UNIX INTRODUCTION

<u>UNIX Commands:</u>

* WARNING: UNIX is case senstive!!!

[] - Indicates an optional component.

mkdir - Makes a directory

- mkdir path-to-desired-new-direcory
- Example1 (to make a directory called JUNK):
  ```
  mkdir JUNK
  ```
- Example2 (to make a sub-directory SYS under directory JUNK):
  ```
  mkdir JUNK/SYS
  ```

* Note: You can only make sub-directories after you've made the directory.

rmdir - Removes a directory

- rmdir path-to-direcory-to-remove
- Example1 (to remove the JUNK sub-directory SYS):
  ```
  rmdir JUNK/SYS
  ```
- Example2 (to remove the directory JUNK):
  ```
  rmdir JUNK
  ```

* Note: You can only remove a directory after you've removed the sub-directories.
* Note: The directory must also be empty before it can be removed (see rm below).

cd - Changes or moves you to a directory

- cd path-to-desired-direcory
- Example1 (to move to the JUNK directory):
  ```
  cd JUNK
  ```
- Example2 (to move to the JUNK sub-directory SYS):
  ```
  cd JUNK/SYS
  ```
- Example3 (to move "back" 1 directory):
  ```
  cd ..
  ```
- Example4 (to move "back" 2 directories):

```
cd ../..
```

ls - Displays the contents of a directory on the screen.

- ls [option] [path] [filename(s)]
- Example1 (to list the current directory's contents):
  ```
  ls
  ```
- Example2 (to list the contents of the JUNK directory):
  ```
  ls JUNK
  ```
- Example3 (to list all ".exe" files in the JUNK directory):
  ```
  ls JUNK/*.exe
  ```

* Note: The options include -a (all files listed, including hidden files), -l (all file data, long display), and others.

cp - Copies a file from 1 location or filename to another.

- cp path1/file1 [drive:]path2/file2
- Example1 (to copy a file from the JUNK directory to the SYS directory)
  ```
  cp JUNK/hw1.txt JUNK/SYS/hw1.txt
  ```
- Example2 (to copy a file from the JUNK directory to the SYS directory and to use a different filename)
  ```
  cp JUNK/hw1.txt JUNK/SYS/hwa.txt
  ```
- Example3 (to copy a file to another file, if you are in the directory already)
  ```
  cp hw1.txt hwa.txt
  ```
- Example4 (to copy a file to another file)
  ```
  cp JUNK/hw1.txt JUNK/hwa.txt
  ```

mv - Moves a file from 1 location/name to another.

- mv path1/file1 path2/file2
- Example1 (to move a file from the JUNK directory to the SYS directory)
  ```
  mv JUNK/hw1.txt JUNK/SYS/hw1.txt
  ```
- Example2 (to move a file from the JUNK directory to the SYS directory and to use a different filename)
  ```
  mv JUNK/hw1.txt JUNK/SYS/hwa.txt
  ```
- Example3 (to move a file to another file, if you are in the directory already)
  ```
  mv hw1.txt hwa.txt
  ```
- Example4 (to move a file to another file - rename it)

```
mv JUNK/hw1.txt JUNK/hwa.txt
```

rm - Removes a file from the disk.

- rm [option] path/file
- Example1 (to remove the file hw.txt from the JUNK directory)

  ```
  rm JUNK/hw.txt
  ```

- Example 2 (to remove a directory, all sub-directories, and all files - rm recurse directories and force removal of all files)

  ```
  rm -rf JUNK
  ```

  * WARNING: rm -rf is a very dangerous command. You can "wipe" out your entire "disk" if you're not carefull. Never, ever use rm -rf when logged in as "root". Been there, done that, regretted it :(

cat - Displays the contents of a file on the screen.

- cat path/filename
- Example1 (to list the file hw.txt in the current directory):

  ```
  cat hw.txt
  ```

- Example2 (to list the file hw.txt in the JUNK directory):

  ```
  cat JUNK/hw.txt
  ```

WILDCARDS

* - Word wildcard.
? - Character wildcard.

* Note: In all of the above cases, wildcards can be used in place (or in conjunction with) filenames.

more - Displays the contents of a file on the screen (only does it page-by-page - you press ANY-KEY to get the next page). More can also be used with cat to do a page-by-page display. For examples see redirect, and pipe below.

pwd - Print working (current) directory (like "cd" in dos).

grep – Searches the specified files for the string specified (grep string filenames → grep help *.cpp).

man - The on-line UNIX manual (try: man man).

Terminal Commands:

Control U - Cancels the line you're typing.

Control C - Interrupts the program currently executing.

Control \ - Interupts the program currently executing (stronger than control C and produces a core dump).

Control Z - Suspends the execution of the current program (execution can be resumed later).

Control S - Stops output sent to terminal (suspends program).

Control Q - Restarts execution of stopped program.

Process Commands:

ps - Displays your processes

ps -f - Displays all info on your processes.

ps -A - Displays info on ALL processes running on the machine.

ps -fA - Displays all info on ALL processes running on the machine.

jobs - Displays jobs (including stopped jobs)

kill - Used to remove a process

kill -pid	- Kills a process.
kill %jobno	- Kills a job.
kill -s 9 pid	- Kills a process by sending the KILL signal (stronger than kill -pid).

fg - Used to restart a stopped job (move it to the forground). fg %jobno or fg pid

bg - Used to move jobs to the backround

bg %jobno or bg pid

REDIRECTION:

I/O includes standard input, standard output, and standard error. Redirection lets you reroute the I/O from any of these to or from a file.

<	redirect standard input
> or 1>	redirect standard output (overwrite)
>> or 1>>	redirect standard output (append)
2>	redirect standard error (overwrite)
2>>	redirect standard error (append)

Format: command < input > output

- cat < file_in > file_out

PIPES:

Redirects output from 1 program into another.

Format: command | command
- ls | more
- cat < source_file | sort
- cat < source_file | sort > result_file

ALIASES:

Aliases can be used to define new names for commonly used commands.

alias newname="unix_command" - Defines newname to be an alias for the UNIX command “unix_command”.

- i.e. alias dir="ls -la"

unalias newname - Removes the alias newname.

- i.e. unalias dir

* Since aliases do not remain in effect if you close the session, you can put commonly used aliases in your ".profile" file.

LINKS:

Links allow the creation of an alias or a knick name for a file. Changes to the link affect the file and the "rm" command removes the link, but not the file. There are 2 types of links:

1. "hard" links - A UNIX file can have multiple "hard" links. "Hard" links and the file must be on the same file system.

   ```
   - link file linking_name
   - ln file linking_name
   ```

2. "soft" links - Aka symbolic link. A symbolic link can be created across file systems. A UNIX file can have multiple symbolic links. Symbolic links can also refer to a directory.

   ```
   - ln -s file linking_path/file.
   ```

* Warning symbolic links can create circular references. Deletion of the linking_path/file results in an undefined link.

unlink - Removes a link ("hard" or "soft").

```
- unlink linked_name
```

PERMISSIONS:

File / directory permissions include read (r), write (w), and execute (x). Permissions can be set for the owner (u), members of the owners group (g), and all others (o). The UNIX command "chmod" is used to change permissions.

File permissions:

r - Required to be able to read a file.

w - Required to be able to write a file
(to edit requires r and w).

x - Required to be able to execute a file. Shell scripts (batch files) require r and x to execute.

Directory permissions.
r - Lets one see what's in the directory, but nothing else.
w - Lets one see what's in the directory, but nothing else??????.
x - Required to do anything with the directory or its contents.

* To view the file permissions in your account use the "ls -la" command (list ALL files using a LONG display) or use the "ls -l" command.