in the background.
- A process in the background will continue to run while you can do other tasks in the terminal.
- To put a process in the background add the "&" character at the end of the command.

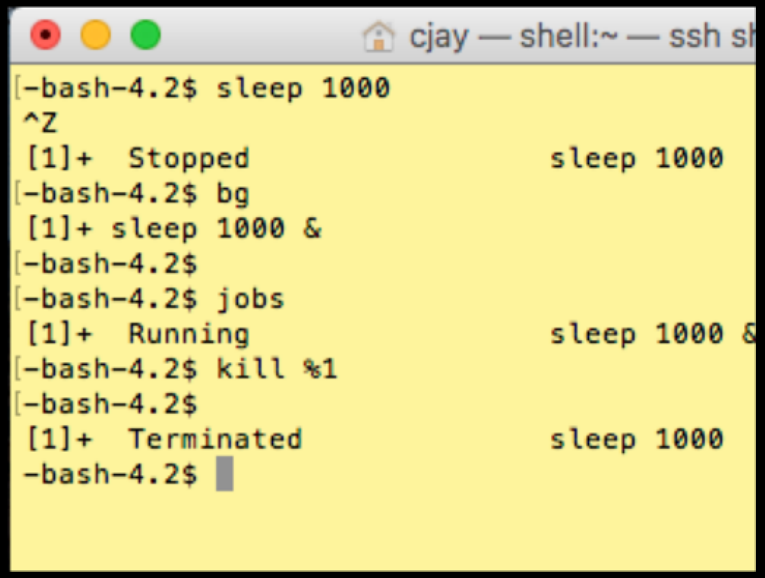
PROCESS AND JOBS CONT.

- For example, the “sleep” command will pause for specified number of seconds before continuing. Type: “sleep 10”. This will pause your terminal for 10 seconds before returning the shell prompt.
- Now lets put the ”sleep” command into the background. Type: “sleep 10 &”. Now the shell prompt returns immediately but the sleep command is still running in the background. It also displays the job ID number. After 10 seconds hit the ”enter” key and you will see the sleep process has finished.

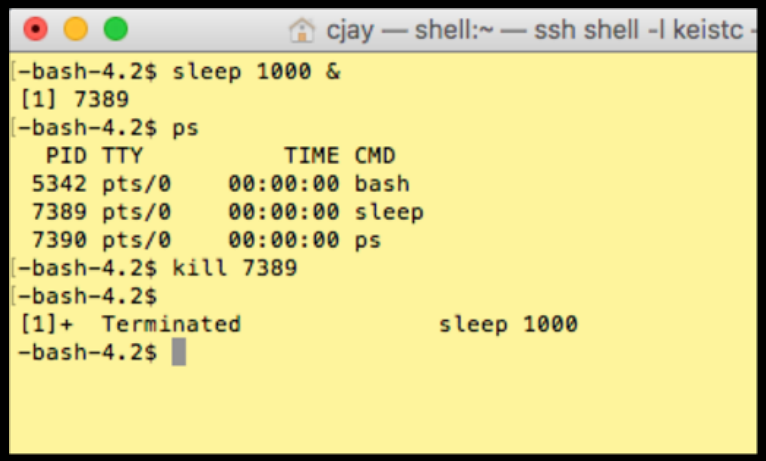


PROCESSES AND JOBS

- To put an existing running process into the background, type ^Z (hold down the Ctrl key and press z key). This will suspend the process and return the shell prompt to you. To put the process into the background type "bg" for background process.
- Lets try this out, type: "sleep 1000". They type ^Z followed by "bg" to put the sleep command into the background.
- To view current processes running, that you own, type the command: "jobs".
- To terminate the running sleep command type: "kill %1".



PROCESSES AND JOBS



- As stated earlier, the “ps” command shows your process status. Lets do the sleep command again and then view the process status with the “ps” command. Type: “sleep 1000 &”. Then type the “ps” command and see the output.
- Let’s now kill the sleep command with the process ID number.

PROCESSES AND JOBS SUMMARY

Command	Description
chmod {mode} file	Change file permissions for file
command &	Run command in background
^c	Kill job running in foreground
^z	Suspend the job running in foreground
bg	Background the suspended job
jobs	List current running jobs
fg %1	Foreground job number 1
Kill %1	Kill job number 1
ps	List current processes
Kill {pid}	Kill process with PID number

OTHER USEFUL COMMANDS

- df – This command shows available space on the server. Type “df –h” to view space in human readable numbers.
- du – Displays number of kilobytes each file/folder is using. Type “du –sh *” to sum up all space used and present in human readable numbers.
- gzip – Used to compress files in order to save on disk space. Type: ”gzip file_name”. This will rename the file by adding “.gz” to the end. To uncompress the file type: “gunzip file_name.gz”
- file – This command will tell you what type of file it is. Type: ”file *”
- diff – This command will show you the difference between two files. Type: “diff file1 file2”.
- history – Your shell will keep an orders list of all the command you have typed. The “history” command will display this list.

OTHER USEFUL COMMANDS CONT.

- which – This command will show you full path of shell commands and other third party software programs. Type: “which matlab”.
- whereis – Similar to “which” command but will locate the binary, source and manual pages if they are any.
- who – Display what other users are logged into the system with you.
- vim – Text editor.
- emacs – Another text editor.
- Xemacs – Graphical text editor of emacs.

UNIX RESOURCES

- A couple of very good YouTube videos on UNIX basics:
  - https://www.youtube.com/watch?v=BjO1BgeuPhE
  - https://www.youtube.com/watch?v=JVBtabkJ4ZE
- Reference:
  - http://www.ee.surrey.ac.uk/Teaching/Unix/unix1.html