

ITEC202 Operating Systems Lab - Linux/Unix Command Line Cheat Sheet

Command	Description
<code>pwd</code>	prints working directory (prints to screen, ie displays the full path, or your location on the filesystem)
<code>ls</code>	lists contents of current directory
<code>ls -l</code>	lists contents of current directory with extra details
<code>ls /home/user/*.txt</code>	lists all files in /home/user ending in .txt
<code>cd</code>	change directory to your home directory
<code>cd ~</code>	change directory to your home directory
<code>cd /scratch/user</code>	change directory to user on scratch
<code>cd -</code>	change directory to the last directory you were in before changing to wherever you are now
<code>mkdir mydir</code>	makes a directory called mydir
<code>rmdir mydir</code>	removes directory called mydir. mydir must be empty
<code>touch myfile</code>	creates a file called myfile. updates the timestamp on the file if it already exists, without modifying its contents
<code>cp myfile myfile2</code>	copies myfile to myfile2. if myfile2 exists, this will overwrite it!
<code>rm myfile</code>	removes file called myfile
<code>rm -f myfile</code>	removes myfile without asking you for confirmation. useful if using wildcards to remove files ***
<code>cp -r dir newdir</code>	copies the whole directory dir to newdir. -r must be specified to copy directory contents recursively
<code>rm -rf mydir</code>	this will delete directory mydir along with all its content without asking you for confirmation! ***
<code>nano</code>	opens a text editor. see ribbon at bottom for help. ^x means CTRL-x. this will exit nano
<code>nano new.txt</code>	opens nano editing a file called new.txt
<code>cat new.txt</code>	displays the contents of new.txt
<code>more new.txt</code>	displays the contents of new.txt screen by screen. spacebar to pagedown, q to quit
<code>head new.txt</code>	displays first 10 lines of new.txt
<code>tail new.txt</code>	displays last 10 lines of new.txt
<code>tail -f new.txt</code>	displays the contents of a file as it grows, starting with the last 10 lines. ctrl-c to quit.
<code>mv myfile newlocdir</code>	moves myfile into the destination directory newlocdir
<code>mv myfile newname</code>	renames file to newname. if a file called newname exists, this will overwrite it!
<code>mv dir subdir</code>	moves the directory called dir to the directory called subdir
<code>mv dir newdirname</code>	renames directory dir to newdirname
<code>top</code>	displays all the processes running on the machine, and shows available resources
<code>du -h --max-depth=1</code>	run this in your home directory to see how much space you are using. don't exceed 5GB
<code>ssh servername</code>	goes to a different server. this could be queso, brie, or provolone
<code>grep pattern files</code>	searches for the pattern in files, and displays lines in those files matching the pattern
<code>date</code>	shows the current date and time
<code>anycommand > myfile</code>	redirects the output of anycommand writing it to a file called myfile
<code>date > timestamp</code>	redirects the output of the date command to a file in the current directory called timestamp
<code>anycommand >> myfile</code>	appends the output of anycommand to a file called myfile
<code>date >> timestamp</code>	appends the current time and date to a file called timestamp. creates the file if it doesn't exist
<code>command1 command2</code>	"pipes" the output of command1 to command2. the pipe is usually shift-backslash key
<code>date grep Tue</code>	displays any line in the output of the date command that matches the pattern Tue. (is it Tuesday?)
<code>tar -zxf archive.tgz</code>	this will extract the contents of the archive called archive.tgz. kind of like unzipping a zipfile. ***
<code>tar -zcf dir.tgz dir</code>	this creates a compressed archive called dir.tgz that contains all the files and directory structure of dir
<code>time anycommand</code>	runs anycommand, timing how long it takes, and displays that time to the screen after completing anycommand
<code>man anycommand</code>	gives you help on anycommand
<code>cal -y</code>	free calendar, courtesy unix
<code>CTRL-c</code>	kills whatever process you're currently doing
<code>CTRL-insert</code>	copies selected text to the windows clipboard (n.b. see above, ctrl-c will kill whatever you're doing)
<code>SHIFT-insert</code>	pastes clipboard contents to terminal

*** = use with extreme caution! you can easily delete or overwrite important files with these.

Absolute vs relative paths.

Let's say you are here: /home/turnersd/scripts/. If you wanted to go to /home/turnersd/, you could type: `cd /home/turnersd/`. Or you could use a relative path. `cd ..` (two periods) will take you one directory "up" to the parent directory of the current directory.

`.` (a single period) means the current directory

`..` (two periods) means the parent directory

`~` means your home directory

A few examples

<code>mv myfile ..</code>	moves myfile to the parent directory
<code>cp myfile ../newname</code>	copies myfile to the parent directory and names the copy newname
<code>cp /home/turnersd/scripts/bstrap.pl .</code>	copies bstrap.pl to "." i.e. to dot, or the current directory you're in
<code>cp myfile ~/subdir/newname</code>	copies myfile to subdir in your home, naming the copy newname
<code>more ../.././myfile</code>	displays screen by screen the content of myfile, which exists 3 directories "up"

Wildcards (use carefully, especially with rm)

`*` matches any character. example: `ls *.pl` lists any file ending with ".pl"; `rm dataset*` will remove all files beginning with "dataset"

`[xyz]` matches any character in the brackets (x, y, or z). example: `cat do[or]m.txt` will display the contents of either doom.txt or dorm.txt