OSU Center for Genome Research and Biocomputing (CGRB), MCB 599: "Introduction to Unix/Linux" (INX_U18)
"Cheat Sheet": Day 3, **Pages:** 4, **Revision:** 2018-07-25, **Instructor:** Matthew Peterson (matthew@cgrb.oregonstate.edu)

Command/Concept	Usage/Syntax	Description	Example

Absolute path	`/<dir>/<dir>/<file>`	Absolute paths to a file or directory always start at the top of the filesystem tree, i.e., `/`, and traverse "down"	# Class directory
			`/ACTF/Course/inx_u18`
			# "Message of the day" file
			`/etc/motd`

Relative path	`<relative to $PWD>`	A relative path can locate a file or directory, which is "relative" to your `$PWD`	# cd to your "projects"
			# directory without
			# specifying an absolute path
			`cd $HOME`
			`cd projects`

.	Special file `.`	Every directory on the filesystem contains a hidden file . (single period). This file represents "here" (*this* directory). Seen by `ls -a`	# Go "here" # (Stay in the same directory) `cd .`
..	Special file `..`	Every directory on the filesystem contains a hidden file .. (two periods). This file represents "go up" (one directory). Seen by `ls -a`	# Go "up" one directory `cd ..`
`less`	`less <path_to_file>`	View the contents of a file. Arrow keys for up/down. Return/Enter key for a single line forward. Space bar for a page forward; `b` for back a page. q to quit out of less. **Note:** The mouse is useless!	# View "Message of the day" `less /etc/motd`

Search for a pattern in a file using `less`	`less <path_to_file>`
	`/<search pattern>`
		Search mode invoked by:
		`/<search pattern>`
		`n` for next match
		`p` for previous match
			# View "Message of the day"
			`less /etc/motd`
			# Search for string: `CGRB`
			`/CGRB`

`less -S`	`less -S <file>`	Ensure "long" lines are not word-wrapped on the display but instead extend "off screen." Scroll left/right with the arrow keys on those lines to see the content. CGRB infrastructure defaults to `-S` without having to specify it.	# Ensure no word-wrap `less -S /etc/bashrc`

`mkdir`	`mkdir <new_dir>`
	`mkdir <validpath>/<new_dir>`
		Create a new directory in your `$PWD` You can create a new directory (outside your `$PWD`) given a valid path	# Create a new directory
			# 'testo1' in your home dir
			`cd $HOME`
			`mkdir testo1`
			# Create a new directory
			# 'testo2' in your home dir
			`mkdir $HOME/testo2`

`mv`	`mv <source> <dest>`	Move or rename a file or directory.	# Rename directory
			# 'testo2' to 'testo3'
			`cd $HOME`
			`mv testo2 testo3`

Move files into a directory	`mv <f1> <f2> <dest>`	Move several files into a pre-existing directory.	`mv f1.txt f2.txt` `f3.txt testo3`

`cp`	`cp <source> <dest>`	Copy files and directories. Similar to `mv` with two differences: The original file is <u>not</u> removed. If you want to copy a directory use the recursive option: `-r`	# f4.txt is copied ("cloned")
			# into the testo3 directory
			`cp f4.txt testo3`
			# Copy the `testo1` dir into
			# the `testo3` dir
			`cp -r testo1 testo3`

`rm`	`rm <path_to_file>`	Removes a file	`rm f4.txt`

`rm -r`
`rm -rf`
	`rm -r <dir>`
	`rm -rf <dir>`
		Removes (recursively) a directory with `-r` Use `-f` (force) to suppress y/n removal confirmations.	# Asks confirmation
			`rm -r testo3`
			# No prompting, quiet
			`rm -rf testo3`

`du`	`du -sh <source>`	Determine size of a file or directory (`-s` summary). "Human readable" size (`-h`), e.g., **K**ilobytes, **M**egabyte, **G**igabyte, and **T**erabyte	# Size of home directory
			cd $HOME
			`du -hs .`
			# Size of projects directory
			`du -sh projects`

`nano`	`nano <path_to_file>`	Text editor. Arrow keys to move around. Ctrl-`o` to save changes. Ctrl-`x` to exit. `nano -w` to turn off word wrapping (for newly edited text).	`nano todo.txt`

`man`	`man <command name>`	Manual page for a command. Uses `less` to view the page.	`man du`
			`man nano`
			`man man`

`info`	`info <command name>`	Info documents for a command; different information than `man` (*possibly* less "cryptic"). Not available for all commands.	`info du`
			`info nano`
			`info man`

`top`	`top`	Displays running processes on the server. Processes sorted by `%CPU`. `%MEM` shows total system memory used.	`top`

Wildcard: `*`	`*`	Matches 0 or more characters.	`ls`
			`f1.txt f2.txt f99.txt`
			# Move all files
			`mv * $HOME/projects`

Wildcard: `?`	`?`	Matches any single character.	`ls`
			`f1.txt f2.txt f99.txt`
			# Move files with single digits
			# Does not move `f99.txt`
			`mv f?.txt`
			`$HOME/projects`

Shell tab completion	`<tab>`	When writing out paths on the command line you can have the shell attempt to "auto-complete" what you are typing (after the first few characters). If there's an ambiguous match it will show all the possible options when you press `<tab>`. Type some more characters that match followed by `<tab>` to continue.	# Press <tab> key `cd /ACTF/Co<tab>` # This auto-completes # to `/ACTF/Course`
Shell history	Up arrow at the command prompt	Loads the previous commands you have run (from your `history`). You can use the left/right arrow keys to edit previous commands; press `<Enter>` to re-run that command.	# For each previous # command repeatedly press: `<up arrow>`