

directory / file commands

```

pwd                - print the working directory
ls -l -alh         - list the contents of the current directory
cd dir           - change directory to dir
mkdir dir        - make directory named dir
touch file       - make a file named file
rm file          - remove a file named file
rm -r dir        - remove directory and contents named dir
rm -f file       - forcefully remove a file named file
rm -rf dir       - forcefully remove a directory dir and contents (careful with this)
mv file1 file2   - move file1 to file2 (used for renaming files)
mv dir1 dir2     - move dir1 to dir2 (used for renaming dirs)
cp file1 file2   - copy file1 to file2
cp -r dir1 dir2  - copy dir1 to dir2
cat file        - display the contents of file to stdout
less file       - display the contents of file fitting within the terminal screen
head -n 10 file  - display the first 10 lines file
tail -n 10 file  - display the last 10 lines of file
sort file       - display the contents of the file with each line sorted
wc -l file       - count the number of lines in file
ln -s target name - create a link to the target file with name

```

system commands

```

w                 - display who is logged in
whoami           - display who you
man command     - display info about command
df -h            - display currant disk usage
du -sh dir      - display disk usage of dir
which app       - display the path to the
                  location of the app
whereis app    - display all possible paths to
                  the app
history          - display all commands that
                  have been run
clear            - clear the terminal of text

```

file permission commands

```

chmod 777 file - set read(r) write(w) and
                  execute(x) for all users
chmod 755 file - set owner to rwx and
                  all other users to rx
chmod 766 file - set owner to rwx and
                  all other users to rw
chmod 644 file - set owner to rw and
                  all other users to r
chmod +x file  - make file executable for
                  all users
chown user file - change the owner of file
                  to user

```

compression commands

```

tar -cf file.tar files - create a tar named file.tar containing files
tar -xf file.tar        - extract the files from file.tar
tar -czf file.tar.gz files - create a tar with Gzip compression
tar -xzf file.tar.gz    - extract a tar using Gzip
tar -cjf file.tar.bz2 files - create a tar with Bzip2 compression
tar -xjf file.tar.bz2   - extract a tar using Bzip2
gzip file               - compresses file and renames it to file.gz
gzip -d file.gz        - decompresses file.gz back to file

```

process commands

`ps -e` - snapshot of processes
`top` - show processes in real time
`kill pid` - kill processes with id `pid`
`pkill name` - kill processes with `name`
`killall name` - kill all processes with `name`
(careful with this)

searching

`grep pattern files` - search for `pattern` in `files`
`grep -r pattern dir` - search for `pattern` in `dir`
`find dir -name "pattern"` - find all files with `pattern` in `name` in `dir`

pipng commands

`cmd > file` - redirect the standard output (stdout) of `cmd` to `file`
`cmd 2> file` - redirect the standard error (stderr) of `cmd` to `file`
`cmd &> file` - redirect the stdout and stderr of a `cmd` to `file`
`cmd >> file` - redirect the stdout of `cmd` to `file` append to file if it exists
`cmd > /dev/null` - discard the stdout of `cmd`
`cmd < file` - redirect the contents of the file to the standard input (stdin) of `cmd`
`cmd <(cmd1)` - redirect the stdout of `cmd1` through a file to `cmd` (useful if `cmd` takes a file input)

`cmd1 | cmd2` - redirect the stdout of `cmd1` to the stdin of `cmd2`
`xargs cmd` - reads data from stdin and executes `cmd` one or more times depending on the input

useful commands

- count the number of unique lines in a file
`cat file.txt | sort -u | wc -l`
- find all files with "assembly" in the name and copy them to a single assembly.txt file
`find . -name "*assembly*" | xargs cat > assembly.txt`
- copy all ".fastq.gz" files from `dir1` to `dir2`
`cp `find dir1 -name ".fastq.gz"` dir2`
- split a multi fasta to individual fasta files
`awk '/^>/{s=++d".fa"} {print > s}' multi.fa`
- convert a fastq file to fasta
`sed -n '1~4s/^@/>/p;2~4p' file.fq > file.fa`
- calculate the mean length of reads in a fastq file
`awk 'NR%4==2{sum+=length($0)}END{print sum/(NR/4)}' input.fastq`
- create a backup of files here all .txt files are backedup as .bak
`find . -name "*.txt" | sed "s/\.txt$//" | xargs -i echo mv {}.txt {}.bak | sh`
- find these and many more at:
<https://github.com/stephenturner/oneliners>