

Table 1: Bash commands and what they mean

| Command  | What it does  |
|--|---|
| <code>ls</code>                                      | list contents of current directory  |
| <code>ls -a</code>                                   | show hidden files too   |
| <code>ls -altr</code>                                | see the last changes made to the files in a directory                               |
| <code>mkdir directory</code>                         | make a new directory  |
| <code>cd directory</code>                            | change directory  |
| <code>cd ..</code>                                   | go back a directory   |
| <code>cd ../../</code>                               | go back two directories   |
| <code>cd ~</code>                                    | go to your root   |
| <code>pwd</code>                                     | print working directory   |
| <code>~/</code>                                      | means your root   |
| <code>.</code>                                       | means the current directory   |
| <code>cp file/to/copy where/newName</code>           | copy a file   |
| <code>cp file/to/copy .</code>                       | copy a file to current directory without changing the name                          |
| <code>cp directory/* .</code>                        | copy all the files in a directory to the current directory                          |
| <code>cp -r directory new/directory</code>           | copy a directory recursively  |
| <code>rm file/to/remove</code>                       | remove a file   |
| <code>rmdir directory</code>                         | remove a directory  |
| <code>rm -rf directory</code>                        | blow away a directory permanently   |
| <code>mv file/to/move where/newName</code>           | moves or renames a file   |
| <code>man command</code>                             | show the manual for a command   |
| <code>cat file/one file/two<br/>&gt; new_file</code> | concatenate two or more files into a new file                                       |
| <code>history</code>                                 | shows a history of your commands  |
| <code>less file/to/see</code>                        | shows one page of a file  |
|  | space turns the page q quits  |
| <code>head file/to/see</code>                        | see the first page of a file  |
| <code>head -n 8 file/to/see</code>                   | see the first 8 lines of a file   |
| <code>tail file/to/see</code>                        | see the last page of a file   |
| <code>tail -n 10 file/to/see</code>                  | see the last 10 lines of a file   |
| <code>grep keyword file/to/search</code>             | search a file for a keyword and print all the lines with that keyword to the screen |
| <code>history   grep keyword</code>                  | search your history for a keyword   |
| <code>grep keyword file/to/search<br/>  wc -l</code> | count the occurrences of lines with a keyword                                       |
| <code>command   less</code>                          | pipe the output of a command to less  |
| Table 1 – Continued on next page                     |   |

| Table 1 – <i>Continued from previous page</i>    |   |
|--|---|
| Command  | What it does  |
| command >> file                                  | append the output of a command to a file                                |
| command > file                                   | writes the output of the command to a file                              |
| !command   | executes the most recent command that starts with the letters you typed |
| echo something                                   | print something to the screen   |
| ls -altr   | see when files in the directory were last altered                       |
| sed -i 's/to replace/new phrase/' file/to/search | find file/to/search phrase in a file                                    |
| grep -Rl keyword                                 | recursively search for a keyword and print the file it was found in     |
| awk '!a[\$0]++' file/to/search                   | get rid of duplicate lines  |
| echo "phrase" >> file/to/append                  | append a phrase to a file   |

Table 2: Emacs commands and what they mean

| Command                     | What it does  |
|-----------------------------|---|
| emacs path/to/file          | enter emacs editor for existing file or creates new file with that name |
| ctrl+x ctrl+s               | save file   |
| ctrl+x ctrl+c y             | save and quit a file  |
| ctrl+x ctrl+c n             | quit without saving   |
| ctrl+w                      | cut a line (highlight line first)                                       |
| ctrl+y                      | paste a line  |
| ctrl+k                      | kills the contents of a line/cut a line                                 |
| ctrl+k ctrl+k ...           | kills however many lines—helpful to copy and paste blocks of code       |
| ctrl+shift+-                | undo  |
| ctrl+u 3 command            | executes the command 3 times  |
| ctrl+x ctrl+f               | find and open a file (at the bottom of the screen)                      |
| ctrl+space                  | set marker  |
| Add text to a block of code |   |
| ctrl+space                  | set marker  |
| ctrl+x                      | set end of rectangle  |
| R                           | format as rectangle   |
| T “text”                    | add text  |
| or k                        | or delete   |

Table 3: Loops in bash

| Command  | What it does  |
|--|---|
| for i in {1..100}<br>do<br>command \$i<br>done   | for 100 iterations<br><br>do this thing(\$i<br>references the index)  |
| if [ condition ]<br>then<br>command<br>else<br>command<br>fi<br>if [ -e file ]<br>for i in `ls -d */`; do cd \$i; command; cd ..; done | check the condition<br>if it's true<br>do this<br>if it's not<br>do this<br>ends if statement<br>check if a file exists<br>iterate over each directory in the current directory |

Table 4: Example of a Bash loop

| Command  | What it does   |
|--|--|
| for i in {1..100}<br>do<br>cd E.\$i<br>if [ -e KPOINTS ]<br>echo \$i<br>getKPoints<br>fi cd ..<br>done | for 100 times<br><br>enter the directory named E.#<br>if KPOINTS doesn't exist<br>print the directory number<br>run the getKPoints script<br>go back one directory<br>close loop |

Table 5: Example of a Bash loop

| Command  | What it does   |
|--|--|
| for i in `ls -d */`<br>do<br>cd \$i<br>if [! -f KPOINTS] | for all directories<br><br>enter the directory named<br>if KPOINTS doesn't exist |

Table 5 – Continued on next page

| Table 5 – <i>Continued from previous page</i>       |   |
|---|---|
| Command   | What it does  |
| <pre> then getKPoints echo \$i fi cd .. done </pre> | <pre> run the getKPoints script print the directory number end if statement go back one directory close loop </pre> |